Interpretation and Analysis of Deep Face Representations: Methods and Applications

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Thrupthi Ann John
201350866
thrupthi.ann@research.iiit.ac.in

Advisors: Prof. C. V. Jawahar
Prof. Vineeth N Balasubramanian

International Institute of Information Technology Hyderabad
500 032, India

December 2024

To

the giants on whose shoulders I stand

and

to those who will stand on my shoulders

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Interpretation and Analysis of Deep Face Representations: Methods and Applications*** by *Thrupthi Ann John* has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. C. V. Jawahar

_____ _____

Date Advisor: Prof. Vineeth N. Balasubramanian

Acknowledgements

”I will instruct you and teach you in the way which you should go;
I will counsel you with My eye upon you. ” - Psalm 32:8

First and foremost, I thank God for the countless blessings and mercies bestowed upon me, without which this achievement would not have been possible. I’m grateful for His unwavering faithfulness through all situations in my life.

My deepest appreciation goes to my advisors, Prof. C. V Jawahar and Prof. Vineeth N Balasubramanian, for their constant support and motivation. Prof. Jawahar, thank you for believing in me, providing steadfast support during challenging times, and teaching me to see the bigger picture. Prof. Vineeth, your technical guidance and willingness to dive into detailed discussions have been invaluable.

I am also grateful to my co-authors, Isha Dua and Riya Gupta. A special thanks to Isha for the endless discussions, constant technical and emotional support, and always being there for me. Words can’t fully express my gratitude.

I feel lucky to have had the support of many friends and colleagues who have contributed significantly to my professional and personal growth. Thanks to my fellow CVIT lab mates: Pritish Mohapatra, Jobin K. V., Aniket Singh, Sourabh Daptadar, Avijit Dasgupta, Sesadri Mazumder, Bhavani Sambaturu, Anand Misra, Yashaswi Verma, Tejaswi Kasarla, Priyam Bakliwal, Vinitha V. S., Swetha Sirnam, Jitendra Yasaswi, Swagatika Panda, and many others. Special thanks to Aditya and Praveen for the countless hours of technical and philosophical discussions and encouragement.

A huge thanks to my parents, sister, and parents-in-law for their unwavering faith and silent support. My heartfelt thanks to my husband, Toju, and my son, David, for their constant encouragement and enthusiasm about my work. Their love and support have been my rock.

I also want to thank the CVIT and IIIT admin staff for their help with paperwork and logistical requests. I’m grateful to all the CVIT and IIIT faculty members and the larger IIIT community for fostering an environment of scientific and academic fervor and for providing a safe space for the expression and exploration of progressive thought and ideals.

Finally, for all those I have mentioned here and the many more I may have missed, I am deeply thankful for your direct or indirect support throughout my PhD journey at IIIT.

Abstract

The rapid growth of deep neural network models in the face domain has led to their adoption in safety-critical applications. However, a crucial limitation hindering their widespread deployment is the lack of comprehensive understanding of how these models work and the inability to explain their decisions. Explainability is essential for ensuring the correctness, reliability, and fairness of AI systems, and there is a growing recognition of its importance across AI applications. Despite the significance of explainability, most current methods are designed for general object recognition tasks and cannot be directly applied to the face domain. Faces are highly structured objects, and face tasks often involve fine-grained details, making them unique and distinct from general object recognition. This thesis aims to bridge the gap in explainability literature for the face domain by providing novel methods for interpreting and analyzing deep face representations.

In this thesis, we embark on a comprehensive journey of interpreting and analyzing deep face representations to uncover the underlying mechanisms behind DNN-based face-processing models. We first visualize face representations and introduce methods to identify functional concepts in face representations using 'cross-task aware filters' (CRAFT). Our approach includes an efficient task-aware pruning method using CRAFTs. We also present state-of-the-art Canonical Saliency Maps (CMS) to pinpoint critical input features. We thoroughly analyze deep face representations to understand the learned features and their functional relevance in different face tasks. To further enhance our understanding of human attention in the context of driving behavior, we investigate driver gaze patterns and develop DashGaze, a large-scale naturalistic driver gaze dataset . Using this dataset, we propose an innovative calibration-free driver gaze estimation algorithm that provides valuable information for studying and predicting driver behavior.

The comprehensive overview, experimental studies, and analyses presented in this thesis contribute to the wider adoption of explainability methods in face-processing tasks, enabling safer and more trustworthy deployment of deep-face algorithms in real-world applications. By shedding light on the inner workings of these models and their biases, this work paves the way for the responsible and ethical development of AI technologies in the face domain.

Contents

List of Figures

List of Tables

List of Related Publications

Journals:

1. **Thrupthi Ann John**, Vineeth N Balasubramanian, and C.V. Jawahar - *Explaining Deep Face Algorithms through Visualization: A Survey* , in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 6, no. 1, pp. 15-29, Jan. 2024, doi: 10.1109/TBIOM.2023.3319837

2. **Thrupthi Ann John**, Vineeth N Balasubramanian, and C.V. Jawahar - *Canonical Saliency Maps: Decoding Deep Face Models*, in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 3, no. 4, pp. 561-572, Oct. 2021, doi: 10.1109/TBIOM.2021.3120758

Conferences:

1. **Thrupthi Ann John**, Isha Dua, Vineeth N Balasubramanian, and C.V. Jawahar - *ETL: Efficient Transfer Learning for Face Tasks*, 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, (VISAPP) 2022.

2. Isha Dua, **Thrupthi Ann John**, Riya Gupta, C.V.Jawahar - *DGAZE: Driver Gaze Mapping on Road* International Conference on Robotics and Automation (IROS) 2020.

Patent (provisional):

1. C V Jawahar, Isha Dua, **Thrupthi Ann John** - *System and Method for Generating Gaze Mapping Dataset and Predicting Gaze Point on Environment*, Indian provisional patent application no. 202041052016

Manuscript under preparation/review:

1. **Thrupthi Ann John**, Vineeth N Balasubramanian, and C.V. Jawahar - *DashGaze: Driver Gaze Through Dashcam*

PART I

Overview

Chapter 1

Introduction

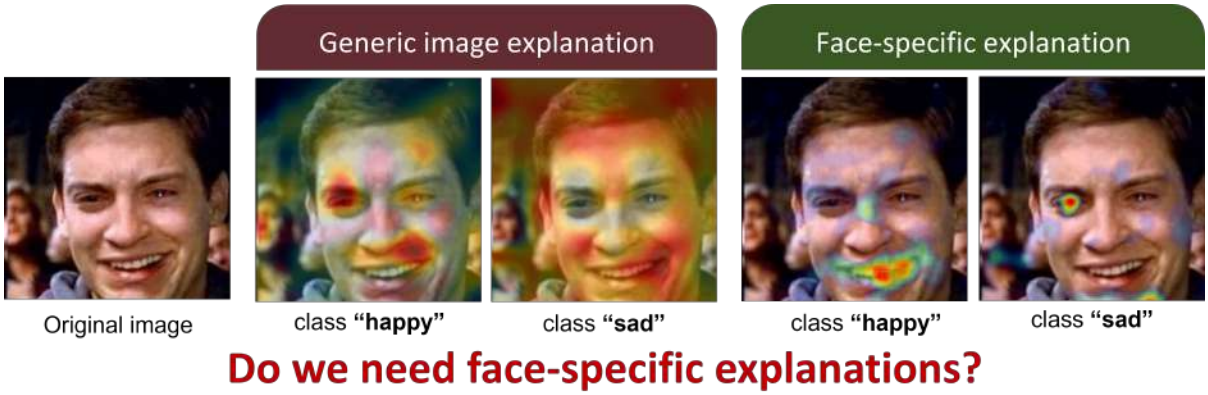


Figure 1.1: Why do we need to study face explainability? In this thesis, we show many examples where generic explainability methods fail on face models due to the unique properties of the face domain.

1.1 Motivation

Deep learning achieves state-of-the-art performance in most computer vision tasks, surpassing earlier methods by a large margin. The performance of deep neural networks is improving in leaps and bounds for face tasks such as face recognition and detection. In 2014, DeepFace [1] approached human performance for the first time on the LFW benchmark [2], a dataset of face images in unconstrained settings (DeepFace: 97.35% vs. Human: 97.53%), using a training dataset of 4 million images. In recent years, the accuracy has increased to 99.8% [3], surpassing human performance on the benchmark. Deep face models are now deemed to be real-world ready. They are used in many critical areas by government agencies, law enforcement, security etc. who may use commercial software or models trained by the agencies for this purpose. Currently, models for face tasks are available from major companies like Microsoft, IBM and Amazon who claim that their models are highly accurate. In this scenario, two crucial questions need to be answered: Do pre-trained models perform as well as they claim, and how do we find the weaknesses existing in these models and improve them. Failures of face models in these criti-

Gender Classifier	Overall Accuracy on all Subjects in Pilot Parlaiments Benchmark (2017)	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	93.7%	94.0%	79.2%	100%	98.3%	20.8%
FACE++	90.0%	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	87.9%	88.0%	65.3%	99.7%	92.9%	34.4%

Figure 1.2: Does high accuracy mean that a model is correct? (Figure taken from www.gendershades.org)

cal areas have far-reaching and devastating consequences. Inaccuracies in facial recognition technology can result in an innocent person being misidentified as a criminal and subjected to unwarranted police scrutiny. Big Brother Watch UK released the Face-Off report [4] highlighting false positive match rates of over 90% for facial recognition technology deployed by the Metropolitan police. A recent study [5] demonstrated that although commercial software solutions report high accuracies (Amazon's Rekognition reports an accuracy of 97%), they demonstrate skin-type and gender biases that go unreported as the benchmarks themselves are skewed. (See Figure 1.2 When performance is reported on public or private databases, they are always subject to the biases inherent in them. The algorithms may then be used in the real world in conditions that differ wildly from the ones they are tested in, causing the algorithms to produce erroneous results. How do we catch such issues at an early stage?

We have seen that a high reported accuracy does not guarantee robustness in real-world applications. We examine interpretability as a key solution, proposing methods to dissect DNN decision-making processes and assess model behavior, ultimately enhancing trustworthiness and resilience in deployment. Interpretability improves the reliability of the deep neural network models. There is less chance of catastrophic failures and dataset biases when there is an intuitive understanding of the neural network models. It helps us understand how the models may behave in the future. It also aids us in improving the performance of existing algorithms and creating new algorithms. Moreover, the opaqueness of deep models restricts their usefulness in highly regulated environments such as healthcare and autonomous driving, which may require the reasoning of the decisions taken by the deep models to be provided. To build trust in deployed intelligent systems, they need to be transparent, i.e., they should be able to explain why they predict what they predict [6]. Interpretable algorithms allow us to responsibly deploy deep face models in the real world, as we will be aware of their characteristics and shortcomings.

This thesis focuses on interpreting and analyzing deep face representations, highlighting the inadequacy of existing general explainability algorithms for the face domain. We introduce cutting-edge face-specific explainability algorithms and comprehensively analyze deep face representations. Moreover, we explore human attention through driver gaze, studying saliency patterns to gain insights into

Figure 1.3: This figure shows some common challenging conditions in face images. Images from the WIDER face database [7]

driver behavior. By addressing these key aspects, the thesis advances the understanding of face models and human attention, paving the way for more robust and interpretable AI systems in the face domain.

1.2 Challenges

As discussed in Section 1.1, there exist many works on deep learning explainability. However, there are several challenges associated with creating and applying explainability methods to the face domain. We describe some of them here:

1. **Privacy and Sensitivity:**
   Faces are the primary means of human identity, making it crucial to handle face data with utmost care and respect for privacy. The collection and processing of faces should be done in a way that does not violate individuals' privacy. Moreover, face applications, particularly in security-related contexts, demand high accuracy and fairness to avoid disastrous consequences. Research has highlighted racial, gender, and age-related biases in face processing systems, making it essential to develop face algorithms that are conscious of and address these biases. To ensure fairness, it is imperative to explain the decisions face algorithms make and mitigate any biases that may arise.

2. **Challenging Variations:**
   Faces in images exhibit considerable variability due to challenging conditions. In crowded scenes, faces may be numerous and small, making detection and recognition challenging. Various factors, such as glasses, masks, and make-up, can obscure faces, adding complexity to face-related tasks. Pose, expression and illumination changes further compound the difficulties in face processing (See Figure 1.3). The high variability in face appearances often leads to intraclass variations that surpass interclass variations, making face tasks intricate and requiring robust algorithms to handle these variations.

3. **The Uniqueness of the Face Domain:**
   The face domain's unique characteristics set it apart from the natural image domain. Faces are highly structured objects with distinct features, colors, and shapes. Consequently, standard vision algorithms cannot be blindly applied without suitable modifications. It becomes crucial to study the impact of these differences on vision algorithms, particularly on their internal representations. Understanding these nuances helps adapt existing methods or develop novel approaches better suited to handle face-related tasks.

4. **Ill-Posed Face Tasks:**
   Facial analysis tasks often involve subtle and complex nuances. Many of these tasks lack clear definitions, and the connection between facial features and their associated tasks can be ambiguous. For instance, facial expressions do not always correspond directly to emotions; a smile may not signify joy, while a frown can indicate a range of emotions such as surprise, anger, or disapproval. Specific attributes like criminal propensity or intelligence may not have any correlation with facial features. We need to be mindful of these complexities and dependencies to process or generate faces effectively. Moreover, the interplay between various face tasks, such as gender-specific expressions or age-related features, necessitates a holistic approach to face processing.

1.3 Scope and Areas of Interest

The scope of this thesis revolves around exploring the explainability of deep face representations, delving into the understanding of what these features represent and how they are filtered through deep models. We aim to identify the functional aspects of these representations and determine which parts of the input features contribute significantly to the final representations.

To achieve this, we investigate existing algorithms for explainability, particularly visualization and structural explainability methods, and assess their applicability to the face domain. We comprehensively demonstrate why many of these algorithms do not work effectively on face models, highlighting the need for face-specific explainability techniques. In response, we propose and present novel saliency algorithms tailored to work specifically on faces, allowing us to gain deeper insights into the workings of deep face models. Understanding the explainability of deep face representations has broader implications. It enables us to make meaningful observations and suggestions to improve current algorithms and identify potential issues when these models fail or behave unexpectedly. By comparing the output of our explainability algorithms to human cognition, we can establish benchmarks and guidelines for assessing the interpretability of deep face models in relation to human perception.

A key aspect of this thesis involves studying the gaze of drivers, which serves as a real-world case study for human attention. To achieve this, we introduce the DashGaze dataset, a large-scale and naturalistic collection of driver gaze data. This dataset allows us to analyze driver behavior under varying conditions, gaining valuable insights into their gaze patterns and attention allocation while driving. We develop the DashGazeNet, a novel appearance-based driver gaze estimation algorithm that relies solely

on the output of a dashcam. By leveraging this model, we can predict the driver's gaze on the road using the driver's face as input. In addition, driver gaze analysis facilitates the development of advanced driver assistance systems and furthers autonomous driving research. We introduce calibration procedures to adapt the model to different drivers and dashcam positions, making the gaze estimation process more accurate and reliable.

In summary, this thesis explores the explainability of deep face representations, particularly in terms of functional interpretations and significant contributions of input features. It encompasses the analysis and comparison of existing algorithms and the development of novel face-specific saliency algorithms. The study of driver gaze using the DashGaze dataset is an essential case study to understand human attention in real-world scenarios. Overall, the findings and insights from this research contribute to the advancement of explainability in deep face models and have broader implications for enhancing driver safety and understanding human perception in AI applications.

1.4 Contributions

The thesis has the following set of significant contributions:

1. **Survey of Explainability in the Face Domain**

   (a) We provide a comprehensive overview of explainability methods, focusing on visualization/structural explainability techniques applied to the face domain. We address the challenges of adapting general explainability methods to face models and offer insights into the workings of face models through these techniques.

   (b) We conduct the first survey of face-specific explainability literature, emphasizing the need for face-specific evaluation methods for explainability.

   (c) We identify factors to make face explainability methods more accessible to AI practitioners by conducting a user survey on the utility of different explainability algorithms.

2. **Functional Concepts in Deep Face Representations**

   (a) We introduce **Cross-Task Aware Filters (CRAFTS)**, convolutional filters in face models that learn to predict related face tasks.

   (b) Using CRAFTS, we propose **ETL**, an efficient task-based pruning and transfer learning procedure.

3. **Canonical Saliency Maps**

   (a) We present a method to standardize face saliency images and project them from image coordinates to face coordinates, yielding more insightful 'canonical heatmaps' that show the relevance of different facial parts to deep face tasks.

(b) We introduce two types of canonical heatmaps: (i) *Canonical Image Saliency* maps, highlighting significant facial areas in specific input images relevant to predictions; and (ii) *Canonical Model Saliency* maps, capturing global characteristics of an entire deep face model while making predictions across data points, facilitating network understanding and potential problem diagnosis.

(c) We explore deep face model workings for various face tasks with different architectures and illustrate how to interpret canonical maps, demonstrating their diagnostic utility by detecting biases arising from using a celebrity face dataset to train a deep gender classification network.

4. **Human Attention through Driver Gaze**.

(a) We introduce **DashGaze** – a novel, large-scale dataset for appearance-based gaze mapping on the road. This dataset is the largest of its kind collected under real driving conditions, providing both driver and road views.

(b) We analyze driver gaze on the road, providing valuable insights into driver behavior under different conditions and variations between drivers.

(c) We present **DashGazeNet** – a lightweight model predicting the driver's road gaze using the driver's face as input. Additionally, we present calibration procedures to adapt the model to different drivers and dashcam positions.

1.5 Organization of the Thesis

The rest of the thesis is organized into four parts.

Part I provides an overview of the thesis, including the contributions and background.

- **Chapter 2** provides a general background on the evolution of face algorithms.

Part II studies deep face representations by visualizing them in various ways.

- **Chapter 3** comprehensively explores existing techniques for visualizing faces in DNNs. Experimental results from popular face models are presented, revealing the visual representations generated during face image processing.
- **Chapter 4** employs feature inversion to uncover retained and discarded information at each layer. A comprehensive overview of feature inversion algorithms is provided, along with experimental results applied to the face domain, yielding valuable insights into various face models.

Part III explores the functional meanings of deep face representations.

- **Chapter 5** studies deep representations in terms of human-recognizable 'concepts' and proposes conceptual meanings for deep face representations.
- **Chapter 6** introduces 'Cross-Task Aware Filters' (CRAFTS) to discover task-based concepts in deep face representations, representing multiple face tasks.
- **Chapter 7** presents algorithms for functional pruning of deep face models.

Part IV uses saliency heatmaps to discover salient input features.

- **Chapter 8** provides an overview of saliency map algorithms, including experimental results applied to the face domain and the need for face-specific saliency algorithms.
- **Chapter 9** introduces Canonical Saliency Maps, a saliency algorithm for face models of any architecture. Extensions for zero-shot learning and face verification are presented.
- **Chapter 10** extends Canonical Saliency Maps to create model-level saliency maps. Extensive results and comparisons are shown, and biases in deep models are discovered using this algorithm.

Part V studies human attention in the context of driver gaze.

- **Chapter 11** introduces the DashGaze dataset, a large-scale, naturalistic driver gaze dataset, and analyses driver gaze under varying conditions using the dataset.
- **Chapter 12** introduces DashGazeNet, an appearance-based driver gaze estimation algorithm that utilizes dashcam output.

Part VI concludes the thesis and discusses future work.

Chapter 2

Background and Related Work

In this chapter, we discuss the background of face processing, explainability, as well as various concepts discussed in this thesis.

2.1 Face Processing

2.1.1 Classical Face Processing

The introduction of the historical Eigenface [8] technique, which relied on principal component analysis for identification, sparked the popularity of face recognition. Early face algorithms, such as PCA and Fisher faces, analysed the full available dataset of faces to generate a low-dimensional face code, then categorised using basic classifiers like nearest neighbour. However, when confronted with uncontrollable face changes, these early techniques failed. In the early days of computer vision in the face domain, datasets like FERET [9] were small and had minimal change across images. The frontal, grayscale photos were aligned and cropped around the faces. They did not account for differences in size, position, or illumination. Compared to the ones we are used to now, the number of images in these datasets was minimal.

Local feature-based techniques like Gabor [10] and local binary patterns [11] were presented in the early 2000s. These handcrafted features improved the performance of classifiers. With the invention of the Viola-Jones detector around this time, face detection gained a substantial boost. It gave competitive object detection performance in real-time. As the early face datasets' performance plateaued, newer datasets were released, each becoming more large and unconstrained. The LFW [2] dataset, which included face photos in unconstrained position and lighting, was released in 2007, marking a watershed moment in face identification. Though there was extensive research on face techniques initially, accuracy improved slowly. Most methods focus on one element of unconstrained facial changes, such as lighting, stance, expression, or disguise. There was no comprehensive strategy in place to address these issues. While these strategies improved their performance on face datasets, they failed in real-world situations.

2.1.2 Face in the Deep Learning Era

AlexNet won the ImageNet competition in 2012, ushering in the era of deep learning in computer vision. Like its predecessor, the artificial neural network, deep learning methods are made up of 'neurons' organised in numerous layers (thus the name 'deep') and linked with weights that may be modified to 'learn' a subject. These levels establish a concept hierarchy, beginning with simple lines and textures and progressing to portions of objects at higher levels. Deep neural networks extract discriminative characteristics and exhibit excellent translation and illumination invariance. A landmark achievement in computer vision in the face domain came when DeepFace [1] achieved the state-of-the-art accuracy on the LFW benchmark [2], approaching human performance on an unconstrained dataset for the first time (DeepFace: 97.35% vs Human: 97.53%), by training a 9-layer model on 4 million facial images. Inspired by this work, the research focus shifted to deep-learning-based approaches, and the accuracy dramatically increased to above 99.80% in just three years. Deep learning has reshaped the research landscape of face tasks in almost all aspects, such as algorithm designs, training/test data sets, application scenarios and even evaluation protocols.

2.1.2.1 Face Recognition

The improvement in the performance of face recognition algorithms mirrored the introduction of progressively more extensive, more complex and less constrained datasets. Previously, researchers trained face algorithms on private databases such as Facebook's Deepface [1] model, trained on 4M images of 4K people and Google's FaceNet [12], trained on 200M images of 3M people. However, the private nature of these databases makes it difficult to reproduce and compare. Many extensive public datasets were released to mitigate this problem, starting with Casia-WebFace [13], which consists of 0.5M images of 10K celebrities collected from the web. Some of the current most extensive databases are MS-Celeb-1M [14], VGGFace2 [15] and Megaface [16], each having over 1 million images.

Inspired by the extraordinary success on the ImageNet [17] challenge, many of the face algorithms use typical CNN architectures such as AlexNet [18], VGGNet [19], GoogleNet [20] and ResNet [21] as their base. Some face tasks such as facial expression recognition (FER) involving a time component also use time-sensitive networks such as RNNs. The performance improvements may be largely attributed to novel loss functions. Object recognition commonly uses softmax loss as the supervision signal in object recognition, as it encourages the separability of features. However, it is less effective on the face domain as the intraclass variations could be larger than inter-class differences. Researchers developed novel losses such as angular/cosine-margin-based losses to increase the separability between features.

2.1.2.2 Other Face Classification Tasks

Many challenging datasets are present for the face detection task. FDDB [22] and WIDER FACE [7] are large databases with face images in natural conditions and various scales. Finding Tiny Faces [20] was seminal work in this area, which addressed all these problems and achieved state-of-the-art on many

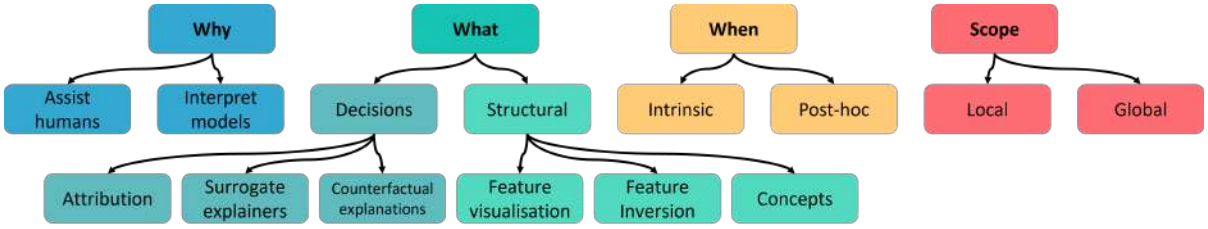


Figure 2.1: An overview of the organization of structural explainability works. We categorise explainability methods across four dimensions: *Why* describes what the end goal of applying the algorithm is, *What* describes the part of the model we are explaining, *When* describes when the explainability algorithm is applied: during or after training, *Scope* describes if the algorithm explains a single input or the entire model.

major datasets.. Their algorithm detected faces on multiple scales by creating a coarse image pyramid and feeding it into a CNN to predict template responses at every resolution, after which they applied non-maximal suppression.

Since the various tasks in the face domain are related, some works focus on learning the various tasks together using multi-task learning. Hyperface [23] is an influential work in this area. They present an algorithm for simultaneous face detection, landmarks localisation, pose estimation and gender recognition. It works by fusing the intermediate layers of a deep CNN using a separate CNN followed by a multi-task learning algorithm that operates on the fused features. The proposed models captured both global and local information in faces and increased each task's performance.

2.2 Explainability

2.2.1 Explainability for AI

As the literature on explainability methods is extensive, there are many ways to classify the algorithms [24–28]. As per Figure 2.1, we classify the methods along four dimensions: **Why**, **What**, **When** and **Scope**. The first dimension, "Why" explores the reason for applying the explanation algorithm. General explanation algorithms are usually applied to explain the working of a deep model. However, in the face domain, explanation algorithms are also used to assist humans in determining the correctness of a model's decisions or make better judgements in face tasks. "Scope" refers to whether the explainability algorithm explains the model for a few inputs (local) or the entire model at once (global).

"When" refers to which part of the training/deployment process the explainability algorithm is applied. Models with intrinsic explainability have interpretability baked into their structure. The network architecture, losses and training procedure need to be changed for intrinsic explainability. They are not model-agnostic. In deep models, intrinsic explainability generally means that each component/neuron of the model represents a single or limited type of well-defined object part. Enforcing this often leads to reduced performance.

On the other hand, post-hoc methods consider the model a black-box and are applied to pre-trained models. They are primarily model-agnostic. They are generally more flexible than intrinsic methods as they do not require modification to the training procedure. However, they tend to be approximate methods as they guess at the 'intention' of the models.

In this thesis, we are concerned with visual explanations primarily for convolutional neural networks, though some of these algorithms can also be applied to other types of models. Consider a feed-forward network consisting of multiple layers, each with several neurons. We pass an image through the input of the network. At each layer, the network transforms the input into its *internal representation* using the *learned weights and filters* of its neurons. Finally, the network produces a *decision* based on the transformed representations. Based on which components of the network we are explaining, methods can be divided into *decision analysis* and *structural interpretation*, which visualises the internal representations and learned weights.

Decision analysis helps us understand what factors go into the predictions made by a deep model and how they change when the input is perturbed. Attribution analysis determines which parts of an input highly influence the outcome. It is usually represented as a saliency heatmap computed for a single input instance or an entire model. Surrogate explainers try to reduce a deep model to a naturally interpretable AI model like decision tree, rule-based model or linear classifier. Counterfactual explanation of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output.

Structural visualisations give insights into the building blocks of the model (the weights and representations) and how they come together to make decisions. We discuss three types of structural visualisations in detail. *Feature visualisation* visualises the internal representations of the different components of a deep model projected into the input domain. They reveal which input patterns strongly activate the components. The overall function of the model can be thought of as a combination of these representations. As the input to a deep model passes through several layers, the model discards superfluous information and keeps only discriminative information. *Feature inversion* projects the internal representations at a layer back to the input domain. This visualisation helps us understand the power of internal representations and how much information is kept or discarded at each layer. The above two types of visualisation often produce vague or unclear results which are hard to interpret. Often, we cannot glean concrete insights into the model unless we are experts in using the visualisation, as the hierarchy of the models does not correspond to the human abstraction of concepts. Several works attempt to align the internal features and representations with human abstractions and constructs. We have grouped these works under the title *'Alignment of Features and Concepts'*.

2.2.2 Discovering Functional Concepts in Deep Representations

Previous works have demonstrated that neural networks can learn auxiliary information beyond their intended task. Long et al. [29] explored the use of intermediate features from networks trained on object detection for image correspondence. Zhou et al. [30] investigated mid-level representations of a CNN

trained on scenes, revealing their association with objects. Donahue et al. [31] showcased the versatility of features trained on image recognition, successfully applying them to tasks such as object recognition, domain adaptation, subcategory recognition, and scene recognition across different datasets. Upchurch et al. [32] utilized final layer features from a face network to manipulate facial attributes. While these works have made significant strides in understanding and visualizing what pre-trained networks have learned for specific tasks, they do not shed light on the network's functionality for related tasks. In Part III, we aim to bridge this gap by examining the auxiliary tasks learned by a network due to the face recognition task, and exploring the extent of their relationship to face recognition.

Other efforts have focused on interpreting the functionality of individual neurons or groups of neurons. Raghu et al. [33] proposed a method to determine the true dimensionality of a layer, revealing it to be much smaller than the number of neurons. Morcos et al. [34] investigated the effect of single neurons on generalization performance by selectively removing neurons from a neural network. Alian and Bengio [35] developed intuition about trained models by employing linear classifiers that utilize hidden units as discriminative features. Bau et al. [36] introduced a method to quantify the interpretability of latent representations in CNNs by evaluating the alignment between individual hidden units and a set of semantic concepts. While these methods primarily focus on interpreting the latent features of deep networks trained on a single task, our work delves into the analysis of latent features containing information about external tasks not encountered during training and explores how these features can be repurposed for such tasks.

2.2.3 Human Cognition of Faces

Extensive research has been conducted on how humans recognize faces. Psychological studies suggest the existence of a conceptual "face space" that aids in face identification. A 2011 study reviewed evidence indicating that the brain uses multiple perceptual norms shaped by visual experience to extract face identity [37]. By encoding faces relative to stored norms, the visual system focuses on what is unique to each individual, enabling discrimination among thousands of faces despite their overall similarity and changes due to aging or health. This "face space" is a multidimensional psychological construct where each face is represented by a location, and the dimensions capture perceived properties of faces. These dimensions may correspond to specific parameters (e.g., head height, face width, or eye distance) or more abstract properties like age or masculinity [38].

Support for a unifying "face space" in primates was provided by a macaque study that recorded neural responses to parametrized faces [39]. The results showed that individual cells in the macaque brain are tuned to specific axes of variation within the face space, while being insensitive to changes along orthogonal axes. Approximately 200 face cells were found sufficient to reconstruct facial images. This challenges the earlier assumption that face cells encode specific facial identities and demonstrates that drastically different appearances can elicit identical responses in single face cells.

A separate 2011 study demonstrated that human face recognition is holistic, relying on the entire face rather than isolated features [40]. It also showed that holistic processing strongly predicts face-

recognition abilities in humans. However, other studies show that certain facial features are critical for recognition, as evidenced by humans' superior performance compared to machines in identifying low-resolution or degraded faces [41]. Familiar faces are recognized more accurately than unfamiliar ones, and specific features, such as the eyebrows, are particularly important cues.

These findings align with our own analysis. Comparing trained network saliency maps to human cognition can reveal instances where the networks rely on incorrect cues for classification. For example, our investigations into gender and age (detailed in Chapter 10) corroborate prior studies on the importance of features like eyes and lips for gender, and eye and mouth corners for age [42–45]. Additionally, our analysis highlights novel insights, such as the significance of eye corners for gender classification due to makeup. Furthermore, we provide a systematic methodology for examining such relationships, enriching the understanding of human and machine face recognition.

2.3 Efficient Deep Learning

In Chapter 7, we create efficient and lightweight models using less data by functionally pruning deep face models. In this section, we discuss works related to lightweight models and transfer learning.

2.3.1 Lightweight Models

While achieving impressive performance, deep learning models often suffer from a high parameter count, leading to energy inefficiency and challenges in deploying them on resource-constrained devices. As a result, researchers have extensively explored various architectures for lightweight convolution models that offer faster training with minimal performance degradation. For instance, GoogLeNet [20] introduced inception modules, which reduce the number of channels in expensive 3x3 convolutions. Building upon this, Xception [46] and MobileNet [47] further improved efficiency by employing completely depthwise separable and sparse 3x3 convolutions. SqueezeNet [48] reduced parameters by downsampling late in the network, ensuring convolution layers have more significant activations. Additionally, models like ResNeXt [49], ShuffleNet [50], Light-CNN [51], MobiFace [52], and SlimCNN [53] proposed dedicated lightweight CNN architectures designed explicitly for face tasks. Another approach to reducing model size is quantized networks [54–58], where extremely low precision is used, replacing arithmetic operations with bitwise operations to reduce memory and power consumption significantly.

Another strategy for model size reduction involves pruning or knowledge distillation. Pruning entails removing connections from a complete network based on a ranking criterion, resulting in a sparse network with comparable performance to the original network. Various studies [59–64] have explored different criteria for ranking convolutional filters and iteratively pruned the bottom k% of filters. Notably, He et al. [65] proposed a lasso regression-based method combined with least-square reconstruction for iterative filter selection. In contrast, our approach utilizes lasso regression to select filters in a single pass. Some works [66, 67] adopted one-shot pruning methods, but they operated at the neuron resolution. A closely related work to transfer learning using pruning is presented by Molchanov et al. [68],

which introduces a criterion based on Taylor expansion to approximate the change in the cost function resulting from pruning network parameters. Additionally, several works have explored pruning techniques through the lens of the "Lottery Ticket Hypothesis" [69], uncovering sub-networks capable of effective training due to their initialization.

On the other hand, knowledge distillation involves training a smaller "student" model by transferring knowledge from a larger pre-trained "teacher" model. Rather than relying on ground truth labels, the student model is trained on the output distribution of the teacher model. Noteworthy works [70–72] have demonstrated impressive performance using knowledge distillation on face tasks such as recognition, detection, and age estimation.

2.3.2 Transfer Learning

Traditional approaches to transfer learning [73–79] involve finetuning a model trained on a base task using a target dataset or task. Extensive studies have been conducted to explore optimal transfer learning policies and practices, employing large-scale experiments across various tasks. For instance, Taskonomy [80] uses a computational approach to recommend the most suitable transfer learning policy for a given set of source and target tasks, while also uncovering structural relationships between vision tasks. Yosinski et al. [81] provide valuable recommendations for effective transfer learning, quantifying the general or specific nature of each layer's features and measuring the "distance" between different tasks through computational analysis.

In Chapter 6, we predict different face attributes using features extracted from a pre-trained face network to study various auxiliary tasks, such as head pose, age, and emotion. This approach has been previously employed by Zhong et al. [82], where networks trained for face recognition were utilized to extract features from specific layers. Subsequently, a binary linear SVM was trained on these features to predict various facial attributes. However, their focus primarily lies in developing state-of-the-art facial attribute predictors rather than comprehending the underlying functionality of face networks.

2.3.3 Efficient Transfer Learning

Although existing approaches in Section 2.3.1 address storage efficiency, computational complexity, and power consumption issues, they are not specifically tailored for task transfer. Recent research in the field of Natural Language Processing (NLP) [83–85] has focused on efficient incremental learning, where the addition of a few neurons per task mitigates catastrophic forgetting, enabling efficient models to achieve performance comparable to separate complete networks for new tasks. Another approach by [86] utilizes knowledge distillation to transfer knowledge from face recognition to non-classification tasks such as alignment and verification by selecting appropriate initializations and targets. Additionally, [68] presents a closely related work combining pruning and transfer learning. Their iterative approach involves alternating between finetuning and pruning until achieving the desired accuracy objective versus compression. However, this iterative process is relatively slow compared to our approach presented in Chapter 7, which accomplishes model transfer in a single step.

2.4 Gaze Mapping

In Part V, we gain insights into human cognition through the lens of driver gaze. In this section, we discuss works related to driver gaze estimation. Gaze datasets pose unique challenges due to the difficulty of annotating eye gaze direction based solely on visual cues. To address this, various approaches, such as using eye trackers or predetermined gaze targets, have been employed to create gaze datasets. We first discuss existing datasets that focus on gaze estimation in general, which are typically compiled using eye trackers or predefined targets. Then, we delve into driver gaze datasets, which specifically aim to study driver behaviors, activities, and attention. These datasets are crucial in understanding driver gaze patterns and developing gaze prediction models for various in-car applications

2.4.1 Gaze Datasets

The task of annotating eye gaze direction based solely on visual cues poses challenges for human annotators. To address this issue, most available gaze datasets are typically compiled using eye-trackers [87–89] or by instructing subjects to focus on predetermined gaze targets [90–92]. These targets may be points on a screen [92–95], or objects in 3D space [90, 91, 96]. NVGaze [97] was collected by asking users to look at targets in a VR headset. The use of eye trackers in gaze data collection has the advantage of providing unconstrained behavioral data. On the other hand, object targets allow the face to remain unobstructed by the eye tracker, facilitating the training of appearance-based gaze models. Rt-GENE [87] combines the benefits of both methods by collecting data using an eye tracker and subsequently inpainting the eye tracker.

2.4.2 Driver Gaze Datasets.

Several datasets have also been collected to particularly study driver behaviors, activities, and attention [98–103]. Our focus is on datasets designed for gaze estimation. Few large-scale datasets concentrate on the task of "gaze zone prediction," [98, 104–106] where the interior of the car is divided into fixed "zones," such as the "rearview mirror" or "dashboard," and the goal is to predict which zone the driver's gaze falls on at any given time. As point-annotation of driver gaze is challenging, some datasets have used markers inside or outside the car [99], while others have been collected using simulators [107, 108] or parked cars [99, 104]. However, these methods do not capture realistic driving behavior. The Dr(eye)ve [109] and LBW [110] datasets address this issue by using eye trackers to collect true behavioral data while the driver is naturally driving on roads.

2.4.3 Driver Gaze Mapping

Methods for driver gaze estimation have been proposed in earlier literature focusing on the task of driver gaze zone estimation, where the driver's gaze is classified based on which specific 'gaze zone' it belongs to [111–113]. In some studies, depth information from RGB-D cameras has been utilized effec-

tively for accurate gaze zone estimation [114, 115]. To address the issue of identity bias, Yu et al. [116] proposed a multimodal approach that utilized geometric features. Context-aware methods that utilize cues from the driver's environment and face for gaze zone prediction have also been proposed. For instance, Stappen et al. [117] introduced 'X-AWARE', a context-aware approach. Additionally, Yuan et al. [118] used a domain prior of typical gaze patterns to auto-calibrate gaze zone estimation. In contrast to gaze zone estimation, some works aim to predict the continuous driver gaze [119–122]. Probabilistic models for gaze estimation using the driver's head pose have also been proposed by Shirpour et al. [123] and Jha et al. [124].

In recent years, a few methods have also been proposed for detecting driver gaze using multiple sensors. Some of these sensors include infra-red cameras [125], RGB-D cameras [114], and dual cameras [122]. Lemley et al. [126] proposed a low-cost solution for driver gaze estimation using noisy cameras. Sonom-Ochir et al. [127] proposed a dashcam-based approach that utilizes an SVM classifier on features extracted from Convolutional Neural Networks (CNNs). To address the challenge of occlusion due to eyeglasses, Rangesh et al. [128] used IR cameras and a gaze-preserving cycle Generative Adversarial Network (GAN) to remove eyeglasses and estimate the driver's gaze accurately.

PART II

Visual Exploration of Deep Face Networks

Chapter 3

Feature Visualization

Human vision possesses remarkable abilities in analyzing scenes by decomposing them into meaningful components, such as objects and their parts. In the case of faces, this decomposition involves identifying facial features like eyes, nose, and mouth. When examining deep neural network (DNN)-based models for face processing, a natural inquiry arises regarding the correspondence between the hierarchical patterns learned by the convolutional layers' filters and human intuition. Specifically, how do the learned patterns differ between processing natural images and faces? Additionally, do these visualizations exhibit discrepancies across face-related tasks, such as recognition and pose estimation?

This chapter examines face visualization methods to enhance our understanding of well-known face models when applied to face images. We provide a comprehensive background on existing techniques for visualizing faces in DNNs. Then, we present experimental results obtained from popular face models, revealing the visual representations generated by these models when processing face images. Our analysis aims to uncover the underlying mechanisms and learned patterns in DNN-based facial analysis, contributing to developing more interpretable and efficient face processing models.

3.1 Introduction

Convolutional neural networks (CNNs) have achieved remarkable performance by learning to extract optimal features directly from the input domain. A CNN typically comprises multiple layers of convolutional filter banks sequentially applied to the input image. Each filter aims to detect a specific pattern, responding with high activation if the pattern is present in the output of the preceding layer. While it is feasible to visualize the filters of the initial layers as they directly operate on the input image, visualizing the patterns learned by higher layer filters is non-trivial, as these filters operate on the outputs of preceding layers.

Definition 3.1.1 (Feature Visualization) *Feature visualization is the process of projecting CNN filters onto the input domain to generate visual representations of these filters.*

In addition to unraveling the hierarchy of learned filters, feature visualization methods are crucial in addressing several pertinent questions, such as: Are there redundant features or layers? Does the DNN

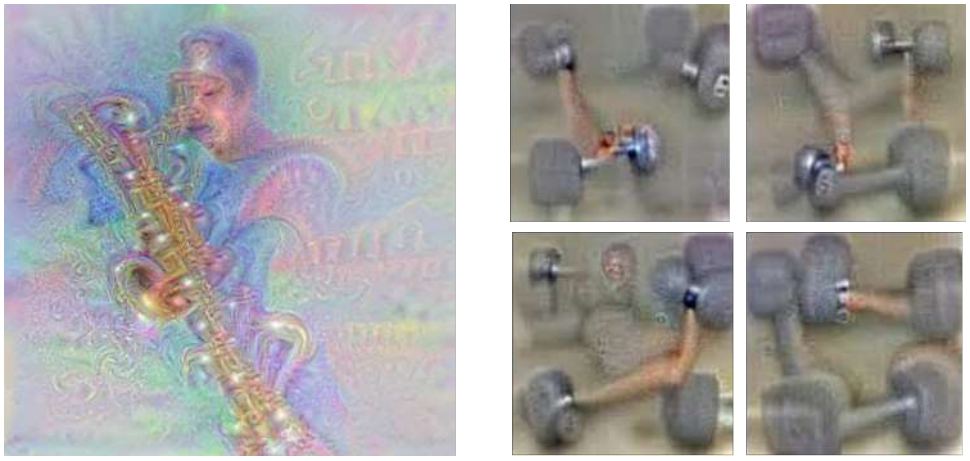


Figure 3.1: Visualizing output classes of deep models sometimes shows spurious learned correlations. *LEFT* Visualizing the GoogLeNet class 'saxophone' also shows a saxophone player. (Image taken from [129]) *RIGHT* Four visualizations of the class 'dumbbell' also shows an arm.(Image taken from [130])

learn biases or incorrect associations from the dataset? A notable example highlighting the potential of feature visualization is demonstrated in [129], where visualizations of the 'saxophone' class revealed the presence of the musician holding the instrument alongside the saxophone itself (refer to Figure 3.1). Such observations underscore the capability of feature visualization techniques in uncovering entanglements and correlations between contextual information and the content being analyzed.

3.1.1 Desiderata for Feature Visualization

We propose three essential properties for effective feature visualization, specifically in the context of face images.

Proposition 3.1.1 (Desirable properties of feature visualization algorithms)

1. *The visualization should be* interpretable in-domain.
2. *The visualization should not have* undue influence from a limited dataset
3. *The algorithm should* show all facets of the unit under study.

