DocFigure: A Dataset for Scientific Document Figure Classification

K. V. Jobin, Ajoy Mondal and C. V. Jawahar
KCIS, CVIT, IIIT-Hyderabad, India
Email: jobin.kv@research.iiit.ac.in, ajoy.mondal@iiit.ac.in, jawahar@iiit.ac.in

***Abstract*—Document figure classification (DFC) is an important stage of a document figure understanding system. The design of a DFC system required a well defined figure categories and dataset. To the best of the author's knowledge, the existing datasets related to classification of figures in the document images are limited with respect to their size and categories [1]–[3]. In this paper, we introduce a scientific figure classification dataset, named as `DocFigure`. The dataset consists of 33K annotated figures of 28 different categories present in the document images which correspond to scientific articles published in the CVPR, ECCV, ICCV, etc. conferences in last several years. Manual annotation of such a large number (33K) of figures is time consuming and cost ineffective. In this article, we design a web based annotation tool which can efficiently assign category labels to large number of figures with the minimum efforts of human annotators. To benchmark our generated dataset on classification task, we propose three baseline classification techniques using deep feature, deep texture feature and combination of both. In our analysis, we found that the combination of both deep feature and deep texture feature is more effective for document figure classification task than the individual features. The dataset and the code are publicly available at https://researchweb.iiit.ac.in/~jobin.kv/projects/**

***Keywords*-Document figure dataset; annotation tool; deep features;**

I. Introduction

Documents contain various types of figures (e.g. Bar chart, Pie chart, Line plot, etc.) to present heterogeneous information in a compact and visual form. This visual representation of complex information helps the reader to easily understand the content of the documents. Better understanding of document also requires understanding of figures present in the documents. However, automatic understanding of these figures is still a complex task. Classification of document figures into various categories like: Graph, Block diagram, Natural image, etc. is the initial task for understanding of those figures. Classification of document figures is also a complex task due to inter-class visual similarity and intra-class visual dis-similarity among figures (refer to Figures 3 and 4). Limited work on document figure (mainly various types of charts) classification has been done in the literature [1]–[7]. The existing methods [1], [4]–[7] based on handcrafted features fail to achieve good accuracy on classification of figures in the document images due to large visual similarity among subordinate categories. To solve the limitation of handcrafted features in figure classification task, recently, some techniques [2], [3] have been developed based on deep features by convolutional neural networks (CNNs).

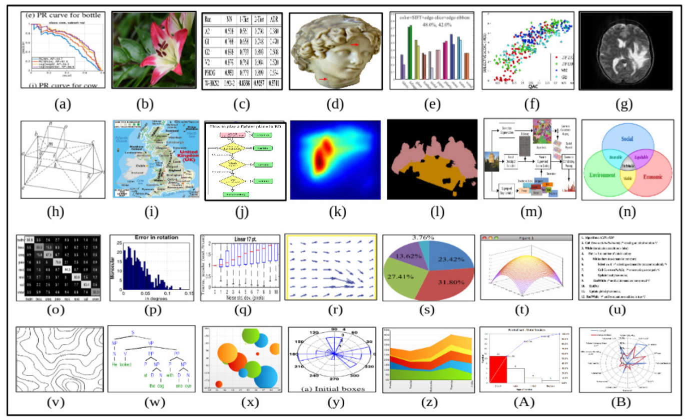


Figure 1: Visual illustration of category wise sample figure images of our `DocFigure` dataset. The 28 categories correspond to (a) Line graph, (b) Natural image, (c) Table, (d) 3D object, (e) Bar plot, (f) Scatter plot, (g) Medical image, (h) Sketch, (i) Geographic map, (j) Flow chart, (k) Heat map, (l) Mask, (m) Block diagram, (n) Venn diagram, (o) Confusion matrix, (p) Histogram, (q) Box plot, (r) Vector plot, (s) Pie chart, (t) Surface plot, (u) Algorithm, (v) Contour plot, (w) Tree diagram, (x) Bubble chart, (y) Polar plot, (z) Area chart, (A) Pareto chart and (B) Radar chart.