Let $\mathcal{D}$ represent the domain of images 'expected' as input by a deep network or its constituent parts. For example, in object detection, $\mathcal{D}$ may consist of natural images, whereas in face recognition, $\mathcal{D}$ includes natural images specifically depicting faces and their components, excluding generated or non-face images. Furthermore, let $\mathcal{T} \subseteq \mathcal{D}$ denote the training set used for training the network, while $\mathbb{R}^{W \times L} - \mathcal{D}$ denotes the collection of out-of-domain and adversarial images [164].

Ideally, our visualization should be situated within the domain $\mathcal{D}$ without being limited to the training set $\mathcal{T}$. Furthermore, it should effectively capture the diverse patterns that the unit within the network

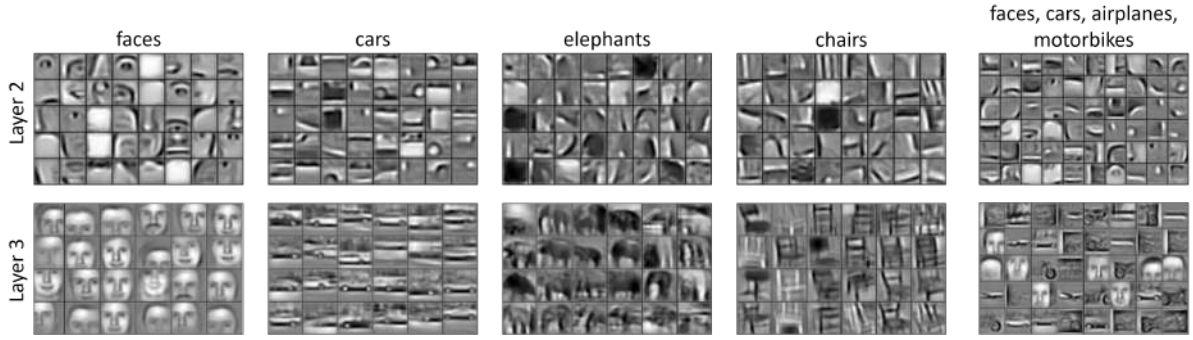


Figure 3.2: Visualization of the second and third layers of deep belief networks trained on specific object categories. Column 5 is trained on a mixture of four object categories. This early visualization demonstrates feature hierarchies in natural object categories. (Image taken from Erhan et al [132])

aims to identify. By adhering to these properties, our proposed visualization technique aims to provide valuable insights into the inner workings of the network, specifically within the context of face analysis tasks.

3.1.2 Summary of Early Feature Visualization Techniques

In earlier studies, the visualization of higher-layer features involved representing them as linear combinations of lower-layer filters [131]. Figure 3.2 shows the layers of a deep belief network visualized as linear combinations of its previous layers. The visualization highlights the hierarchies present in various object categories.

However, this approach is primarily practical for the first layer of a CNN, as it does not account for the non-linearities present within the network. To address this limitation, a more flexible formulation considers feature visualization as a search for input images that elicit a high response from a specific neural unit. Erhan et al. introduced the concept of activation maximization [132], which employs gradient ascent to optimize an input image such that the activation or response of a neural unit is maximized. This method enables more comprehensive visualization of higher-layer features, accommodating the non-linear dynamics of the network. We will discuss activation maximization in detail in Section 3.2.

3.2 Activation Maximization

Activation maximization serves as the prevailing framework for feature visualization, with recent studies focusing on addressing the limitations of the original formulation through the exploration of various regularization terms. Here is a generalized expression for feature visualization:

Definition 3.2.1 (Activation Maximization)

$$x^* = \arg\max_{x \in X} \phi(x) - \lambda R \tag{3.1}$$

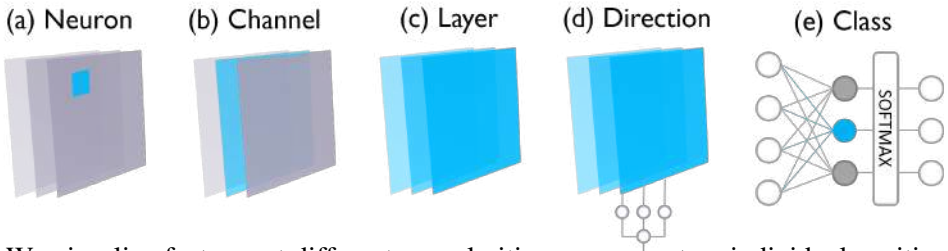


Figure 3.3: We visualize features at different granularities: neurons at an individual position, an entire channel, linear combinations of channels, or an entire layer. We can also visualize classes (inspired by [133]).

Here, X denotes the search space of the input, $\phi(x)$ represents the activation of the neural unit under investigation, and R is a regularization term that accounts for the quality or diversity of the visualization.

In practice, the search space X commonly takes the form of $\mathbb{R}^{W \times L \times C}$ or crops from the dataset images $\mathcal{T}$ [134].

The objective function in Equation (3.1) can be maximized effectively by applying gradient ascent. By computing the gradient of the activation with respect to the image and iteratively adjusting the image in the direction of increasing activation, activation maximization provides a flexible and informative means of dissecting neural networks. This framework enables the visualization of combinations of neurons by modifying the objective function accordingly. The critical advantage of activation maximization lies in its ability to generate visualizations that align with the preferences of the targeted neuron, unconstrained by the images present in the dataset. However, a notable drawback is that the resulting visualizations often exhibit high-frequency details and lack meaningful interpretations. To mitigate these issues, numerous research efforts have explored regularization methods to produce high-quality visualizations.

Another limitation of activation maximization is its failure to capture all aspects of a neuron. A neuron may exhibit sensitivity to multiple types of output [133]. To address this challenge, some approaches introduce a "diversity term" within the objective function. In certain instances, activation maximization is employed with an image dictionary, effectively mitigating the drawbacks of both approaches. By combining these methods, researchers aim to overcome the deficiencies of activation maximization and achieve improved visualizations with greater comprehensiveness and meaningfulness.

3.2.1 Granularity of Feature Visualization

The neural unit under investigation, denoted as ϕ, can be defined as a specific channel at a particular position on the input image or as an entire channel [135]. Additionally, it is possible to seek inputs that maximize the activation of any neuron within a layer, as depicted in Figure 3.3(c) [130, 136]. Such visualizations provide insights into the patterns that each convolutional filter can detect within an input image. Notably, the algorithm can select any channel from the layers and optimize its activation at every

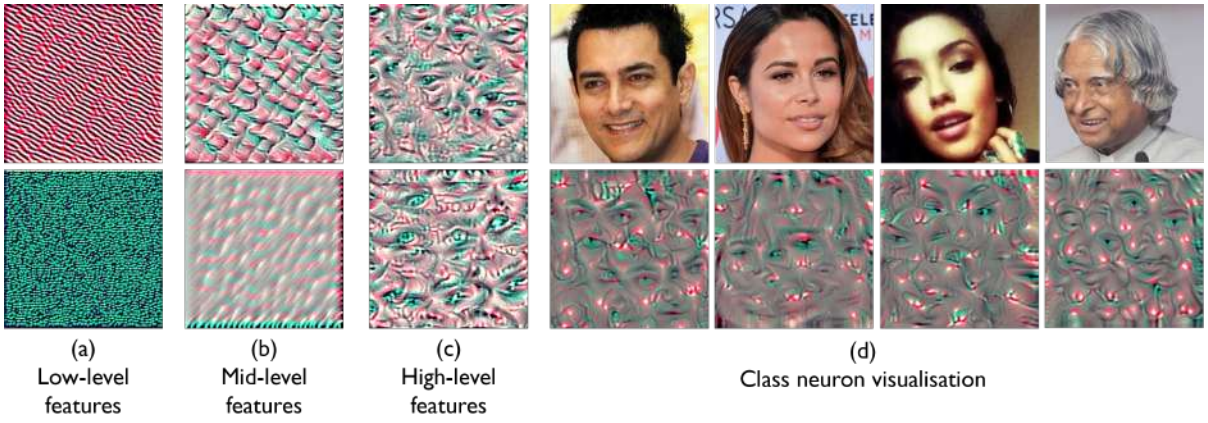


Figure 3.4: Overview of feature representations of the VGG-Face network [19]. (a) Early conv layers (1 to 6) show simple patterns. (b) Middle-layer patterns (7 to 9) show more complex patterns. (c) Later convolutional layers (10 to 13) show facial features and parts of the face. (d) Class neurons capture facial identities. These visualizations were created using the activation maximization algorithm with L-2 regularization.

position within the image, resulting in dream-like visualizations where prominent features are accentuated. The concept of channel visualizations stems from the assumption that individual channels serve as a distinct basis, enabling the extraction of semantic information, which proves particularly valuable (refer to Chapter 5). By jointly optimizing multiple channels [133], we gain insights into the interplay between neurons. Furthermore, visualizing one of the output neurons (before the softmax operation) is commonly referred to as 'class visualization' (Figure 3.3(e)), aiming to identify a representative image for an entire class.

3.3 Results of Visualizing Deep Face Features

This section presents the results of feature visualization applied to deep face models targeting three crucial face tasks: face recognition, head pose recognition and emotion recognition. The investigated models utilize the VGG-16 architecture, which encompasses 13 convolutional layers followed by three fully connected layers. Through our extensive analysis, we unveil the hierarchical characteristics of facial features and elucidate the distinctions among the feature representations employed in different face tasks.

3.3.1 Facial Feature Hierarchy

Convolutional networks learn features that exhibit a hierarchical structure, with elements repeated in the input image and a progression from simple to complex patterns. In this study, we employed activation maximization to visualize different layers of the VGG-Face model [19]. We comprehensively summarize the observed features in Figure 3.4. Our findings demonstrate that deep face models ac-

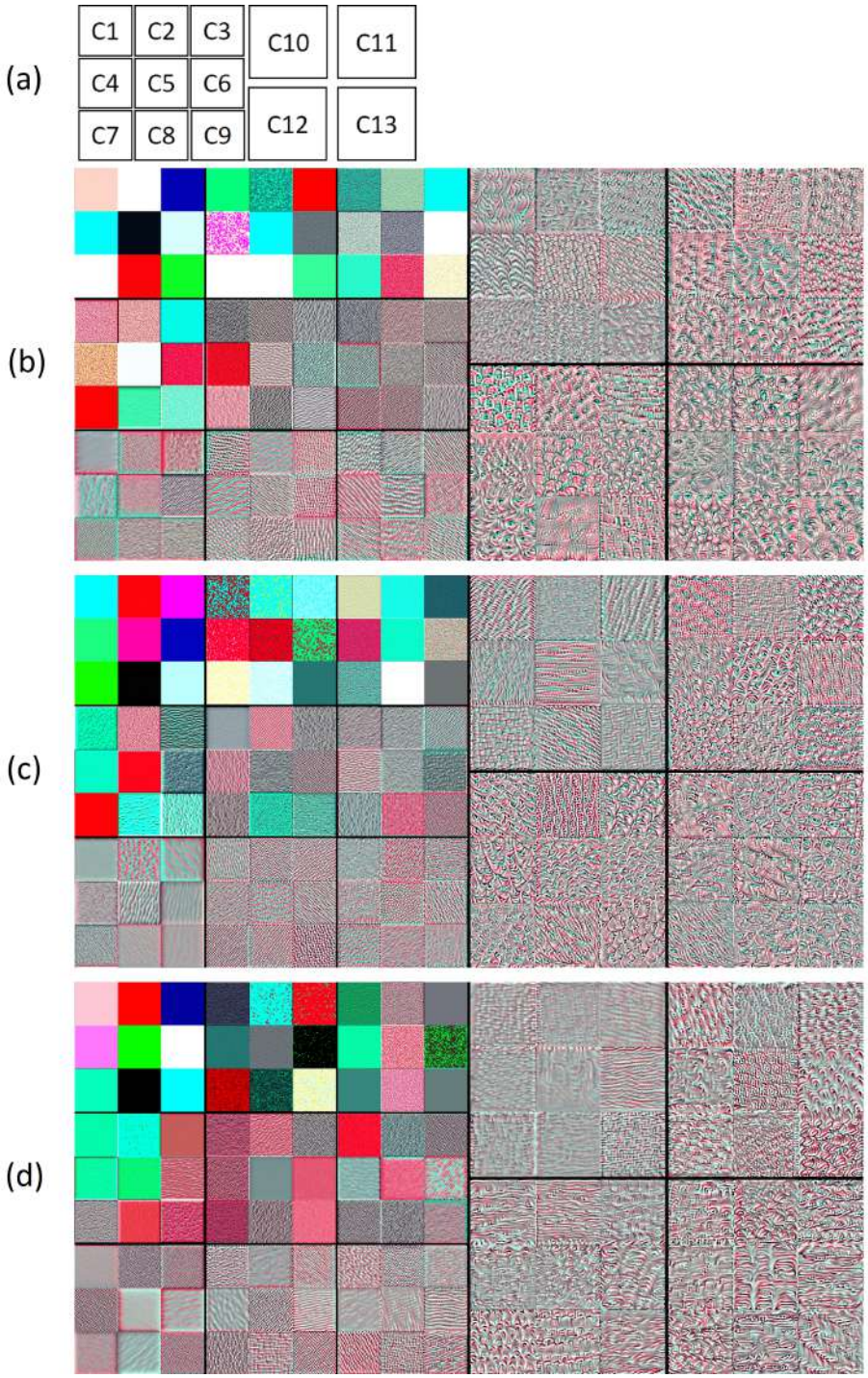


Figure 3.5: Hierarchical Visualization of VGG-16 Networks: Face Recognition, Head Pose Estimation, and Emotion Recognition. Nine randomly selected filters from each layer are depicted, providing insights into the distinct characteristics learned by these models. (a) Order of layers (b) Face recognition model (c) Head pose model (d) Emotion recognition model (Best seen electronically. Zoom in to see the details)

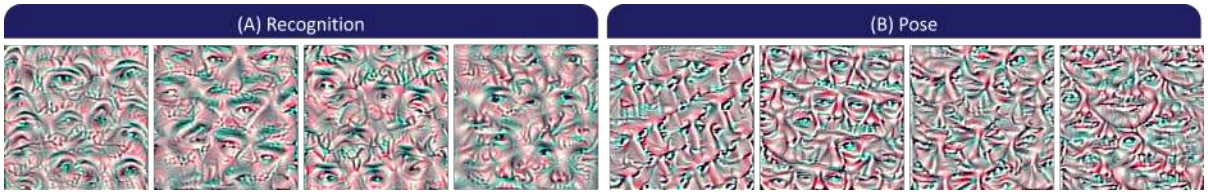


Figure 3.6: Activation patterns of last convolution layer filters in face recognition and head pose networks. *LEFT* Sample patterns that activate filters in the last convolution layer of the VGG-Face recognition network [19], displaying distinct shapes of eyes and nose, potentially representing different identities. *RIGHT* Activation patterns of filters in a head pose VGG-16 network, revealing variations in feature poses rather than their shapes.

quire hierarchical features from face images. The initial layers capture elementary patterns like those observed in object classification networks, while higher layers discern intricate patterns composed of these elemental components. Intermediate layers specialize in facial features such as eyes and nose, while neurons in the fully connected layers learn specific facial parts. Interestingly, the neurons in the last layer develop representations of the whole face. This observation aligns with the work of Zhong and Deng [137], who constructed a feature visualization dictionary using Deconvnet [134] to highlight the hierarchy within facial features.

Figure 3.5 showcases the complete hierarchy of features for three VGG-16 models trained for face recognition, head pose estimation, and emotion recognition, respectively. Remarkably, the level of detail per layer remains consistent across all three models: the initial four layers predominantly capture simple colors and patterns, the subsequent five layers discern complex patterns, and the final four layers reveal recognizable facial features.

3.3.2 Visualizing Features of Diverse Face Tasks

Figure 3.6 provides evidence that face models of distinct tasks acquire different features. Within the higher layers of a face recognition network, multiple filters specialize in capturing patterns associated with various eye and nose shapes, as these traits contribute significantly to differentiating identities. Conversely, although feature shapes exhibit limited variation in the higher layers of a head pose network, they encompass facial components in different poses. Our comprehensive investigation consistently reveals that visualizations of face models emphasize the interplay between *part geometry* and *task objective*, two complementary factors crucial for understanding facial representations.

3.3.3 Class Visualization

The final layer of a discriminative network contains enough information to produce highly detailed representational images of classes. Although these images may differ from a human's conception of the class, the discriminative features specific to each class are prominently highlighted. However, there are instances where the visualizations include details that are not strictly representative of the class.

For example, as shown in Figure 3.1, the 'saxophone' class visualization from Imagenet shows a vague saxophone player also [129]. Similarly, the visualization of the 'dumbbell' class included forearms [130], indicating potential dataset biases. These observations suggest that certain parts of objects can occur multiple times in seemingly random locations within the visualizations. Notably, when we feed images with the repetitions removed into the model, the confidence of the discriminator decreases, indicating a focus on discriminative parts rather than the global structure of the object [138].

Figure 3.4(e) showcases the class visualizations of various identities from the VGG-Face network. The top row presents facial images of different classes, while the bottom row displays the corresponding class visualizations. It is evident that the visualizations do not capture the global structure of the face, and multiple instances of eyes, nose, and other facial features appear at different locations within the images. Despite the lack of preserved facial structure, the visualizations retain local identifying characteristics such as the shape of eyes, eyebrows, nose, and the distance between the eyes. These discriminative features represent the learned characteristics of the classes by the VGG-Face network. In contrast to object class visualizations, visualizations of face classes typically lack spurious correlations, as face models are usually trained on isolated face images obtained by cropping with a face detector.

3.4 Regularization of Activation Maximization

Framing feature visualization as an optimization problem to maximize the activation of a unit ('activation maximization') is powerful and versatile. However, this approach poses several challenges when directly optimizing an image using gradient ascent. The search for an optimal image within the $\mathbb{R}^{W \times L}$ space often becomes trapped in locally optimal solutions filled with adversarial noise and extraneous high-frequency details, mainly when regularization terms are absent from the objective function. To address these limitations, the formulation of activation maximization incorporates a regularization term (refer to Equation 3.1) that guides visualizations towards the domain $\mathcal{D}$ of the neural network. In this section, we delve into two key applications of regularization: clearer visualization and enhancing visualization diversity.

3.4.1 Regularization for Better Visualization

Regularization methods incorporating a "natural image prior" into the objective function have become a prominent focus in recent feature visualization techniques, ensuring that the generated visualizations fall within the dataset space $\mathcal{D}$. Figure 3.7 illustrates the spectrum of regularization functions, ranging from strong to weak. These methods can be categorized into dataset-free approaches and dataset-based approaches.

Unconstrained optimization often leads to visualizations with high-frequency details. This phenomenon can be attributed to strided convolutions and pooling operations, which introduce high-frequency patterns in the gradient [142]. Dataset-free regularizations attempt to mimic the statistical properties of natural images by penalizing high frequencies or adding robustness to transformations. For instance,

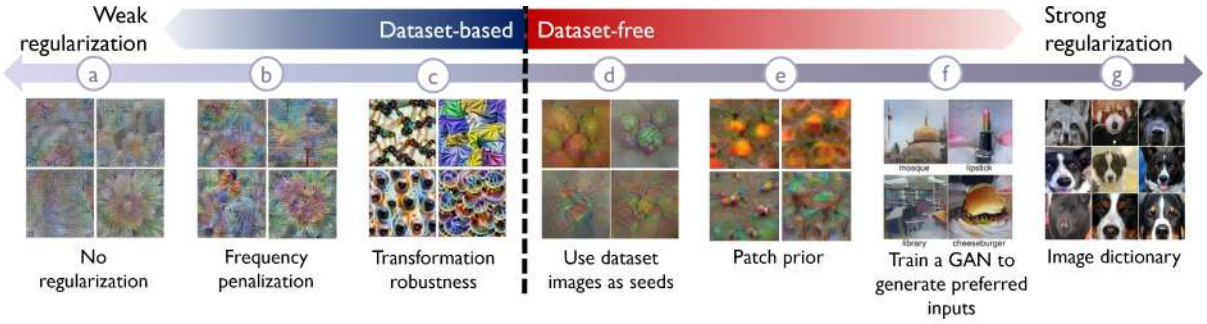


Figure 3.7: Spectrum of regularizations for feature visualization, depicting two categories of methods. *LEFT* Data-independent approaches *RIGHT* Dataset-informed methods. Image sources: (a), (b), (c) and (g) are from [133]; (d) [139] (e) [140] (f) [141]

Figure 3.8: Visualization of a single neuron highlighting its multiple facets through two techniques: incorporating a diversity term [133] (top) and initializing the visualization with diverse cluster means [139] (bottom). The right-most column showcases sample images from the dataset that activate the neuron. (Please zoom in for detailed observation.)

Simonyan et al. [135] employed activation maximization with an L2-regularization term to generate representative class visualizations. By maximizing the unnormalized class scores before softmax, they effectively minimize the scores of other classes. Gaussian blur and bilateral filters have also been utilized as frequency penalization techniques [129, 133, 143, 144].

Incorporating transformation robustness with frequency penalization has shown promising results in generating visually appealing and distinct visualizations [129, 133]. Transformation robustness aims to find examples that maintain high activation for the optimization target even under slight transformations [133], such as jittering, upsampling, or rotation. This approach is commonly employed alongside frequency penalization in feature visualization. Notably, the specific choice of regularization and its implementation details can significantly impact the resulting visualizations, as extensively examined by Olah et al. [133].

Dataset-based regularizations generally yield higher-quality visualizations but may exclude certain regions within the space $\mathcal{D} - \mathcal{T}$. One straightforward method involves selecting image parts from the training dataset with the highest filter response [133, 134, 140]. Nguyen et al. [141] trained a Generative Adversarial Network (GAN) on the input dataset and explored the GAN latent space to identify suitable images. Another approach involves seeding the visualization process with dataset images [139] or patches extracted from dataset images [140].

3.4.2 Adding Diversity to Visualizations

The regularization functions described above primarily aim to enhance the visual quality of the generated images. However, another crucial aspect of feature visualization is capturing the diversity of images produced by the algorithms. Convolutional filters often respond to multiple types of images, and studying only a single facet of a filter may not provide a comprehensive understanding.

Certain regularization functions have been developed to increase the diversity among the visualizations generated by the algorithms. Nguyen et al. proposed a data-centric approach to enhance diversity in visualizations [139]. Their method involves clustering images of a specific class and determining the mean image for each cluster. These mean images are then utilized as initial inputs for the activation maximization process, guiding the algorithm to focus on specific facets of the convolutional filter rather than producing a mixture of all facets. However, this approach restricts the exploration of how filters react to images beyond those in the dataset.

Alternatively, Olah et al. introduced a "diversity term" inspired by neural style transfer [136] into the objective function to foster diversity among the generated visualizations [133]. They achieved this by computing the negative pairwise cosine similarity of the gram matrix of channel responses from previous visualizations.

$$G_{i,j} = \sum_{x,y} \text{layer}_n[\text{x, y, i}] \cdot \text{layer}_n[\text{x, y, j}] \tag{3.2}$$

Here, $G_{i,j}$ is the dot product between the (flattened) response of filter i and filter j. The diversity term was formulated as the negative pairwise cosine similarity of these pairs of visualizations.

$$C_{\text{diversity}} = -\sum_{a}\sum_{b \neq a} \frac{\text{vec}(G_a) \cdot \text{vec}(G_b)}{\|\text{vec}(G_a)\| \ \ \|\text{vec}(G_b)\|} \tag{3.3}$$

Although this simple approach resulted in visualizations with varying degrees of diversity depending on the target layer, it is essential to exercise caution as the diversity term may unintentionally introduce artifacts by pushing the visualizations apart from each other.

3.5 Summary

This chapter explored the visualization methods employed to gain insights into face models when applied to face images. A comprehensive overview of existing techniques for visualizing faces in deep neural networks was presented. Subsequently, experimental findings obtained from popular face models shed light on the representations generated during the processing of face images.

The results obtained through our analysis highlight several key observations. Firstly, we observed a hierarchical organization of face features, wherein lower levels capture simple patterns, followed by complex patterns, facial features, and ultimately the entire face. This hierarchy underscores the progressive complexity and abstraction of the learned representations within these models. Our investigations also revealed that the levels of detail exhibited by the features across different face tasks were remarkably similar. For face recognition and head pose estimation, the models demonstrated consistent patterns in capturing discriminative features specific to the respective tasks. For face recognition, these features primarily revolved around the shape of the nose and eyes, while head pose recognition emphasized variations in facial features across different poses.

By uncovering the underlying mechanisms and learned patterns in deep neural network-based facial analysis, our analysis contributes to developing more interpretable and efficient face processing models. These findings provide valuable insights into the inner workings of face models and further our understanding of how these models extract and represent facial information.

Building on the insights gained from feature visualization, the next chapter transitions to *feature inversion*, a complementary approach that focuses on reconstructing input images from the learned representations. While feature visualization highlights the foundational patterns a model has learned, feature inversion reveals how the model interprets and reconstructs an input image based on these representations. This allows for a deeper understanding of the relationship between encoded features and the visual cues that drive the model's predictions.

Chapter 4

Feature Inversion

In the previous section, we explored the remarkable insights offered by feature visualization, enabling us to unveil the underlying ”bases” of a convolutional neural network. Feature inversion emerges as a closely related approach, aiming to project the deep features corresponding to an input image back to the input domain. This methodology allows us to observe the input image through the same ”lens” as the deep network.

This chapter delves into the intriguing realm of feature inversion, investigating its motivation, formulation, and implications. By undertaking a comprehensive analysis, we aim to deepen our understanding of this powerful technique and its relevance in unraveling the inner workings of deep networks. We present the results of applying feature inversion to deep-face models. We gain unique insights into how different layers of a face model perceive and interpret the input image. Through this investigation, we seek to unravel the visual hierarchies and representations within face models, shedding light on how these models ”see” and interpret facial features.

4.1 Introduction

In a deep face network, as an image traverses through different layers, the network progressively extracts and refines discriminative features while disregarding unwanted factors, such as lighting and pose information for a face recognition task. Feature visualization allows us to uncover the retained and discarded information at each layer, shedding light on the crucial aspects of images for various tasks. It provides valuable insights into the inner workings of the network and its decision-making processes. Feature inversion provides insights into the information preservation and discarding process within each layer of a deep neural network, thereby highlighting the discriminative aspects of images for various tasks. Additionally, it offers clues about the generality of the representations at each level, which can be helpful for transfer learning and fine-tuning purposes. Understanding the inner workings of the network through feature visualization enhances its applicability and performance.

On the other hand, feature inversion is a valuable tool for investigating why a network may misclassify or misinterpret specific images. By inverting the features of an image, we can observe how an object recognition model perceives a boat instead of a train or how an object detection model identifies

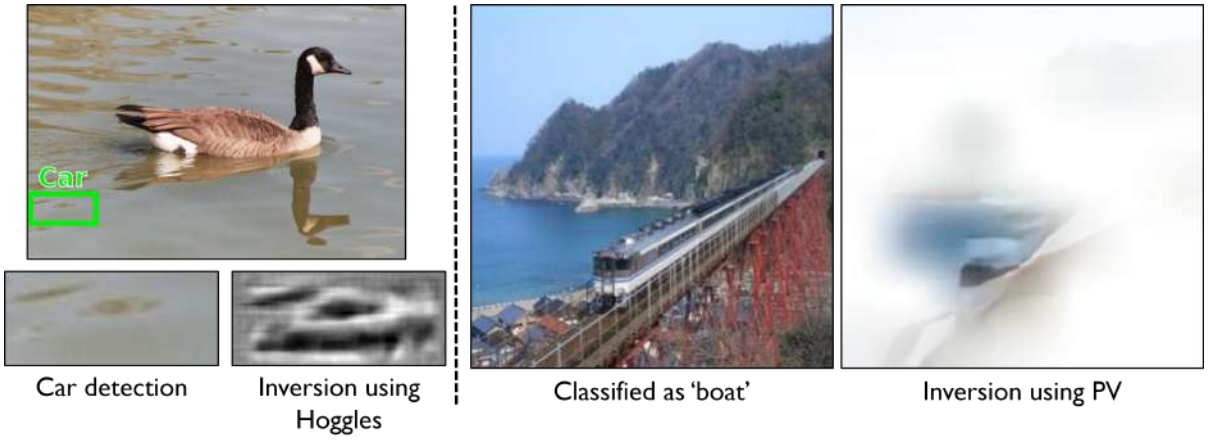


Figure 4.1: Examples of deep network 'mistakes' elucidated through feature inversion. *LEFT* HOGgles [145] is employed to elucidate an erroneous 'car' detection made by HOG features. *RIGHT* Perception Visualization (PV) [146] is utilized to clarify a misclassification of a 'boat' by deep features.

a car within the context of ocean waves (See Figure 4.1). Such insights provided by feature inversion help uncover the reasons behind model behavior and can guide network training and performance improvements. Another critical application of feature inversion is in safeguarding user privacy and security. Intelligent systems may store feature descriptors extracted from user images instead of retaining the complete images to address privacy concerns. Feature inversion techniques aid in deciphering the information encoded within these descriptors and assessing the potential risks of reverse-engineering user-identifying systems such as facial recognition or location tracking [147]. Furthermore, feature inversion has sparked creative endeavors, such as Deep Dream [130, 148] and Style Transfer [136], which leverage the technique to generate digital art. These approaches allow for collaborative creation between humans and AI, enabling the fusion of different artistic styles or the generation of entirely novel artworks.

4.2 Feature Inversion as an Optimization Problem

Early research in this domain concentrated on the inversion of local image descriptors such as HOG [149], SIFT [150], and local binary descriptors [151, 152]. D'Angelo et al. [153] proposed an analytical solution to invert local binary descriptors by formulating inversion as an optimization problem with TV norm as a regularizer. Weinzaepfel et al. [154] and Vondrick et al. [145] employed patches extracted from an extensive dataset to reconstruct input images based on SIFT and HOG descriptors, respectively. Present methodologies adopt a generic optimization framework, which will be explored in this section.

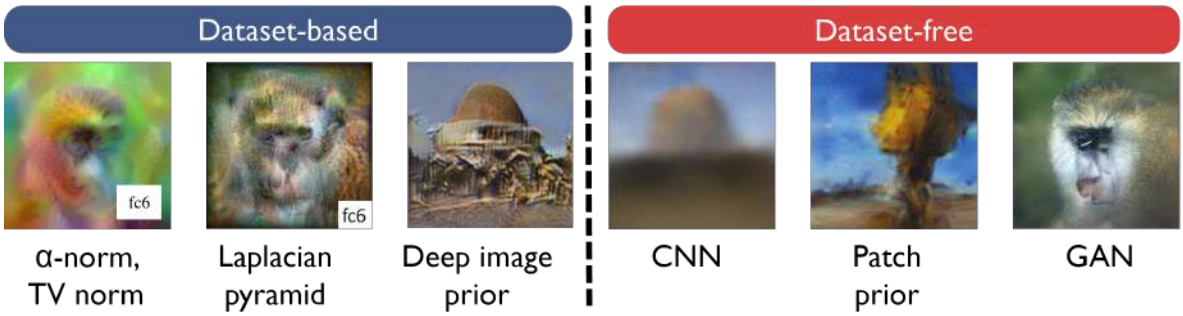


Figure 4.2: This image showcases the two main categories of regularizations commonly employed in feature inversion: dataset-based and dataset-free approaches. These regularizations are used to bring the visualizations into the natural image domain and enhance the quality of the inverted features.

4.2.1 Formulation

The process of feature inversion can be formulated as a general optimization problem as follows: Let x^0 represent an input image, and $\phi(.)$ denote the function responsible for extracting features from a specified layer of a deep network. Consequently, $\phi^0 = \phi(x^0)$ corresponds to the features we aim to invert. The objective is to discover a 'pre-image' x^* that yields features $\phi(x^*)$ closely aligned with the provided features ϕ^0. Analogous to Equation 3.1, feature inversion can be represented as an optimization function that seeks an image minimizing the distance between its features and the given feature vector [155]:

$$x^* = \underset{x \in \mathbb{R}^{W \times L \times C}}{\arg\min} \; L(\phi(x), \phi^0) + \lambda R \tag{4.1}$$

Here, L denotes the loss function utilized to quantify the dissimilarity between the features of the input image and the visualized image, often represented by $\left\|\phi(x) - \phi^0\right\|_2$. Additionally, R represents a regularization function that ensures the resulting visualization aligns with the desired characteristics necessary for effective explainability.

4.2.2 Desiderata of Feature Inversion Algorithms

Before delving into the results obtained from feature inversion methods applied to deep face models, it is vital to outline the desired characteristics of the inverted image, denoted as x^*:

Proposition 4.2.1 (Desired properties of Feature Inversion Algorithms)

1. ***Closeness of $\phi(x^*)$ to ϕ^0*** *The features extracted from x^* should exhibit a high degree of similarity to the original input features, as dictated by the concept of feature inversion.*

2. ***Closeness of x^* to x^0:*** *While not an obligatory criterion for explainability, we leverage feature inversion to assess the extent to which the inverted image aligns with the original input image,*

even though this is not explicitly included in the objective function (Equation 4.1). Due to the abstraction of features across layers, the mapping between representations and input images often becomes one-to-many, making x^* inherently distinct from x^0.

3. ***x^* within the input image domain $\mathcal{D}$:*** *If Equation 4.1 is directly optimized, x^* tends to exhibit high-frequency noise akin to activation maximization. Consequently, existing methods incorporate regularization functions that aim to confine the generated image within the input image domain $\mathcal{D}$, mitigating undesirable artifacts.*

4. ***Utilization of dataset images as input domain priors:*** *In cases where the regularization term R incorporates dataset images in some manner, it is crucial to ensure that the dataset employed is identical to the training dataset of the deep network, preserving consistency and fidelity.*

4.2.3 Regularization

Regularization functions are essential in feature inversion to mitigate the generation of noisy and adversarial images when directly optimizing the inversion equation. Similar to activation maximization, various regularization techniques have been proposed to ensure that the visualizations align with the characteristics of natural images. Both dataset-free and dataset-based approaches have been explored. Figure 4.2 shows examples of popular regularization functions.

Among dataset-free techniques, Mahendran and Vedaldi [155] introduced α-norm and TV norm as regularization functions to penalize high frequencies, as depicted in the top row of Figure 4.2. However, these regularizations demonstrate limited effectiveness in restoring color information. Singh and Namboodiri [156] utilized Laplacian pyramids to facilitate coarse-to-fine inversion, enabling the recovery of recognizable feature inversion across different layers of deep networks. Another dataset-free approach, known as "Deep Image Prior," [157] leverages the structure of a convolutional generator network as a prior for natural images. As shown on the right side of Figure 4.2, Deep Image Prior produces high-quality reconstructions without relying on input dataset information.

Deconvnet [134] and guided backpropagation [158] techniques lie at the intersection of feature visualization and inversion. Instead of inverting the full feature map of a layer, these methods focus on inverting a single activation map. Dosovitskiy and Brox [159] introduced a simple dataset-based approach, training a convolutional neural network (CNN) to invert the features of each layer using feature-image pairs from a dataset. As illustrated in Figure 4.2, the resulting pre-images exhibit blurriness and heavy influence from the dataset used for training the inversion networks. Other works [140, 145, 154] employ a natural image prior by stitching together representative patches.

Dosovitskiy and Brox extended their approach by training a generative adversarial network (GAN) for feature inversion [160]. The training process involves three losses: an l_2 norm between the input image and the generated pre-image, an l_2 norm between the features to be inverted and those of the

Figure 4.3: This image demonstrates the inversion process applied to a mandrill image using different Convolutional Neural Network (CNN) layers. The earlier layers of the network exhibit a remarkable resemblance to the original image, achieving near-perfect recreation. However, as we progress towards the later layers, the resulting images depict the abstract "concept" of a mandrill rather than a faithful representation. This observation highlights the transformation and abstraction of features as we move deeper into the CNN architecture. (Image excerpt from [155])

pre-image, and a standard adversarial loss. This combined loss formulation is depicted in Equation 4.2.

$$\begin{aligned} L = \lambda_\alpha \sum_i \left\| G(\phi_i^0) - x_i^0 \right\|_2^2 \\ + \lambda_\beta \sum_i \left\| \phi(G(\phi_i^0)) - \phi(x_i^0) \right\|_2^2 + \lambda_\gamma L_{adv} \end{aligned} \tag{4.2}$$

Giulivi et al. [146] introduced a hybrid approach that combines feature inversion and image saliency to visually represent the input space perceived by the deep neural network (DNN). They employed a CNN to invert a feature vector using a loss function incorporating l_2 norms between the input and pre-image, the features and the target features, as well as the structural similarity index measure (SSIM).

$$\begin{aligned} L = \lambda_\alpha \sum_i \left\| x_i^0 - x_i \right\|_2^2 + \lambda_\beta \left\| \phi(x_i) - \phi^0 \right\|_2^2 \\ + \lambda\gamma \sum_i \left\| SSIM(x_i^0, x_i) \right\|_2^2 \end{aligned} \tag{4.3}$$

The pre-image was multiplied with the GradCAM heatmap to focus on areas relevant to class prediction, as demonstrated in Figure 4.1.

In the subsequent sections, we will analyze and discuss the results of these feature inversion techniques on deep face models, evaluating their effectiveness and shedding light on the interpretability of the inversion process.

4.3 Results of Feature Inversion on Deep Face Representations

This section presents the findings of our study on feature inversion techniques applied to deep face representations. We conduct a comprehensive analysis by comparing the results obtained from different layers, face tasks, and inversion methods. The primary objective is to investigate the information preservation and discarding characteristics of each layer within a convolutional neural network (CNN). Furthermore, we explore how the specific training tasks of the network influence the information en-

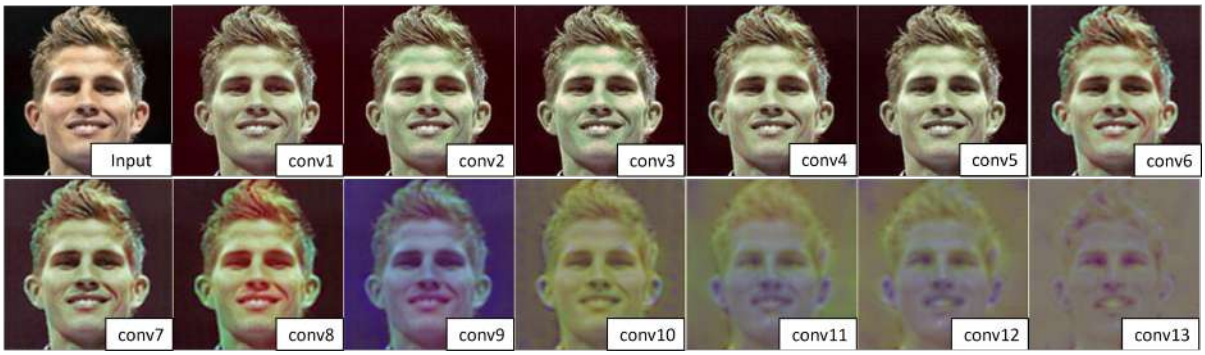


Figure 4.4: This image showcases the process of feature inversion applied to each convolutional layer of VGG-Face [19] in order to reconstruct a face. The regularization techniques, L2 and TV-norm, were utilized to guide the inversion process and improve the fidelity of the reconstructed face. The input face image, provided for reference, is depicted in the top left corner.

coded in the deep representations. All experiments in this section are performed on the VGG-16 architecture [161]. We examine the outcomes of feature inversion on three distinct networks: face recognition [19], head pose recognition, and emotion recognition.

4.3.1 Exploring Feature Inversion across Layers of Face Recognition Models

In this section, we examine feature inversions from different layers of a face recognition model and analyze the transition from concrete to abstract visualizations. The results of inverting the representations of all the convolutional layers of a VGG-Face network [19] are given in Figure 4.4. The study uses L2-norm and TV-norm [155] as regularizers.

Similar to feature visualization (Refer to Section 3.3) , the visualizations obtained through feature inversion show a progression from more concrete features to abstract representations as we move down the layers. The early layers of the face recognition model capture detailed information that allows for near-perfect reconstruction of the input face. This observation aligns with the findings from feature visualization studies. We note a similar transition from concrete to abstract representations when comparing feature inversions from face recognition models to those from object recognition networks. However, the face representation inversions exhibit a clearer and more pronounced manifestation of 'faces.' This distinction can be attributed to the fact that face recognition networks are specifically trained on face images, resulting in specialized representations with a narrower range of variation.

The insights gained from this study have implications for transfer learning and fine-tuning to different tasks. Understanding the level of abstraction in the representations across layers helps determine which layers to freeze when applying transfer learning. By freezing layers only up to the point where the desired information is not excessively abstracted, we can ensure effective transfer of knowledge while avoiding the loss of task-specific details.

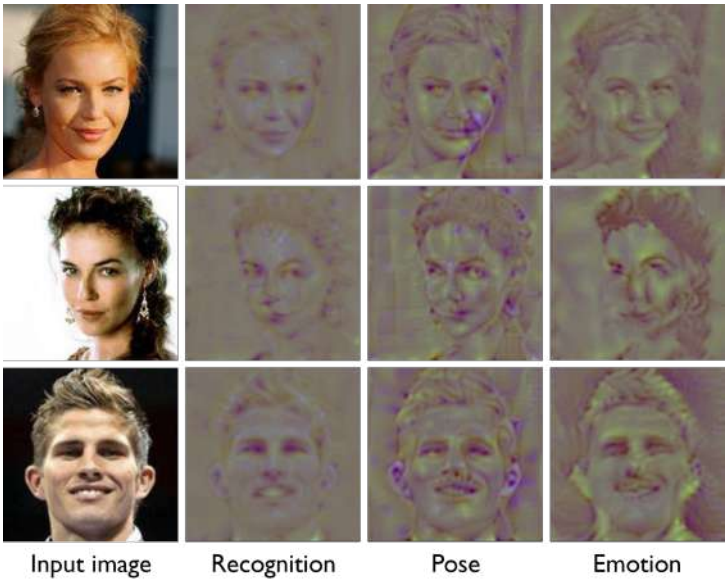


Figure 4.5: This image illustrates the process of feature inversion applied to the last convolutional layer of VGG-16 networks trained for recognition, head pose estimation and emotion recognition. The visualizations showcase the inverted representations, providing insights into the specific features these networks focus on for accurate recognition, head pose estimation, and emotion classification. This analysis highlights the networks' task-specific nature and ability to selectively retain attributes crucial for effective performance in their respective domains.

4.3.2 Comparing Feature Inversions across Face Tasks

In this section, we investigate the characteristics of face representations across different tasks. Our analysis includes three face-related tasks: face recognition, head pose estimation and emotion recognition. We utilize the VGG-16 architecture as the backbone for our experiments. Figure 4.5 showcases the feature inversion results obtained from the face models of these tasks for three input face images. Our observations shed light on the specific information each task's network captured.

Across all tasks, we find that the last convolutional layer of the network holds significant discriminative information. In the case of the face recognition model, the inverted features emphasize the shape and proportions of facial components, such as the eyes and nose. This aligns with the model's objective of accurately identifying individuals. In contrast, the head pose estimation network does not faithfully reconstruct the precise shape of facial features. Instead, it preserves the highlights and shadows, emphasizing the three-dimensional information of the face, with a particular emphasis on the contours of the nose and facial outlines. This highlights the network's focus on capturing head orientation rather than fine-grained facial details. For the emotion recognition model, the inverted features exhibit exaggerated curves in the face and eyebrows that emphasize expressions. This reflects the network's emphasis on capturing expressive facial features crucial for detecting and classifying emotions. Consequently, the inverted images from the emotion model tend to exhibit amplified emotional expressions.

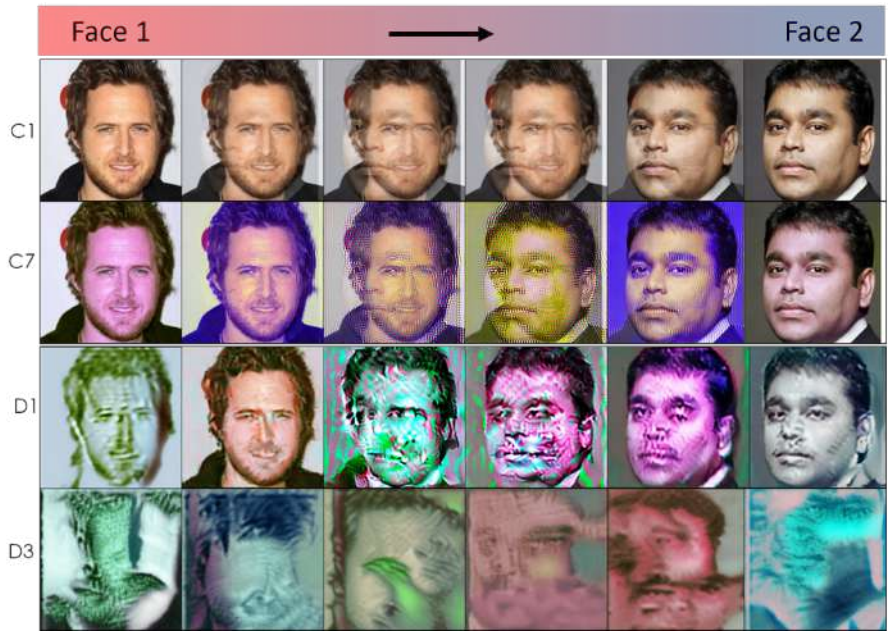


Figure 4.6: This image demonstrates the process of interpolating between two faces in the feature space and subsequently inverting the interpolated features back to the image space using deep image prior [157]. The interpolation is performed by smoothly transitioning the feature vectors between the two faces. Notably, the features of the convolutional layers exhibit abrupt changes, causing a noticeable jump in identities from one face to the other. Conversely, the fully connected layers exhibit a more seamless interpolation, resulting in a visually smoother transition between the two faces.

Such observations have implications for tasks involving multi-task learning or meta-learning on face image data. Understanding the specific information encoded by different face models can aid in designing effective learning strategies and selecting appropriate network architectures. By leveraging the unique characteristics of each task's representations, we can enhance the performance and interpretability of face-related applications.

4.3.3 Interpolation in Feature Space

In addition to visualizing pre-images of feature vectors corresponding to natural images, exploring the pre-images of features that do not align with any specific natural image can provide valuable insights. Figure 4.6 presents the results of inverting features obtained through interpolation between the features of two distinct face images. These features were extracted from various layers of the VGG-Face network [19]. Notably, we observe distinct behaviors between the convolutional and fully connected layers regarding feature interpolation. The interpolation performed on the convolutional layer features produces images that resemble a combination of the original images, indicating that these layers capture low-level visual elements shared by both images. In contrast, the dense layers generate pre-images with more semantically meaningful interpolation, suggesting that these layers capture higher-level semantic information contributing to the overall appearance and characteristics of the face.

Furthermore, their respective roles in capturing local and global features reflect the disparity in interpolation characteristics between convolutional and fully connected layers. Specifically, the convolutional layers exhibit abrupt changes, resulting in a noticeable jump in identities during face interpolation. On the other hand, the fully connected layers exhibit a more seamless interpolation, yielding visually smoother transitions. This observation underscores the contrasting roles of these layers and highlights the importance of considering their layer-specific characteristics when interpreting and utilizing deep representations. Our experiments demonstrate that while the initial and middle layers can faithfully recreate the input image, the deeper layers focus on extracting class-discriminative information relevant to the specific task under study. This emphasizes the role of deeper layers in capturing abstract and task-specific representations.

4.3.4 Comparision of Feature Inversion Methods

In this section, we investigate the impact of various natural image priors on feature inversion and analyze the outcomes obtained from different methods. The results, depicted in Figure 4.7, are obtained by inverting representations from multiple layers of a VGG-Face network [19] using three input images. For each input image, we employ three different methods for feature inversion: L2-norm and TV norm [155] for the top layer, training a dedicated CNN [159] for the middle layer, and utilizing the Deep Image Prior [157] for the third layer.

Our observations reveal that applying L2-norm and TV-norm [155](top rows) alone fails to reproduce the color information present in the later layers. Additionally, when inverting fully connected layers, the resulting visualizations exhibit multiple faces and features randomly distributed throughout the image. The middle rows demonstrate inversions using CNNs trained to generate face images based on their corresponding representations [159]. While these inversions remain faithful to the original images, they exhibit a certain degree of blurriness. Moreover, the fidelity of these visualizations is highly influenced by the dataset on which the CNNs were trained, suggesting a tendency towards overfitting the specific representations. In contrast, using Deep Image Prior [157] (bottom rows) as a robust natural image prior, independent of any specific dataset, yields promising results. The early layers produce accurate reconstructions, closely resembling the input faces. However, the visualizations from the last layer exhibit a more abstract representation of the face, emphasizing the extraction of high-level features.

4.4 Summary

In this section, we have explored feature inversion, delving into its formulation, motivation, and implications. Through a comprehensive analysis, we have aimed to deepen our understanding of this powerful technique and its significance in unraveling the inner workings of deep networks.

A key takeaway from our study is the complementary nature of feature inversion and feature visualization. Both techniques provide similar insights into deep face representations, particularly regarding the local and global information captured by lower and higher layers, respectively. As we invert repre-

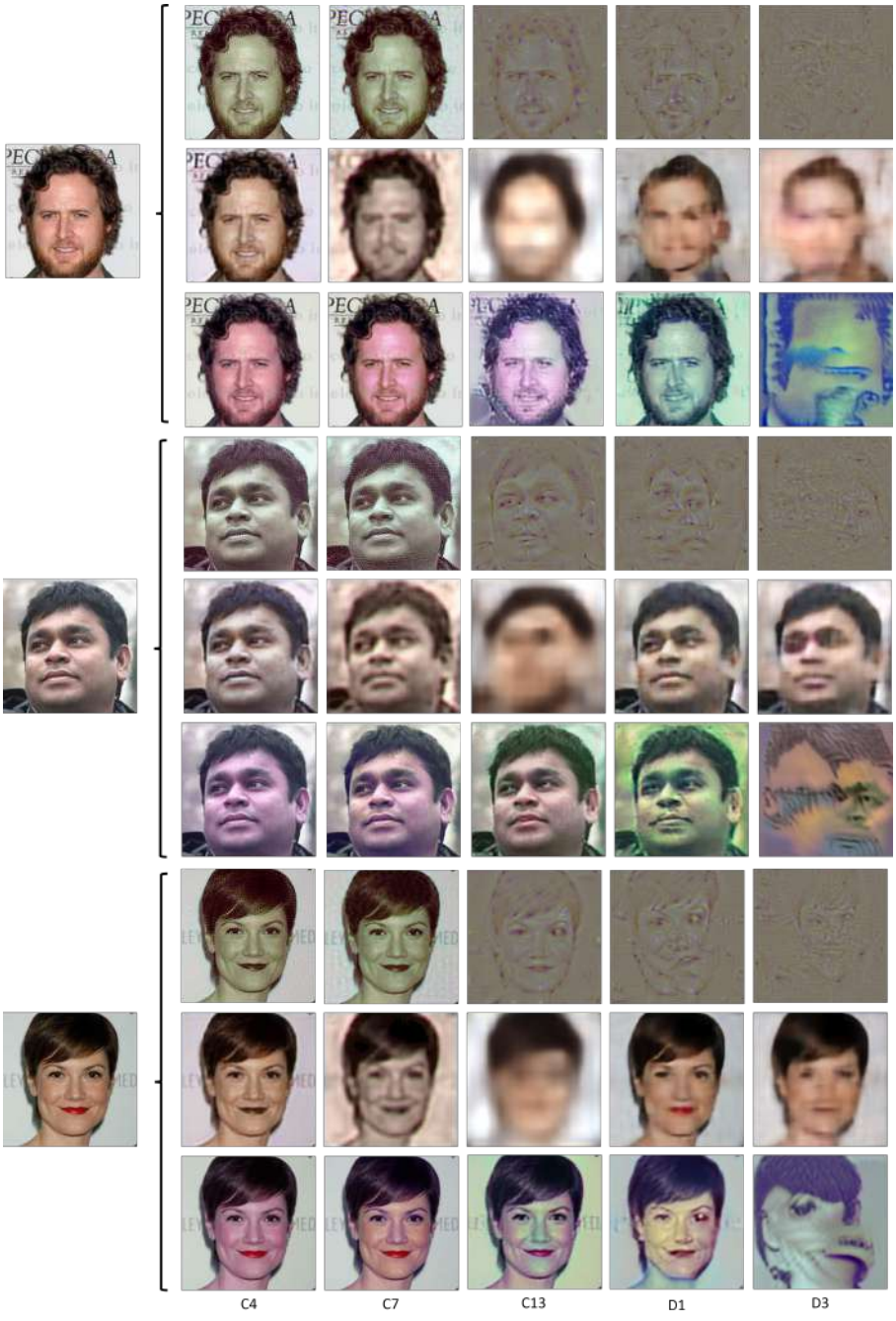


Figure 4.7: This image illustrates feature inversion using three distinct methods applied to different layers of the VGG-16 architecture. For each input face image, the rows correspond to a specific input face, while the columns represent different network layers. In the top row, feature inversion is performed utilizing L2 and TV-norm regularization [155]. The middle row demonstrates inversion achieved by training a convolutional neural network (CNN) specifically for feature inversion [159]. Lastly, the bottom row showcases feature inversion accomplished through the deep image prior approach [157].

sentations from lower layers to higher layers, we observe a gradual abstraction of the input image. Our findings also highlight that the convolutional layers of the deep-face models contain sufficient information to reconstruct the face faithfully. However, the type of information retained within these layers is influenced by the specific task the model is trained for.

In summary, this chapter has comprehensively explored feature inversion for deep face representations. Our findings contribute to our understanding of the inner workings of deep networks and their interpretation of facial features. The insights gained from this analysis have implications for various applications, such as face recognition, transfer learning, and multi-task learning.

PART III

Functional Exploration of Deep Face Representations

Chapter 5

Functional Concepts of Deep Representations

Part II of this thesis has explored individual units within a convolutional neural network (CNN), aiming to uncover their symbolic representations through visualizations that maximize unit activation or invert their representations. However, in many instances, the patterns observed in these representations are abstract, defying our ability to articulate them in ordinary language as they do not align with how humans naturally conceive concepts. For example, a neuron may respond to image patches that represent 'nose' or 'left-facing' or 'rough skin.' Still, it is not immediately apparent by looking at a visualization of the neuron. Consequently, this chapter investigates the functions exhibited by neural net units in terms of concepts more readily understandable to human cognition.

In this chapter, our focus shifts from visual interpretations of unit activation to unraveling these units' underlying functions and purposes and relating them to concepts and ideas that align with human understanding. By reviewing relevant research papers and studies, we aim to bridge the gap between the abstract nature of CNN unit representations and our innate comprehension of the world.

5.1 Introduction

Previous sections of this thesis have focused on visualizing images that activate individual units to their maximum extent or inverting their representations. However, the resulting patterns often prove too abstract to be easily expressed in familiar human terms. Therefore, this section examines how to establish meaningful connections between unit functions and concepts humans can readily understand.

An important aspect to consider is the nature of concepts themselves. While concepts are closely tied to object categorization, we must examine whether any shared property can effectively serve as a concept. Rosch [162] argues that human categorization should not be considered the arbitrary product of historical accident or whim but rather the result of psychological principles of categorization. Thus, some latent metrics can make some concepts better than others.

Here are some desired properties of concept definitions as described in [163]:

Definition 5.1.1 (Desiderata of Functional Concepts)

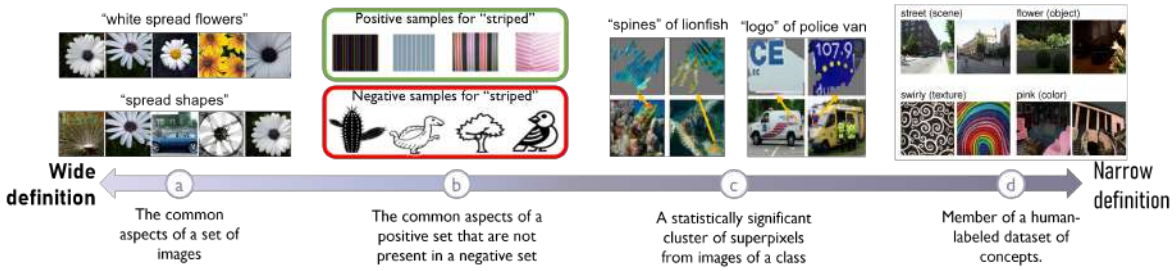


Figure 5.1: Various Approaches to Defining a 'Concept' (a) [164] (b) TCAV [165] (c) ACE [163] (d) Broden [166]

Meaningfulness *A concept is semantically meaningful on its own. Different individuals should associate similar meanings to the concept.*

Coherency *Examples of a concept should be perceptually similar to each other while being different from examples of other concepts.*

Importance *A concept is "important" for predicting a class if its presence is necessary to predict samples in that class accurately.*

5.2 Definition of Concepts

This section delves into the different approaches used to define concepts. Figure 5.1 shows the spectrum of concept definitions from weak to strong.We explore two primary methods: mining concepts from collections of images and curating a list of concepts.

In Figures 5.1(a) and 5.1(c), we observe visual representations of concepts that have been extracted through the process of mining from a dataset. These mining algorithms employ statistical analysis techniques to identify recurring patterns and assess their significance within the dataset. By leveraging this data-driven approach, these algorithms aim to uncover common themes or visual elements that are relevant to the dataset under investigation. Visualizing these mined concepts provides valuable insights into the underlying patterns and structures within the dataset, facilitating a deeper understanding of the visual information captured within the images.

Figures 5.1(b) and 5.1(d) present examples of curated concepts that human experts have meticulously selected. These concepts encompass a range of image attributes, including object categories, colors, textures, poses, and other relevant visual features. Once a comprehensive list of such attributes is obtained, one straightforward approach to assigning a concept to a neural unit involves activation maximization 3. The validity of the inferred concept can then be assessed by collecting images containing the suspected concept and passing them through the neural network. If the concept truly corresponds to the unit, a discernible spike in the activation magnitude is expected to occur [167]. However, this simplistic approach encounters several inherent challenges. Firstly, it is a time-consuming process that necessitates significant manual effort. Furthermore, multiple units exhibit spikes for the same image in many cases,

suggesting the potential presence of other concepts represented by the image. Unfortunately, since not all units correspond to human-interpretable concepts, there are no definitive means of verifying these additional concepts. Moreover, this method cannot identify concepts that multiple units may jointly represent.

In this section, we undertake a comprehensive examination of diverse concept definitions and methodologies employed for the purpose of assigning concepts to neural units.

5.2.1 Mined Concepts

In the context of mined concepts, these can be broadly defined as common attributes shared among a collection of images (Fig 5.1(a)) []. Examples of such concepts may include abstract notions like "top round stroke" or "lower left loop" in the case of MNIST digits or more concrete descriptors such as "white flowers" or "dogs with brown heads" for images from the ImageNet dataset.

Ghorbani et al. [163] proposed a more systematic approach for automatically mining concepts from image datasets, known as Automatic Concept-based Evaluation (ACE), as illustrated in Figure 5.1(c). ACE algorithmically searches for salient groups of superpixels within the dataset that meet specific criteria for being considered a concept. ACE identifies segments related to textures, object parts, and objects by utilizing superpixel segmentation at various resolutions. These segments are then clustered based on their feature-space similarity, with each cluster representing a potential concept. ACE incorporates the concept activation vectors (CAVs) introduced in TCAV [165] to identify salient concepts for each output class.

Ghorbani et al. defined several tests based on human perception to evaluate the coherence of the mined concepts. The first test, the "Intruder Test," involves presenting six images and asking human participants to identify the image that differs semantically from the rest. Coherent concepts are indicated if participants consistently select the intruder image. Another test involves displaying four concept segments and four randomly selected segments from images of the same class. Participants are then prompted to select the most "meaningful" grouping of image segments and describe their choice using a single word. The degree of agreement among participants, measured by the number of individuals using the same word to describe a group (after controlling for synonyms), serves as a measure of concept coherence. These metrics contribute to establishing a "concept" as a categorization that garners consensus among multiple individuals. A drawback of this method is that only concepts that can be defined as parts of images are considered.

5.2.2 Curated Concepts

The Broadly and Densely Labeled (Broden) dataset, introduced by Bau et al. [166], offers a comprehensive collection of diverse visual elements, including objects, scenes, object parts, textures, colors, and materials, within varying contextual settings. Each class in the Broden dataset is associated with an English word, and labels are merged based on shared synonyms, disregarding positional distinctions such as 'left' and 'top'. Numerous studies have adopted this dataset as a reliable reference for

defining concepts. However, it is worth noting that the Broden dataset solely encompasses concepts that correspond to individual images or specific regions within an image. Consequently, any statistical analyses performed using this dataset will be constrained to the concepts explicitly defined within its predetermined scope.

An alternative approach to defining concepts is presented by Kim et al. [165] in their work on Quantitative Testing with Concept Activation Vectors (TCAV). As shown in Figure 5.1(b), TCAV employs a more flexible concept definition, wherein each concept is characterized by two sets of images: one set containing the concept in all images, and another set in which the concept is absent from all images. This relaxed definition allows for a more nuanced exploration of concepts and their associations within neural networks.

5.3 Conceptual Units of Deep Models

Should a concept be linked to a single neuron, or can random directions of neuron concepts hold equal significance? The answer to this question largely depends on defining ”concept” within a specific context. Szegedy et al. [164] hypothesized that random directions possess semantic meaning comparable to that of individual neurons. To investigate this, they conducted experiments comparing images that evoked high activations in individual units with images that corresponded to random directions of those units. The results demonstrated that both groups exhibited semantic meaning. However, it is worth noting that their definition of ”concept” was rather broad, emphasizing the shared properties among a set of images. The concepts included “postures”, “spread shapes”, and “round green or yellow objects”. Consequently, the extent to which such a concept captures strong semantics may raise concerns.

In contrast, other studies [166, 168] leaned towards the perspective that neurons possess specialized semantic meaning. They adopted a curated set of concepts derived from the Broden Dataset [166] as their definition of concepts. Notably, they discovered that the natural basis, i.e., individual units, corresponded to more distinct concepts than orthogonal rotations of the basis, even though both exhibited similar discriminative power. However, it is essential to consider that this observation might be influenced by the limited range of concepts provided by the Broden dataset.

Fong et al. [169] introduced a different viewpoint, suggesting that while random directions of neural units may lack interpretability comparable to individual units, there exist particular directions that are more interpretable. Their argument was rooted in the notion that the number of available feature channels is typically much smaller than the number of different concepts a neural network may need to encode to comprehend a complex visual scene. This implies that the representation must employ combinations of filter responses to represent concepts, indicating a distributed nature of representation effectively. Fong et al. also employed the Broden dataset as their corpus of concepts, albeit disregarding the scene and texture labels. Hence, the precise definition of concepts becomes a critical factor in this context.

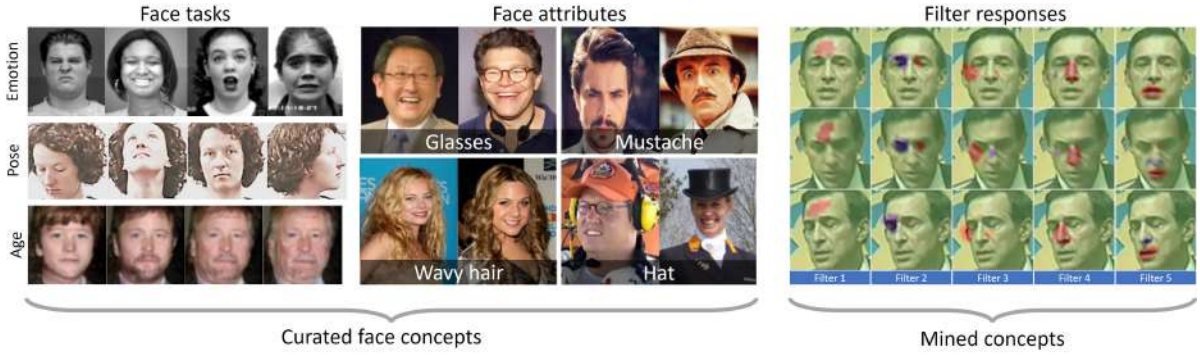


Figure 5.2: Listing of potential concepts for the face domain. *Image sources:* Emotion: [170], Head pose: [171], Age: [172], Face attributes: [173], Filter responses: [174].

5.4 Functional Concepts in Deep Face Models

The above concept-mining approaches have not been applied much to the face domain. Based on the above definitions and discussions, possible concepts for face image data include identity, pose, gender, race, facial hair and accessories. CelebA [173] is a dataset with labels for 40 facial attributes, which could also be considered concepts. Facial parts, head pose, emotions and facial action units also constitute relevant concepts. Face concepts could also be defined based on the task of the pre-trained network, as in what may be concepts useful for classification or the specific task at hand. In Figure 5.2, we show possible definitions for facial concepts. Facial concepts may correspond to facial features like nose and eyebrows or be unique shapes or textures. In Chapter 6, we discuss algorithms to assign concepts to neural units of deep face networks.

In the only work in the face domain to the best of our knowledge, Yin et al. [174] follow a different approach towards interpretability. Instead of 'finding' concepts corresponding to each convolutional filter, their modified training procedure pushes each filter to represent a concept. Their approach uses a Siamese network with two branches sharing weights. The first branch gets a face image as input, and the second gets the same image superimposed with a synthetic occlusion. Along with the recognition loss, they introduce two new losses that encourage the filter representation to have a more consistent semantic meaning and require the filters to be insensitive to occlusions. The two losses ensure filter response locations are distributed across the face, and each filter concentrates on local face parts. Figure 5.2 (bottom) shows some filter responses from a model trained with this technique. We observe that each filter responds to a specific face feature regardless of the identity or pose of the face. This makes each filter's representation more 'concrete' and more straightforward to describe in words, thus being easier for humans to conceptualize.

5.5 Summary

In conclusion, this explored the underlying functions and purposes of units within convolutional neural networks (CNNs) and their connection to human-understandable concepts. This chapter explored the spectrum of functional concept definitions, ranging from the broad notion of ”any shared characteristic of a group of images” to the more specific ”one of the characteristics from a given dataset.” This understanding helps us establish a framework for defining and interpreting functional concepts within deep representations. Furthermore, we highlighted that a functional concept could be linked to an individual neuron or a collection of neurons within a layer. In the context of faces, we identified potential collections of concepts associated with facial representations. These include face attributes, face tasks, and facial features.

In the subsequent chapter, our exploration delves deeper into the functions of convolutional filter groups within deep face models. We closely examine the intricate relationship between functional units across various face models and analyze their interplay. By unraveling these intricate mechanisms, we aim to gain further insights into the complex nature of deep face representations.

Chapter 6

Task-Based Concepts in Deep Face Models

In the previous chapter, we explored the functions and purposes of units in CNNs and their relevance to human-understandable concepts. By exploring a spectrum of functional concept definitions, from general shared characteristics of images to curated lists of concepts, we developed a framework for defining and interpreting these concepts within deep representations. Additionally, we discovered that functional concepts can be attributed to individual neurons or groups of neurons within a layer. Specifically, within the domain of faces, we identified potential collections of concepts related to face attributes, face tasks, and facial features.

In this chapter, we further explore task-based concepts in deep face networks. We examine the close relationship among all tasks in the face domain and investigate their interplay. Additionally, we investigate the assignment of task concepts to neural units in deep face models and the visualization methods used to elucidate these concepts. This comprehensive exploration aims to enhance our understanding of the functional aspects that drive deep face networks.

6.1 Introduction

A well-known fact is that tasks in the face domain are closely related to one another. Despite the variability in face images, different face tasks, such as recognition, pose estimation, age estimation, and emotion detection, operate on similar input data. These tasks aim to discern subtle differences among the images. We leverage these relationships to gain insights into face networks. These tasks pose challenges as face images can be similar, and fine-grained classification is often required. Additionally, diverse variations, such as expression, pose, and accessories, must be considered for each face task.

In this chapter, we explore face tasks and their interrelationships. Table 6.1 shows the details of the tasks we used. Face recognition/verification is a critical problem in this domain, benefiting from large available datasets. However, other face-related tasks, such as head pose, age estimation, and emotion detection, lack extensive datasets. We investigate the potential of utilizing face recognition datasets to improve performance on other face tasks. Moreover, we thoroughly examine the relationships among these face tasks.

Task	Type	Classes	Label Source
Identity	Categorical	10177 classes	CelebA [173]
Gender	Categorical	2 classes	CelebA [173]
Facial Hair	Multilabel	5 classes: 5 o'clock shadow, goatee, sideburns, moustache, no beard	CelebA [173]
Accessories	Multilabel	5 classes: earrings, hat, necklace, necktie, eyeglasses	CelebA [173]
Age	Categorical	10 classes	Imdb-Wiki [175]
Emotions	Categorical	7 classes: angry, disgust, fear, happy, sad, surprise,neutral	Fer13 [176]
Head pose	Categorical	(9 classes)	3DMM [177]

Table 6.1: List of different tasks and corresponding labels. The labels for the first four tasks are provided with the CelebA data set. The other labels were obtained using known methods [175–177]

We also uncover that deep representations in face models contain information about other face tasks without explicit training. In this chapter, we develop a methodology to discover these cross-task aware filters. Our findings deepen our understanding of the face domain and provide a framework for efficient transfer learning and task-based pruning of deep face models, which will be discussed in the next chapter.

6.2 Relationship Between Face Tasks

Let a dictionary of face tasks be defined by $F = \{f_1, f_2, \ldots, f_n\}$. Let f' be the primary task. Then the set of satellite tasks are denoted by $F - \{f'\}$. We train a network model $\mathcal{M}$ on f' and use its features to regress $f^t \in F - \{f'\}$ for a satellite task. For example, we can train a network on the primary task of face recognition and use its features to regress for satellite tasks such as age, head pose and emotion detection. To this end, we consider a convolutional layer of model $\mathcal{M}$ with, say, k filters. Let the activation map of layer l on image I be denoted by $A^l(I)$, and have size $k \times u \times v$, where each activation map is of size $u \times v$. We hypothesize that, unlike contemporary transfer learning methodologies (that finetune the weights or input these activation maps through further layers of a different network), a simple linear regression model is sufficient to obtain the predicted label of the satellite task, f^t. Our procedure is outlined in Algorithm 1. First, we take the activation map of a convolutional layer and perform global average pooling on it. This is then used as a feature vector to regress the satellite tasks. A large data set typically trains the primary task, as with any other deep face network. However, owing to the simplicity of our satellite task model, limited data is sufficient to train the satellite model using linear regression.

A data set with ground truth for all considered face tasks is essential to validate the above-mentioned method. We used the CelebA data set [173], which consists of 202,599 images of all experiments in this chapter. The labels for identity, gender, accessories, and facial hair are available as a part of the data set.

Algorithm 1: Training Satellite Face Task Model from Primary Task Model

Input:
Face image data set for satellite task $f^t \in F - \{f'\}$: $\{I_1, \ldots, I_n\}$ with corresponding ground truth $Y^t = \{y_1^t, \ldots, y_n^t\}$
$\mathcal{M}$ is a model obtained by training for the primary face task, f'.
$A^l(I_j)$ is the activation map (size $u \times v$) of layer l with k filters on image I_j, $j = 1, \cdots, n$
Output: Regression model W^t for f^t

1 $W^t = \arg\min_W \sum_{j=1}^{n} \frac{1}{2} \left\| w^T A^l(I_j) - y_j^t \right\|_2^2$

Figure 6.1: Sample predictions obtained on CelebA data set using linear regression on the activation maps of a CNN trained for face recognition. The green text shows correct predictions and red text shows incorrect predictions.

We generated the ground truth using known methods for age, emotion, and pose. The ground truth for age was obtained using the method DEX: Deep EXpectation of apparent age from a single image [175]. This method used a VGG16 architecture and was trained on the IMDB-WIKI data set, which consists of 0.5 million images of celebrities crawled from IMDB and Wikipedia. The ages obtained using this method were binned into ten bins, each with ten ages. The head pose was obtained by registering the face to a 3D face model using linear pose fitting [177]. The model is a low-resolution shape-only version of the Surrey Morphable Face Model. The yaw and pitch values were binned into nine bins ranging from top-left to bottom-right. Figure 6.4 depicts the binned pose values. For emotion, a VGG-16 model was trained on FER 2013 data set [176] with seven classes. (See Table 6.1 for the details of the considered data set).

We consider the seven face tasks in Table 6.1. The entire CelebA data set was divided into 50% train, 25% validation, and 25% test sets. We used a pre-trained VGG Face model [19] and finetuned it for a considered primary task. The ground truth for the satellite tasks was created by taking a subset of 20,000 images from CelebA ($\approx 10\%$ of the data set). This was also divided into 50% train, 25% validation, and 25% test sets. All our reported results are obtained by averaging over three random trials from different partitions of the satellite data. We converted the continuous regression outputs into categorical attributes for each task. For binary classes such as gender, our output was regressed to a value between 0 and 1,

	Age	Gender	Emotion	Facial Hair	Accessories	Pose
Face Recognition	42.5	97.1	46.21	94.34	92.65	85.22
Age	61.723	93.88	36.5	93.4	91.51	86.94
Gender	38.23	98.37	40.75	94.16	92.52	89.36
Emotion	29.21	87.8	69.014	91.88	91.16	87.48
Facial Hair	38.41	96.76	47.22	96.14	92.584	88.16
Accesssories	38.26	96.46	39	93.56	94.76	87.77
Pose	36.90	95.05	40.79	93.38	93.43	96.62

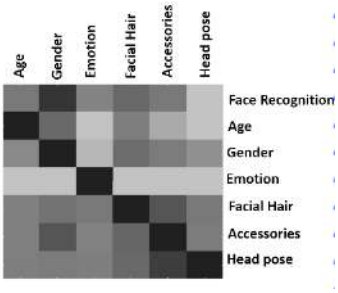


Figure 6.2: Table (left) shows the results of transferring tasks using the proposed method. Each row corresponds to the primary task, and the columns correspond to the satellite task. We report the accuracy obtained when transferring a network pre-trained on the primary task to the considered satellite task. The diagonal cells show the accuracy obtained while training the primary task. The figure (right) shows a heatmap of the transfer capability of one face task to another based on this methodology (darker is better; for example, face recognition models can regress gender very well, while age estimation models are one of the least capable of estimating emotions).

and a threshold (learned on the validation set) was used to decide the label on the test set. We regressed to a continuous label space based on the original labels for multicategory classes such as age and pose. We then binned it using the same criteria we used for training the primary networks.

To determine how well a particular transfer took place, we compared the performance of our models learned on satellite tasks to the accuracy obtained by a network that was trained explicitly on the same task as the primary. For example, we want to compare the accuracy obtained by transferring a network trained for face recognition to the gender task. We do this by comparing the regression accuracy of face recognition→gender with the network trained on the full data set for gender. This is measured as the percentage reduction in performance when changing from the full data set to the subset.

Figure 6.2(left) shows the results of transferring tasks using regression. The activations were regressed to continuous labels, which were then binned to get the accuracy. For the emotion detection task, we used linear classification. The primary tasks are represented by each row, which was then transferred to each satellite task represented by the columns. The accuracy obtained by a network trained for the primary task is denoted in the cells where the primary and secondary tasks are the same. For each satellite task, the percentage reduction in performance compared to the network trained on the corresponding primary task is also captured in Figure 6.2 (right), with lower values (darker cells) being better. We show qualitative results obtained using our regression algorithm in Figure 6.1.

We notice that networks trained on primary tasks give better results while regressing with tasks with which the primary tasks may have some correlation. For example, a model trained on gender recognition as the primary task gives good results for facial hair estimation and vice versa (supports common knowledge). Similarly, the accessories and gender estimation tasks are strongly correlated because certain accessories, such as neck tie, earrings, and necklace, correlate strongly with gender. On the other hand, emotion gives low accuracy for all other tasks since emotion is usually learned independently from other facial attributes. Face recognition gives very good results for gender, facial

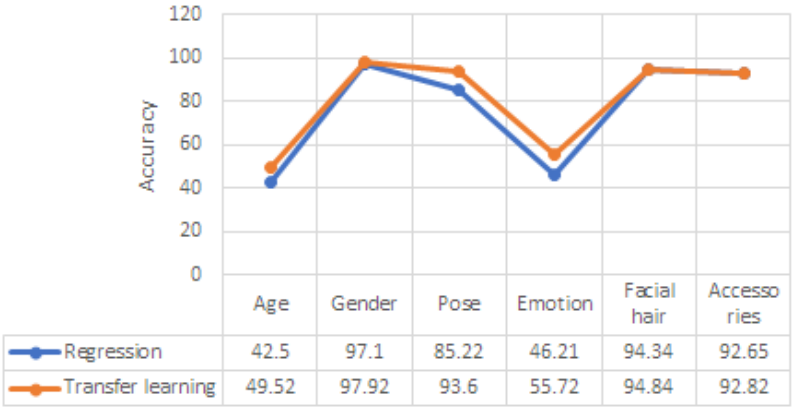


Figure 6.3: The graph compares regression and transfer learning for a network pre-trained on face recognition and transferred to six other tasks. We can see that regression and transfer learning results are very close, allowing us to replace transfer learning with our method effectively.

hair and accessories since these vary from individual to individual. Face recognition does not give the best results for pose because face recognition has to be invariant to pose. Curiously, age is regressed well by the face recognition network. This may be due to biases in the data set, where images belonging to each individual do not have a large range of ages.

6.2.1 Relation to Transfer Learning

We conducted experiments to examine how well our regression method compares to using transfer learning on various tasks. For this setting, we used networks pre-trained on the face recognition tasks and reinitialized all the fully connected layers. We then froze the convolutional layers and trained the linear layers for the satellite tasks. The results can be seen in figure 6.3. We can see that the regression results are close to the transfer results. Thus, we can use our simple regression method to find task relationships instead of doing expensive transfer learning for each task. Our regression method takes 10 seconds to run for a single task instead of transfer learning, which takes 780 seconds. We thus achieve a speed-up of **78X** using our method.

6.3 Cross-Task Concepts in Face Recognition

An intriguing question in the field of computer vision pertains to whether models trained on a specific face task, such as recognition, inherently encapsulate information about other related face tasks. Empirical investigations have shown that certain convolutional filters within face recognition models exhibit the ability to predict additional face-related tasks, including head pose, age, and gender, without the need for additional supervision. In this study, we coin the term ”Cross-task Aware Filters” (CRAFTs) to describe these filters and aim to demonstrate their existence through an experiment.

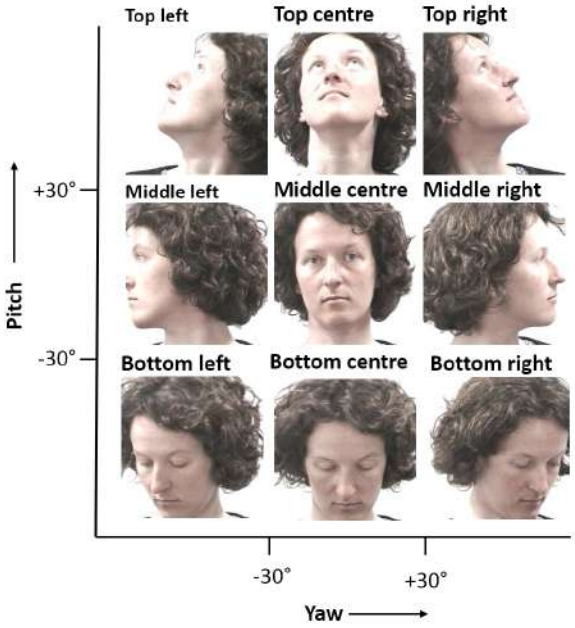


Figure 6.4: Classes for head pose task in Table 6.1. The yaw and pitch were divided into bins with 60° bin size.

To illustrate the presence of CRAFTs, we employ the widely used VGG-Face model [19], which has been trained on an extensive dataset of 2.6 million face images specifically for face recognition. To identify CRAFTs associated with head pose, we leverage the Head Pose Image Database [171], a benchmark dataset containing face images that maintain constant attributes except for variations in head pose. By passing the images from this dataset through the VGG-Face model, we examine the mean activation of each filter in the final convolutional layer in relation to the yaw of the head. Figure 6.5 illustrates the findings, demonstrating highly correlated activations of certain filters with respect to yaw. Some filters exhibit heightened responses for front-facing images, while others display significant activations for face images with pronounced head rotation. Remarkably, these CRAFTs emerge within the face recognition model without additional supervision or explicit training for yaw estimation. Consequently, these identified CRAFTs can be effectively utilized for transferring the model's capabilities to predict the yaw of the head.

6.4 Finding Cross-Task Aware Filters

This section presents a comprehensive methodology that identifies the optimal sets of Cross-task Aware Filters (CRAFTs) and investigates the intricate relationship between the primary face task and secondary face tasks. CRAFTs can be considered as task-based facial concepts of the neural units of a deep face model (See Chapter 5).

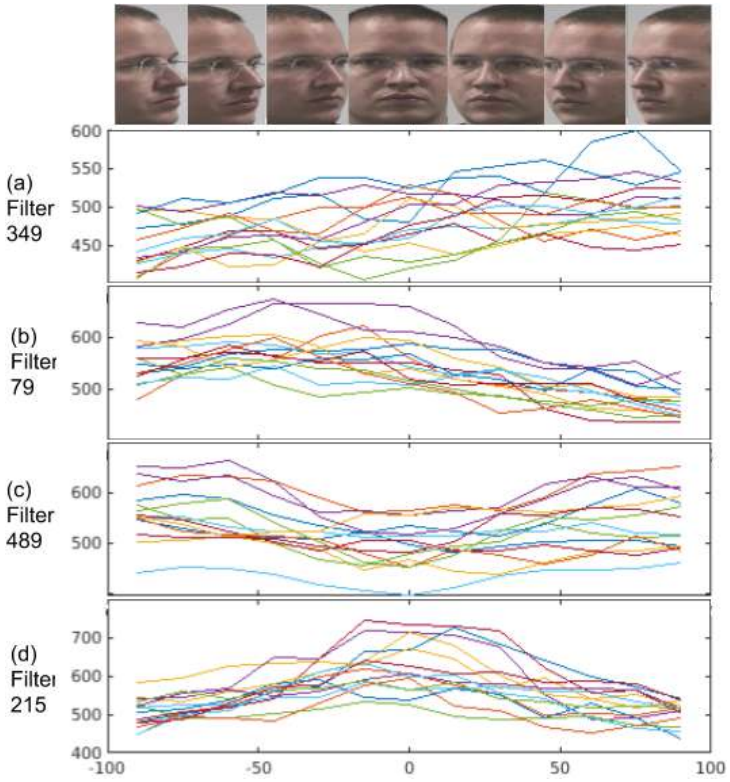


Figure 6.5: The response of some convolutional filters of the VGG-Face model are correlated to the yaw of the face, despite the model not being trained with head pose. The figure shows a few filters from the last layer of VGG-Face whose average responses have a high correlation to the yaw angle on Head Pose Image Database for different identities. The different lines in each graph represent 15 different identities: (a) high activation for left-facing faces; (b) high response for faces facing right; (c) high response for sideways faces; (d) high response for frontal faces

6.4.1 Finding Optimal Sets of CRAFTs

We now find the optimal CRAFT sets which predict secondary tasks like Age, Head Pose, Gender and Emotion. Let $D = (I, Y)$ be a dataset for a secondary task where $I \in \mathbb{R}^{N \times 3 \times W \times H}$ is the set of N dataset images and $Y \in \mathbb{R}^N$ is the corresponding ground-truth values. Consider the l^{th} convolutional layer of a model ϕ having weights $W \in \mathbb{R}^{C_{l+1} \times C_l \times k \times k}$, which has C_{l+1} output channels/filters. Let $\phi_l(I) \in \mathbb{R}^{N \times C_{l+1} \times w \times h}$ be the activation of layer l. Let $X \in \mathbb{R}^{N \times C}$ be the average activations:

$$X = \frac{1}{wh} \sum_{i=1}^{w} \sum_{j=1}^{h} \phi_l(I)[:, :, i, j] \tag{6.1}$$

We need to choose groups of filters whose activations are highly correlated with Y. One way to do this is to rank each filter group based on a correlation coefficient ρ and pick the highest-ranked filters.

$$\rho_c = \left| \frac{Cov(X[:, c], Y)}{\sigma_{X[:,c]} \sigma_Y} \right| \tag{6.2}$$

where $X[:, c] \in \mathbb{R}^N$ is the activation of the c^{th} filter. However, individually picking filters results in a greedy solution as we do not consider the interdependence of filters. Instead of exhaustively checking all groups of filters in a layer, we use LASSO [178], an L_1-regularized regression method which selects a subset of filters that best predict Y using the objective:

$$\min_{\beta_0, \beta} \left(\frac{1}{2N} \sum_{i=1}^{N} (Y_i - \beta_0 - X_i^T \beta_i^T)^2 + \lambda \sum_{j=1}^{p} |\beta_i| \right) \tag{6.3}$$

where Y_i is the ground truth of sample i, $X_i \in \mathbb{R}^{C_{l+1}}$ is the global average-pooled activation of sample i, $\beta \in \mathbb{R}^{C_{l+1}}$ is the LASSO regression weight vector and λ is a non-negative regularization parameter which determines the sparseness of the regression weights β. The number of filters chosen decreases with an increase in λ, as more coefficients of β become zero.

6.4.2 Cross-Task Concepts in Face Recognition

The last layer of a VGG-Face [19] model trained for face recognition consists of 512 filters, each potentially carrying information about different face tasks. How many of these encode cross-task concepts? We applied Equation 6.3 for the secondary tasks age, head pose, eyeglasses and facial hair to find CRAFTs related to the particular tasks. The distribution of these tasks across the 512 filters is depicted in Figure 6.6. Our observations reveal that while numerous filters demonstrate task-specific relevance, a subset exhibit a more general nature, allowing for easy adaptation to solve various face tasks. Furthermore, we find that during the fine-tuning process of a pre-trained network for a specific face task, the task-specific filters that lack relevance for other tasks can be pruned, resulting in a highly streamlined

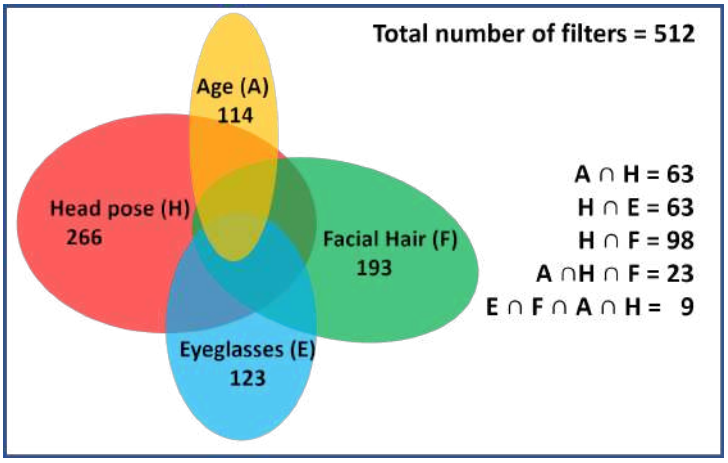


Figure 6.6: This figure illustrates the distribution of information about different face tasks within the last convolutional layer of a face recognition network. The outer rectangle represents all the filters in this layer, while the ovals indicate filters that encode information about specific tasks. Notably, we observe that most face tasks are represented by a small subset of filters, suggesting the presence of redundancy. This finding offers insights into the potential for network compression by eliminating redundant filters, thereby enhancing efficiency without compromising task performance..

network with minimal performance reduction. The details of this application are further discussed in Chapter 7.