As per author's knowledge goes, the existing datasets (e.g. FigureSeer [3], Revision [1], Deepchart [2], Karthikeyani and Nagarajan [8], Prasad *et al.* [7], Huang and Tan [9], Zhou and Tan [6]) on the classification of document figures (mainly charts) are limited with respect to both the samples (less than or equal to 5K except FigureSeer) and category labels (less than or equal to 10). All these datasets are created by downloading figure images from web. Table I in Section II, highlights the statistics of existing figure classification datasets. In this article, we introduce a dataset containing 33K document figures annotated with 28 category labels, named as `DocFigure`. Figure 1 displays the category wise sample figure images in our dataset. Table II in Section IV highlights the comparison of our dataset with the existing datasets with respect to category labels and samples. This database is created by extracting various figures form $20K$ scientific articles (correspond to 130K document page images) published in various (e.g. CVPR, ECCV, ICCV, etc.) conferences during several years using an existing technique: PDFFigures 2.0 [10]. Manual annotation of large number of (33K) figure images is time consuming and cost ineffective. Here, we propose a web based annotation tool to efficiently assign category labels to the figure images. For this purpose, after extraction of figures, few (50) sample images of each category are annotated manually. We consider those few annotated samples and concept of incremental learning [11]; and minimal effort of human annotators to annotate the remaining figures. Finally, we generate annotated DocFigure dataset.

In particular, the contributions of this paper are as follows:

- We introduce a dataset: DocFigure containing annotated 28 categories of 33K figure images for document figure classification task.
- We design a web based annotation tool to efficiently assign category labels to the document figures using the concept of incremental learning and minimal efforts of human annotators.
- We propose three baselines based on deep feature, deep texture feature and combination of both to validate our generated dataset for document figure classification task.

II. Related Datasets

Some of the datasets exist in literature for figure classification in document images are Figureseer [3], Revision [1], Deepchart [2], Karthikeyani and Nagarajan [8], Prasad *et al.* [7], Huang and Tan [9], Zhou and Tan [6]. All these datasets are created by downloading images from the web. Except Figureseer [3] with 30.6K images, all other datasets contain limited number of figure images (less than or equal to 5K). However, only a part of Figureseer [3] dataset (i.e. 1K images) with ground truth is available for figure understanding task. Each of these datasets have limited number of document figure categories (less than or equal to 10). Table I presents a list of existing datasets dedicated to document figure classification task.

Dataset	**No. of Labels**	**Total Images**
Zhou and Tan [6]	3	1190
Huang and Tan [9]	4	200
Deepchart [2]	5	5000
Figureseer [3]	5	30600
Prasad *et al.* [7]	5	653
Karthikeyani *et al.* [8]	8	155
Revision [1]	10	2000

Table I: Statistics of the existing datasets for document figure classification task.

III. State-of-the-Arts Approaches for Document Figure Classification

Various types of figures like charts, tables and images are used to visually represent a wide range of textual information in books, scientific articles, newspapers, etc.

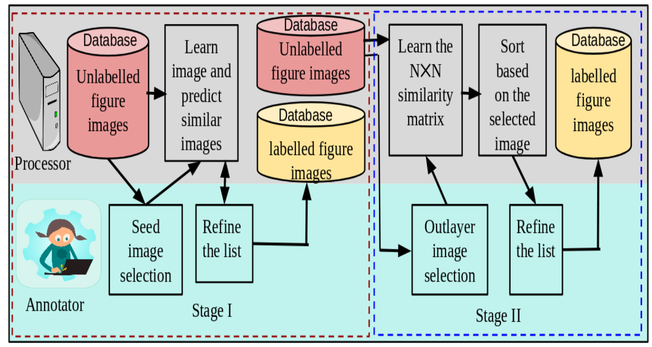


Figure 2: Work-flow for generation of DocFigure dataset. Annotation is done using two stages. In stage I, part of the dataset is annotated using incremental learning and annotator. In stage II, remaining part is annotated based on similarity score between the rest of the images and help of annotator. Finally, complete annotated dataset is generated.

Text recognition using optical character recognition (OCR) is the primary process for understanding the content of the document images. However, due to increasing use of figures in the document images, figure recognition is an important sub-task for OCR for better and complete understanding the content of the document images [12]. In early works [1], [4]–[7], different handcrafted features are used to recognize various types of charts in the document images.

Zhou *et al.* [4], [5] considered Hough transformation to recognize bar charts in the document images. Prasad et al. [7] considered SIFT and HOG feature to recognize five different types of chart images. Due to the large visual similarity among subordinate categories, the handcrafted features fail to achieve good accuracy on document figures classification task.

To solve the limitation of handcrafted features for figure classification task, recently, Kavasidis et al. [13] proposed a saliency based convolutional neural network (CNN) for localizing different types of figures in the document images. This work is limited for localizing tables, bar charts and pie charts. Tang *et al.* [2] proposed a novel framework (DeepChart) to classify charts by combining (CNNs) and deep belief networks (DBNs). The authors experimentally established that their method is far better than the hand-crafted feature based chart classification techniques. In the same direction, Siegel *et al.* [3] proposed various kind of document figure classification algorithm using deep features.

IV. DocFigure Dataset

Our generated dataset DocFigure consists of 33K figure images of 28 different category labels. Table II highlights the comparison of DocFigure dataset with existing datasets: Figureseer [3], Revision [1], Deepchart [2] with respect to category labels and samples. From Table, it is observed that the DocFigure dataset is a superset of all the existing categories of Figureseer [3], Revision [1] and Deepchart [2].