6.5 Summary

In conclusion, this chapter has comprehensively explored task-based concepts in deep face networks, shedding light on the interrelationships between different face tasks. Through our investigations, we have gained insights into assigning task concepts to neural units in deep face models and developed a methodology to identify cross-task aware filters. These filters have revealed valuable information about other face tasks within the deep representations of the models.

Our findings highlight that while a significant number of filters exhibit task-specific relevance, a subset of filters demonstrates a more general nature, allowing for their seamless adaptation to solve a variety of face tasks. We have determined that this subset of relevant filters constitutes a small proportion of the layer, indicating potential opportunities for developing techniques that optimize and streamline the model architecture, which will be explored in the next chapter.

Chapter 7

Functional Pruning of Deep Face Models

In the previous chapter, we explored the intricacies of task-based concepts in deep face networks, specifically focusing on identifying cross-task aware filters. Our investigations unveiled that within each layer, only a fraction of the filters is dedicated to specific tasks, yet their inclusion is sufficient to achieve comparable accuracy in those tasks. The present chapter harnesses this knowledge to facilitate the development of efficient and compact neural networks through functional pruning. By targeting functionally redundant filters for removal, we aim to create streamlined models that maintain their performance while reducing computational overhead and model size.

7.1 Introduction

Deep face models are renowned for their demanding data and computational resource requirements. While the availability of massive face datasets like VGG-Face2 [15] (3M images) and Ms-Celeb-1M [173] (10M images) partially addresses the data challenge for tasks such as recognition, secondary tasks like age or emotion recognition suffer from limited publicly available data due to the difficulties in data collection and annotation. Consequently, transfer learning has gained popularity, where a model trained on a ”primary task” with abundant data is adapted to a secondary task through finetuning. However, the resulting model remains large, and computationally intensive, and the transfer learning process often falls short in leveraging the learned filter information from the primary model.

In response to these challenges, this chapter introduces ETL (Efficient Transfer Learning): a method designed to address the abovementioned limitations in deep face models. The foundation of ETL lies in comprehending the impact of different filters within a convolutional layer of a primary model regarding secondary tasks for which the model was not initially trained, as discussed in Chapter 6. Using lasso regression, we identify convolutional filters in the primary model that are irrelevant to the secondary task, subsequently removing them in a single-pass pruning step. The resulting sparse model is then finetuned for the specific secondary task. The proposed approach significantly reduces training time compared to training from scratch or standard transfer learning and yields computationally efficient models without sacrificing performance. This transfer learning technique finds applications in various

domains, including ADAS (Advanced Driver Assistance Systems) [108, 179, 180], where the efficient implementation of face algorithms in real-time applications is crucial.

Our proposed approach offers several advantages:

Rapid transfer learning: Unlike other pruning methods that iteratively prune filters and finetune the model, our non-iterative approach identifies all non-relevant filters in a single pass using lasso regression.

Light-weight models: Our approach achieves a high compression ratio, resulting in faster training times and real-time inference without compromising accuracy. This is particularly important for deploying models on low-powered edge devices.

Fewer data requirements: ETL leverages existing filters from primary models to train models on tasks with limited available data.

To validate the effectiveness of our proposed approach, we conducted extensive experiments comparing it to the standard transfer learning algorithm. We present the results on multiple face datasets, encompassing secondary tasks such as age, gender, emotions, and head pose, where large datasets are lacking.

7.2 Methodology

In Chapter 6, we explored the identification of cross-task aware filters (CRAFTs) within the convolutional layers of deep face networks. This section focuses on leveraging CRAFTs to efficiently prune and train deep face models for auxiliary tasks. To achieve this, we introduce characteristic curves for each convolutional layer within the model, enabling us to assess the layer's affinity to the auxiliary task. We automatically establish a sparsity constant for each layer by determining a γ parameter for each curve. Finally, we present our one-shot pruning and transfer-learning algorithm, capitalizing on these insights.

7.2.1 Characteristic Curves

In Chapter 6, Equation 6.3 presented the determination of cross-task aware filters (CRAFTs) using the sparsity parameter λ. However, selecting an appropriate λ for each layer is challenging, as changes in λ do not directly impact the error. This section introduces a global hyperparameter that balances sparsity and error, eliminating the need for sensitivity parameters specific to each layer.

We construct "characteristic curves" for each layer to investigate the relationship between error and sparsity. These curves depict the sparsity of filters against the error for different values of λ. We train multiple LASSO models, varying λ such that the largest λ renders all coefficients zero and selecting the remaining λ values using a geometric sequence where the largest λ value is 1E+4 times the smallest λ value. Section 7.4 illustrates characteristic curves for various tasks. Notably, we observe that

certain curves exhibit a flat-bottomed shape, indicating minimal changes in error as sparsity increases. Moreover, the shape of the characteristic curves varies depending on the specific secondary task.

To determine the knee point on the characteristic curve (k), corresponding to the λ value maximizing sparsity while maintaining the error within acceptable limits, we introduce a global parameter γ. This parameter represents the maximum allowable increase in error.

$$\begin{aligned} k = &\min_i num(i) \text{ such that} \\ &\text{RMSE}(i) - \min(r) < \gamma(\max(r) - \min(r)) \end{aligned} \tag{7.1}$$

Equation 7.1 outlines the calculation of k, where i represents the λ value at a particular point on the characteristic curve. The knee point k is determined as the minimum i value for which the difference between the RMS error ($RMSE(i)$) and the minimum and maximum RMS error values ($\min(r)$ and $\max(r)$, respectively) falls below γ times the range between $\min(r)$ and $\max(r)$. The choice of γ influences the size and error of the transferred model, with higher values resulting in larger models but lower errors, and vice versa. Finally, we calculate the λ value at the knee point k for each layer using the selected γ parameter.

7.2.1.1 Predicting discrete attributes

Our method can be extended to study discrete binary attributes like gender. We learn continuous lasso regression between 0 and 1 on the training set and learn the best threshold on the validation set. We apply a threshold to the regression results on the test set to get binary classes. The characteristic curve of each layer is obtained by plotting the number of filters versus accuracy on the test set. It is better to have a lower number of filters with higher accuracies.

7.2.2 Pruning the Model

In this methodological step, we focus on discarding filters that were not selected by the LASSO model with $\lambda = k$ for each convolutional layer of the model. We adopt the procedure outlined in [59] as our guiding framework. Let us consider the l^{th} convolutional layer, denoted as ϕ, with a kernel size of $k_l \times k_l$. The weight matrix W associated with this layer has dimensions $C_{l+1} \times C_l \times k_l \times k_l$, where C_l represents the number of input channels for layer l and C_{l+1} denotes either the number of output channels for layer l or the number of input channels for layer $l+1$. Similarly, the weight matrix of the $(l+1)^{th}$ layer is represented as $W_{l+1} \in \mathbb{R}^{C_{l+2} \times C_{l+1} \times k_{l+1} \times k_{l+1}}$.

To remove the i^{th} filter from layer l, we eliminate the corresponding output channel weight, denoted as $W_l[i, :, :, :]$. Additionally, the input channel weight $W_{l+1}[:, i, :, :]$ is removed from layer $l+1$. This process is performed simultaneously across all layers, based on selecting the LASSO model at the knee point k. Let β represent the weight vector obtained from the LASSO regression, and let $t \in \{1 \cdots C_{l+1}\}$ be the index of the filters chosen when $\lambda = k$, representing the non-zero coefficients of β. The updated weight vector for layer l, denoted as W_l', is given by:

Algorithm 2: Create sparse model by removing selected filters

Input: Model ϕ with L convolutional layers having weights $\{W_1^{\phi}, W_2^{\phi}, ...W_L^{\phi}\}$, regression weights $\{\beta_1, \beta_2, ...\beta_L\}$ of knee-point LASSO models for each layer

Output: Sparse model ϕ'

$\phi' \leftarrow$ copy of ϕ with weights $\{W_1^{\phi'}, W_2^{\phi'}, ...W_L^{\phi'}\}$
for *each convolutional layer l of ϕ* **do**
  $n_l \leftarrow$ number of non-zero elements of β_l
  for *each non-zero element i in β_l and j=1 to n_l* **do**
    $W_l^{\phi'}[j, :, :, :] \leftarrow \beta_l[i] W_l^{\phi}[i, :, :, :]$
    if $l < L$ **then**
      $W_{l+1}^{\phi'}[:, j, :, :] \leftarrow W_{l+1}^{\phi'}[:, i, :, :]$
    else
      Let $W_{L+1}^{\phi} \in \mathbb{R}^{C_{L+1} \times C_{L+2}}$ be the first linear layer of model ϕ
      $W_{L+1}^{\phi'}[j, :] \leftarrow W_{L+1}^{\phi}[i, :]$
    end
  end
end

$$W_l' = \beta[t] W_l[t, :, :, :] \tag{7.2}$$

For a more detailed algorithm description, please refer to Algorithm 1.

7.2.2.1 Pruning LightCNN Models

Filter pruning algorithms must be modified when applied to architectures that differ significantly from the VGG architecture. In this subsection, we describe the procedure for pruning the LightCNN-9 architecture, which introduces a specialized operation called the Max-Feature-Map (MFM) operation. The MFM 2/1 layer in LightCNN-9 consists of an underlying convolutional layer with i_l input channels and $2o_l$ output channels, along with the MFM operator that combines the output channels by taking the maximum value across channels. This ensures that the overall layer output has only o_l channels. Let X denote the convolutional layer output with dimensions $2o_l \times h \times w$. The MFM operator is defined as follows:

$$\hat{x}_{i,j}^{p} = \max(x_{i,j}^{p}, x_{i,j}^{p+o_l}) \tag{7.3}$$

Here, $\hat{x}_{i,j}^{p}$ represents the (i, j)-th element of channel k in the output. When we aim to retain a set $D = \{f_1, f_2, ..., f_d\}$ of channels out of the o_l available channels, we need to preserve both the D and $D + o_l$ output channels from the convolutional layer, as well as the corresponding input channels in the subsequent layer.

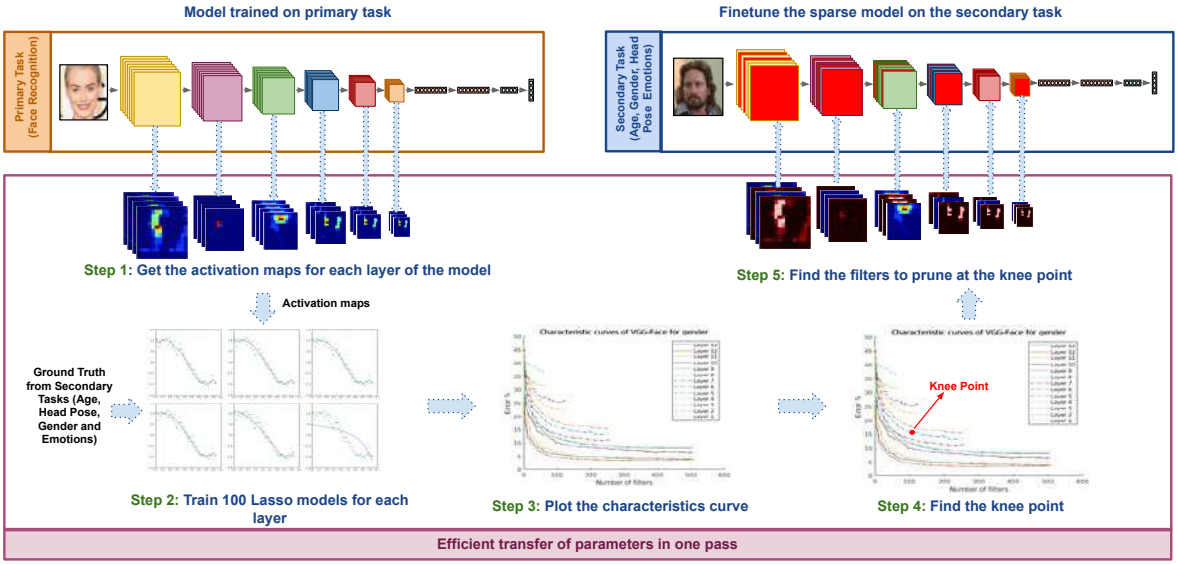


Figure 7.1: Pipeline for efficient transfer of parameters from a model trained on a primary task, such as face recognition, to a model for secondary tasks, including gender, emotion, head pose, and age, in a single pass. The Efficient Transfer Learning (ETL) technique selectively preserves task-related filters, leading to a highly sparse network that facilitates efficient training of face-related tasks.

In addition to the MFM layers, LightCNN-9 includes group layers consisting of two MFM layers. Let's consider a group layer with i_l input channels, o_l output channels, and a $k \times k$ filter size. The first MFM layer consists of a 1×1 convolutional layer with i_l input and output channels. The second MFM layer has i_l input channels, o_l output channels, and a convolutional size of $k \times k$. To retain D filters within a group, we preserve the corresponding D filters from the second MFM layer as described earlier. Additionally, we retain D filters from the first MFM layer of the following group.

7.2.3 Efficient Transfer Learning

Our complete pipeline is given in Figure 7.1. We begin with an initial model ϕ pre-trained on a primary task D_1. Let $D_2 = (I, Y)$ be the secondary task. We first pass the dataset images I through the model ϕ and collect the activations at each layer $\{X_1, X_2, ..., X_L\}$ according to Equation 6.1. We then plot the characteristics curve and find the knee-point k_l for each layer using Equation 7.1. We generate a sparse model ϕ' by keeping only filters corresponding to the non-zero coefficients of the regression weights β_l of the LASSO models with $\lambda = k_l$, according to Algorithm 1. Finally, we finetune the sparse model ϕ' on the dataset D_2 to obtain the efficient transferred model ϕ^*.

7.3 Details of Datasets and Models

This section provides a comprehensive overview of the datasets and tasks utilized in the experiments conducted in Sections 7.4 and 7.5. It also presents an overview of the deep models analyzed in this study.

7.3.1 Datasets and Tasks

This section comprehensively describes the datasets employed and the corresponding face tasks tackled in the experiments conducted. Table 6.1 provides a summary of the utilized datasets. For the gender and eyeglasses recognition task, we utilize the Celeb-A dataset [173]. This dataset encompasses over 202,000 labeled facial images with diverse attributes, including gender, race, facial hair, and accessories. We randomly select a subset of 20,000 images from this extensive dataset, ensuring an equal distribution between male and female images. The subset is divided into 10,000 training images, 5,000 validation images, and 5,000 test images. The training phase involves training regression models, while thresholds for binary classification are determined based on the validation set.

The age prediction task relies on the AgeDB dataset [181], which contains more than 15,000 images annotated with ages ranging from 1 to 101. This task is framed as a continuous prediction problem, aiming to predict the precise age of individuals. To facilitate experimentation, we partition the dataset, allocating 75% of the images for training and the remaining portion for testing. For the emotion prediction task, the AFEW-VA dataset [182, 183] is utilized. This dataset comprises clips extracted from feature films, with per-frame annotation of valence and arousal. Our experiment treats each frame as an individual image and focuses on predicting the valence attribute. 75% of the dataset is allocated for training, while the remaining portion is designated for testing. To tackle the head pose task, we employ the Annotated Facial Landmarks in the Wild (AFLW) dataset [184]. AFLW consists of a substantial collection of "in the wild" facial images and annotations of yaw, pitch, and roll attributes. Our specific experiment concentrates on predicting the yaw angle, measured in degrees. The dataset is partitioned, with 75% of the images designated for training purposes and the remaining portion reserved for testing.

7.3.2 Deep Models for Experimentation

This section provides an overview of the four models employed in our experiments, detailed in the subsequent sections.

VGG-16 Architecture: The primary model utilized in our experiments is the VGG-16 architecture, a convolutional neural network (CNN) consisting of 16 layers. The architecture comprises 13 convolutional layers followed by rectified linear unit (ReLU) activation functions. The convolutional layers predominantly employ 3x3 filters. To capture salient information and reduce spatial dimensions, max pooling layers with a 2x2 filter size and a stride of 2 are incorporated. We employ two variants of this architecture: the VGG-Face model, specifically designed for face recognition, and the VGG image recognition model trained on the Imagenet dataset.

Task	VGG ImageNet → task	VGG-Face → task	dedicated network
AFLW yaw (MSE)	2165711.5	441.09	1906.19
AgeDB age (MSE)	154.00	98.55	73.45
Celeb-A gender (Accuracy)	91.88%	96.66%	96.14
Celeb-A eyeglasses (Accuracy)	93.82%	95.38%	94.76%

Table 7.1: Comparison of best performance obtained using VGG (ImageNet) and VGG-Face with dedicated trained networks. For AFLW and AgeDB, performance is measured in MSE (lower is better). For Celeb-A, performance is measured in accuracy (higher is better)

FaceNet: FaceNet introduces a system for embedding face images into a 128-dimensional space for face verification using triplet loss. For our experiments, we utilize the pre-trained model with the nn4 architecture provided by OpenFace. This model comprises several convolutional layers followed by seven inception layers. In our investigation, we focus on the activations of the inception layers, treating each inception layer as a single convolutional layer.

LightCNN: The LightCNN paper proposes a lightweight CNN architecture designed to generate a 256-dimensional "universal face representation" for face recognition. A distinguishing feature of this architecture is the introduction of the Max-Feature-Map (MFM) layer. The paper presents various LightCNN architectures of varying sizes. For our experiment, we employ the LightCNN-9 pre-trained network, which consists of one MFM layer, four group layers, and a fully connected layer. Pooling layers are interspersed within the architecture.

7.4 Analysis of Characteristic Curves

In this section, we analyze the characteristics of various face recognition models by calculating characteristic curves for five face tasks, as described in Section 7.3. By studying these characteristic curves, we gain insights into the overlap between the primary face recognition task and secondary tasks. The results for the prediction of secondary tasks using CRAFTs are summarized in Table 7.1. The VGG-Face CRAFTs perform better than dedicated models explicitly trained for the secondary tasks.

Figure 7.2 compares characteristic curves between VGG-Face and VGG-Imagenet for four face tasks. Several observations can be made: While VGG-Face outperforms VGG-Imagenet in terms of overall performance, the shapes of the curves remain consistent. The shape of a curve reflects how secondary task information is distributed within a layer. A sharp 'elbow' in the curve, as observed in the cases of gender and eyeglasses, indicates that only a few CRAFTs are necessary, enabling the prediction of the task using a minimal number of filters. Conversely, a gradual curve like the one for valence suggests that all filters within the layer contribute to the task prediction.

The lowest mean squared error (MSE) for age prediction is 98.55, implying that ages can be predicted within $\pm$ 10 years. Table 7.1 reveals that VGG-Imagenet also yields similar results, suggesting that age

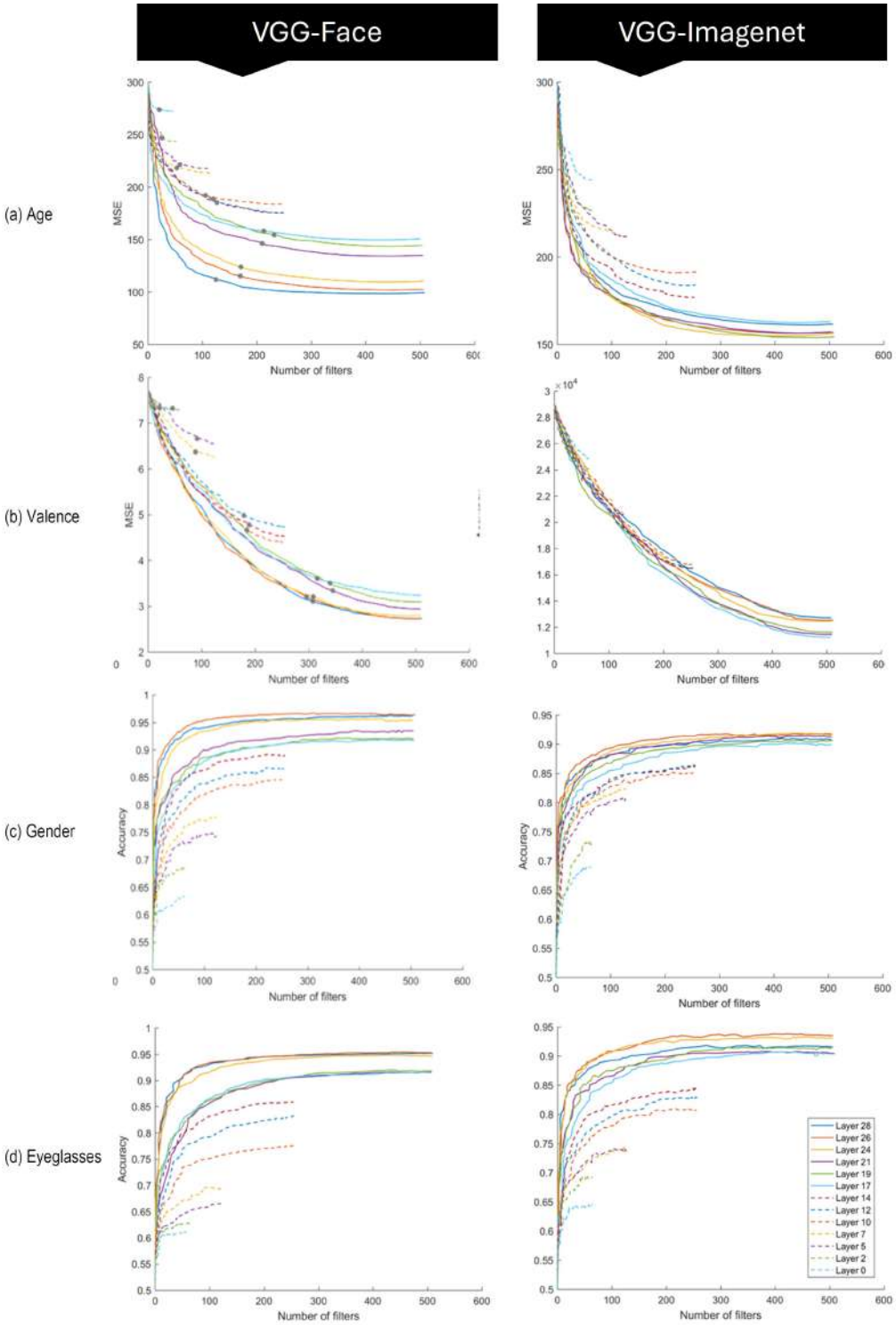


Figure 7.2: Characteristic curves of VGG-16 for various face tasks. The left column shows the characteristic curves of VGG-Face [19], while the right column displays the characteristic curves of VGG-16 trained on Imagenet [161]. Each row corresponds to a specific face task: (a) Age using the AgeDB dataset [181], (b) Valence using the AFEW-VA dataset [182, 183], (c) Gender using the Celeb-A dataset [173], and (d) Presence or absence of eyeglasses using the Celeb-A dataset [173]. The error of age and valence is measured in MSE (lower is better). Gender and eye glasses are binary attributes and are measured as accuracy (higher is better)

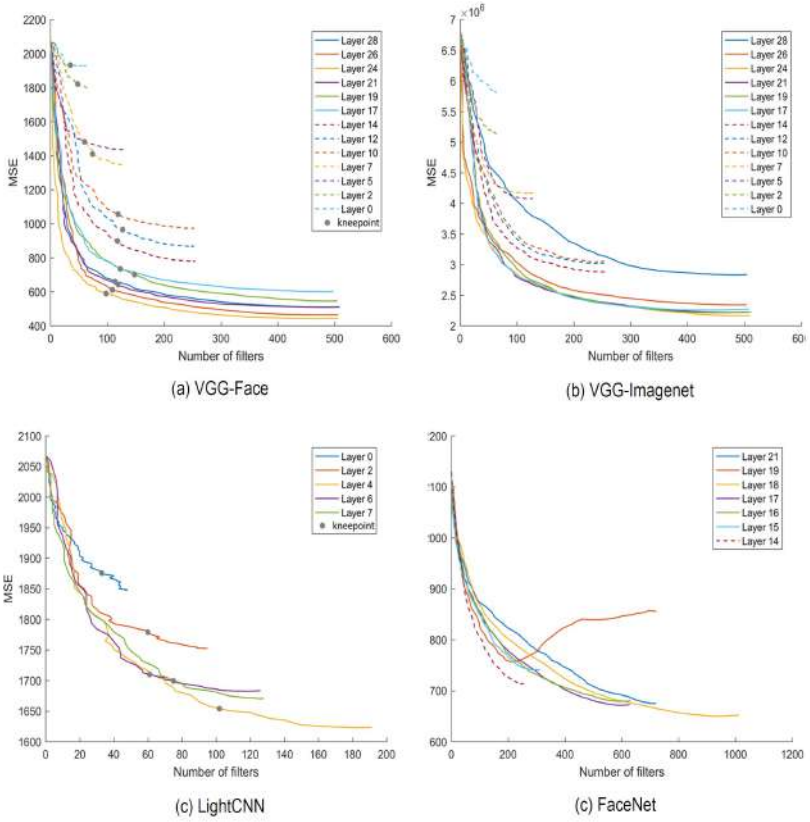


Figure 7.3: Characteristic curves of various deep models for the yaw task using the AFLW dataset [184] The curves correspond to the following models: (a) VGG-Face [19], (b) VGG trained on Imagenet [161], (c) LightCNN [51], and (d) FaceNet. [185]. The error is measured in MSE (lower is better).

might not strongly correlate with the face recognition task. Previous age prediction methods often relied on facial geometry or texture, which could also be attributes learned by the image recognition task.

Emotional valence prediction proves to be a highly challenging task closely tied to facial expressions. The discrepancy of around 10^3 between VGG-Face and VGG-Imagenet indicates that face recognition captures crucial transferable information for emotion prediction, which is not learned by generic image recognition. Interestingly, the highest accuracy obtained from VGG-Face originates from the last few layers, whereas it occurs in the middle layers for VGG-Imagenet. This discrepancy suggests that the transferable information is most concentrated in these respective layers, with a noticeable divergence thereafter. The accuracy achieved in detecting eyeglasses is comparable for both VGG-Face and VGG-Imagenet, as eyeglasses are considered objects and share common visual characteristics. Figure 7.3 further presents the characteristic curves for the head pose task across four different models, demonstrating the versatility of our algorithm in analyzing diverse CNN architectures.

7.5 Results

This section presents the results obtained from our Efficient Transfer Learning (ETL) procedure utilizing functional pruning. We aim to demonstrate ETL's efficacy in achieving rapid and parameter-efficient transfer learning for face tasks, surpassing the baseline transfer learning method. Through experiments conducted on multiple face datasets, we showcase how the ETL models preserve up to 99.5% of the baseline accuracy while significantly reducing the size of the baseline model by 97%. This reduction translates to a remarkable 94% decrease in CPU inference time, highlighting the efficiency gains achieved by the ETL approach. Our experimental setup is given in Section 7.3.

7.5.1 Evaluation Metrics

In our experiments, we employ multiple evaluation metrics to assess and compare the performance of our methods. These metrics include accuracy, FLOPs (floating-point operations) for a forward pass, the number of parameters in the model, and inference time.

FLOPs are calculated as the total number of multiplication operations required during a forward pass. For a model comprising M convolutional layers and N linear layers, the FLOPs are computed as the sum of the FLOPs for each layer. Specifically, the FLOPs for a convolutional layer l are determined by the number of output channels o_l, input channels i_l, kernel size $k_l \times k_l$, and the activation map dimensions $o_l \times w_l \times h_l$. Similarly, the FLOPs for a linear layer l depend on the number of input features n_input_l and output features n_output_l.

Task	Model	Accuracy (%)	FLOPs ($\times 10^{11}$)	Size (MB)	Inference Time (ms)
Gender	Baseline	**97.06**	7.38	4.84E+02	5.469
	ETL	96.62 (0.5%)	**3.26** (55.8%)	**2.47E01** (95.5%)	**0.528** (90.3%)
Emotion	Baseline	**65.92**	7.38	4.84E+02	5.527
	ETL	55.16 (16.3%)	**2.06** (72.1%)	**1.56E01** (97.2%)	**0.373** (94.2%)
Head Pose	Baseline	**95.7**	7.38	4.84E+02	5.5169
	ETL	94.58 (1.2%)	**4.25** (42.4%)	**3.23E01** (94.2%)	**0.4626** (91.6%)
Age	Baseline	**51.8**	7.38	4.84E+02	5.313
	ETL	46.96 (9.3%)	**4.16** (43.6%)	**3.17E01** (94.3%)	**0.47** (91.2%)

Table 7.2: The table shows the comparison of ETL (sparse fine tuning) with baseline transfer learning (full fine tuning) in terms of accuracy, FLOPS, size, and inference time per image on CPU for different face tasks, including gender, emotion, head pose, and age.The percentage reduction in metrics is given in the brackets. We observe a significant drop in model size, which leads to faster inference time with a slight loss in the model's accuracy.

$$FLOPS = \sum_{i=1}^{M} ConvFLOPS_i + \sum_{j=1}^{N} LinFLOPS_j \tag{7.4}$$

$$ConvFLOPS_l = o_l \times i_l \times k_l \times k_l \times w_l \times h_l \tag{7.5}$$

$$LinFLOPS_l = n\_input_l \times n\_output_l \tag{7.6}$$

The size of the network refers to the cumulative size of its stored parameters, including weights and biases across different layers. This size affects the resources and training time required for deep model training. Inference time denotes the duration needed to predict the output of a single image during testing on a CPU. Lower inference times indicate the feasibility of deploying the deep model on devices with limited resources.

7.5.2 Results of Efficient Transfer Learning

We compare our proposed ETL procedure with baseline transfer learning for the tasks of gender, emotion, head pose and age. Table 7.2 summarizes the results of ETL with $\gamma = 0.01$. We observe a significant reduction of up to 97% in size and 72% in the computational complexity without much loss of accuracy, as we can remove many convolutional filters from each layer without impacting the performance. We observe from Figure 7.2 that the characteristics curves for gender and head pose are flat, indicating that most of the information about secondary tasks exists in very few filters of each convolutional layer of the VGG-Face network. Thus, the performance of the ETL models reaches up to 99.9% of the baseline models. The characteristics curves for emotion and age are not as flat, resulting in a higher performance drop.

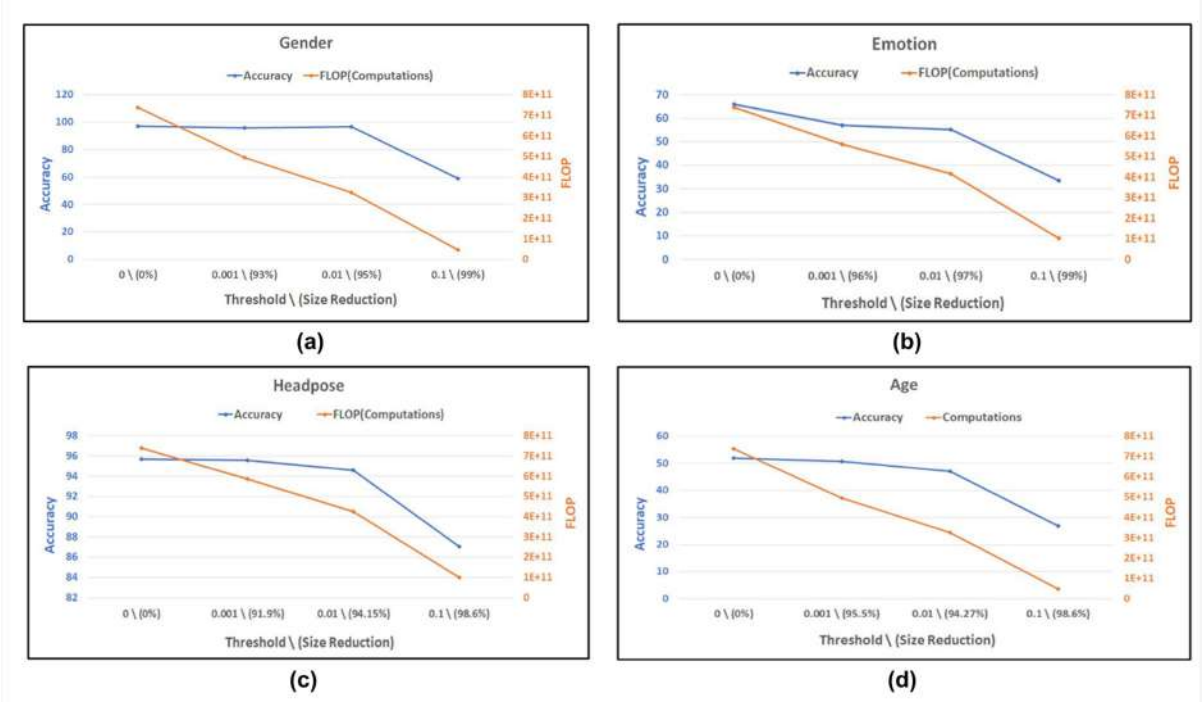


Figure 7.4: The four figures show the accuracy and computational complexity for the VGG-Face model pruned with different thresholds(γ). For each task, we varied the threshold from 0.1 to 0.001. A threshold of 0 indicates an unpruned network, and a threshold of 0.1 corresponds to a highly sparse network with 99% of filters pruned. We have shown the accuracy on the left axis and computational cost (number of flops) on the right axis. The X-axis shows the percentage reduction in size along with the respective threshold values on the X-axis. The four figures correspond to the different face tasks: a) Gender b) Emotion c) Head pose d) Age.

7.5.3 Influence of γ Parameter

The value of γ controls the model compactness; higher γ results in fewer parameters at a possible cost to the performance. To explore this trade-off, we consider different γ values and compare their effect on accuracy, FLOPs and size to the baseline. Figure 7.4 presents our results. Using VGGFace as the base network, we applied our ETL procedure for four tasks: gender, emotion, head pose and age with γ values of 0, 0.1, 0.01 and 0.001. The figure shows that the FLOPs reduce monotonically as γ changes. We observe that as γ increases, the model size and computational complexity reduces significantly with only a minor reduction of accuracy. Thus, the threshold is a reliable way to tune the ETL algorithm and get the desired compromise between compression ratio and accuracy. Our experiments observed a γ value of 0.01 as ideal.

7.5.4 Training and Inference Time

Figure 7.5(a) shows the training time per epoch for different values of γ, which reduces with an increase in γ as fewer filters get chosen. We observe a per-epoch reduction of 32% for $\gamma = 0.01$ for

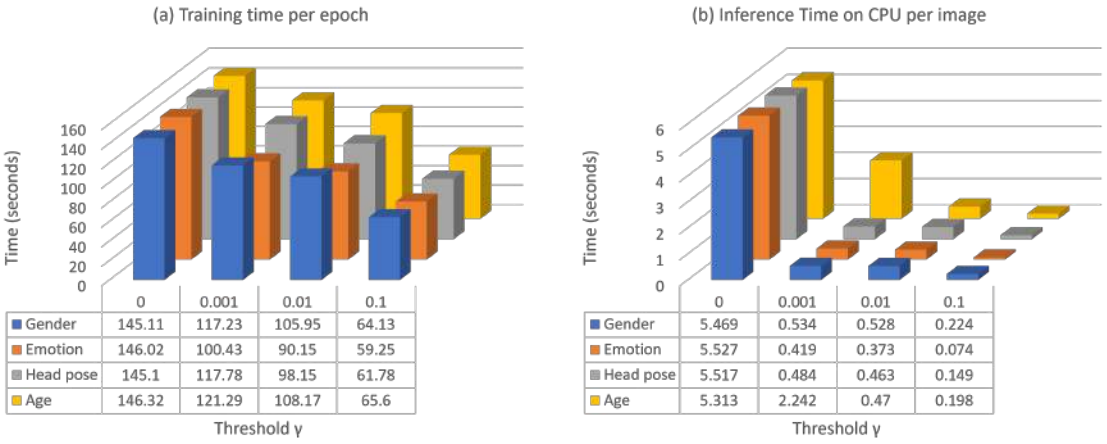


Figure 7.5: Inference and training times for ETL. (a) Training time per epoch on GPU for threshold values between 0 to 0.1 (b) Inference time on CPU per image at different threshold values. The increase in threshold value results in higher real-time performance.

head pose. This speeds up the finetuning step, resulting in accelerated transfer learning. Figure 7.5(b) presents the inference time on CPU per image at different γ values. A dramatic decrease in inference time of 90% enables the ETL models to perform inference in real-time, which is important for deploying on low-powered edge devices.

7.6 Summary

In this chapter, we have introduced a practical approach to gain deeper insights into face networks and the face recognition task by utilizing Cross-Task Aware Filters (CRAFTs). We have enhanced our understanding of these networks by studying various face attributes and their connection to face recognition.

Moreover, we have demonstrated the practical application of CRAFTs in Efficient Transfer Learning (ETL). ETL offers a streamlined procedure for transfer learning of face tasks, employing functional pruning techniques on deep face models. This approach enables the creation of lightweight and accurate models for face-related tasks, even with limited datasets. Notably, ETL requires only a single tunable hyperparameter, simplifying the process and allowing for predictable and user-friendly implementation.

One key advantage of ETL is its ability to achieve high compression ratios, making real-time inference on CPUs feasible. This capability is crucial for deploying deep models on resource-constrained edge devices, expanding the practicality and accessibility of face recognition technology.

PART IV

Discovering Salient Facial Features

Chapter 8

An Overview of Saliency Maps

Previous sections have provided insights into understanding facial features through visualization and concept matching. In this chapter, we shift our focus to determining crucial facial features that significantly influence the outcomes of neural networks. Notably, we explore utilizing saliency maps to assess the importance of input image features, specifically pixels, in the face domain. This chapter provides an overview of different saliency maps and their application in facial analysis. We also discuss evaluation protocols to assess saliency maps' quality and effectiveness.

8.1 Introduction

The prediction of a model depends on the features of the input sample, and not all features have equal importance. Saliency or attribution maps attribute the model's prediction to specific input features. Typically represented as heatmaps over the input sample pixels, these maps highlight pixels that positively or negatively contribute to the final prediction. The primary goal of saliency maps is to provide insight into the model's decision-making process. While many works focus on saliency maps, only a few are tailored explicitly to face networks. General saliency algorithms often require modifications to work effectively on face images. Existing algorithms may not offer new information, as highlighted by GradCAM applied to various faces with different attributes (Figure 1).

In this chapter, we explore the properties of desirable interpretable saliency maps. We provide an overview of different algorithm classes for generating saliency maps and examine their applications in enhancing human decision-making. Additionally, we discuss objective metrics used to measure the fidelity of saliency maps.

8.1.1 Utility of Saliency Maps in the Face Domain

Saliency maps play a crucial role in understanding the decision-making process of deep models. By visualizing the most salient input features, we can gain insights into the rationale behind the model's predictions. Additionally, saliency maps are valuable tools for uncovering implicit biases in a model. When a prediction is incorrect, saliency maps can help identify the specific features that led to the error. An illustrative example by Selvaraju et al. using GradCAM [6] showed that biased models tend to rely

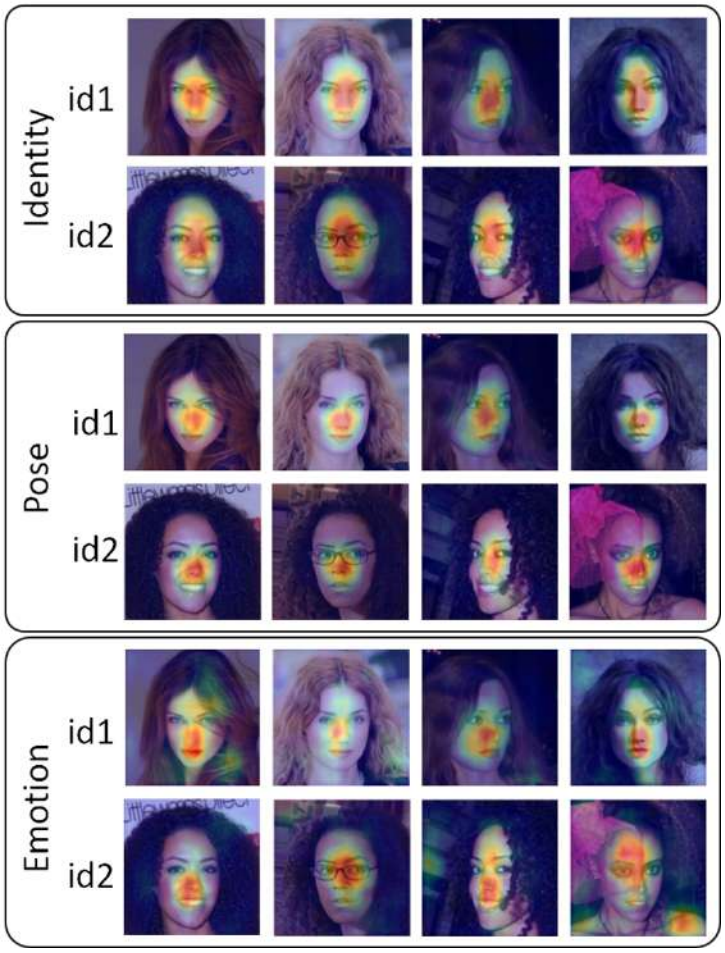


Figure 8.1: Saliency algorithms tailored for general object recognition struggle to produce meaningful results when applied to faces. In this image, we visualize GradCAM [6] applied to three models trained on distinct face tasks. Despite variations in identity, expression, and pose, the algorithm predominantly emphasizes the center of the face. Original images sourced from the VGG-Face dataset [19].

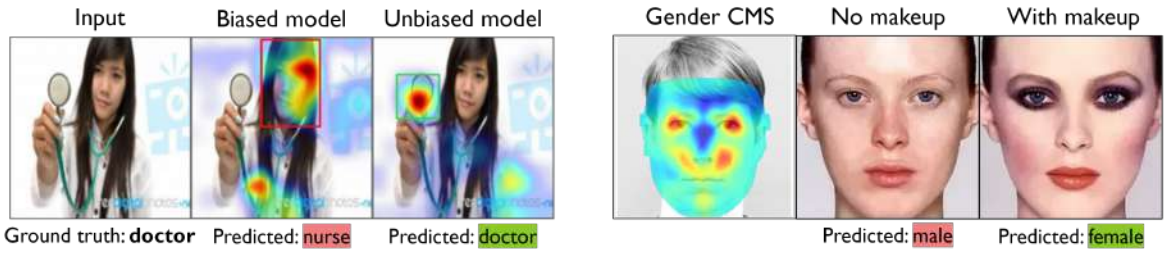


Figure 8.2: The utility of saliency maps in comprehending incorrect predictions is demonstrated through two instances of biased predictions. *(Left:)* GradCAM highlights how a biased model bases its occupation classification on facial and appearance features, while an unbiased model incorporates the tools in the input image (Image adapted from [6]). *(Right:)* In the case of gender classification, a biased model misclassifies a woman's face as male due to the absence of makeup. The CMS (saliency) map reveals that the model relies heavily on the eyes for gender classification (Image adapted from [186]).

more on facial and appearance features when classifying occupations, while unbiased models focus on other relevant attributes (Figure 8.2).

Furthermore, saliency maps facilitate a deeper exploration of the nature of a task and highlight discriminative parts of an image that are relevant to a specific class. In facial recognition, certain facial features might resemble another person's, and saliency maps can reveal which features are essential for recognition. Similarly, in emotion recognition, saliency maps can show that the model concentrates on specific regions like the eyes while recognizing emotions, even if other parts of the face contradict the emotion label.

8.1.2 Considerations and Desiderata of Saliency Maps

There are three considerations for saliency algorithms: *helpfulness* [187, 188], *trustworthiness* [189] and *fidelity*.

Several studies have suggested empirical methods for measuring the helpfulness of saliency maps and explanations in general. Doshi-Velez and Kim [187, 188] mention five factors related to the usefulness of explanations to humans, in essence measuring the simplicity of the explanations and how intuitive they are to interpret. Silva et al. [189] introduce the 'three Cs of interpretability', which deal with the trustworthiness of an explanation, in addition to its simplicity:

Completeness: Users should be able to apply the explanation to cases where it is known and can validate it.

Correctness : The explanation should be accurate.

Compactness: The explanation should be succinct. This condition was related to rule-based explanations. Regarding saliency maps, this can mean that the level of detail and number of discontinuous chunks should be manageable [188].

The fidelity of an explanation may be evaluated by the three axioms provided by Sundarajan et al. [190]:

Sensitivity: If some input samples differ only in one feature but produce different predictions, the attribution of that feature cannot be zero. Similarly, if the inclusion or exclusion of a feature does not change the prediction, the attribution of that feature must be zero.

Implementation Invariance: Two models are functionally invariant if they produce the same prediction for the same inputs. This axiom states that the attribution of two functionally invariant models should be the same for the same inputs. In other words, the explanation should not be implementation-dependent.

Completeness: Different from the previous condition, completeness states that the attributions of various features of an input image should sum up to the difference in prediction between the input image and a baseline image with no relevant features.

We discuss objective metrics for fidelity in Section 8.6.

These considerations and evaluation protocols ensure saliency maps' reliability, interpretability, and accuracy, enabling their practical application in the face domain and beyond.

8.1.3 Classification of Saliency Maps

We broadly classify existing saliency algorithms into perturbation-based and backpropagation-based algorithms, of which gradient-based algorithms are a subclass. Figure 8.3 compares saliency maps for faces. Next, we present an overview of some popular saliency algorithm classes.

8.2 Perturbation-Based Saliency Maps

Perturbation-based methods find the saliency of the input features compared to a baseline sample by perturbing input features and observing the effect on output. These algorithms are architecture and implementation-agnostic but computationally expensive, as they pass the input sample through the model several times with different perturbations.

8.2.1 Occlusion Maps

Occlusion maps [134] systematically slide a window over an input sample, flipping the pixels to the baseline image and observing the change in output class confidence. The saliency of a patch is given by:

$$H_p = f(x) - f(x \odot (1 - p)) \tag{8.1}$$

where x is the input sample, p is the patch and $f(\cdot)$ is the class confidence. Occlusion maps produce interpretable and intuitive maps whose coarseness can be controlled by the window size. Zhong and Deng [137] used occlusions at predefined locations to calculate the importance of coarse facial features for face recognition. They also highlighted the difference between similar faces by occluding the same features of two images at once and calculating the feature distance [196]. John et al. systematically occluded the input face image and projected the resulting saliency map onto a neutral frontal face [186]. This 'canonicalization' procedure allowed them to collate multiple saliency maps to obtain model-level saliency maps, We discuss this method in Chapter 9.

8.2.2 Shapley Values

Shapley values [197] provide a theoretical framework for perturbation-based methods which obeys the axioms of completeness, symmetry (two features must be attributed equally if they have an equal effect on the output) and sensitivity. Shapley values consider the input features to be players in a coalition game where the outcome is the payout. The algorithm fairly distributes the payout between each player

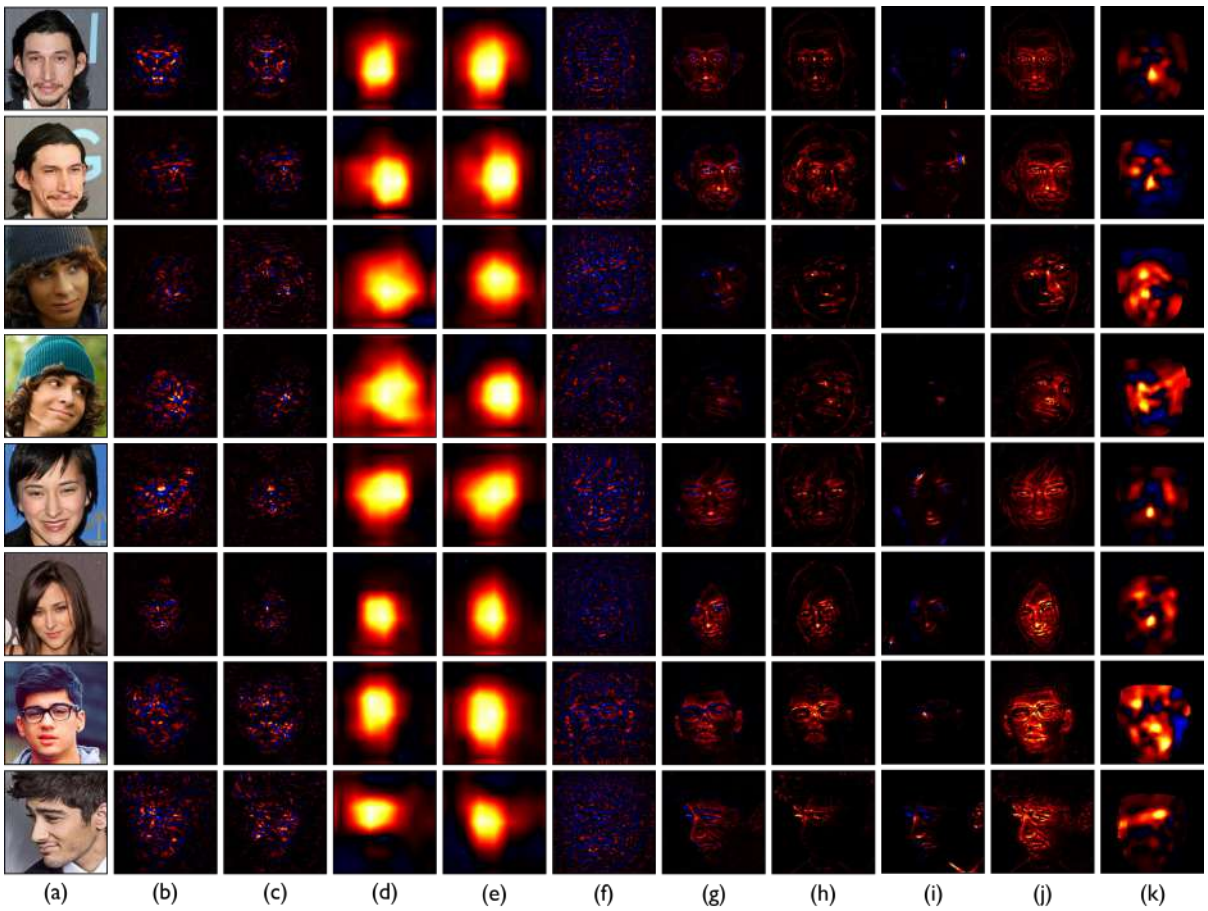


Figure 8.3: Comparison of various saliency visualization methods on the VGG-Face model [19] for the task of face recognition. Each image shows the ground truth class as the target. The color scale indicates positive saliency from red to yellow, negative saliency from blue to cyan, and neutral regions in black. (a) Original image; (b) Smoothgrad [191]; (c) Integrated gradients [190]; (d) GradCAM [6]; (e) ScoreCAM [192]; (f) Deconvolution [134]; (g) Guided Backpropagation [6]; (h) LRP [193]; (i) DeepLIFT [194]; (j) Excitation Backprop [195]; (k) Canonical Saliency maps [186]. The original images are taken from the VGG-Face dataset [19]. Rows (1, 2), (3, 4), (5, 6) and (7, 8) have the same identity. Original images are taken from the VGG-Face dataset [19].

based on their contribution. Given a model f and n features, the payout for a feature i is computed as:

$$\phi_i(f) = \sum_{S \subseteq \{1 \ldots n\}/i} \frac{|S|!(n - |S| - 1)!}{n!} \big(f(S \cup \{i\}) - f(S)\big) \tag{8.2}$$

where S is a subset of the features and x is the vector of feature values of the input instance. While Shapley values satisfy many theoretical constraints, they are not practical to calculate for input with many features like image data. Variations such as SHAP and Kernel SHAP [198] make the computation more practical.

8.2.3 Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanation (LIME) [199] uses local surrogate models to explain individual predictions. LIME generates a dataset by perturbing the input sample's features and recording the model's corresponding predictions, given a model and an input sample. LIME then trains an interpretable model on this generated dataset. This learned model is a good approximation of the model locally but not globally (local fidelity).

8.3 Backpropagation-Based Saliency Maps

Backpropagation-based algorithms start with a trained model and a desired output. A high gradient at the desired output is then backpropagated to the model elements, starting with the layers closest to the output, all the way down to the input. This procedure finally assigns a score to each input feature based on their contribution to the output. They are easy to calculate as they require only a single backward pass. However, they tend not to be implementation-agnostic by their very nature.

8.3.1 Layer-wise Relevance Propagation

Layer-wise relevance propagation (LRP) [193] uses the structure of layered neural networks to assign prediction score to elements of a layer, ensuring that the relevance flowing into a unit should equal the relevance flowing out to a lower layer. If we observe column (h) of Figure 8.3, we notice that the output of LRP on faces visually resembles guided backpropagation. LRP tends to highlight image edges, and in most cases, it highlights all facial features equally. The high number of separate salient areas makes it difficult to interpret.

8.3.2 Excitation Backprop

Excitation Backprop or EBP [195] uses a probabilistic winner-takes-all formulation. Let the relevance of a prediction be specified by the prior distribution $P(A_0)$ over the output neurons. Let $P(A_t|A_{t-1})$ be the probability of selecting neuron A_t in layer t as the winning neuron given A_{t-1}

is selected in the layer before. We calculate the marginal winning probability as:

$$P(a_i) = \sum_{a_j \in P_i} P(a_i|a_j)P(a_j) \tag{8.3}$$

which gives the relevance of each neuron. Here, a_i denotes a specific neuron, and A_t refers to a variable over the neurons). This work specifies the conditional winning probability based on the weights between each neuron and the activation. Contrastive EBP (c-EBP) is a variation that calculates the contrastive saliency between pairs of classes. Column (j) of Figure 8.3 shows the results of applying unmodified EBP on face images. We observe that it highlights almost the entire face. Thus, unmodified EBP does not give actionable insights on face images. Researchers have proposed modified versions of EBP designed to work on faces. Castanon and Byrne [200] used a variant of Excitation Backprop' (EBP) and 'contrastive Excitation Backprop (cEBP) [195] called 'truncated cEBP' to compute the network saliency on faces, where the network attention signal is propagated from the output neurons back to the input pixels. Williford et al. [201] focused on explaining the matches returned by a facial matcher to understand why a probe was matched with one identity over another. The unit for explanation is a triplet of (probe, matching face and non-matching face). They adapted EBP, cEBP and tcEBP by using triplet loss instead of cross-entropy loss.

8.3.3 DeepLIFT

DeepLIFT [194] breaks down the difference between a neuron's output and reference output to all the upstream neurons connected to it such that the sum of contributions equals the difference between output and reference output. Column (i) of Figure 8.3 shows DeepLIFT applied to faces. Compared to LRP, DeepLIFT has fewer salient areas highlighted and thus is easier to interpret. However, the results do not seem to agree with other saliency map methods.

8.4 Gradient-Based Saliency Maps

The gradient of the output with respect to an input pixel represents how much difference a tiny change in the pixel would make to the output. Thus it may be used to highlight salient pixels [135]. Gradient-based saliency maps are noisy and discontinuous due to the 'shattered gradient effect' [202]. For deep rectifier networks, the discontinuities in the gradient increase exponentially with number of layers. Also, the gradient values change much more rapidly than the corresponding change in input [203], adding more noise to the heatmap. Moreover, these methods are not implementation-independent.

8.4.1 Variations of Gradient-Based Saliency Maps

Smilkov et al. proposed an improvement called 'SmoothGrad' [191], which attempted to reduce the noise by averaging over several saliency maps of the same input image, each produced after adding

some random noise to the input image. Column (b) of Figure 8.3 shows that the results are highly discontinuous and challenging to interpret even after smoothing. Gradient-based methods break the axiom of 'Sensitivity' as the prediction function may flatten at the input and have zero gradients despite the function value at the input being different from the baseline. Sundararajan et al. [190] proposed 'Integrated Gradients', which preserves sensitivity. They start with a baseline image and calculate the gradient at equally spaced intervals on the straight-line path between the baseline image and the input image. They then integrate these gradients to get the final saliency heatmap. The integrated gradient along the i^th dimension is given by:

$$H_i = (x_i - x_i^0) \int_{\alpha=0}^{1} \frac{\partial f(x^0 + \alpha(x - x^0))}{\partial x_i} d\alpha \tag{8.4}$$

where x is the input sample and x^0 is the baseline image. In practice, integrated gradients look similar to SmoothGrad with many disparate areas (Column c). Although there are theoretical guarantees, our experiments showed it to not be as useful for the face domain, where one object dominates the entire image.

8.4.2 Class Activation Mapping

Class Activation Mapping (CAM) is a category of gradient-based saliency algorithms which focuses on localizing the objects of importance in the input image. Zhou et al. [204] proposed highlighting salient objects without discontinuities for architectures where the feature maps directly preceded the softmax layers. Selvaraju et al. [6] extended the algorithm to cover a wide range of architectures in their popular method, '*Grad-CAM*'. Given a class c and an output neuron y^c, it sums the weighted feature maps A_k of a layer of interest:

$$H^c = ReLU\Big(\sum_k \alpha_k^c A^k\Big) \tag{8.5}$$

where each feature map is weighted by the gradient of the class output with respect to the activation maps as follows:

$$\alpha_k^c = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{8.6}$$

Finally, the map is upsampled using bilinear interpolation to match the input image's size. Many extensions of Grad-CAM exist, such as GradCAM++ [205] and ScoreCAM [192]. Columns (d) and (e) of Figure 8.3 show the results Grad-CAM and ScoreCAM on faces. Although they successfully localize the object of interest, they do not provide finer details, rendering them not as useful for face images.

8.5 Enhancing Human Decision-Making through Saliency Maps

Saliency maps play a crucial role in explaining model decisions, but they can also go beyond mere explanation by assisting humans in making better decisions and verifying model outputs. We investigate

how saliency maps can aid humans in specific face processing tasks and improve the accuracy of their judgments.

8.5.1 Saliency Maps as Decision-Making Aids

In this context, we explore explainability methods that go beyond explaining model behavior to assist humans in verifying and improving decisions. Zee et al. [206] utilized existing explainability techniques to enhance human face recognition performance, specifically for distinguishing between similar-looking celebrity faces. They trained CNNs for recognition and verification tasks involving these identities and used explainability visualizations to identify key facial regions responsible for identity changes. By instructing novice participants to focus on specific areas like the forehead and cheekbones, the participants achieved higher accuracy in distinguishing between identities.

Another study by Zhong and Deng [196] designed a visualization method to aid human evaluators in identifying individuals attempting to breach biometric systems using similar-looking faces. They systematically occluded portions of aligned face pairs and mapped the resulting drop in cosine distance to a heatmap. Normalizing this heatmap with a learned threshold highlighted differences in negative pairs while not affecting positive pairs. This visualization allowed human evaluators to increase their accuracy from 76% to 83%. These findings demonstrate the potential of explainability methods in improving human decision-making and verification tasks.

8.5.2 Optimal Detail in Saliency Maps for Human Interpretability

In Figure 8.3, we compare different saliency map methods applied to the VGG-Face model for face recognition. Each algorithm highlights varying levels of detail. Integrated Gradients, Guided Backprop, and DeepLIFT produce fine-grained maps akin to edge detectors. On the other hand, class activation mapping-based approaches yield coarser results, typically using layers known for visualizations. Class-level saliency maps are less informative for face images since they usually contain only one object (i.e., the face). Algorithms displaying mid-level detail, such as EBP and CSM, appear to strike a balance, effectively capturing the relevant facial features at an appropriate size. Hence, careful attention to the level of detail is essential when designing saliency algorithms for faces.

8.5.3 User Survey: Assessing the Most Useful Type of Interpretability

In this thesis, we have explored various explainability algorithms aimed at achieving correctness and clarity. However, a domain gap often exists between the developers (practitioners) and users (consumers) of these explainable deep learning methods. Consumers may lack in-depth knowledge of algorithms and face challenges in making accurate conclusions about their models. To bridge this gap, we conducted a user study to assess the effectiveness of popular explainability strategies: saliency maps, feature inversion, and feature visualization.

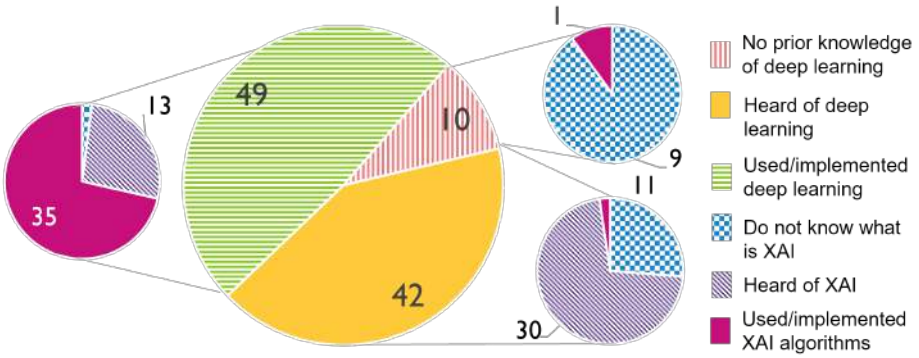


Figure 8.4: Demographic of participants who responded to the survey in our user study.

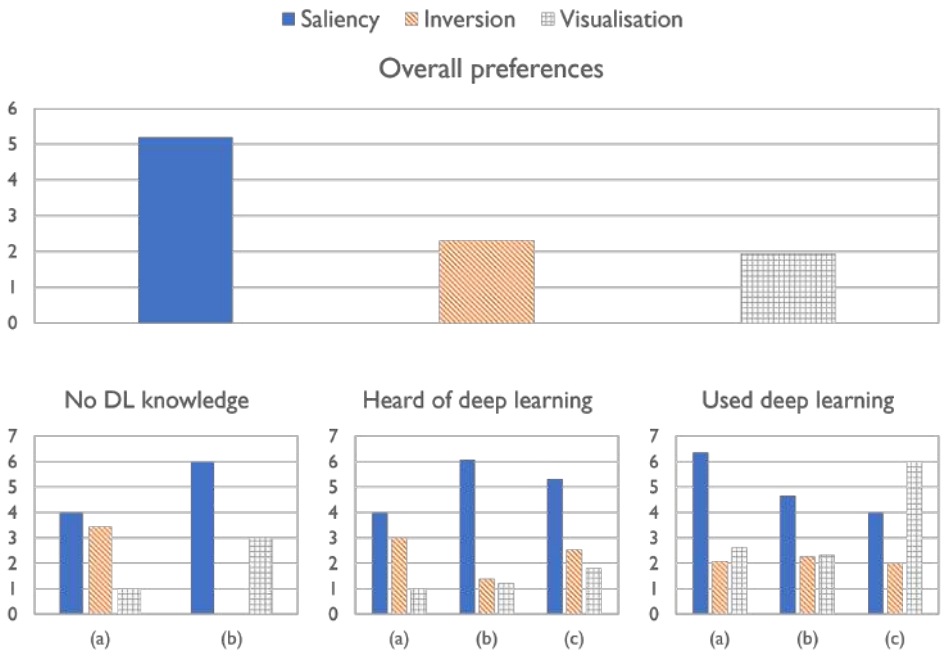


Figure 8.5: User study results: Aggregated preference and preferences categorized by participants' familiarity with deep learning. Top row: Overall aggregated preference. Bottom row: Preferences of participants based on their familiarity with deep learning, including (a) participants unfamiliar with explainability, (b) participants aware of the need for explainability but with no prior usage, and (c) participants with experience using or implementing explainability algorithms.

Using VGG16 models trained for gender, age, and expression, we generated explanations for eight input images and presented them to 101 participants, including deep learning practitioners and users of explainability methods. The participants were asked to select the explanation(s) that best helped them understand the model predictions. Results depicted in Figure 8.5 revealed that saliency maps were preferred by most respondents, except those who were experienced in using or implementing explainability algorithms. Additionally, feature inversion was favored over feature visualization in many cases, except by a group with expertise in deep learning and explainability algorithms, who preferred feature visualization.

The outcomes highlight a preference gap between practitioners and consumers of explainability algorithms. Saliency maps seem more beneficial for an overall model view, while feature visualization provides a valuable internal perspective applicable to model developers. Moreover, saliency maps may offer greater interpretability than feature visualization and feature inversion. These findings underscore the importance of considering usability and including end consumers in developing explainability methods tailored for specific face processing tasks. By addressing these usability concerns, we can enhance the adoption and practicality of explainability techniques for a broader audience.

8.6 Evaluation Protocols for Saliency Maps

The fidelity of a saliency map is often assessed by comparing the attribution of each input feature to a 'baseline input,' which represents a neutral or absent input feature. This comparison allows us to determine the importance of individual features for the model's prediction. In this section, we introduce objective metrics used to measure the fidelity of saliency maps. These metrics help evaluate the accuracy and reliability of the saliency maps in explaining model decisions.

Mask and evaluate: Chattopadhyay et al. [205] proposed a protocol to measure the impact of unimportant areas in an input image using saliency maps. However, this method is not well-suited for faces, as face images typically have a single object at the center, and face recognition models focus on different facial parts. To address this limitation, John et al. [186] introduced a modified approach for face images. Instead of masking unimportant areas, they mask important parts of the image and observe the resulting drop in prediction confidence. This adaptation allows for a more accurate assessment of saliency in face images. Additionally, John et al. normalized the sum of heatmap pixels to prevent saliency algorithms from artificially covering a large area to manipulate the results.

Hiding game: This approach is a variant of the 'Mask and evaluate' technique, where pixels are flipped gradually based on their saliency [200]. This process is also known as 'pixel flipping' [207] or 'insertion and deletion metrics' [192]. In the insertion metric, salient pixels are added gradually to a baseline image, and the rate of confidence increase indicates the quality of the saliency map. Conversely, in the deletion metric, salient pixels from the input image are gradually replaced with

baseline image pixels, and the speed at which the confidence decreases is measured compared to other saliency methods or random pixel flipping.

Pointing game: This is a supervised test that evaluates the localization accuracy of a saliency method for relevant regions in an image. The process involves extracting the maximum point from the saliency map and verifying if it falls within the ground truth bounding box of the object [195]. Wang et al. [192] expanded this measure by summing all heatmap pixels inside the object's bounding box. However, a major limitation of this method is that it may not align with human expectations if the model's ground truth reasoning differs. Additionally, it is more suitable for simple tasks where salient points are expected to fall inside the object [208] strictly.

Inpainting game for faces: Williford et al. [201] developed a curated database where predefined features (e.g., nose, left eye, left eyebrow) are replaced with features from another identity, creating customized datasets for each network. The inpainting game involves presenting a saliency algorithm with a triplet of the probe, mate, and inpainted non-mate, and tasking it with estimating a discriminative saliency map to identify pixels belonging to regions discriminative for the mate. The saliency threshold is used to replace pixels classified as salient with pixels from the "inpainted probe." These "blended probes" are then organized by the tested network as the original identity or the inpainted non-mate identity. This measure is suitable for face models, where high-performing deep learning models accurately assign more saliency to inpainted regions that change the identity of the blended probes without increasing the false alarm rate of the pixel salience classification.

Randomized sanity checks: This measure assesses the trustworthiness of saliency visualizations. Adebayo et al. proposed two sanity checks to ensure the accuracy of saliency algorithms: the *Model parameter randomization test* and the *Data randomization test* [209]. The *Model parameter randomization test* compares the saliency map on a trained model with that generated on a randomly initialized model. If the algorithm depends on the model's learned parameters, there should be a significant difference between the two maps. Randomizing weights from the top to the bottom layer progressively yields more random saliency maps.The *Data randomization test* compares the saliency map of a model trained on a labeled dataset with one trained on the same dataset but with randomly assigned labels. If the saliency method depends on data labeling, the saliency maps should show drastic differences.

In conclusion, the above discussion emphasizes the importance of employing face-specific saliency visualization methods to capture the intricacies of face tasks. Similarly, it is crucial to carefully select evaluation metrics that are relevant and appropriate for face images. By doing so, we can ensure accurate and meaningful assessments of the saliency maps' performance in the context of face-related applications.

8.7 Summary

In conclusion, this chapter provided a comprehensive overview of saliency algorithms, focusing on their application in the face domain. Saliency maps are crucial in understanding deep learning models and providing insights into their decision-making processes. We explored various types of saliency algorithms, including gradient- and perturbation-based methods. Each approach offers unique strengths and is suited for different interpretability needs.

Furthermore, we discussed the importance of face-specific saliency algorithms, as generic methods may not effectively capture the intricacies of face processing tasks. Face images have distinct characteristics and pose unique challenges for saliency visualization, necessitating specialized approaches to ensure accurate and meaningful explanations. Moreover, we examined the evaluation protocols for saliency maps, emphasizing the need for relevant metrics tailored to face images. It is essential to choose appropriate evaluation methods that align with the specific requirements of the face domain, ensuring the accuracy and reliability of saliency maps. Additionally, we explored how saliency maps can enhance human decision-making. By assisting humans in understanding model predictions and verifying their correctness, saliency maps offer valuable aids for face recognition and identity verification tasks. The optimal level of detail in saliency maps emerged as a critical factor in improving human interpretability and decision-making, and various saliency algorithms were compared based on their ability to provide informative and intuitive visualizations. In the subsequent chapters, we explore saliency visualization methods tailored to faces that overcome the shortcomings faced by generic saliency methods.

Chapter 9

Canonical Face Saliency Maps

9.1 Introduction

In previous chapters, we explored various visualization methods to enhance the interpretability of deep neural networks. However, most of these methods were primarily developed for object recognition tasks, with limited applications in the face domain [137, 196]. The unique properties of face images, such as high structure and fine-grained classification, pose challenges in applying traditional saliency methods designed for generic object recognition to face tasks. In Chapter 8, Figure 8.1. Face images are typically pre-processed to center around the face of interest, containing only one face per image, rendering the question of "where in the image" less relevant than "where on the face."

This chapter introduces a straightforward yet highly effective "standardization" process tailored explicitly for visualizing deep learning models in face processing to address these challenges. This process converts image coordinates to face coordinates, yielding more practical and meaningful saliency maps. By leveraging the inherent structure of faces, we project the saliency maps onto a standard frontal face, resulting in what we refer to as "Canonical Saliency Maps." These canonical saliency maps are independent of image coordinates and can be further processed to facilitate image comparison and observation of trends. This chapter will explore the methodology behind canonical image saliency maps and evaluate their effectiveness in various face processing scenarios.

9.1.1 Importance of Facial Features for Recognition

In this initial experiment, we aimed to assess the relative importance of different facial features for the task of face recognition. We selected 150 random images from the VGG-Face dataset and considered four key facial features: eyebrows, eyes, nose, and mouth. To occlude each feature, we utilized facial landmarks (refer to Figure 9.1). Subsequently, we measured the face recognition confidence of the VGG-Face model for each occluded image, comparing it to the confidence in the unoccluded image. The experiment results, illustrated in Figure 9.2, revealed that the nose is the most discriminative feature for face recognition, which goes against human intuition. The top part of the face, particularly the eyebrows, is known to be crucial for human face recognition [41]. Additionally, we observed that the mouth plays a less significant role in face recognition.

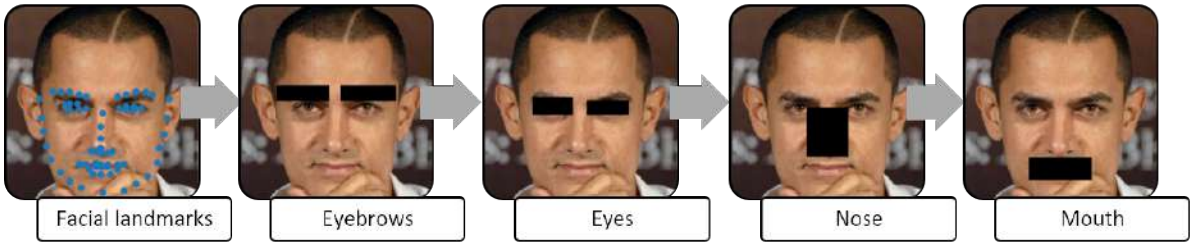


Figure 9.1: Initial experiment of Section 9.1.1, showing occlusion of facial features (eyebrows, eyes, nose, and mouth) using rough calculations of feature locations based on facial landmarks.

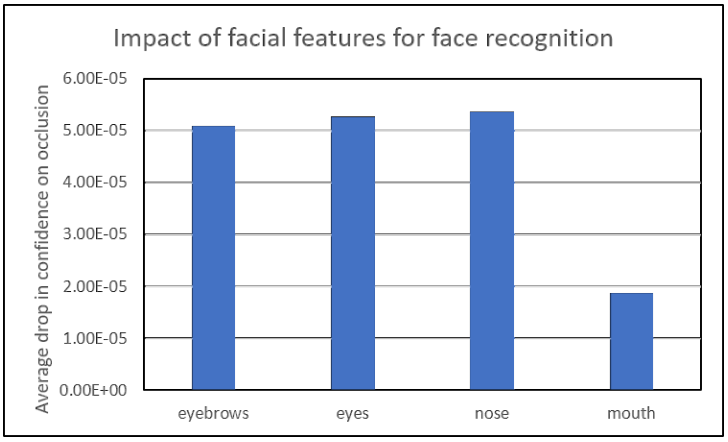


Figure 9.2: Results of the initial experiment, demonstrating the relative importance of various facial features based on the drop in confidence of the VGG-Face model. The eyes, eyebrows, and nose show high significance, while the mouth exhibits low significance.

This procedure solely assesses the importance of predefined facial features for face tasks. In the Methodology section, we illustrate how our canonical image saliency maps provide a fine-grained analysis of the significance of various facial regions.

9.2 Methodology

The methodology proposed in this chapter aims to create 'Canonical Image Saliency' (CIS) maps by projecting saliency maps onto a canonical neutral face. Here, 'neutral face' refers to average frontal face without any expression, lighting or exaggerated features to minimize distractions. This process enables us to understand how face models approach their tasks. Our approach is based on mapping the sensitivity of classification confidence through systematic occlusion of different image regions, as described in [134]. The resulting drop in confidence is visualized on a heatmap, generating occlusion maps highlighting influential parts of the image. We create canonical saliency maps by projecting these occlusion maps onto a neutral face.

Like other occlusion-based saliency map methods, given an image $I \in \mathbb{R}^{W_I \times H_I}$ and the coordinates (i, j), the importance of a patch ($|i - x| < \frac{sz}{2}\ \forall x < W_I$, $|j - y| < \frac{sz}{2}\ \forall y < H_I$) is given as follows:

$$S_{i,j} = \phi(I, c) - \phi(I \odot B_{i,j}, c) \tag{9.1}$$

where $\phi(I, c)$ is the confidence of class c for image I and $B_{i,j} \in \{0, 1\}^{W_I \times H^I}$ is a mask such that:

$$B_{i,j}[x][y] = 0 \text{ if } |i - x| < \frac{sz}{2} \text{ and } |j - y| < \frac{sz}{2} \tag{9.2}$$

$$= 1 \text{ otherwise} \tag{9.3}$$

and sz is the size of the patch, which is a hyperparameter.

9.2.1 Alignment to Canonical Face

In order to capture the finer details of the parts of an image a trained DNN model looks at, we compute our saliency map on a standard neutral frontal face image $F \in \mathbb{R}^{W_F \times H_F}$ called the *canonical face*, which helps compare saliency maps on a standardized platform.

We find an one-to-one mapping between the input face image and the canonical face image by fitting a 3D morphable model (3DMM) [210] using the procedure used by PR-Net [211]. In particular, we use a convolutional neural network to regress a UV positional map from the input image, which gives the depth for a set of fixed points on the UV map of the face. For details of this procedure, please see [211]. Let $M \in \mathbb{R}^{N \times 3}$ be a set of N 3D points representing the 3DMMM. We fit it on the input image I and the canonical image F to obtain the set of 2D points $M_I \in \mathbb{R}^{N \times 2}$ and $M_F \in \mathbb{R}^{N \times 2}$ as the projection of M on I and F respectively. Thus, we have a 1:1 dense mapping of points from I to F such that $I[M_I[n, 1]][M_I[n, 2]]$ refers to the same facial feature as $F[M_F[n, 1]][M_F[n, 2]]\ \forall n \in \{1.2, \cdots, N\}$.

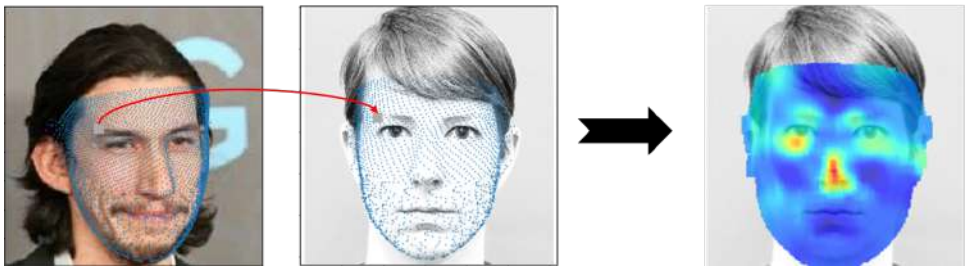


Figure 9.3: Procedure of computing Canonical Image Saliency (CIS) map. First, the input face is densely aligned. Each part of the input face is occluded with a small patch and the classification confidence is obtained. The drop in confidence is plotted on the same face location on a neutral face image to obtain the Canonical Image Saliency map

9.2.2 Mapping Discriminative Areas

The Canonical Image Saliency (CIS) map is generated by accumulating the drop in confidence at each point of the dense alignment matrix M_I and recording it on the corresponding location of F on an intermediate matrix $P^* \in \mathbb{R}^{W_F \times H_F}$ as follows:

$$\begin{aligned} P^*_{M_F[n,1],M_F[n,2]} &= P^*_{M_F[n,1],M_F[n,2]} \\ &\quad + S_{M_I[n,1],M_I[n,2]} \\ \forall n < N & \end{aligned} \tag{9.4}$$

where $P^*_{M_F[n,1],M_F[n,2]}$ is the patch around the point $(M_F[n,1], M_F[n,2])$ on the heatmap P, and $S_{M_I[n,1],M_I[n,2]}$ is the drop in confidence in the patch around point $(M_I[n,1], M_I[n,2])$ calculated according to Equation 9.1.

9.2.3 Density Normalization

Note that an equi-spaced grid on a 3-dimensional face may not correspond to equi-spaced grid on a 2D projection of the face. For example, on a frontal face image, the points on the sides of the face may be more spatially concentrated due to the curvature of the face. The heatmap values in these regions will hence be higher due to the concentration. We hence introduce a normalization step that keeps track of the number of times a pixel on an image is occluded, when performing the occlusion heatmap on the mesh. Let $N \in \mathbb{R}^{W_F \times H_F}$ be a matrix which stores the count of times each pixel of P^* was updated. The final CIS map is calculated as follows:

$$P = P^* \oslash N \tag{9.5}$$

where $\oslash$ represents element-wise division. Figure 9.4 shows the effect of density normalization on the CIS map.

Our algorithm is summarized as follows:

Algorithm 3: Canonical Image Saliency Map

Input: • input image I of size $W_I \times H_I$ • input mesh M_I of size $N \times 3$
• frontal image F of size $W_F \times H_F$
• frontal mesh M_F of size $N \times 3$
• model ϕ: deep model to find saliency where $\phi(I, c)$ gives the confidence of I for class c
• target class C of the input image I
• sz: size of occlusion square

Output: heatmap P of size $W_F \times H_F$

$P \leftarrow \{0\}^{W_F \times H_F}$
$N \leftarrow \{0\}^{W_F \times H_F}$
$fsz \leftarrow fsz \times \frac{H_F}{H_I}$
for $i \leftarrow 0$ ***to*** n **do**
  $I^* \leftarrow I$
  $I^*[M_I[i,0] - \frac{sz}{2} : M_I[i,0] + \frac{sz}{2}][M_I[i,1] - \frac{sz}{2} : M_I[i,1] + \frac{sz}{2}] \leftarrow 0$
  $x_F, y_F \leftarrow M_F[i,0], M_F[i,1]$
  $P[x_F - \frac{fsz}{2} : x_F + \frac{fsz}{2}][y_F - \frac{fsz}{2} : y_F + \frac{fsz}{2}] += \phi(I, C) - \phi(I^*, C)$
  $N[x_F - \frac{fsz}{2} : x_F + \frac{fsz}{2}][y_F - \frac{fsz}{2} : y_F + \frac{fsz}{2}] += 1$
end
$P[N = 0] \leftarrow 0$
$N[N = 0] \leftarrow 1$
$P \leftarrow P \oslash N$
return P

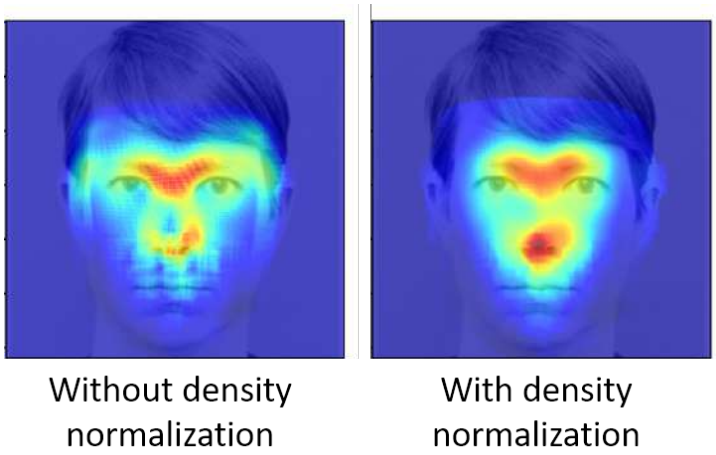


Figure 9.4: Effect of applying density normalization to the heatmap. Without density normalization, the nose is not highlighted despite it being a discriminative feature, mainly because the density of points on the nose is low

9.2.4 Application to Non-Classification Tasks

Canonical saliency maps can be generated for any face model which has a measure of confidence associated with each input image. Our method can be adapted to non-classification models by defining an appropriate confidence function. Here, we define the confidence function for two commonly-used face tasks: zero-shot recognition using nearest neighbour and face verification.

9.2.4.1 Zero-shot Face Recognition

Here, the query image q is assigned the label of the image from the training set whose features have the highest cosine similarity with the features of the query image [212]. We define the confidence of classification in this setting as follows:

$$S_{q,c} = \frac{A.Q}{\|A\|\|Q\|} \tag{9.6}$$

where c is the ground truth label of q, Q is the feature of q and A is the feature of the closest training set image with label c. This new confidence function can replace the class confidence ϕ in Equation 9.1.

9.2.4.2 Face Verification

Here, a pair of face images has the same identity if the cosine similarity between their features is more than a threshold calculated on the training set [212]. We define the confidence in this setting as follows:

$$S_{q_1,q_2,c} = c \times (\tau - \frac{Q_1.Q_2}{\|Q_1\|\|Q_2\|}) \tag{9.7}$$

where $c \in \{-1, 1\}$ is the verification ground truth label, τ is the verification threshold, and Q_1 and Q_2 are the features of the image pair q_1 and $q2$.

9.3 Experiments and Results

In this section, we present the qualitative results of our visualization method. We begin by showcasing visual examples of our saliency maps in Section 9.3.1. Next, we demonstrate the results of face verification saliency maps, as discussed in Section 9.2.4.2 in Section 9.3.2. Additionally, we perform a sanity check to ensure the correctness of our results in Section 9.3.3. Finally, we conduct an ablation study to determine the optimal hyperparameter for occlusion size in Section 9.3.4. All experiments were conducted on the VGG-Face dataset [19].

9.3.1 Qualitative Results

We compare the saliency maps produced by various popular saliency methods in Figure 9.5. Columns (b) and (c) in the figure show results of methods that use the magnitude of gradients to produce a heatmap. These heatmaps are scattered, making it difficult to see the details and interpret classification results. Guided backpropagation, shown in column (d), shows the finer details of the face, but is not class-sensitive, thus reducing their utility for interpretation. Columns (f), (g) and (h), corresponding to GradCAM [6], GradCAM++ [205] and ScoreCAM [192], are class-specific, but most commonly highlight the central area of a face making them uninformative across different face processing tasks. Column (e) represents the results of Guided GradCAM++, obtained by multiplying the output of guided backpropagation with the GradCAM++ heatmap, shows fine details while highlighting the class-discriminative area of the face. Occlusion maps in column (i) of Figure 9.5 seem to give the most informative results for our use case. This method maps each region of the image's impact on the classification, in effect mapping out how representative of the class each region is. It produces a more non-trivial heatmap showing finer details than the other heatmaps. The heatmap resolution can also be adjusted by changing the occlusion and stride size, and the method can be used with any architecture and loss function. Our visualization method is hence built on occlusion maps given this inference from our studies on face images. Results of the canonical saliency maps shown in Figure 9.6 demonstrate that our canonicalization process effectively transfers the occlusion heatmap onto a neutral frontal face without altering any details.

9.3.2 Results on Face Verfication

Canonical saliency maps were created for face verification following the procedure in Section 9. This canonicalization is crucial as it ensures that the same regions are occluded in both faces simultaneously, avoiding noise in the heatmap. The Euclidean distance between the images is measured at each occlusion step, and the resulting values are added to the heatmap. The heatmap is multiplied by -1 to

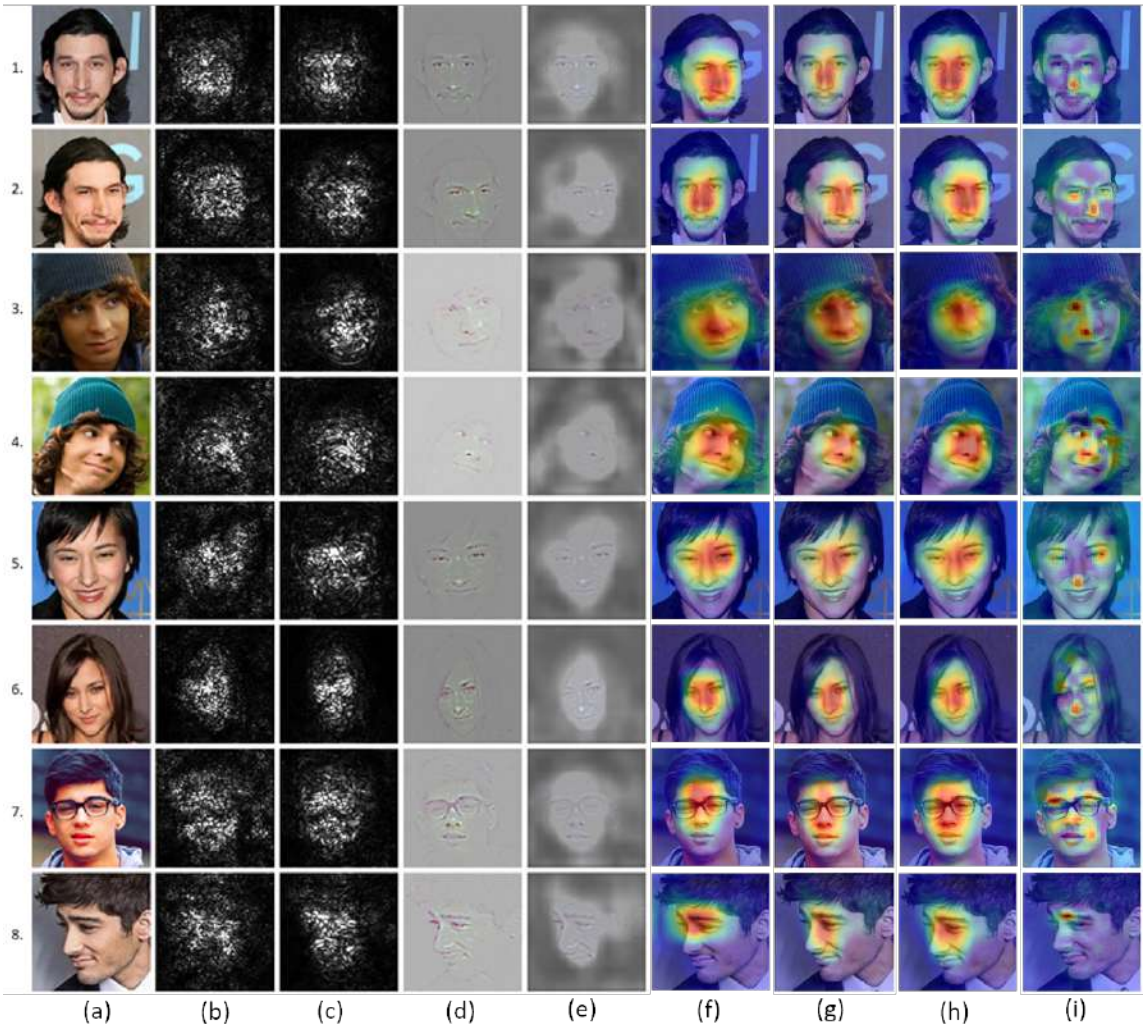


Figure 9.5: Comparison of different saliency visualization methods applied to the VGG-Face model [19] for face recognition. Each visualization is targeted toward the ground truth class of the corresponding image. (a) Original image; (b) Vanilla gradients [135]; (c) Smooth-grad [191]; (d) Guided Backpropagation [158]; (e) Guided GradCAM++ [205]; (f) GradCAM [6]; (g) GradCAM++ [205]; (h) Score-CAM [192]; (i) Occlusion map [134]. Images are taken from the VGG-Face dataset [19]. Rows (1, 2), (3, 4), (5, 6) and (7, 8) have the same identity. (Best viewed in color)

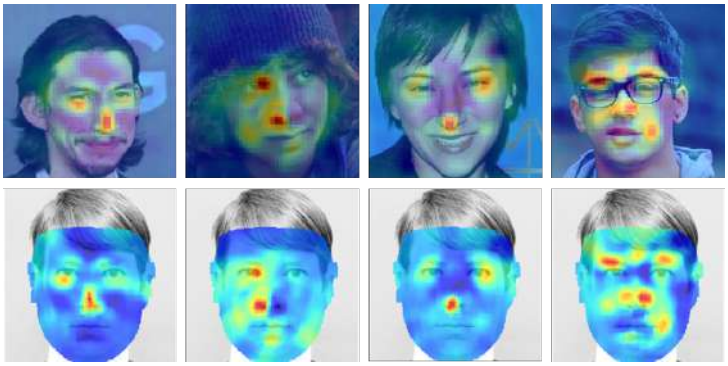


Figure 9.6: Comparison of occlusion maps (top row) and canonical saliency maps (bottom row) for various face images.

highlight areas with the most significant differences. Although inherent differences between images of the same identity can cause some noise, any change that affects identity should result in a larger distance in features than these inherent differences.

Highly controlled experiments were conducted using two images of the same person taken under different conditions, with variations in pose, expression, and hairstyle to demonstrate the effectiveness of this visualization method. In one image, a facial feature was deliberately altered to change the appearance while keeping the rest of the image constant. The similarity heatmap between the original image and the altered image was computed, and the heatmap was used to identify regions where the feature difference lessened when occluded. Results in Figure 9.7 show that the altered parts are generally highlighted. In some cases, additional regions may be highlighted with extreme changes in pose or lighting. For example, in the third row, only one eye is highlighted, while the eye in the shadow is not, possibly because the network did not pay attention to the far side of the face due to being in shadow.

9.3.3 Sanity Check Using Randomization

We conducted a sanity check on our saliency maps following the method proposed by [209]. We progressively randomized the layers of our trained model, starting from the output layer, and observed the changes in the generated saliency maps. A method passes the sanity check if the progressive randomization increases the randomness of the corresponding visualizations. In Figure 9.8, we present the results of our experiment, showing that as more layers are randomized, the visualization becomes more randomized, indicating that our method successfully passes the sanity check.

9.3.4 Ablation

We conduct a qualitative ablation study to investigate the impact of the occluding patch size on the generated Canonical Image Saliency (CIS) maps. The dense face alignment algorithm provides many vertices, leading to significant computation time for heatmap generation at each vertex. To address this, we introduce a tunable 'stride' parameter, which omits vertices at regular intervals to speed up the pro-

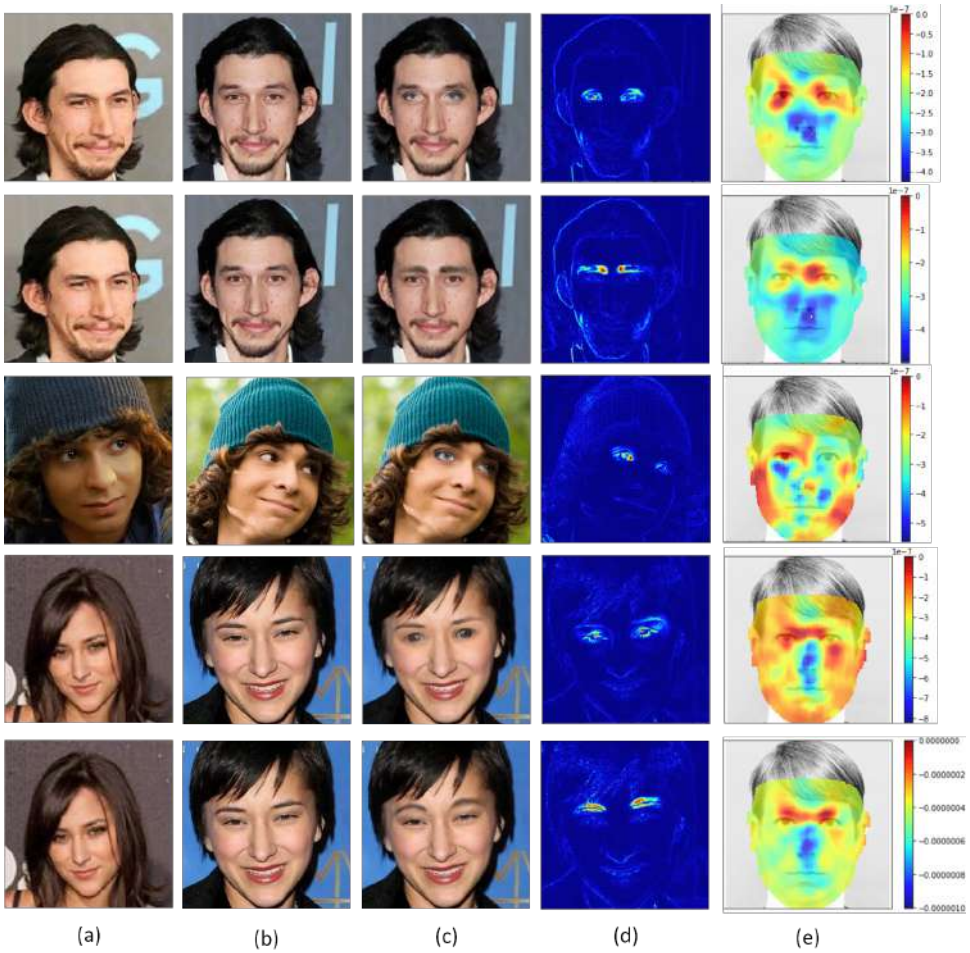


Figure 9.7: Images from the VGG-Face dataset were modified with photo editing to examine differences in pose and expression. Canonical saliency maps highlight regions with the highest Euclidean distance between features, indicating the altered facial parts. The third row highlights only one eye, suggesting reduced attention due to the other eye's orientation away from the camera. (a) Unaltered image 1, (b) Unaltered image 2, (c) Image 2 with altered features, (d) Difference image showing alterations, (e) Verification heatmap between images (a) and (c).

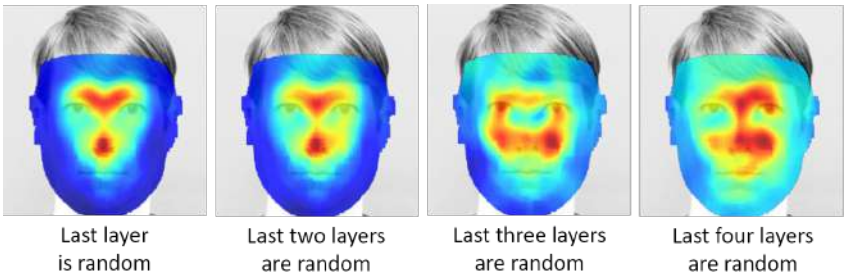


Figure 9.8: Sanity check on our Canonical Saliency Map visualization method. We progressively randomized the layers of the VGG-16 face model, starting with the output layer, following the procedure described in [209]. The map becomes progressively more randomized as layers get randomized, indicating that our method successfully passes the sanity check.(a) Last layer randomized; (b) Last two layers randomized; (c) Last three layers randomized; (d) Last four layers randomized

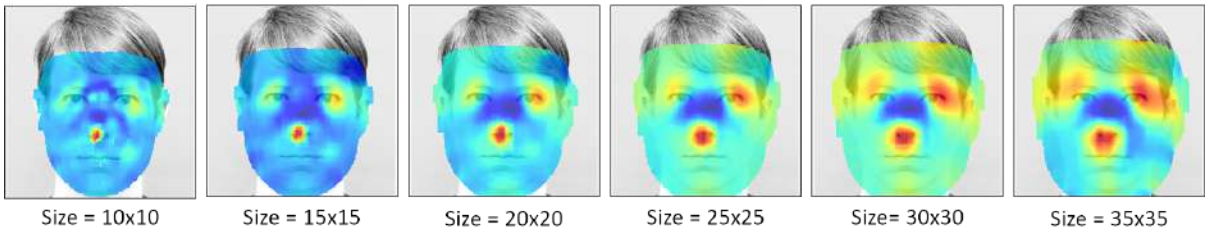


Figure 9.9: Variation of Canonical Image Saliency maps with different occluding patch sizes. A patch size of 15×15 was used in all other experiments.

cess. Smaller strides are chosen for smaller occluding patches to avoid gaps in the visualization, while larger strides can be used for bigger occluding patches without compromising visualization quality. Figure 9.9 demonstrates how changing the patch size affects the CIS maps generated from the same input image. Increasing the patch size results in fuzzier maps, but the general patterns remain consistent. Our method provides valuable insights regardless of the occluding patch size, although smaller patches offer higher resolution. For this work, we use a patch size of 15×15 as it strikes a good balance between heatmap resolution and computation time.

9.4 Summary

In conclusion, this chapter introduced a novel approach for visualizing and understanding the decision-making process of deep learning models in face recognition tasks. The creation of 'Canonical Image Saliency' (CIS) maps through the projection of saliency maps onto a standard neutral face enables better interpretability and comparability across different images. The chapter outlined the procedure for aligning occlusion maps to 'canonical' faces while preserving essential details. Additionally, it demonstrated the generation of canonical saliency maps for non-classification tasks, such as verification and zero-shot learning. Extensive qualitative experiments showcased the efficacy of the proposed approach compared to traditional saliency maps. Our method successfully passed a sanity check, ensuring the correctness and reliability of the obtained results.

Canonical saliency maps hold great potential for various applications, particularly in generating model-level saliency maps. The subsequent chapter explores how the process of making face saliency maps 'canonical' enables us to combine them, providing deeper insights into model behavior and decision-making processes.

Chapter 10

Canonical Model Saliency Maps for Faces

10.1 Introduction

In this chapter, we extend the concept of Canonical Saliency Maps introduced in the previous chapter to create Canonical Model Saliency Maps (CMS). CMS is a novel approach that allows us to aggregate and observe significant facial regions across an entire deep face model, providing a holistic understanding of model attention. We can capture trends and patterns in model decision-making processes by aligning occlusion-based saliency maps to a canonical face model.

A single occlusion map may contain variations caused by differences in image settings, making it challenging to comprehend the overall behavior of the model. The aggregation process in CMS averages out the effects of individual image variations, revealing the critical facial areas for the model (See Figure 10.1). We can observe attention trends in face models through CMS maps, which are not readily apparent from single image saliency maps. This alignment process enables meaningful comparison and aggregation of saliency maps, making CMS the first model-level saliency map.

This chapter presents the methodology to obtain CMS maps and showcase their application on various face models, including popular off-the-shelf models. By analyzing the CMS maps, we gain insight into face models and uncover inherent biases in their decision-making processes.

10.2 Methodology

Building upon the Canonical Image Saliency Maps (CIS) introduced in Chapter 9, we now present the methodology to create Canonical Model Saliency (CMS) Maps, which provide model-level saliency visualizations highlighting the facial regions that influence the model across all test images for a specific task (e.g., gender recognition, age recognition).

Given a test set D consisting of images $\{I_1, I_2, I_3, ...\}$ with variations in factors like pose, lighting, or expressions, the CMS map is obtained by averaging the CIS maps of each image in the test set:

$$V = \frac{1}{N} \sum_i P_i \qquad \forall I \in D \tag{10.1}$$

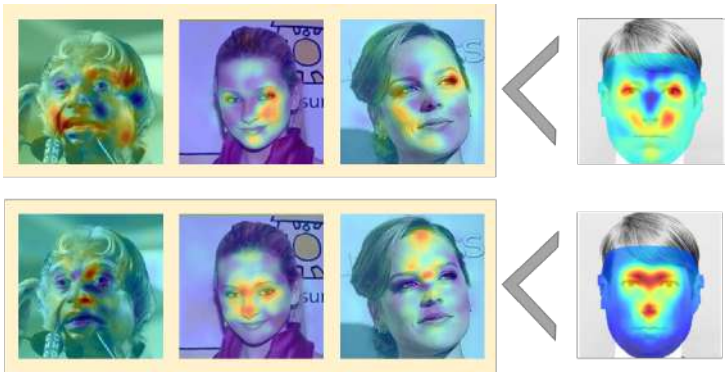


Figure 10.1: Comparison of individual occlusion maps of gender (first row) and recognition (second row) with their respective cumulative model saliency maps on the right. Individual occlusion maps exhibit variations due to pose, occlusion, and lighting, making it challenging to obtain a comprehensive understanding. Aggregating heatmaps eliminates minor differences caused by image conditions, facilitating valuable insights.

In Equation 10.1, P_i represents the CIS map of image $I_i \in D$, and N is the total number of images in the test set. The CMS map aggregates the saliency information from individual images, enabling a comprehensive understanding of the significant facial features influencing the model across the entire test set. While simple averaging is used in this study for generating CMS maps, alternative aggregation techniques could be explored in future research to enhance model-level analysis further (see Section 10.3.5 for more details).

10.2.1 Datasets and Models

Our main experiments are conducted on five different models trained for five tasks. The details of these models are given in Table 10.1. Models for Expression, Pose, Gender and Age were obtained by finetuning the VGG-Face model on the Celeb-A [173] dataset. The ground truth for gender was provided in the dataset. For age, emotion and pose, we generated the ground truth using known methods. The ground truth for age was obtained using the method DEX: Deep EXpectation of apparent age from a single image [175]. This method uses a VGG16 architecture and was trained on the IMDB-WIKI data set which consists of 0.5 million images of celebrities crawled from IMDB and Wikipedia. The ages obtained using this method were binned into 10 bins, each with 10 ages. Head pose was obtained by registering the face to a 3D face model using linear pose fitting [213]. The model is a low-resolution shape-only version of the Surrey Morphable Face Model. The yaw and pitch values were binned into 9 bins ranging from top-left to bottom-right. Figure 10.17 shows the binned pose values. The ground truth was obtained for emotion using a VGG-16 model trained on FER 2013 data set [214] with 7 classes. The accuracy for the recognition models are reported on the LFW dataset. The accuracy for other models are reported on a test partition of the Celeb-A dataset. We used three additional models for our experiments on gender detailed in Sections 10.3.2. The details of these models are given in Table 10.2. In addition,

Task	Architecture	Training	Accuracy %
Recognition	VGG-16 [161]	VGG-Face [19]	98.95 on LFW [17]
Recognition	LightCNN-9	Casia-WebFace MS-Celeb-1M	98.8 on LFW
Expression	VGG-16	Celeb-A using FER13 [214]	69.01
Pose	VGG-16	Celeb-A using 3DMM [213]	96.62
Gender	VGG-16	Celeb-A [173]	98.37
Age	VGG-16	Celeb-A using IMDB-Wiki [175]	61.72

Table 10.1: Details of deep face models used in this work

Model name	Implementation	Base Architecture
VGG Gender	Trained by authors on CelebA	VGG-16
Fairface [215]	https://github.com/dchen236/FairFace	resnet34
DEX [175]	https://github.com/siriusdemon/pytorch-DEX	VGG-16
CPG [216]	https://github.com/ivclab/CPG	spherenet

Table 10.2: Details of the deep gender models used for Figures 10.8 and 10.3

we have conducted experiments on the Labelled Faces in the Wild dataset [17] and the LightCNN model for face recognition [51].

10.3 Experiments and Results

10.3.1 Qualitative Results

In this section, we analyze various CMS maps of the models presented in Section 10.2.1.

10.3.1.1 Evaluation of Canonical Model Saliency Maps on Various Face Tasks

In this experiment, we applied our algorithm to five models trained for different face tasks: classification, expression, head pose, age, and gender. Figure 10.2 displays the resulting Canonical Model Saliency (CMS) maps. Remarkably, models of the same architecture but trained for distinct tasks exhibit varying focuses on different facial regions. For instance, recognition models emphasize the eye-nose triangle, downplaying the mouth and chin. Surprisingly, gender models find the corners of the eyes most discriminative. The head pose model prioritizes the nose, while the expression model focuses on the area between the eyebrows. Additionally, the age model incorporates various facial features. These CMS maps provide valuable insights into face tasks and the characteristics of deep models that handle them, as discussed in Section 10.4.

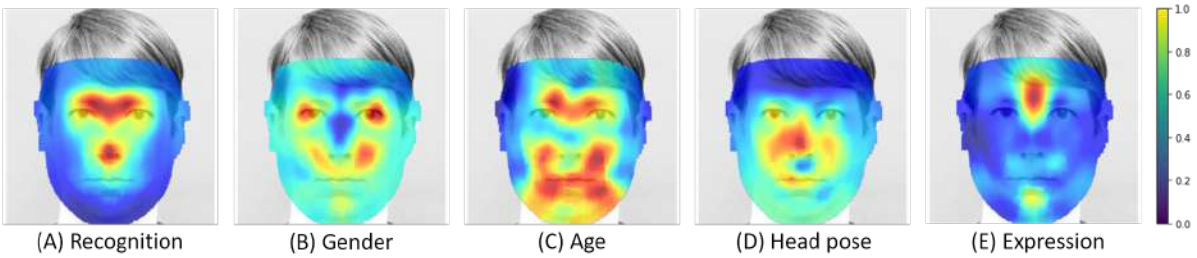


Figure 10.2: Canonical Model Saliency (CMS) maps demonstrate that different face classification tasks do not attribute equal importance to all parts of the face. These maps reveal significant facial areas influencing deep model decisions, aiding in the understanding of model behavior and identification of potential biases. In the heatmaps, red denotes high importance, while blue indicates low importance.

.

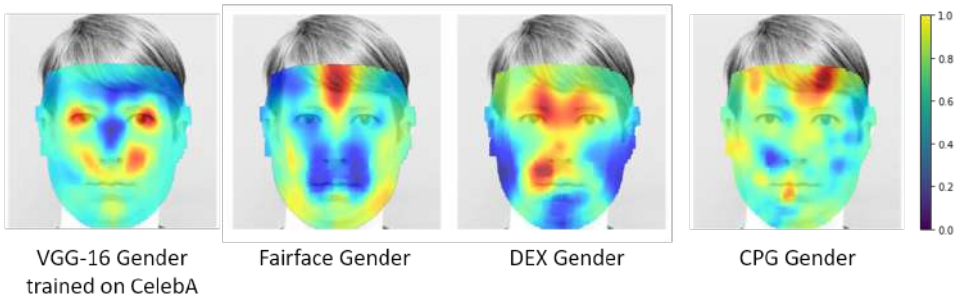


Figure 10.3: We compare CMS maps obtained from various off-the-shelf deep gender models

10.3.1.2 Canonical Model Saliency Maps on Various Gender Models

We computed Canonical Model Saliency (CMS) maps for four pre-trained face gender models with different architectures [175,215,216], as shown in Figure 10.3. Despite differences between the models, they primarily focus on the eye-nose triangle for gender recognition while de-emphasizing the mouth. This aligns with expectations, as the mouth is highly deformable and may not provide reliable cues for recognition. The variations in CMS maps indicate that they are influenced not only by the face task but also by factors like architecture, dataset, and training methodology. These maps are model-specific and can be used to diagnose biases and other model-related issues.

10.3.2 Quantitative Results

We conduct an objective evaluation of the faithfulness of our method on two datasets: CelebA and LFW [17] and compare it with three popular saliency visualizations: GradCAM [6], GradCAM++ [205] and ScoreCAM [192]. Similar to [192, 205], we measure the confidence drop of explanation images produced by pixel-wise multiplication of the saliency heatmap with the base image. In particular, we utilize a 'negative explanation image' by darkening the relevant areas of the base image. Unlike the task of object recognition, face images have a single object at the center of the image, and models trained on

Figure 10.4: Comparing the impact of using positive and negative saliency maps. Face models have higher confidence in Figure (a) than in Figure (b), although Figure (b) highlights more relevant features. As shown in Figure (c), negative saliency maps emphasize relevant features to cause a larger drop in confidence. Thus, we use negative saliency maps where darkening relevant features should cause a larger drop in confidence. This also ensures enough context for the model to interpret the face holistically. Additionally, negative maps prevent incorrect interpretations, as seen in Figure (d), where the heatmap misses the face but using normal explanation maps might result in high metric scores.

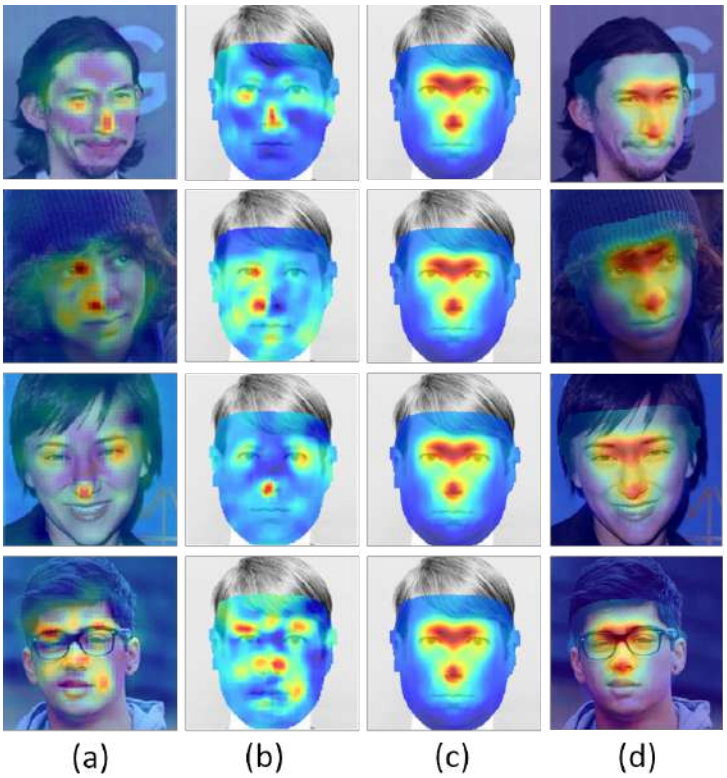


Figure 10.5: Column (a) shows Occlusion Maps used for saliency visualization; Column (b) shows Canonical Image Saliency (CIS) maps. CIS maps are a projection of occlusion maps onto a canonical frontal face; Column (c) shows Canonical Model Saliency (CMS) maps. These maps are generated for a model as a whole and hence do not vary with input; Column (d) shows the CMS maps reprojected back onto the input face.

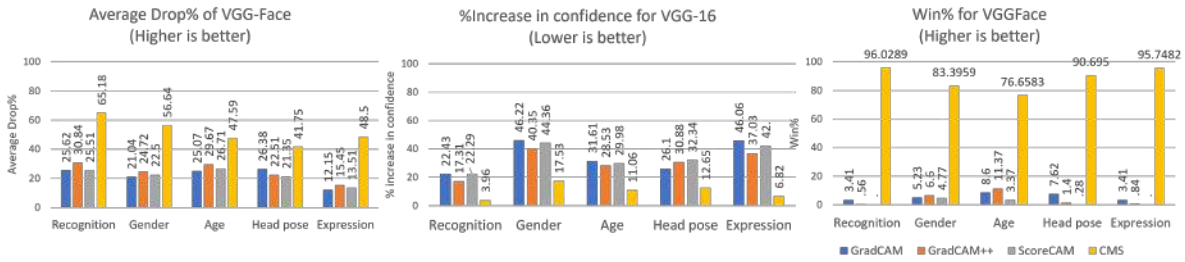


Figure 10.6: Results for Average Drop %, % Increase in Confidence and Win % of VGG-16 on Celeb-A for the tasks of recognition, gender, age, head pose and expression.

Figure 10.7: Results for Average Drop %, % Increase in Confidence and Win % of the explanations generated by Grad-CAM, Grad-CAM++, ScoreCAM and CMS on LFW for the VGG-16 model.

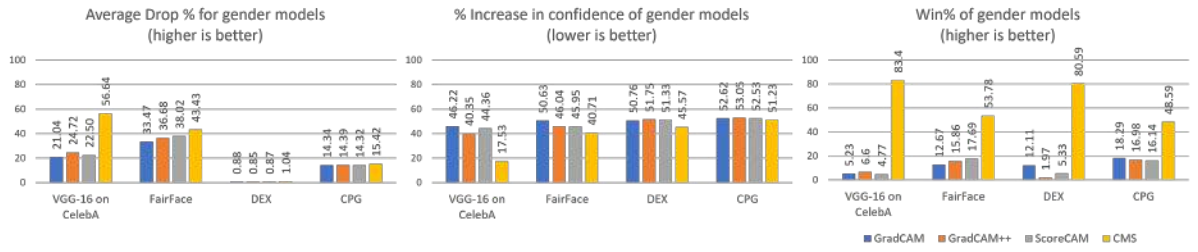


Figure 10.8: Results for Average Drop %, % Increase in Confidence and Win % of the explanations generated by Grad-CAM, Grad-CAM++, ScoreCAM and CMS on CelebA for various deep face gender models

face images focus on different parts of the face image. In this process, saliency maps sometimes fail to detect the face completely (see Figure 10.4). Using negative explanation maps addresses such concerns. The negative explanation image E is given by:

$$E = (1 - H) \otimes I \tag{10.2}$$

where H is the heatmap, I is the base image and $\otimes$ represents pixel-wise multiplication. The heatmaps are first normalized to a range of [0,1] and the heatmaps for all the methods are standardized to have the same sum of pixels for each image:

$$H' = \frac{h - \min(h)}{\max(h) - \min(h)}; H = \frac{s}{\Sigma H'} H' \tag{10.3}$$

where h is the original heatmap, s is a scalar which is the same for all heatmaps of the same image, and H is the final heatmap which is used to create negative explanation maps. Normalizing the heatmaps in this way ensures that no visualization method gets an advantage of highlighting a large area of the input image, as only the discriminative parts should be highlighted.

We adopt the three metrics used in [205] with negative explanation images:

Average Drop %: The confidence of an image when passed through a model is expected to decrease when the most discriminative parts are covered. We measure the drop in confidence when compared to the unmodified image as:

$$\frac{1}{N} \sum_{n=1}^{N} \max(0, \frac{M(I_n) - M(E_n)}{M(I_N)} \times 100 \tag{10.4}$$

where $M(E_n)$ and $M(I_n)$ are the confidence values of the n^{th} explanation image and original image respectively. A high Average Drop % value indicates that the heatmap accurately highlights the most discriminative parts of the image.

% Increase in confidence: In some images, covering the highlighted parts may result in an undesired increase in confidence for the original image. We measure the number of such images using this measure as follows:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{M(E_n) > M(I_n)} \times 100 \tag{10.5}$$

where $\mathbb{I}$ is the indicator function which returns 1 if $M(E_n) > M(I_n)$ and 0 otherwise. A low score in this metric is better.

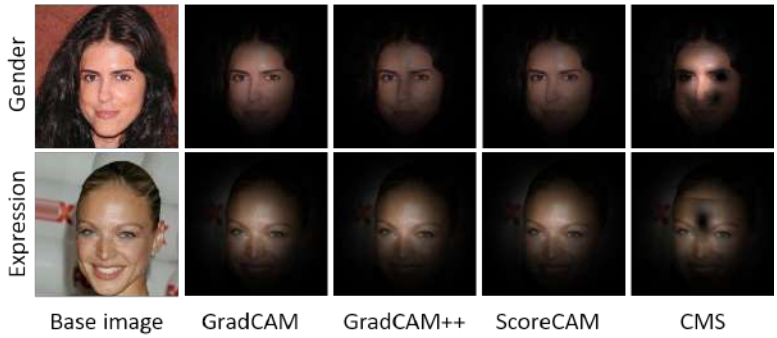


Figure 10.9: Samples of figures used in our survey (see Section 10.3.3)

Win %: Here, we compare all four methods and measure which produces the greatest drop in confidence for a given test image. For example, Win % of CMS is calculated as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{M(E_n^{CMS})<(M(E_n^{GradCAM}),\ M(E_n^{GradCAM++}),M(E_n^{ScoreCAM}))} \times 100 \tag{10.6}$$

where the indicator returns one if the explanation map produced by CMS has the lowest confidence, the sum of Win % across all the visualization methods for a single task should add up to 100.

We conduct three experiments for quantitative evaluation. First, we calculate the above metrics on VGG-16 for the tasks of recognition, gender, age, head pose, and expression on the CelebA dataset. For a fair comparison, we use our maps projected back onto the input image (Col (d) of Figure 10.5). Figure 10.6 shows our results and a comparison with other visualization methods. Our method outperforms all other methods in all metrics. The Win % shows that for most images, removing the explanation map given by our method causes the highest drop in confidence (the higher the better). Secondly, we repeat the experiment on the LFW [17] dataset using the VGG-Face network, using the same experimental settings as above. We show the results in Figure 10.7. Our method also outperforms all other methods by a large margin in all quantitative metrics, showing that our method generalizes across datasets. We also compare our saliency methods on various off-the-shelf gender models. We use pretrained models from [175, 215, 216] and evaluate our metrics on CelebA-subset. Our results are shown in Figures 10.8. Once again, our method outperforms all other methods on all metrics. We show the CMS maps obtained using the various networks in Figure 10.3.

10.3.3 Human Perception

We conducted a user survey to assess the human interpretability of our saliency maps compared to other visualization methods, explicitly focusing on the tasks of gender and expression. The survey involved 96 images, each evaluated by 154 participants not involved in our work. Participants were

Figure 10.10: All base images used for our user survey (see Section 10.3.3)

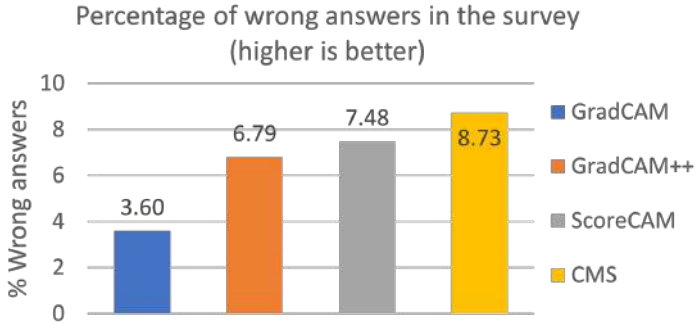


Figure 10.11: Results for user survey on the perception of gender and emotion on explanation maps. We used 12 base images modified using GradCAM, GradCAM++, ScoreCAM and CMS. The users had to pick binary labels for each image (male-female, happy-sad). Each question was answered by 143 people who were not involved in this project

Figure 10.12: Ablation study on the effect of different types of alignment. Shown are the Average Drop%, % Increase in confidence and Win % for three different types of alignment on the LFW dataset: 1. Canonical face 2. Keypoint-based alignment 3. No alignment.

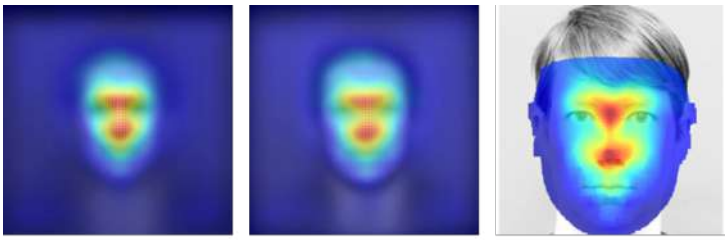


Figure 10.13: Ablation study on the effect of different types of alignment. Shown are the model saliency maps for three different types of alignment on the LFW dataset: (a) No alignment, superimposed on the average image of LFW; (b) Keypoint-based alignment, superimposed on the average image of LFW-funneled; and (c) CMS superimposed on the canonical face.

presented with four negative explanation maps for each image generated by different saliency visualization methods, including GradCAM, GradCAM++, ScoreCAM, and reprojected CMS maps. Vignettes were applied to hide contextual information. Participants were asked to make binary choices (e.g., male-female or happy-sad) for each image. The percentage of wrong answers was used to measure the effectiveness of the visualization method, as better methods should hide crucial information and make interpretation more difficult. The results in Figure 10.11 indicate that our method outperformed other methods, as it achieved a higher percentage of wrong answers, indicating better concealment of the most crucial and discriminative facial areas. The samples used for the survey are given in Figure 10.9. Some examples of survey images are given in Figure 10.9.

10.3.4 Why Align to Canonical Face

This section investigates the importance of using a canonical face alignment instead of a keypoint-based alignment or raw image pixel positions. Canonical face alignment ensures the accuracy of model saliency maps when aggregating individual image saliency maps. Without precise alignment, the accumulated changes in position can lead to inaccuracies in the final model saliency map.

To demonstrate this effect, we conducted an ablation study on the Labeled Faces in the Wild (LFW) dataset. We compare three types of alignment: 1) no alignment; 2) keypoint-based alignment using LFW images aligned with keypoints; and 3) canonical face alignment, as used in previous Canonical Model Saliency (CMS) experiments.

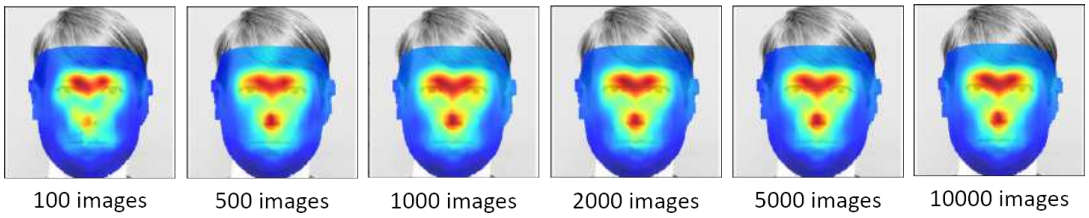


Figure 10.14: Ablation study to study the effect of the number of images used to create a CMS map. CMS maps for recognition using 100, 500, 1000, 2000, 5000 and 10000 random CIS maps

We generate image saliency maps for the first case by sliding an occlusion window over the entire input image. In the second case, we use LFW images aligned with keypoints as input for generating image saliency maps. We use the same canonical face alignment in the third case as in the previous CMS experiments. We create model saliency maps for each case by averaging individual image saliency maps and calculating quantitative metrics. Figure 10.12 shows that canonical alignment outperforms keypoint-based alignment and no alignment in all cases. Figure 10.13 shows the model saliency maps for all three cases.

Using canonical faces improves the accuracy of model saliency maps and reduces computation cost. Since we precisely know which parts of the image need to be occluded, we avoid sliding the occlusion patch over the entire image, resulting in lower computational overhead.

10.3.5 Ablation: Number of Images

In this experiment, we investigate the convergence of Canonical Model Saliency (CMS) maps by aggregating Canonical Image Saliency (CIS) maps over varying numbers of images. The goal is to determine how many images are sufficient to obtain a stable CMS map for a fully trained model.

In practice, only a few images are needed to generate a stable CMS map. This indicates that face networks consistently rely on a small set of facial features, and the canonical visualizations remain consistent across images. Figure 10.14 illustrates this trend, where the CMS map becomes stable and converges to a final state after analyzing as few as 100 random images. With the addition of more images, such as 1000, the CMS map remains practically unchanged, demonstrating that convergence is achieved relatively quickly.

10.4 Analysis

In this section, we present an analysis of the proposed method including ablation studies and discussions

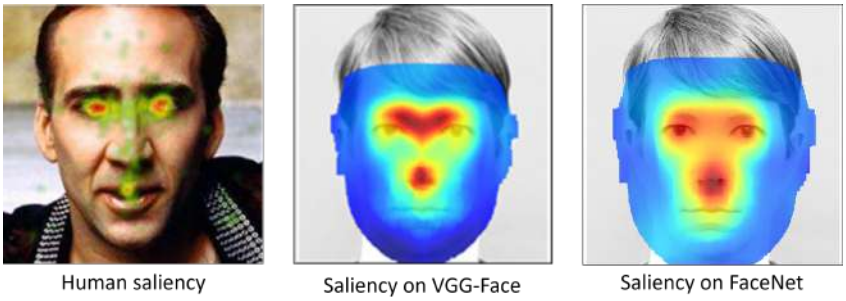


Figure 10.15: Comparison of face recognition CMS of VGG-Face and LightCNN with human gaze saliency. *LEFT*: Eye gaze fixation patterns when humans are asked to view faces freely. Image taken from [217] *MIDDLE*: CMS of VGG-Face [19] *RIGHT* CMS of LightCNN [51]

Figure 10.16: Make-up matters! The figure shows the classification confidence of a gender model on the same person with and without eye makeup. The top row shows the confidence for the 'female' classification and the bottom row shows the confidence for the 'male' classification. The ground truth label is given below each pair of images.

10.4.1 Important Facial Regions for Recognition and Emotion

In Figure 10.15, we compare the recognition Canonical Model Saliency (CMS) with eye gaze saliency. The CMS map indicates that the nose is the most crucial feature for face recognition, which aligns with our initial experiment in Chapter 9. Interestingly, this finding differs from human face recognition, where the eyes and eyebrows play a more significant role. Humans rarely rely on the nose for identifying individuals and instead, focus on the facial features in the upper half of the face. However, the network also places relevance on the region around the eyes, indicating its importance in identifying people.

Regarding expression classification, the relevant facial regions are not surprising. The mouth is the most essential feature for recognizing expressions, along with the region between the eyebrows, which captures expressions such as the furrowing of eyebrows and raised eyebrows.

10.4.2 Effect of Make-up on Gender Classification

The Canonical Model Saliency (CMS) maps for the gender model revealed interesting insights (Figure 10.2E). We expected the heatmap to highlight areas around the mouth, jaw, and cheeks, which typically contain facial hair cues and different bone structures for different genders. Surprisingly, the

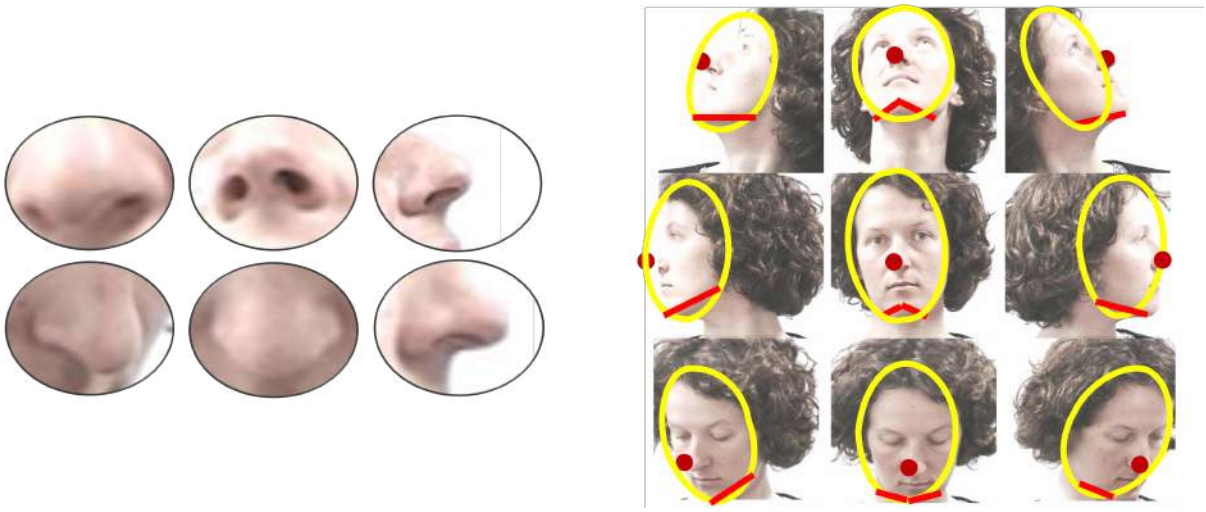


Figure 10.17: Close-ups of the nose tip in this figure reveal valuable cues for face pose estimation. The nose, along with the jawline, provides insights into the 3D orientation of the face. Additionally, the consistent location of the nose tip within a specific quadrant of the face area indicates its relevance to the same 3D orientation.

map primarily focused on the eye corners. We hypothesize that this behavior is due to the model being finetuned on the CelebA dataset [173], consisting of images of celebrities who extensively use makeup. Thus, the model learned to rely on eye makeup as a cue for gender classification. However, this reliance on makeup could lead to biased performance for different demographic distributions, as observed in commercial face models' failure to detect gender for females and different races accurately [5]. This highlights the importance of detecting and addressing dataset biases, as they can significantly impact the performance of deep models.

To test our hypothesis, we conducted a qualitative experiment using images of people with and without eye makeup from the internet. These images were passed through the gender model, and we observed the confidence scores for 'male' and 'female' classifications. As presented in Figure 10.16, in all cases, there was a drop in confidence for the 'male' classification when men wore makeup and a smaller drop in confidence for the 'female' classification when women did not wear makeup. In some cases, the drop in confidence was substantial enough to change the original classification result, particularly for males of Asian origin, especially those from the Far East. This demonstrates that eye makeup significantly affects the performance of the gender model, leading to skewed results for certain ethnicities.

10.4.3 Nose as a Strong Cue for Head Pose Detection

The shape of the nose changes with the pose of the face and its placement on the face is affected by the 3D orientation of the face (Figure 10.17A). The nose is located at the center of the face, and its position consistently changes with the head's orientation. The head pose can be accurately detected

from the shape of the nose and the quadrant of the face in which the nose tip is located (along with the jawline), especially when there are only nine classes (as shown in Figure 10.17). The nose thus serves as the most significant cue for head pose detection, a finding supported by the CMS map shown in Figure 10.2D.

10.4.4 Age Prediction: Facial Cues Distributed Across Multiple Areas

The CMS map for age (Figure 10.2F) shows that the cues for age are present in multiple areas of the face. Some of the distinctive features for age may be skin tightness around the eyes and jaws, wrinkles and receding hairline. Pre-deep learning methods used the geometry or texture of the face for age prediction [218], thus corroborating our finding on why age-related cues are found all over the face.

10.4.5 Robustness in Deep Models

Robustness is a crucial property of deep learning models, ensuring that slight variations in input images, whether caused by noise or natural variations, do not significantly impact the model's accuracy. Models that rely on a limited set of cues may be more prone to errors when faced with diverse input images. In contrast, models that consider multiple cues are generally more robust and can better handle variations in the input data. Canonical Model Saliency (CMS) maps reveal the specific facial areas that deep models focus on, allowing us to estimate their robustness. A model concentrating on only a few facial areas is likely less robust than one considering many facial features. Less robust models may make mistakes when facing extreme occlusion, lighting, and other deviations. For instance, we observed this in our gender model (Section 10.4.2), which was sensitive to changes in facial appearance due to makeup.

10.5 Summary

In this chapter, we introduced Canonical Model Saliency (CMS) maps, which provide valuable insights into the decision-making process of deep face models. By aggregating Canonical Image Saliency (CIS) maps over multiple images, CMS maps highlight the facial areas that significantly influence the model's predictions across the entire test set. We demonstrated the effectiveness of CMS maps on various face models, revealing distinctive patterns for different tasks such as gender, expression, head pose, age, and recognition.

The CMS maps allowed us to observe how different face models focus on specific facial regions for different tasks. For instance, recognition CMS maps emphasized the importance of the eye-nose triangle, while gender models fixated mainly on the corners of the eyes, possibly due to dataset biases. Head pose models relied significantly on the shape of the nose to determine the 3D orientation of the face. Moreover, CMS maps showed that age-related cues are distributed across multiple areas of the face, consistent with previous non-deep learning methods.

We conducted a user survey to evaluate the human interpretability of our saliency maps compared to other visualization methods. The results showed that our method performed better at hiding the most crucial and discriminative facial areas, indicating the potential of CMS maps in understanding facial attributes through deep models.

We explored the significance of using a canonical face alignment when generating CMS maps. The alignment process ensured accurate and stable CMS maps, with fewer computation costs compared to other alignment techniques. The robustness of a model was also analyzed through CMS maps. Models that focus on a few facial areas were found to be less robust, making them susceptible to mistakes when faced with diverse input variations. This observation was exemplified in our gender model's sensitivity to changes in facial appearance due to makeup, potentially arising from dataset biases.

Overall, Canonical Model Saliency maps provide crucial insights into the functioning of deep face models and help to uncover biases and challenges inherent in these models. This chapter has demonstrated the power of Canonical Model Saliency maps in uncovering important insights and potential biases in face recognition models, driving us towards more robust and transparent deep learning solutions for facial analysis tasks.

PART V

Human Visual Saliency Using Gaze

Chapter 11

DashGaze - A Naturalistic Driver Gaze Dataset for Appearance-Based Gaze Estimation

11.1 Introduction

The previous chapters attempted to understand deep face representations using various explainability methods. Saliency maps allow us to conduct useful analysis by comparing the facial areas important to the network to the areas that are expected to be important to classify the task. However, the challenge herein is - how do we obtain the 'correct' expectations to compare the network's saliency map to? One may consider human cognition a benchmark for what a deep network should see. Human saliency or what they find most important is often measured as the areas of an image on which a human pays attention to. This is in turn measured using eye movements - a human will fixate more on areas of images they attend to. In the following chapters, we examine the particular case of driver attention by capturing driver gaze.

A promising approach for gaze monitoring is appearance-based driver gaze mapping, where a computer vision model predicts the driver's gaze using images captured from a dashcam. However, the current datasets for appearance-based driver gaze estimation are lacking in several respects. Some

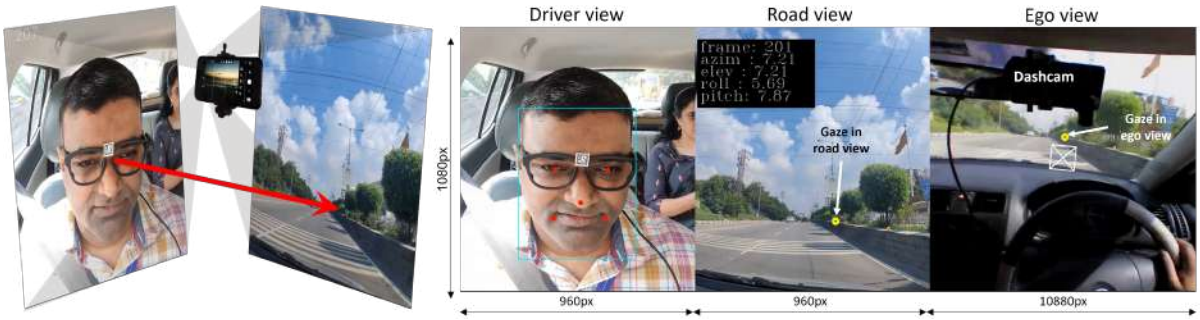


Figure 11.1: *LEFT*: The problem statement: given the road view and driver view, estimate the driver gaze point on the road using only a dashcam. *RIGHT*: An overview of our dataset DashGaze. It consists of three views: driver view, road view and egocentric view. The 2D gaze point, blink and fixation, and IMU data is provided for each frame.

datasets do not have variation in the position of the gaze cameras and hence are not general to any dashcam [109, 110]. Other datasets are collected in stationary vehicles [99, 104] or lab settings [108], which do not capture driving behavior. Additionally, many existing datasets provide ground truth as coarse 'gaze zones' inside the car instead of fine gaze points on the road [98, 99, 104–106]. Many datasets do not provide paired images of the driver and the road [98, 99, 104–106, 109]. This work aims to lower the barrier to adopting gaze monitoring technology by reducing the hardware required to a single uncalibrated dashcam.

We introduce DashGaze: a large-scale, naturalistic driver gaze dataset on which we can train appearance-based models that work without any special equipment except a dashcam. Our dataset consists of 33 driving sessions with 28 unique drivers. The dataset has over 900,000 frames, each a tuple of the road view, driver view, and driver's egocentric view (refer to Figure 11.1). Additionally, we provide the gaze ground truth in the egocentric view and road view, along with blink, fixation, and IMU (Inertial Measurement Unit) data. DashGaze is captured on Indian roads and is designed to be highly realistic. It contains samples collected in various types of weather, roads, and traffic conditions. The primary use of the DashGaze dataset is to train dashcam-based gaze models. The dataset may also be used to study long-term driver behavior, as each session lasts more than 10 minutes.

This chapter outlines the DashGaze dataset, compares it with other driver gaze datasets, and comprehensively describes our data collection process. Additionally, we conduct extensive driver gaze analysis to gain insights into driver behavior and saliency. The DashGaze dataset opens up new possibilities for driver gaze research and appearance-based gaze estimation from dashcam footage.

11.2 Dataset Acquisition

In this section, we describe the data acquisition process for our dataset, which adheres to the Institutional Review Board (IRB) protocols. All drivers included in the dataset are over 18 years of age and possess a valid driving license. Before data collection, each participant signed a consent form, providing their agreement to share their data and take part in the study. Throughout the data collection sessions, an instructor occupied the front seat to ensure safety. Drivers were instructed to drive naturally, and in some video captures, one or more passengers were present in the back seat to introduce additional variation to the dataset.

11.2.1 DGaze: Data Collection in the Lab

Our previous dataset, DGaze [108], was the first driver gaze dataset to include both road and driver views. We collected the DGaze dataset in a controlled lab environment to obtain data without using obstructive eye trackers. The subjects sat in front of a backdrop that mimicked the interior of a car, and a video of the road view was projected in front of them. The projected video was captured using dashboard-mounted cameras from real cars driving under various traffic conditions and times of the day. To simulate a dashboard-mounted phone, we mounted a mobile phone on a tripod in front of the subject

Figure 11.2: Lab setup for the collection of the DGaze dataset [108]. A driver sits in front of a static backdrop depicting the interior of a car. A road view is projected in front of the driver. A dashcam captures both the driver view and the projected view simultaneously.

and used its front and rear cameras simultaneously to capture both the driver and projected road views. To collect the dataset, we annotated various moving points on the projected video, and the subjects were asked to look at these points. While this setup provided a simple way to collect gaze data, it did not fully capture realistic driver behavioral gaze. As a result, we have developed the DashGaze dataset to address this limitation by collecting data in a completely naturalistic setting. The setup for DashGaze is depicted in Figure 11.2.

11.2.2 Hardware Configuration

In our data collection process, we employ the Pupil Invisible eye tracker [219] to track driver gaze. The eye tracker provides 2D gaze location in the driver's ego-centric view and information on blinks, fixations, IMU data, and gaze angles. The eye tracker is connected to a OnePlus 8 phone and is powered by the phone and records the gaze information. The OnePlus 8 phone is mounted on the car windscreen, capturing both the road view through the rear camera and the driver's view using the front camera (see Figure 11.3). For enhanced dataset variation and generality, we deliberately changed the position of the dashcam during different data collection sessions. The right side of Figure 11.3 illustrates three exemplar positions of the dashcam relative to the driver.

11.2.3 Temporal Synchronization

In our data collection process, we initiate the dash-cam first, followed by the eye tracker. We employ audio alignment to find their temporal disparity to synchronize the two data streams. The ego-centric video is sampled at 30Hz, while the gaze and IMU data are sampled at 200Hz. To match each gaze and

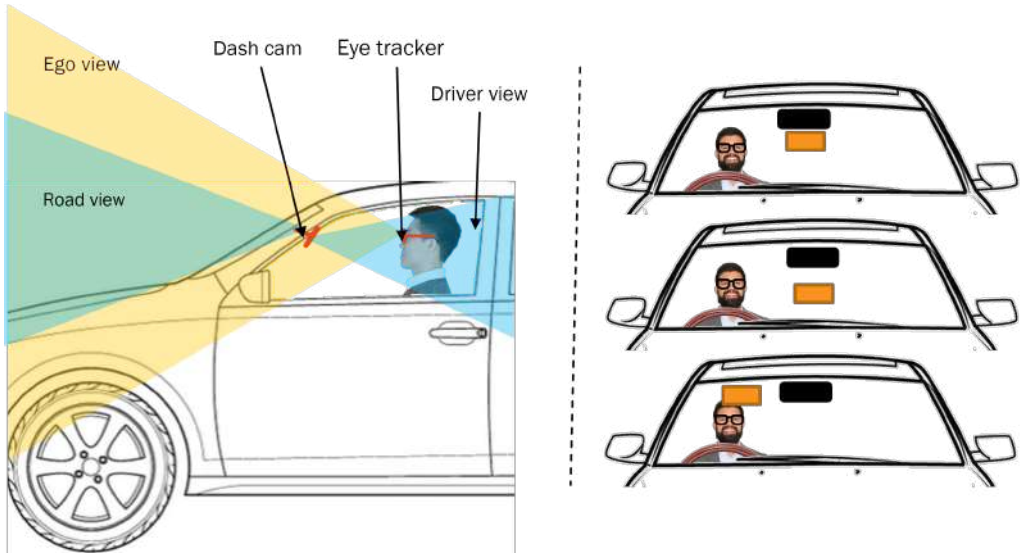


Figure 11.3: Data collection setup for the DashGaze dataset. *(LEFT)* Our data collection setup involves three cameras. A wide-angle camera on the eye-tracker captures the driver's ego view. We use a OnePlus 8 smartphone as our dashcam, fixed on the windscreen. The smartphone's front camera captures the driver's face, while the rear camera records the road view. *(RIGHT)* To enhance dataset diversity, we varied the position of the dashcam in different sessions. The figure displays three positions of the dashcam (in orange): under the rear-view mirror (top), in the center of the windscreen (middle), and the top right corner of the windscreen (bottom).

IMU sample to the nearest video frame, we associate them with the corresponding frame by averaging all the gaze and IMU samples corresponding to a single video frame. This procedure results in a single value per frame, ensuring data alignment and consistency.

11.2.4 Spatial Alignment

The eye tracker provides the 2D gaze location in the ego-centric view. The ego camera on the eye tracker uses a wide-angle lens. To transfer the gaze to the road view, we first undistort the ego-centric video and gaze points using the intrinsic camera parameters provided by the device. We transfer the gaze point to the road view by computing a homography between the road view and ego-centric view using ORB keypoints matched with the MLESAC algorithm (a variation of RANSAC). However, in some cases, we do not obtain enough matches between road and ego frames due to resolution differences, blurred frames, and view differences caused by the driver's head movements.

To address this challenge, we leverage the fact that matching two consecutive frames of the same stream (road or ego) is easier. We use this information to 'propagate' the good homographic transformation obtained between road-ego frame pairs to all frame pairs where good matches are impossible. The process of gaze transfer is illustrated in Figure 11.4, and the detailed algorithm for gaze transfer is provided in Algorithm 6.

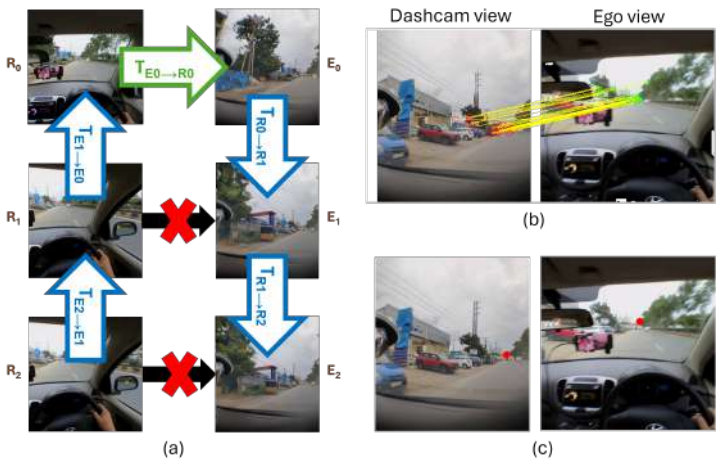


Figure 11.4: (a) Illustration of our spatial alignment algorithm. Here, we could find a good homography from ego frame to road frame at timestep 0, but not at timesteps 1 and 2. We transfer the gaze from E_2 to R_2 through E_0 and R_0. (b) An example of matches between road and ego frames (c) Transfer of gaze (in red) from ego to road frame using the homography found in (b)

Algorithm 4: Indirect gaze transfer between a road and ego frame

Input: Ego frames $\{E_0, E_1, \cdots, E_n\}$, corresponding road frames $\{R_0, R_1, \cdots, R_n\}$, augmented gaze points on the ego frames $\{G_{E_0}, G_{E_1}, \cdots, G_{E_n}\}$, where G_{E_i} is of the form $[x, y, 1]^T$. The homography transformation matrix between E_0 and R_0 is $T_{E_0 \to R_0} \in \mathbb{R}^{3\times3}$, which is computed using ORB and MLESAC.

Output: Gaze locations on road frames $\{G_{D_0}, G_{D_1}, \cdots, G_{D_n}\}$

$G_{D_0} \leftarrow G_{E_0} \times T_{E_0 \to D_0}$

for $i \in \{1, 2, ...n\}$ **do**

  Find $T_{E_{i+1} \to E_i}$ and $T_{D_i \to D_{i+1}}$ using ORB and MLESAC.

  $T_{E_i \to D_i} \leftarrow T_{E_{i+1} \to E_i} \times T_{E_i \to D_i} \times T_{D_i \to D_{i+1}}$

  $G_{D_i} \leftarrow G_{E_i} \times T_{E_i \to D_i}$

end

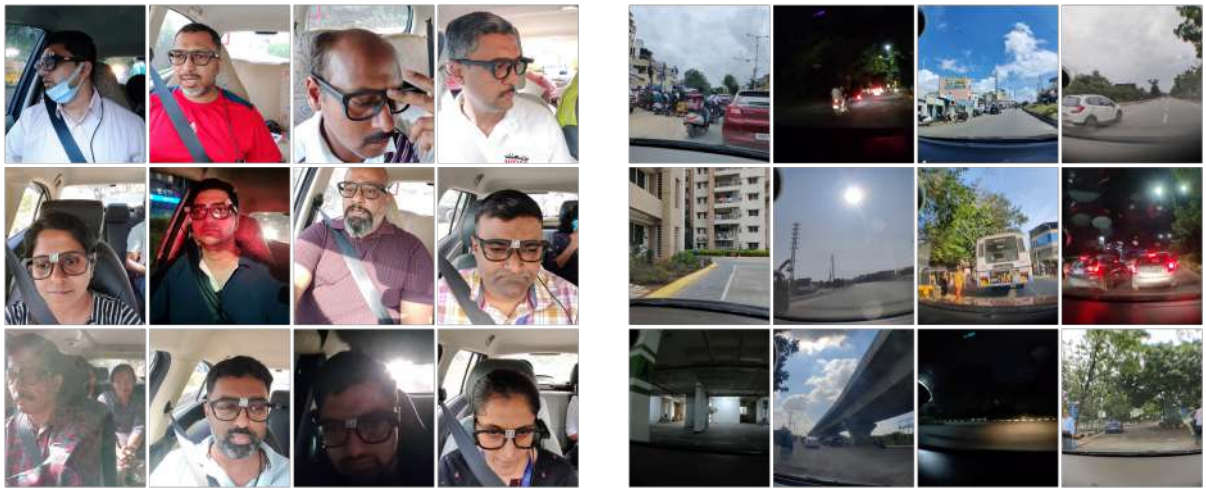
Dataset	Driving	Dashcam	Natural	Global South	Continuous	Road View	Driver View	Ego View
Gaze360 [90]	✘	✘	✔	✘	✘	✘	✘	✘
Rt-GENE [87]	✘	✘	✔	✘	✔	✘	✘	✘
MoGAZE [91]	✘	✘	✘	✘	✔	✘	✘	✘
HUMBI [220]	✘	✘	✘	✘	✔	✘	✘	✘
DG-Unicamp [105]	✔	✘	✔	✘	✘	✘	✔	✘
DMD [98]	✔	✘	✔	✘	✘	✘	✔	✘
DGW [104]	✔	✘	✔	✘	✘	✘	✔	✘
AutoPOSE [107]	✔	✘	✘	✘	✘	✘	✔	✘
LISA [106]	✔	✘	✔	✘	✘	✔	✔	✘
MDM [99]	✔	✘	✔	✘	✘	✔	✔	✘
Dr(eye)ve [109]	✔	✘	✔	✘	✔	✔	✘	✘
LBW [110]	✔	✘	✔	✘	✔	✔	✔	✘
DGAZE [108]	✔	✔	✘	✔	✔	✔	✔	✘
DashGaze	✔	✔	✔	✔	✔	✔	✔	✔

Table 11.1: Comparison of the DashGaze dataset with other gaze datasets on the following parameters: 1. Is it a driving dataset; 2. Is dashcam the modality of capture; 3. Does the data represent natural behaviour or is it collected in simulation? 4. Does the dataset capture unstructured road events typical to the Global South? 6. Is the gaze ground truth continuous or categorical?; 7. Is the road view available?; 8. Is the driver view available?; 9. Is the driver's ego view available?; Evidently, our dataset scores a 'Yes' on all these parameters.

11.3 Dataset Statistics

The DashGaze dataset is the largest naturalistic driving dataset to date, featuring driver and road-facing cameras. A summary of existing eye gaze datasets is given in Table 11.1, where we compare various desirable attributes for a driver gaze dataset. The DashGaze dataset is the only true behavioral driver gaze dataset that provides the road, driver, and ego views. The dataset comprises 28 participants and 32 unique drives, each lasting over 10 minutes. The DashGaze dataset supports long-term driver behavior modeling. With over 0.9 million frames and over 10 hours of video, it offers substantial content for analysis and research. (Refer to Table 11.2 for size comparison). The data collection is designed to encompass diverse and naturalistic scenarios and involves five different cars, showcasing variations in lighting, driver heights, and the presence of passengers in the driver frames. Road frames demonstrate various weather conditions, traffic scenarios, environments, and times of the day. Participant ages range from 18 to 62, contributing to a broader representation of driver demographics.

Comparing the mean frame of DashGaze with two other datasets (see Figure 11.8), it becomes evident that DashGaze exhibits higher variation. The dataset includes six sessions with a single driver, two of which were collected at night, while four sessions feature drivers wearing glasses.



(a) Driver view (b) Road view

Figure 11.5: Data samples showing variations in the DasjGaze dataset w.r.t. lighting, pose, camera position, traffic and weather conditions.

Dataset	Gaze	Subjects	Size
LISA [106]	Gaze zones	10	47K frames
MDM [99]	Markers	59	50.2 hrs *
Dr(eye)ve [109]	Continuous	8	0.5M frames (6 hrs)
LBW [110]	Continuous	28	0.1M frames (7 hrs)
DashGaze	Continuous	28	0.9M frames (10 hrs)

Table 11.2: A comparison of recent driver gaze datasets which feature subjects driving on roads. *For MDM, only a portion of the dataset is on the road. Naturalistic gaze is not provided

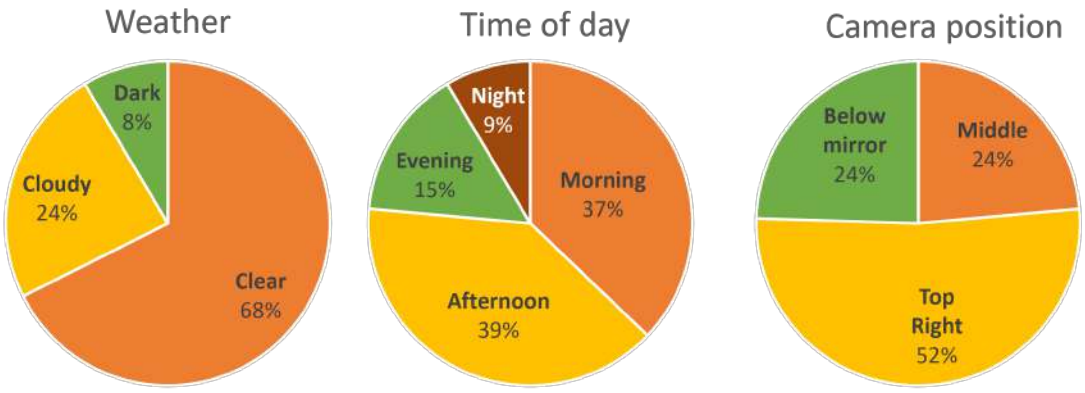


Figure 11.6: Distribution of DashGaze dataset with respect to weather, time of day and camera position

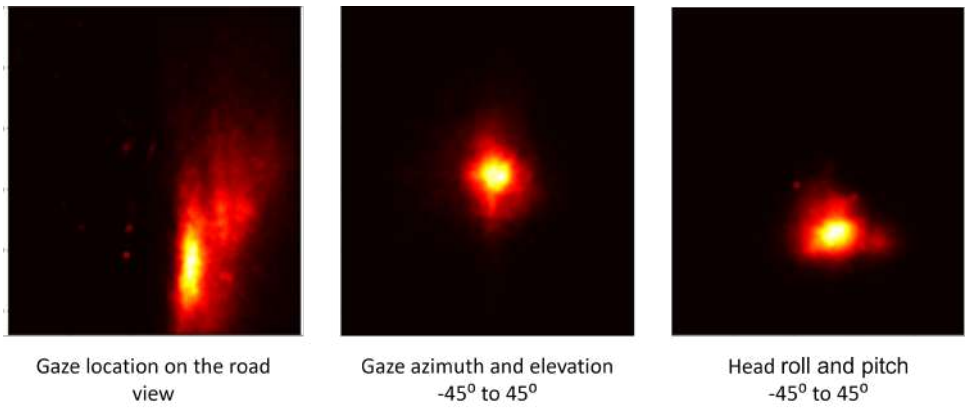


Figure 11.7: Distribution of gaze targets on road view (left), gaze azimuth and elevation (middle) and head pose (right) in the DashGaze dataset.

Figure 11.8: Mean driver frames of three driver gaze datasets that have driver-facing cameras. We can see that our dataset has more variation than other similar datasets, where driver is generally in the same position of the captured frame.

Figure 11.9: Mean road frames of DashGaze compared to the Dr(eye)ve dataset [109]. While the Dr(eye)ve dataset is collected mostly on highways as evidenced by the mean frame, our road frames have high diversity.

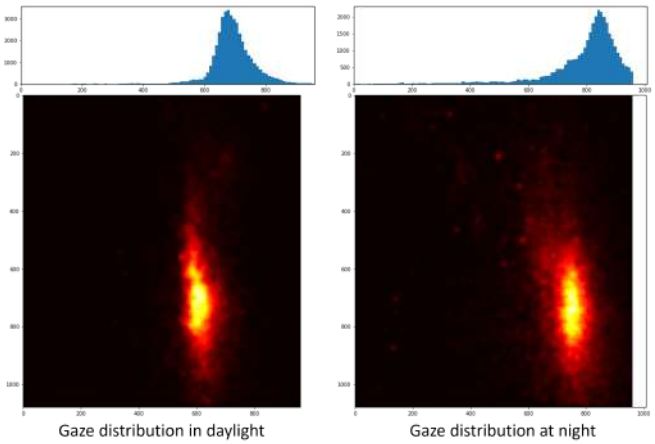


Figure 11.10: Distribution of gaze for the same driver in the morning versus night

The dataset encompasses a wide distribution of gaze angles, head poses, and gaze locations (see Figure 11.7). Furthermore, it covers three distinct weather conditions and dashcam positions, making it more comprehensive in capturing real-world driving situations. The distribution of the dataset concerning weather, time of day, and camera position is illustrated in Figure 11.6.

11.4 Analysis of Driver Gaze

11.4.1 Gaze Bias Towards Vanishing Point

Our analysis of driver gaze in the DashGaze dataset reveals an interesting bias towards the vanishing point (see Figure 11.7). The vanishing point is shifted to the right in our dataset due to the camera's

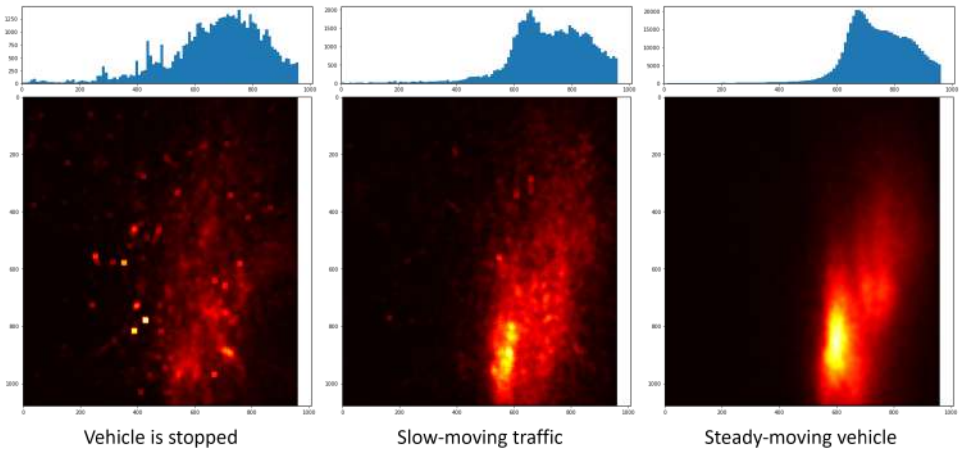


Figure 11.11: Gaze distribution at different speeds of movement

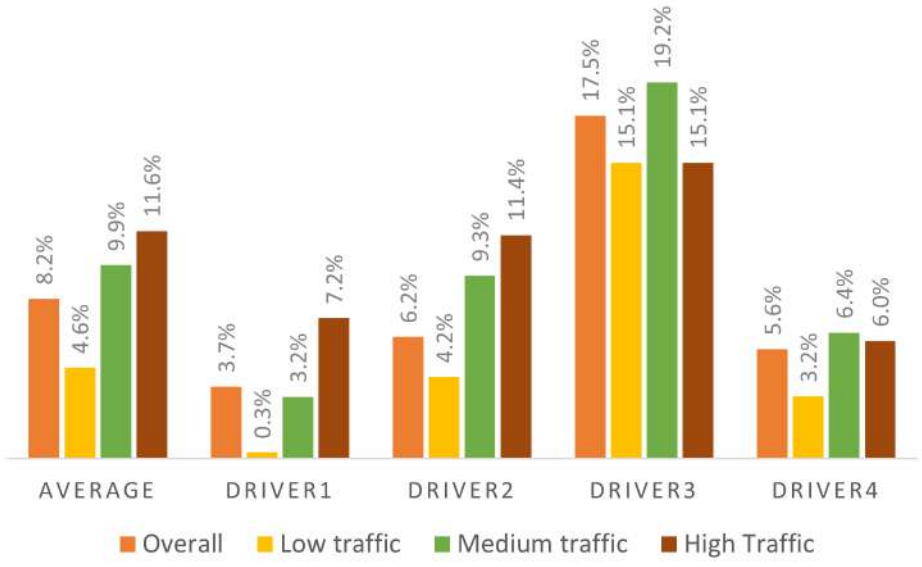


Figure 11.12: Percentage of time drivers look at an object in different traffic conditions

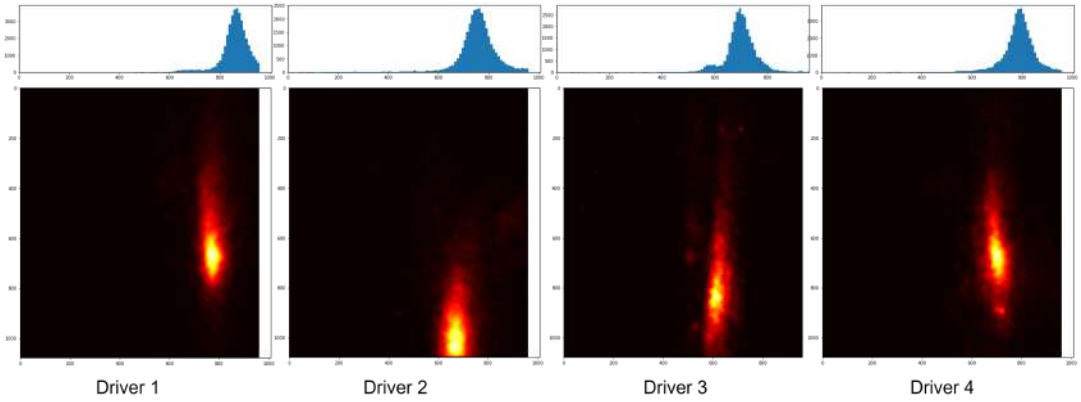


Figure 11.13: Gaze distribution for four different drivers

placement and the right-hand driving position, causing the vanishing point to be shifted to the right. This observation aligns with previous gaze studies [109, 221, 222]. Additionally, our study highlights a significant vertical spread of the gaze, likely influenced by the diverse driving situations present in the dataset, including busy roads and stopped traffic.

Furthermore, our investigation indicates that drivers focus less on specific objects, such as cars or pedestrians, than the road or background (constituting only 8.2% of the gaze). Figure 11.12 illustrates how this percentage varies under different traffic conditions, revealing that drivers tend to look more at objects during heavy traffic situations, which aligns with expected behaviors.

11.4.2 Effect of Lighting on Gaze Distribution

We analyzed the effect of lighting on gaze distribution. Figure 11.10 shows the gaze distribution of the same driver during the day and night. We observe that the horizontal spread of gaze is greater during the night. This contrasts with a previous study [223], which found that in simulated driving, there was a slightly less horizontal deviation of gaze during night scenes than in daylight scenes.

11.4.3 Effect of Traffic on Gaze Distribution

The horizontal distribution of gaze becomes consolidated towards one point as the speed of the vehicle increases. In Figure 11.11, the driver's gaze is scattered over the entire field of view when the vehicle is stopped due to traffic. According to [224], the horizontal deviation decreases as the speed increases. Figure 11.11 confirms this observation for our dataset.

11.4.4 Characteristic Gaze of Individual Drivers

Figure 11.13 shows the gaze patterns of four drivers. The perceived differences in their distribution are due to camera placement and driver height. We found no significant deviations between individual gaze patterns at different lighting conditions or speeds. However, We noted that individual drivers differ

in observing objects like vehicles and pedestrians on the road. Figure 11.12 shows that the gaze of some drivers falls on objects significantly more than others.

11.5 Summary

In this chapter, we introduced DashGaze, a pioneering naturalistic driver gaze dataset aimed at advancing appearance-based driver gaze estimation using only uncalibrated dashcam footage. DashGaze provides a diverse collection of 900,000 frames from 33 driving sessions with 28 unique drivers, each lasting over 10 minutes. The dataset includes road views, driver views, and driver's egocentric views, along with gaze ground truth, blink, fixation, and IMU data. By comparing DashGaze with other driver gaze datasets, we demonstrated its advantages regarding realism, diversity, and paired images of the driver and road. Our data collection procedure utilized a single dashcam and an eye tracker, capturing true behavioral data during naturalistic driving scenarios. We emphasized the potential of DashGaze for explaining driver attention using explainability methods. We gained insights into driver behavior under various driving situations by exploring driver gaze biases towards vanishing points and observing significant vertical spread. The dataset's richness allowed us to study gaze behavior in different weather, time of day, and camera positions.

In conclusion, DashGaze offers a valuable resource for training dashcam-based gaze models and studying long-term driver behavior. The dataset's unique characteristics facilitate research in understanding driver attention, providing opportunities for further advancements in driver safety and assistive technologies. The next chapter studies methods to estimate the driver gaze using appearance-based methods and applies them to the DashGaze dataset.

Chapter 12

Appearance-Based Driver Gaze Estimation

12.1 Introduction

In the previous chapter, we introduced the DashGaze dataset to study driver attention saliency. In this chapter, we present a novel method for estimating driver gaze using only a dashcam.

Driver gaze monitoring has become an essential aspect of driving safety in recent years due to its potential to reduce the risk of driver distraction and improve situational awareness. A study by the National Highway Traffic Safety Administration (NHTSA) found that driver inattention was a major factor in 10% of fatal crashes in 2020 [225]. In another study, researchers found that the average driver takes their eyes off the road for approximately 4.6 seconds while texting, which is enough time to travel the length of a football field at 55 mph [226]. These findings highlight the critical role of driver attention in preventing accidents and underscore the need for effective gaze monitoring solutions. However, the lack of practical solutions has hindered the implementation of gaze monitoring in vehicles. Currently, its implementation depends on various factors, including cost and reliability. The cost can include cameras, sensors, and software that must be integrated with the vehicle's existing systems. Eye trackers, the most commonly used technology for gaze monitoring, are both cumbersome and expensive, which limits their feasibility for mass production and widespread adoption.

We present DashGazeNet: our baseline model for dashcam-based gaze estimation. It is a multi-branch CNN-based model that has two stages. The first stage estimates the gaze angle with respect to the driver, and the second stage calculates the 2D gaze point on the road. We achieve a state-of-the-art gaze angle estimation error of 0.6 degrees and a gaze point error of 279 pixels, approximately 5 meters, on unseen sessions and drivers.

12.2 DashGazeNet: Methodology

We aim to estimate the target location of the driver's gaze on the road using only the driver-view dashcam as input. This setting ensures maximum flexibility during deployment and ensures that the road video does not bias our model to predict only salient objects. The position of the dashcam and driver's head may vary for each session and driver. We aim to ensure that our model is calibration-free

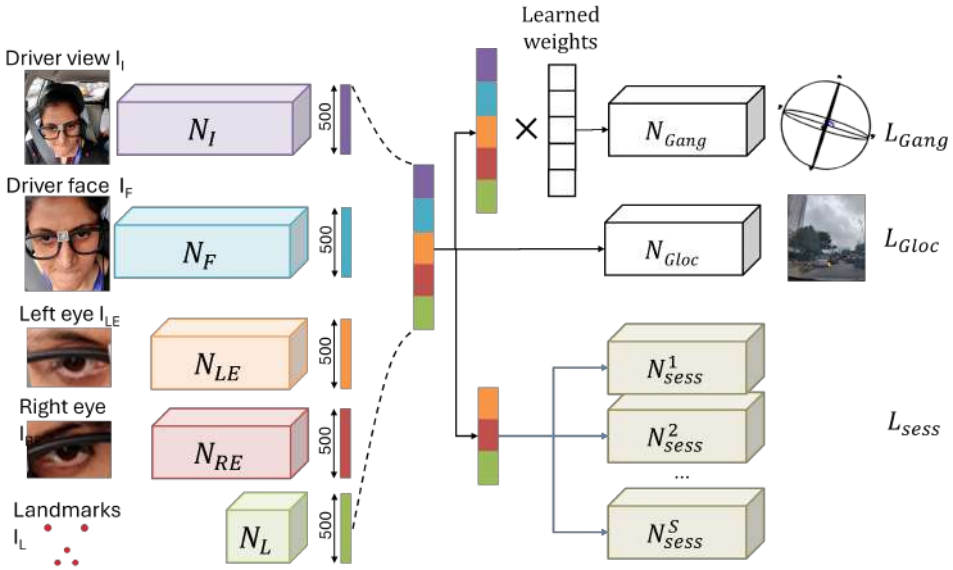


Figure 12.1: The DashGazeNet architecture, depicted in the accompanying image, employs five input branches, including driver view, face, left eye, right eye, and landmarks, to estimate gaze angle and location using a multi-task learning approach. Notably, the architecture includes a separate self-supervised session branch for each dataset session, thereby enhancing the accuracy and precision of the model.

and works out of the box for most settings and drivers. The DashGazeNet model is a multi-branch multi-task, calibration-free model for appearance-based driver gaze prediction (see Figure 12.1 for the overall architecture). We describe our architecture and method below.

12.2.1 Multi-Branch Input

Similar to [108], we designed our model with multiple input branches. The first branch $N_I(I_I \in \mathbb{R}^{3\times224\times224})$ has a ResNet50 architecture and takes the entire driver-view image as input. We down-sample the input image to 224×224 pixels. The face branch $N_F(I_F \in \mathbb{R}^{3\times224\times224}$ has a ResNet50 architecture and takes in an image of the driver's cropped face resized to 224×224 pixels. We detect and crop the face using MTCNN [227]. The left-eye and right-eye branches $N_{LE}(I_{LE} \in \mathbb{R}^{3\times50\times100})$ and $N_{RE}(I_{LE} \in \mathbb{R}^{3\times50\times100})$ have four convolutional layers followed by a linear layer. The landmark branch $N_L(I_L \in \mathbb{R}^1 0)$ is a fully connected branch that takes the x and y locations of five facial landmarks provided by MTCNN. All the input branches $N_I, N_F, N_{LE}, N_{RE}, and N_L$ output feature vectors of length 500.

12.2.2 Gaze Location Prediction

The gaze location branch N_{Gloc} is the main output branch of the DashGazeNet. It outputs the x and y location of the gaze with respect to the road view. Accurate gaze estimation requires knowledge of the gaze angle, head pose, and head position. Hence, we concatenate the features of the input branches

as input to the gaze location branch.

$$I_{Gloc} = N_F(I_F) \oplus N_{LE}(I_{LE}) \oplus N_{RE}(I_{RE}) \oplus N_I I_I \oplus N_L(I_L) \tag{12.1}$$

We use mean squared error as a loss for gaze location.

$$L_{Gloc}(N_{Gloc}(I_{Gloc}), G) = \sum_{i=1}^{D} (N_{Gloc}(I_{Gloc})_i - G_i)^2 \tag{12.2}$$

where D is the number of samples, and G is the ground truth gaze location.

12.2.3 Gaze Angle Prediction

The gaze angle branch N_{Gang} learns the azimuth and elevation of the gaze vector with respect to the face (not the driver camera). Recent studies show that the appearance of the left and right eyes differ for the same angle, and thus it is beneficial to use both eyes for gaze estimation [87, 228]. In our dataset, we notice that sometimes the left or right eye or face may not be fully visible. Hence we combine the features of all the input branches with learned weights:

$$\begin{aligned} I_{Gang} =& w_F * N_F(I_F) + w_{LE} * N_{LE}(I_{LE}) + w_{RE} * N_{RE}(I_{RE}) \\ &+ w_I * N_I(I_I) + w_L * N_L(I_L) \end{aligned} \tag{12.3}$$

where all the weights w are learned.

We learn the gaze location and the eye gaze angle (azimuth and elevation) in a multi-task manner to achieve better performance and learn more robust and universal representations [229]. Thus, the combined loss is:

$$L_G = L_{Gloc} + \lambda L_{Gang} \tag{12.4}$$

where λ is a hyperparameter.

12.2.4 Calibration-Free Estimation

As stated in Section 11.2, the positions of the driver's head and the dashcam angle will change between sessions. We aim for the DashGazeNet to adapt between sessions without calibration. We include a *session loss* to learn the difference between sessions. The session branch N_{sess} is a fully-connected network that uses the left and right eye images and the five facial landmark points. We don't provide the driver's face, or driver view image to impede the model from learning shortcuts to distinguish between sessions. We create a separate session branch N^s_{sess} for each session s in the training set without sharing weights. Each session branch learns the self-supervised task of detecting translation in the facial landmark points.

Let $M^s \in \mathbb{R}^B$ be the session mask for session s, where b is the batch size.

$$\begin{aligned} M_i^s &= 1 \quad \forall\, I_i \in s \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{12.5}$$

We concatenate the features of the left eye, right eye, and landmark branches as input:

$$I_{sess}^s = M^s * (I_{LE} \oplus I_{RE} \oplus I_L) \tag{12.6}$$

We alternate batches with normal and augmented facial landmarks, where each sample's landmark is translated by a random number between -20 and 20. All landmark points of a single sample are translated by the same amount. We use cross-entropy loss to learn whether a sample is translated or not.

$$\begin{aligned} &L_{sess}^s(N_{sess}^s(I_{sess}^s), y) = \\ &-(y \log(N_{sess}^s(I_{sess}^s)) + (1-y) \log(1 - N_{sess}^s(I_{sess}^s))) \end{aligned} \tag{12.7}$$

where y is 1 for translated samples and 0 otherwise. The total training loss for our model is given by

$$L = L_B + \sum_{s=1}^{S} L_{sess}^s \tag{12.8}$$

We consider all incoming samples as belonging to a single session during inference. We discard the trained session branches and create a new session branch N_{sess}^i. As we aim to have calibration-free gaze estimation, we use test-time adaptation [230] to adapt to the new settings. For each batch of training samples, we obtain the gaze location using Equation 12.1. We then follow the above procedure to provide alternate batches of translated landmarks. We do a single iteration of weight updation for each batch using the loss in Equation 12.7. This teaches the model to adapt to the test conditions.

12.3 Experiments and Results

This section comprehensively evaluates DashGazeNet's performance for gaze angle and location estimation on the DashGaze dataset. Our model surpasses other gaze models in accurately predicting driver gaze. The average pixel error for gaze location is 279.11 pixels, while the prediction error for gaze angle is 7.91° and 6.86° for azimuth and elevation, respectively. We present qualitative results for the DashGazeNet architecture, including an ablation study that assesses the various components of the model. Additionally, we visualize the input feature saliency for gaze prediction. Our experiments employ 25 sessions for training and 8 sessions for testing, with different drivers in the test sessions compared to the training sessions.

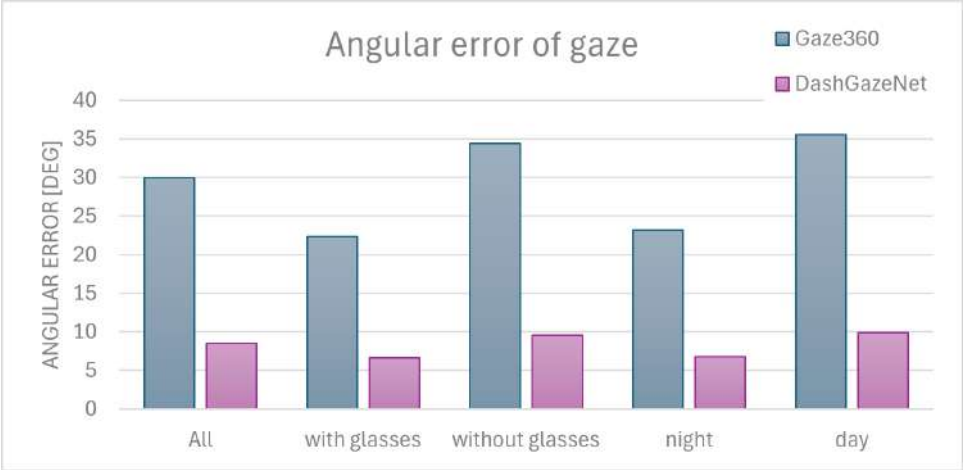


Figure 12.2: Graph comparing the angular error (in degrees) of gaze estimation across different methods (lower values indicate better performance). Errors are calculated for the entire dataset, as well as for subsets where the driver is wearing lenses, not wearing lenses, during daytime sessions, and nighttime sessions.

12.3.1 Experimental Setup

We split our data into 82% train, 10% validation and 8% test. While splitting the data, we ensured to not split a driving session into different splits. We ensured that the sessions in the test split had a variety of lightings, drivers, camera positions and cars.

Our model outputs both gaze angles (azimuth and elevation) and gaze locations (x and y coordinates relative to the road view). We quantify gaze error as the mean angular deviation (in degrees) between the predicted and ground truth gaze directions. The location error is calculated as the root mean squared error (RMSE) between the predicted and ground truth coordinates.

$$E_{Ang} = \frac{1}{N}\Sigma_i cos^{-1}(\mathbf{g}_i^T \hat{\mathbf{g}}_i) \tag{12.9}$$

$$E_{Loc} = \frac{1}{N}\Sigma_i \sqrt{(\mathbf{l}_i - \hat{\mathbf{l}}_i)^2} \tag{12.10}$$

where g_i and l_i are the i_{th} predictions for gaze angle and location, and $\hat{g}_i$ and $\hat{l}_i$ are the ground truth. N stands for the number of samples.

12.3.2 Baseline Gaze Estimation Results

We compare our gaze angle estimates with Gaze360 [90], an off-the-shelf gaze estimator designed for 'in-the-wild' scenarios. We use the static models provided by the Gaze360 authors, which are trained on the Gaze360 dataset. Since our model outputs gaze angles relative to the face rather than the driver's view, we employ a head pose estimator [231, 232] to estimate the driver's head pose and adjust our gaze angles accordingly. Figure 12.2 presents the results of this experiment. The angular error across the

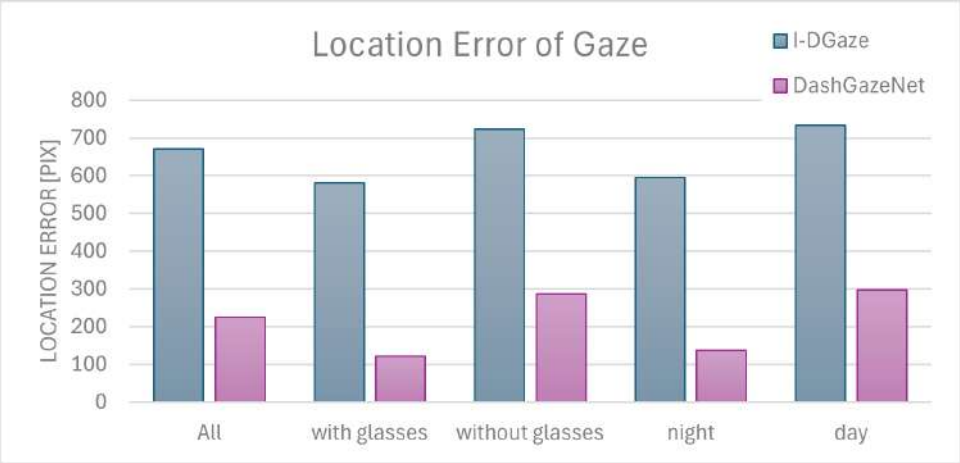


Figure 12.3: Graph comparing the location error (in pixels) of gaze estimation across different methods (lower values indicate better performance). Errors are calculated for the entire dataset, as well as for subsets where the driver is wearing lenses, not wearing lenses, during daytime sessions, and nighttime sessions.

Figure 12.4: Qualitative results of DashGazeNet on randomly picked samples. Top row: Qualitative results of gaze location estimation. The green dots represent the ground truth, and the red dots represent the predicted gaze location. Bottom row: Qualitative results of gaze angle location.

entire test set is 8.48°$\pm$ 0.04°for DashGazeNet, compared to 29.97°$\pm$ 0.11°for Gaze360. The figure also illustrates the gaze estimation performance with and without glasses and during both night and daytime driving. In all cases, DashGazeNet significantly outperforms Gaze360.

We compare our gaze location estimation with the I-DGaze model [108]. Like DashGazeNet, I-DGaze uses only the driver-side video to infer the two-dimensional gaze target location on the road video. We trained I-DGaze on the Dash-Gaze train set with the same hyperparameters as in the original work. As we see from the results in Figure 12.3, our model outperforms I-DGaze in all cases. For the entire dataset, the pixel error of DashGazeNet is 225.06 ± 1.32, whereas I-DGaze has an error of 669.93 ± 1.96 In comparison, the I-Dgaze model reported an error of 186.89 pixels on the DGaze dataset [108]. This shows that the Dash-Gaze dataset is collected in a more unconstrained setting than DGaze and requires more contextual information to estimate gaze on the Dash-Gaze dataset.

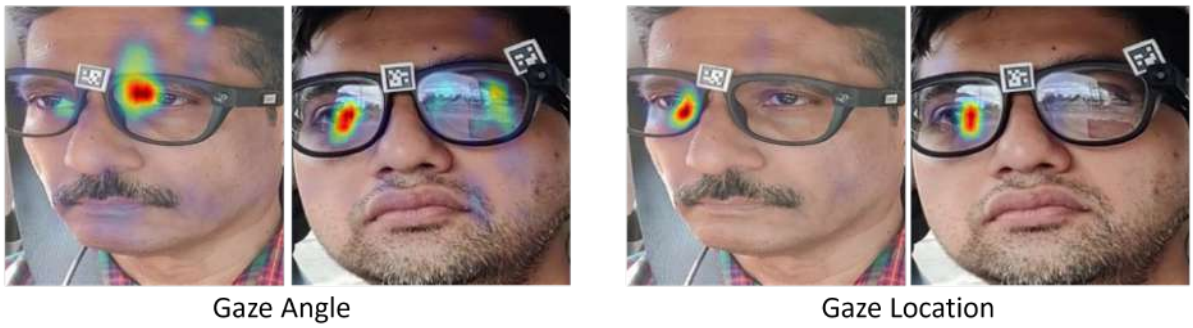


Figure 12.5: Visualization of the DashGazeNet model using occlusion maps on the input image. The left images show the saliency map for gaze angle prediction and the right images show the saliency map for gaze location prediction.

	Variation	Azimuth	Elevation	Location
1	No TTT	8.12°	6.15°	319.32pix
2	TTT all features	8.13°	6.80°	296.07pix
3	eyes + landmarks	9.67°	6.37°	287.69pix
4	weighted features	8.07°	**6.14**°	1047.22pix
5	**DashGazeNet**	**7.91**°	6.86°	**279.11pix**

Table 12.1: Results of ablation study on the DashGazeNet model. The numbers are errors (lower is better)

Figure 12.4 shows some qualitative results of gaze prediction on random samples. We see that the predicted gaze is near or overlapping the ground truth gaze in most cases, despite different drivers and dashcam positions. We also note that the gaze prediction is robust to various lighting and occlusions.

12.3.3 Visualization of Model Attention

We visualize the saliency of face input pixels for gaze angle and gaze location estimation in Figure 12.5. We systematically occluded windows of the input and mapped the difference in output between the occluded and unoccluded inputs onto a heatmap. We observe that the eyes are the most important features for both gaze angle and gaze location.

12.4 Ablation Studies

This section examines the impact of various parts of the DashGazeNet architecture on the model's performance. In Table 12.1, we see the results of four variants of our model. The first row shows the results of a variant that does not use test time adaptation. This model cannot predict the gaze location well on unseen sessions. The second row uses a variant of test time training where the session branch was asked to predict which session a sample came from during training. During inference, a new session

branch predicted if a sample was translated. The session branch took the concatenated feature vectors of all input branches as input. The third-row variant is the same as the second row except that only the eye and landmark features were used for training the session model. We see a marked improvement in the gaze location prediction in this variation. The fourth model variant learned weights for each input branch and combined the input branch features using weighted addition. While this improves the performance of gaze angle estimation, gaze location prediction does not work. Finally, we have our model, the DashGazeNet. It is similar to the third variant, except it uses weighted addition to combine the features for the gaze angle branch. In addition, one session branch for each session during training predicts whether a sample has been translated. This architecture gives the best results for gaze angle and gaze location predictions.

12.5 Discussion: Potential Societal Impact

Driver gaze tracking offers significant benefits for road safety and infrastructure design. Advanced Driver Assistance Systems (ADAS) may be built to utilize gaze tracking to alert drivers in case of inattention, thereby preventing accidents. These systems can potentially enhance situational awareness and response times by identifying potential accident-causing events that the driver may have missed. Gaze tracking may also be crucial in driver education, providing feedback to improve attentiveness and reaction. Analyzing driver gaze data can contribute to designing better roads and signage and optimizing their placement and visibility. It also informs improvements in road lighting, ensuring that critical information is visible, thus enhancing overall driving safety.

However, gaze-tracking technology raises significant privacy concerns, particularly regarding appearance-based methods that can be deployed without a person's consent. Unauthorized use of gaze tracking can lead to severe privacy violations, especially if the data is exploited for market research, capturing personal interests, or targeting advertisements without the individual's knowledge. In the context of driving, malicious actors could misuse gaze tracking data to design distracting signboards, intentionally diverting drivers' attention and increasing the risk of accidents. These potential abuses highlight the need for stringent regulations and ethical guidelines to protect individuals' privacy and ensure that gaze-tracking technology is used responsibly and transparently.

12.6 Summary

In this chapter, we presented DashGazeNet, a novel method for estimating driver gaze using only a dashcam. Our approach addresses the critical need for effective gaze monitoring solutions to enhance driving safety by reducing driver distraction and improving situational awareness. We introduced DashGazeNet as a multi-branch CNN-based model with two stages: gaze angle estimation concerning the driver and 2D gaze point estimation on the road.

Through extensive experimentation, we demonstrated the effectiveness of DashGazeNet, achieving state-of-the-art performance in gaze angle estimation with an error of 0.6 degrees and 2D gaze point estimation with an error of 279 pixels on unseen sessions and drivers. These impressive results showcase the potential of our approach for practical and cost-effective gaze monitoring solutions that can be readily integrated into existing dashcam systems.

By utilizing a dashcam-based approach, we aim to lower the barrier to adopting gaze monitoring technology for the automotive industry and make it more feasible for mass production and widespread implementation. Our method offers a valuable step towards safer driving practices, as it can help mitigate the risks associated with driver inattention and reduce the number of fatal crashes caused by distracted driving.

PART VI

Conclusion

Chapter 13

Summary and Future Works

This thesis explores deep face representations using explainability methods and functional concepts. It investigates the workings of deep models by visualizing face representations and developing novel saliency algorithms. Additionally, the thesis explores human visual saliency using the driver's gaze. We introduce a large-scale driver gaze dataset and propose an appearance-based model for driver gaze estimation using only a dashcam. In this chapter, we provide a detailed summary of the thesis and discuss possible future directions.

13.1 Summary

This thesis presents an in-depth investigation into deep face representations and their explainability. It is divided into four interconnected parts, each shedding light on different aspects of deep face representations.

In the first part, the thesis explores deep face representations using visualization algorithms. Chapter 3 provides a comprehensive background on existing feature visualization techniques and experimental results for face models, revealing the visual representations generated by these models. Chapter 3 introduces feature inversion algorithms, uncovering retained and discarded information at each layer and gaining valuable insights into various face models. The first part concludes with a user survey identifying factors to make face explainability methods more accessible to AI practitioners.

In the second part, the thesis delves into the functional concepts inherent in deep face representations.Chapter 5 discusses methods to define and find functional concepts in deep representations. Chapter 6 introduces Cross-Task Aware Filters (CRAFTS): convolutional filters of a face model that learn to predict related face tasks. Chapter 7 presents ETL, an efficient task-based pruning and transfer learning procedure using CRAFTS.

The third part investigates algorithms to discover salient input features. Chapter 8 presents an overview of saliency visualization algorithms for deep models. We discuss insights into deep face models and the limitations of applying generic saliency algorithms to the face domain Chapter 9 presents Canonical Image Saliency maps, a novel method that standardizes face saliency images and projects them to face coordinates. Chapter 10 introduces Canonical Model Saliency maps, providing valuable

insights into the working of deep face models across different architectures. The algorithm's diagnostic utility is demonstrated by detecting biases in gender classification using a celebrity face dataset.

The fourth part focuses on understanding human attention through the driver gaze. Chapter 11 introduces DashGaze, the largest naturalistic driver gaze dataset capturing driver and road views under real driving conditions. Analyzing driver gaze patterns reveals valuable insights into driver behavior under varying conditions. Chapter 12 presents DashGazeNet, an appearance-based driver gaze estimation model using only a dashcam's output.

13.2 Conclusion

In conclusion, this thesis significantly contributes to the face and biometrics community. It begins with a pioneering survey on face explainability methods, shedding light on the need for domain-specific algorithms and face-specific evaluation techniques. Introducing state-of-the-art face saliency algorithms and efficient transfer learning procedures further enhances our understanding of deep face representations and enables better model interpretation. The introduction of DashGaze, a large-scale driver gaze dataset, opens new avenues for studying human saliency and driver behavior under various conditions. The appearance-based driver gaze estimation model, DashGazeNet, built using the dataset, provides a practical and lightweight solution for gaze monitoring using only a dashcam.

The increasing deployment of deep face models in critical applications, such as security and law enforcement, raises significant concerns about the absence of tailored explainability methods. The potential ramifications of biased or inaccurate face models necessitate comprehensive analysis and understanding of these models prior to their deployment. By identifying this evident gap in the field, our thesis urges researchers and practitioners to delve deeper into face explainability, creating algorithms that are effective and interpretable to laypeople.

13.3 Future Directions

The thesis opens up many promising directions as listed below:

- **Explainability for generative AI:** The focus of the explainability in this thesis has primarily been on discriminative AI tasks, encompassing classification, regression, verification, and one-shot learning. However, the landscape of AI has witnessed a surge in generative models, driven by transformers and diffusion models. These large-scale models, with billions of parameters trained on massive datasets, can now generate data virtually indistinguishable from real samples. Notably, the face domain has seen substantial advancements in generative AI, exemplified by deepfake technology, face hallucination, and lip-sync applications. Despite these strides, there remains a paucity of research on explainability in these domains. As these technologies proliferate, developing domain-specific explainability methods to comprehend and validate their outputs becomes imperative, ensuring accountability and ethical deployment.

- **Explainability of complex social behaviours** While our current work has primarily focused on face tasks involving single faces in images, the future direction of explainability holds immense potential in extending its scope to more complex tasks with social interactions encompassing multiple faces. Tasks such as Visual Question Answering (VQA), Gaze Lock Detection (Look at Each Other or LAEO [233]), and understanding and summarizing movies [234, 235] involve images and videos with multiple moving individuals engaged in intricate interactions. Additionally, the domain of driver behavior analysis using gaze data also falls under this ambit. For instance, we envision addressing questions like, "Why did the driver swerve?" with an answer like, "Because they did not notice the pedestrian on time.". We must develop novel explainability methods to disentangle individual faces' contributions and interactions to achieve this extension. Exploring ethical implications becomes paramount when dealing with complex social interactions. Addressing fairness, bias, and privacy issues is crucial, especially in applications involving multiple individuals and sensitive social contexts. Explainability can play a pivotal role in ensuring the transparency and accountability of such AI systems, fostering trust among users and stakeholders.

- **Improve training using human attention** A promising future direction in AI research is to leverage human attention to enhance the training of deep models. By incorporating human-in-the-loop mechanisms, we can harness the unique capabilities of human attention to inform and guide machine-learning models. One potential application lies in improving recognition tasks, where deep models may benefit from learning more human-like attention patterns, such as defining characteristics of individuals for face recognition, instead of solely focusing on the eye-nose triangle. Additionally, in self-driving AI systems, human attention data can serve as a valuable guide to creating more reliable and safe autonomous driving systems. For example, using human attention as a supervision signal, the self-driving AI can be trained to prioritize important elements on the road that humans would naturally attend to, thus enhancing the AI's decision-making process. However, careful consideration is needed to determine the appropriateness of human attention data, as biases may color it. Additional research is essential to ensure that the integration of human attention in AI training is done ethically and responsibly to create more interpretable and human-like AI systems that inspire trust and acceptance in various domains.

- **Analysis and prediction of behaviour** In Chapter 11, we have collected a large dataset called DashGaze, which has dense gaze annotations for long drives. A future direction involves leveraging our gaze dataset to gain insights into long-term driver behavior through gaze fixations. Analyzing gaze patterns over extended periods can reveal trends and tendencies in attention allocation during different driving scenarios. Additionally, the dataset offers opportunities for predicting driver behavior based on gaze, aiding in developing advanced driver assistance systems and autonomous vehicles. Integrating multimodal data and employing deep learning techniques can enhance predictive capabilities, leading to personalized and adaptive driver behavior predic-

tion systems. Overall, this research holds promise for advancing human-vehicle interaction and autonomous driving, ultimately contributing to safer and more efficient transportation systems.

- **Assistive technologies for drivers using gaze** The future direction of developing assistive technologies for drivers using driver gaze is highly promising for enhancing road safety and driving experience. Analyzing driver gaze patterns makes detecting inattention and drowsiness in real-time possible. When the system recognizes signs of drowsiness, it can alert the driver to take a break or adopt necessary safety measures. Additionally, the system can act as an extra set of vigilant eyes, pointing out potential hazards, such as pedestrians or cars about to jump in front, which the driver may not have noticed. This proactive warning system can significantly reduce the risk of accidents and improve overall road safety. Integrating driver gaze data with existing driver assistance systems and autonomous vehicles can create a holistic and adaptive approach to assistive technologies, leading to safer and more efficient transportation for everyone on the road.

Bibliography

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2014, pp. 1701–1708.

[2] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," 2008.

[3] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021.

[4] Big Brother Watch UK. Face off. [Online]. Available: https://bigbrotherwatch.org.uk/all-campaigns/face-off-campaign/

[5] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on Fairness, Accountability and Transparency (ACM FAccT)*, 2018.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *International Conference on Computer Vision (ICCV)*, 2017.

[7] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 5525–5533.

[8] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1991, pp. 586–591.

[9] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and vision computing (IMAVIS)*, vol. 16, no. 5, pp. 295–306, 1998.

[10] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Transactions on Image processing (TIP)*, vol. 11, no. 4, pp. 467–476, 2002.

[11] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, pp. 2037–2041, 2006.

[12] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015.

[13] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[14] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 87–102.

[15] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[16] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4873–4882.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2009.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NeurIPS)*, 2012, pp. 1097–1105.

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference (BMVC)*, 2015.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 1–9.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[22] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of MAssachusetts, Amherst, Tech. Rep., 2010.

[23] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 41, no. 1, pp. 121–135, 2017.

[24] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," *Machine Learning and Knowledge Extraction*, 2021.

[25] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, 2019.

[26] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *arXiv preprint arXiv:2006.11371*, 2020.

[27] A. Shahroudnejad, "A survey on understanding, visualizations, and explanation of deep neural networks," *arXiv preprint arXiv:2102.01792*, 2021.

[28] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI)*, 2021.

[29] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 1601–1609.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[31] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning (ICML)*, 2014, pp. 647–655.

[32] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," pp. 7064–7073, 2017.

[33] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6078–6087.

[34] A. S. Morcos, D. G. T. Barrett, N. C. Rabinowitz, and M. Botvinick, "On the importance of single directions for generalization," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 961–969.

[35] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *CoRR*, 2016.

[36] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] G. Rhodes and D. A. Leopold, "Adaptive norm-based coding of face identity."

[38] T. Valentine, M. B. Lewis, and P. J. Hills, "Face-space: A unifying concept in face recognition research," vol. 69, no. 10, pp. 1996–2019.

[39] L. Chang and D. Y. Tsao, "The code for facial identity in the primate brain," vol. 169, no. 6, pp. 1013–1028.e14.

[40] J. J. Richler, O. S. Cheung, and I. Gauthier, "Holistic processing predicts face recognition," vol. 22, no. 4, pp. 464–471.

[41] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.

[42] H. Han, C. Otto, X. Liu, and A. K. Jain, "Demographic estimation from face images: Human vs. machine performance," *PAMI*, 2014.

[43] T. Ezure, E. Yagi, N. Kunizawa, T. Hirao, and S. Amano, "Comparison of sagging at the cheek and lower eyelid between male and female faces," *Skin Research and Technology*, 2011.

[44] K. Tsukahara, T. Fujimura, Y. Yoshida, T. Kitahara, M. Hotta, S. Moriwaki, P. S. Witt, F. A. Simion, and Y. Takema, "Comparison of age-related changes in wrinkling and sagging of the skin in caucasian females and in japanese females," *International Journal of Cosmetic Science*, 2004.

[45] K. Tsukahara, K. Sugata, O. Osanai, A. Ohuchi, Y. Miyauchi, M. Takizawa, M. Hotta, and T. Kitahara, "Comparison of age-related changes in facial wrinkles and sagging in the skin of japanese, chinese and thai women," *Journal of dermatological science*, 2007.

[46] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[48] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size," *ArXiv*, 2016.

[49] S. Hitawala, "Evaluating resnext model architecture for image classification," *ArXiv*, 2018.

[50] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[51] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, 2018.

[52] C. N. Duong, K. G. Quach, N. T. H. Le, N. Nguyen, and K. Luu, "Mobiface: A lightweight deep learning face recognition on mobile devices," *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019.

[53] A. Sharma and H. Foroosh, "Slim-cnn: A light-weight cnn for face attribute prediction," *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020.

[54] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *Journal of Machine Learning Research (JMLR)*, vol. 18, no. 187, pp. 1–30, 2018.

[55] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, "Compressing deep convolutional networks using vector quantization," *ArXiv*, 2014.

[56] M. Kim and P. Smaragdis, "Bitwise neural networks for efficient single-channel source separation," pp. 701–705, 2018.

[57] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[58] D. Miyashita, E. H. Lee, and B. Murmann, "Convolutional neural networks using logarithmic data representation," *ArXiv*, 2016.

[59] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Graf, "Pruning filters for efficient convnets," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[60] J.-H. Luo, H. Zhang, H. yu Zhou, C.-W. Xie, J. Wu, and W. Lin, "Thinet: Pruning cnn filters for a thinner net," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 41, pp. 2525–2538, 2018.

[61] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4335–4344, 2018.

[62] X. Ding, G. Ding, J. Han, and S. Tang, "Auto-balanced filter pruning for efficient convolutional neural networks," in *AAAI*, 2018.

[63] R. V. Soelen and J. W. Sheppard, "Using winning lottery tickets in transfer learning for convolutional neural networks," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[64] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

[65] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1398–1406, 2017.

[66] N. Lee, T. Ajanthan, and P. H. S. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *ArXiv*, vol. abs/1810.02340, 2019.

[67] M. S. Zhang and B. C. Stadie, "One-shot pruning of recurrent neural networks by jacobian spectrum evaluation," *ArXiv*, vol. abs/1912.00120, 2020.

[68] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient transfer learning," *ArXiv*, vol. abs/1611.06440, 2016.

[69] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv: Learning*, 2019.

[70] H. Jin, S. Zhang, X. Zhu, Y. Tang, Z. Lei, and S. Li, "Learning lightweight face detector with knowledge distillation," *2019 International Conference on Biometrics (ICB)*, 2019.

[71] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, "Effective training of convolutional neural networks for face-based gender and age prediction," *Pattern Recognition (PR)*, 2017.

[72] C. N. Duong, K. Luu, K. G. Quach, and N. T. H. Le, "Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks," *ArXiv*, 2019.

[73] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012.

[74] Y. Bengio, A. Bergeron, N. Boulanger-Lewandowski, T. Breuel, Y. Chherawala, M. Cisse, D. Erhan, J. Eustache, X. Glorot, X. Muller *et al.*, "Deep learners benefit more from out-of-distribution examples," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[75] R. Caruana, "Learning many related tasks at the same time with backpropagation," in *Advances in neural information processing systems (NeurIPS)*, 1995.

[76] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *International Conference on Computer Vision (ICCV)*, 2011.

[77] J. J. Lim, R. R. Salakhutdinov, and A. Torralba, "Transfer learning by borrowing examples for multiclass object detection," in *Advances in neural information processing systems (NeurIPS)*, 2011.

[78] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[79] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2010.

[80] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[81] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NeurIPS)*, 2014.

[82] Y. Zhong, J. Sullivan, and H. Li, "Face attribute prediction using off-the-shelf cnn features," in *2016 International Conference on Biometrics (ICB)*. IEEE, 2016, pp. 1–7.

[83] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning (ICML)*, 2019.

[84] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," in *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, 2021.

[85] H. Zhang, H. Zhao, C. Liu, and D. Yu, "Task-to-task transfer learning with parameter-efficient adapter," in *CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)*, 2020.

[86] C. Wang and X. Lan, "Model distillation with knowledge transfer in face classification, alignment and verification," *ArXiv*, 2017.

[87] T. Fischer, H. J. Chang, and Y. Demiris, "RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments," in *European Conference on Computer Vision (ECCV)*, 2018.

[88] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters (RAL)*, vol. 5, no. 2, pp. 676–682, 2020.

[89] R. Kothari, Z. Yang, C. Kanan, R. Bailey, J. B. Pelz, and G. J. Diaz, "Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities," *Scientific reports*, vol. 10, no. 1, p. 2539, 2020.

[90] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[91] P. Kratzer, S. Bihlmaier, N. Balachandra Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice, "Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze," *IEEE Robotics and Automation Letters (RAL)*, 2020.

[92] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2176–2184.

[93] S. He, H. R. Tavakoli, A. Borji, and N. Pugeault, "Human attention in image captioning: Dataset and analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8529–8538.

[94] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 41, no. 1, pp. 162–175, 2017.

[95] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *European Conference on Computer Vision (ECCV) Proceedings, Part V 16*. Springer, 2020, pp. 365–381.

[96] H. Tomas, M. Reyes, R. Dionido, M. Ty, J. Casimiro, R. Atienza, and R. Guinto, "Goo: A dataset for gaze object prediction in retail environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[97] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–12.

[98] J. D. Ortega, N. Kose, P. Cañas, M.-a. Chao, A. Unnervik, M. Nieto, O. Otaegui, and L. Salgado, "DMD : A Large-Scale Multi-Modal Driver Monitoring Dataset for Attention and Alertness Analysis," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2020.

[99] S. Jha, M. F. Marzban, T. Hu, M. H. Mahmoud, N. Al-Dhahir, and C. Busso, "The multimodal driver monitoring database: A naturalistic corpus to study driver attention," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2021.

[100] A. Jain, H. S. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena, "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture," *arXiv preprint arXiv:1601.00740*, 2016.

[101] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. Reiß, M. Voit, and R. Stiefelhagen, "Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2801–2810.

[102] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li, "Dada-2000: Can driving accident be predicted by driver attention*f* analyzed by a benchmark," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4303–4309.

[103] T. Wu, N. Martelaro, S. Stent, J. Ortiz, and W. Ju, "Learning when agents can talk to drivers using the inagt dataset and multisensor fusion," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–28, 2021.

[104] S. Ghosh, A. Dhall, G. Sharma, S. Gupta, and N. Sebe, "Speak2label: Using domain knowledge for creating a large scale driver gaze zone estimation dataset," *ICCVW*, 2021.

[105] R. F. Ribeiro and P. D. Costa, "Driver gaze zone dataset with depth data," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 2019, pp. 1–5.

[106] S. Vora, A. Rangesh, and M. M. Trivedi, "Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis," *IEEE Transactions on Intelligent Vehicles (IV)*, vol. 3, no. 3, pp. 254–265, 2018.

[107] M. Selim, A. Firintepe, A. Pagani, and D. Stricker, "Autopose: Large-scale automotive driver head pose and gaze dataset with deep head orientation baseline," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020. [Online]. Available: http://autopose.dfki.de

[108] I. Dua, T. A. John, R. Gupta, and C. V. Jawahar, "Dgaze: Driver gaze mapping on road," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2020.

[109] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, "Predicting the driver's focus of attention: the dr(eye)ve project," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.

[110] I. Kasahara, S. Stent, and H. S. Park, "Look both ways: Self-supervising driver gaze estimation and road scene saliency," in *European Conference on Computer Vision (ECCV)*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 126–142.

[111] Y. Yang, C. Liu, F. Chang, Y. Lu, and H. Liu, "Driver gaze zone estimation via head pose fusion assisted supervision and eye region weighted encoding," *IEEE Transactions on Consumer Electronics*, vol. 67, pp. 275–284, 2021.

[112] S. Vora, A. Rangesh, and M. M. Trivedi, "On generalizing driver gaze zone estimation using convolutional neural networks," *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 849–854, 2017.

[113] Y. Zhang, X. Yang, and Z. Ma, "Driver's gaze zone estimation method: A four-channel convolutional neural network model," *Proceedings of the 2020 2nd International Conference on Big-data Service and Intelligent Computation (BDSIC)*, 2020.

[114] Y. Wang, G. Yuan, and X. Fu, "Driver's head pose and gaze zone estimation based on multi-zone templates registration and multi-frame point cloud fusion," *Sensors (Basel, Switzerland)*, vol. 22, 2022.

[115] Y. Wang, G. Yuan, Z. Mi, J. Peng, X. Ding, Z. Liang, and X. Fu, "Continuous driver's gaze zone estimation using rgb-d camera," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[116] Z. Yu, X. Huang, X. Zhang, H. Shen, Q. Li, W. Deng, J.-B. Tang, Y. Yang, and J. Ye, "A multimodal approach for driver gaze prediction to remove identity bias," *Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI)*, 2020.

[117] L. Stappen, G. Rizos, and B. Schuller, "X-aware: Context-aware human-environment attention fusion for driver gaze prediction in the wild," *Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI)*, 2020.

[118] G. Yuan, Y. Wang, H. Yan, and X. Fu, "Self-calibrated driver gaze estimation via gaze pattern learning," *Knowledge-Based Systems*, vol. 235, p. 107630, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121008923

[119] S. M. Shah, Z.-L. Sun, K. Zaman, A. Hussain, M. Shoaib, and L. Pei, "A driver gaze estimation method based on deep learning," *Sensors (Basel, Switzerland)*, vol. 22, 2022.

[120] B. Vasli, S. Martin, and M. M. Trivedi, "On driver gaze estimation: Explorations and fusion of geometric and data driven approaches," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 655–660, 2016.

[121] Z. Hu, C. Lv, P. Hang, C. Huang, and Y. Xing, "Data-driven estimation of driver attention using calibration-free eye gaze and scene features," *IEEE Transactions on Industrial Electronics*, vol. 69, pp. 1800–1808, 2021.

[122] L. Yang, K. Dong, A. J. Dmitruk, J. Brighton, and Y. Zhao, "A dual-cameras-based driver gaze mapping system with an application on non-driving activities monitoring," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 21, pp. 4318–4327, 2020.

[123] M. Shirpour, S. S. Beauchemin, and M. A. Bauer, "A probabilistic model for visual driver gaze approximation from head pose estimation," *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, pp. 1–6, 2020.

[124] S. K. Jha and C. Busso, "Probabilistic estimation of the driver's gaze from head orientation and position," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[125] S. Mohan and M. R. Phirke, "Eye gaze estimation invisible and ir spectrum for driver monitoring system," *Signal & Image Processing : An International Journal*, vol. 11, pp. 1–20, 2020.

[126] J. Lemley, A. Kar, A. Drimbarean, and P. M. Corcoran, "Convolutional neural network implementation for eye-gaze estimation on low-quality consumer imaging systems," *IEEE Transactions on Consumer Electronics*, vol. 65, pp. 179a–187, 2019.

[127] U. Sonom-Ochir, S. Karungaru, K. Terada, and A. Ayush, "Appearance-based driver's gaze mapping using a dash camera," *2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS)*, pp. 1–5, 2022.

[128] A. Rangesh, B. Zhang, and M. M. Trivedi, "Gaze preserving cyclegans for eyeglass removal and persistent gaze estimation," *IEEE Transactions on Intelligent Vehicles (IV)*, vol. 7, pp. 377–386, 2020.

[129] A. M. Oygard. Visualizing googlenet classes. [Online]. Available: https://www.auduno.com/2015/07/29/visualizing-googlenet-classes/

[130] A. Mordvintsev, C. Olah, and M. Tyka. (2015) Deepdream - a code example for visualizing neural networks. [Online]. Available: http://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html

[131] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *International Conference on Machine Learning (ICML)*, 2009.

[132] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, 2009.

[133] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017.

[134] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, 2014.

[135] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[136] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[137] M. D. Zeiler and R. Fergus, "Exploring features and attributes in deep face recognition using visualization techniques," in *Automatic Face and Gesture Recognition (FG)*, 2019.

[138] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 427–436.

[139] A. M. Nguyen, J. Yosinski, and J. Clune, "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks," *CoRR*, 2016.

[140] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding intra-class knowledge inside cnn," *CoRR Vol abs/1507.02379*, 2015.

[141] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[142] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[143] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[144] M. Tyka. Class visualization with bilateral filters. [Online]. Available: https://mtyka.github.io/deepdream/2016/02/05/bilateral-class-vis.html

[145] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *International Conference on Computer Vision (ICCV)*, 2013.

[146] L. Giulivi, M. J. Carman, and G. Boracchi, "Perception visualization: Seeing through the eyes of a dnn," in *British Machine Vision Conference (BMVC)*, 2021.

[147] D. Dangwal, "Mitigating reverse engineering attacks on local feature descriptors," in *British Machine Vision Conference (BMVC)*, 2021.

[148] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google Research Blog. Retrieved June*, 2015.

[149] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[150] G. Lowe, "Sift-the scale invariant feature transform," *International Journal of Computer Vision (ICCV)*, 2004.

[151] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)*, 2011.

[152] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2012.

[153] E. d'Angelo, A. Alahi, and P. Vandergheynst, "Beyond bits: Reconstructing images from local binary descriptors," in *International Conference on Pattern Recognition (ICPR)*, 2012.

[154] P. Weinzaepfel, H. Jégou, and P. Pérez, "Reconstructing an image from its local descriptors," in *IEEE International Conference on Pattern Recognition (CVPR)*, 2011.

[155] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[156] A. Singh and A. Namboodiri, "Laplacian pyramids for deep feature inversion," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 286–290.

[157] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9446–9454.

[158] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *CoRR abs/1412.6806*, 2014.

[159] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4829–4837.

[160] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *Advances in neural information processing systems (NeurIPS)*, 2016.

[161] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014.

[162] E. Rosch, "Principles of categorization," *Concepts: Core Readings*, 1999.

[163] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[164] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, 2013.

[165] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International Conference on Machine Learning (ICML)*, 2018.

[166] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[167] F. M. Graetz. (2019) How to visualize convolutional features in 40 lines of code. [Online]. Available: https://towardsdatascience.com/how-to-visualize-convolutional-features-in-40-lines-of-code-70b7d87b0030

[168] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.

[169] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[170] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops (CVPR-W)*. IEEE, 2010, pp. 94–101.

[171] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial features," in *ICPR International Workshop on Visual Observation of Deictic Gestures*. Citeseer, 2004.

[172] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2089–2093.

[173] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[174] B. Yin, L. Tran, H. Li, X. Shen, and X. Liu, "Towards interpretable face recognition," in *International Conference on Computer Vision (ICCV)*, 2019.

[175] R. Rothe, R. Timofte, and L. V. Gool, "Dex: Deep expectation of apparent age from a single image," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015.

[176] A. T. Lopes, E. de Aguiar, A. F. D. Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order," *Pattern Recognition (PR)*, 2017.

[177] G. Hu, F. Yan, J. Kittler, W. Christmas, C. H. Chan, Z. Feng, and P. Huber, "Efficient 3d morphable face model fitting," *Pattern Recognition (PR)*, 2017.

[178] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996.

[179] I. Dua, A. U. Nambi, C. V. Jawahar, and V. N. Padmanabhan, "Autorate: How attentive is the driver?" *2019 14th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–8, 2019.

[180] I. Dua, A. U. Nambi, C. V. Jawahar, and V. N. Padmanabhan, "Evaluation and visualization of driver inattention rating from facial features," *IEEE Transactions on Biometrics, Behavior, and Identity Science (T-BIOM)*, 2020.

[181] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "Agedb: the first manually collected, in-the-wild age database," in *Proceedings of IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR-W)*, 2017.

[182] J. Kossaifi, G. Tzimiropoulos, S. Todorovic, and M. Pantic, "Afew-va database for valence and arousal estimation in-the-wild," *Image and Vision Computing (IMAVIS)*, 2017.

[183] A. Dhall, R. Goecke, S. Lucy and T. Gedeon, "Collecting large, richly annotated facial-expression databases from movies," *IEEE MultiMedia*, 2012.

[184] M. Koestinger, P. Wohlhart, P. M. Roth and H. Bischof, "Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization," in *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (BeFIT)*, 2011.

[185] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[186] T. A. John, V. N. Balasubramanian, and C. V. Jawahar, "Canonical saliency maps: Decoding deep face models," *IEEE Transactions on Biometrics, Behavior, and Identity Science (T-BIOM)*, vol. 3, no. 4, pp. 561–572, 2021.

[187] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[188] F. Doshi-Velez and B. Kim, "Considerations for evaluation and generalization in interpretable machine learning," in *Explainable and interpretable models in computer vision and machine learning*, 2018.

[189] W. Silva, K. Fernandes, M. J. Cardoso, and J. S. Cardoso, "Towards complementary explanations using deep neural networks," in *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, 2018.

[190] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning (ICML)*, 2017.

[191] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *International Conference on Machine Learning (ICML)*, 2017.

[192] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[193] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, 2015.

[194] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Computer Vision (ICCV)*, 2017.

[195] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision (IJCV)*, 2018.

[196] Y. Zhong and W. Deng, "Deep difference analysis in similar-looking face recognition," in *International Conference on Pattern Recognition (ICPR)*, 2018.

[197] L. S. Shapley, "17. a value for n-person games," in *Contributions to the Theory of Games (AM-28), Volume II*, 2016.

[198] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[199] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[200] G. Castanon and J. Byrne, "Visualizing and quantifying discriminative features for face recognition," in *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2018.

[201] J. R. Williford, B. B. May, and J. Byrne, "Explainable face recognition," in *European Conference on Computer Vision (ECCV)*, 2020.

[202] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *International Conference on Machine Learning (ICML)*, 2017.

[203] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proceedings of the IEEE*, 2021.

[204] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[205] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Improved visual explanations for deep convolutional networks," *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[206] T. Zee, G. Gali, and I. Nwogu, "Enhancing human face recognition with an interpretable neural network," in *International Conference on Computer Vision -Workshop (ICCVW)*, 2019.

[207] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2016.

[208] J. S. Kim, G. Plumb, and A. Talwalkar, "Sanity simulations for saliency methods," *arXiv preprint arXiv:2105.06506*, 2021.

[209] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems (NeurIPS)*, 2018.

[210] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2009.

[211] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," in *European Conference on Computer Vision (ECCV)*, 2018.

[212] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[213] P. Huber, G. Hu, J. R. Tena, P. Mortazavian, W. P. Koppen, W. J. Christmas, M. Rätsch, and J. Kittler, "A multiresolution 3d morphable face model and fitting framework," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2016.

[214] P.-L. Carrier, A. Courville, I. J. Goodfellow, M. Mirza, and Y. Bengio, "Fer-2013 face database," *Technical report*, 2013.

[215] K. Karkkainen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[216] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," in *Advances in neural information processing systems (NeurIPS)*, 2019.

[217] M. Xu, Y. Ren, and Z. Wang, "Learning to predict saliency on face images," *International Conference on Computer Vision (ICCV)*, 2015.

[218] M. Georgopoulos, Y. Panagakis, and M. Pantic, "Modeling of facial aging and kinship: A survey," *Image and Vision Computing (IMAVIS)*, vol. 80, pp. 58–79, 2018.

[219] M. Tonsen, C. K. Baumann, and K. Dierkes, "A high-level description and performance evaluation of pupil invisible," 2020.

[220] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park, "Humbi: A large multiview dataset of human body expressions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (PAMI)*, 2020.

[221] A. Borji, M. Feng, and H. Lu, "Vanishing point attracts gaze in free-viewing and visual search tasks," *Journal of Vision*, vol. 16, no. 14, pp. 18–18, 2016.

[222] Y. Ueda, Y. Kamakura, and J. Saiki, "Eye movements converge on vanishing points during visual search," *Japanese Psychological Research*, vol. 59, no. 2, pp. 109–121, 2017.

[223] P. Konstantopoulos, P. Chapman, and D. Crundall, "Driver's visual attention as a function of driving experience and visibility. using a driving simulator to explore drivers' eye movements in day, night and rain driving," *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 827–834, 2010.

[224] J. Rogé, T. Pébayle, E. Lambilliotte, F. Spitzenstetter, D. Giselbrecht, and A. Muzet, "Influence of age, speed and duration of monotonous driving task in traffic on the driver's useful visual field," *Vision research*, vol. 44, no. 23, pp. 2737–2744, 2004.

[225] National Highway Traffic Safety Administration. (2022) Overview of motor vehicle crashes in 2020. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813266

[226] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. A. Perez, M. Buchanan-King, and J. M. Hankey, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proceedings of the National Academy of Sciences*, vol. 113, pp. 2636 – 2641, 2016.

[227] B. Jiang, Q. Ren, F. Dai, J. Xiong, J. Yang, and G. Gui, "Multi-task cascaded convolutional neural networks for real-time dynamic face recognition method," in *Communications, Signal*

Processing, and Systems: Proceedings of the 2018 CSPS Volume III: Systems 7th. Springer, 2020, pp. 59–66.

[228] Y. Cheng, F. Lu, and X. Zhang, "Appearance-based gaze estimation via evaluation-guided asymmetric regression," in *European Conference on Computer Vision (ECCV)*, 2018.

[229] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[230] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 9229–9248.

[231] T. Hempel, A. A. Abdelrahman, and A. Al-Hamadi, "Toward robust and unconstrained full range of rotation head pose estimation," *IEEE Transactions on Image Processing*, vol. 33, pp. 2377–2387, 2024.

[232] T. Hempel, A. A. Abdelrahman, and A. Al-Hamadi, "6d rotation representation for unconstrained head pose estimation," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 2496–2500.

[233] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman, "Laeo-net++: revisiting people Looking At Each Other in videos," in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2020.

[234] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Book2movie: Aligning video scenes with book chapters," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1827–1835, 2015.

[235] M. Tapaswi, M. Bäuml, and R. Stiefelhagen, "Storygraphs: Visualizing character interactions as a timeline," *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 827–834, 2014.