Category	**Datasets**			
	Deepchart [2]	Figureseer [3]	Revision [1]	`DocFigure`
Line graph	✓	✓	✓	✓ 9022
Natural image	-	-	-	✓ 3676
Tables	✓	-	✓	✓ 1899
3D object	-	-	-	✓ 1369
Bar plot	✓	✓	✓	✓ 1196
Scatter plot	✓	✓	✓	✓ 1138
Medical image	-	-	-	✓ 1128
Sketch	-	-	-	✓ 1105
Geographic map	-	-	-	✓ 1078
Flow chart	✓	✓	-	✓ 1074
Heat map	-	-	-	✓ 1073
Mask	-	-	-	✓ 1055
Block diagram	-	-	-	✓ 1024
Venn diagram	-	-	✓	✓ 889
Confusion matrix	-	-	-	✓ 811
Histogram	-	-	-	✓ 783
Box plot	-	-	-	✓ 605
Vector plot	-	-	-	✓ 576
Pie chart	-	-	✓	✓ 440
Surface plot	-	-	-	✓ 395
Algorithm	-	✓	-	✓ 392
Contour plot	-	-	✓	✓ 368
Tree diagram	-	-	-	✓ 360
Bubble chart	-	-	-	✓ 339
Polar plot	-	-	-	✓ 338
Area chart	-	-	✓	✓ 318
Pareto chart	-	-	✓	✓ 311
Radar chart	-	-	✓	✓ 309
Total samples	5K	30.6K	2K	33K

Table II: Comparison of our `DocFigure` dataset with existing Deepchart [2], Figureseer [3] and Revision [1] datasets with respect to category labels and samples. The last column indicate the number of images in each class.

Although, Figureseer [3] dataset contains 30.6K figure images, however only 1K figure images are publicly available for figure understanding task.

In this section, we discuss about the creation of our `DocFigure` dataset consisting of mainly three steps: collection of scientific documents, extraction of figures and its annotation. Figure 2 displays the work-flow for annotation of `DocFigure` dataset. Each of these steps is discussed in the following subsections in details.

A. Collection of Scientific Documents

We choose various conferences (e.g. CVPR, ECCV, ICCV, etc.) which publish various scientific articles in computer vision area to collect documents. Most of the published articles contains various types of figures such as natural images, medical images, charts, tables, etc. To create our dataset, we collect 20K published articles in the mentioned conferences . Finally, we have 130K single page document images corresponding to 20K articles.

B. Extraction of Figures

We consider an existing algorithm: PDFFigure 2.0 [10] to extract various types of figures from the set of collected articles. Note that PDFFigure 2.0 algorithm is designed to work on raster PDFs. This pruned the collection to 13.4K articles among total 20K articles which are in the form of raster PDF. We choose those articles to extract figures using PDFFigure 2.0. The algorithm analyzes the structure of each individual page by detecting captions, graphical elements and chunks of body text and finally localizes the figures and tables by reasoning about the empty regions within text.

The figures present in the scientific documents can be a collection of various categories of sub-figures, we call it as compound figure. PDFFigure 2.0 tool is unable to localize individual sub-figures in compound figure. To localize individual sub-figures in a compound figure, we use a similar concept described in [3] to iteratively decompose into sub-figures by identifying valid axis-aligned splits using the following criteria: (i) both resulting regions must have an aspect ratio between $1 : c_1$ and $c_1 : 1$ where $c_1 = 5$, (ii) ratio of the areas of the resulting regions must be between $1 : c_2$ and $c_2 : 1$ where $c_2 = 2.5$. The first criterion ensures that we avoid to split it into extremely narrow sub-figures (this is happened due to accidentally split-off an axis or legend label). The second criterion enforces a weak symmetry constraint between the resulting halves (as sub-figures are all often approximately of the same sizes). Finally, we obtained $143K$ figures from the $13.4K$ articles. Due to the limitation of PDFFigure 2.0 and iterative decomposition of compound figures, 32% of total extracted figures are erroneous (i.e. the figures are over segmented or they contain text regions).

C. Assignment of Category Labels to the Extracted Figures

Manual annotation i.e. assign category labels to $96K$ figures is a time consuming and cost ineffective job. Here, we propose an efficient way to assign category labels to a large number of sub-figures using the concept of incremental learning [11]. For this purpose, we develop a web based annotation tool. Initially, we manually assign category labels to randomly selected 50 figures of each category and we term this set of annotated figures as `initial training set`. Therefore, a deep feature called FC-CNN (refer to Section VI-A) descriptor corresponding to manually annotated figures are obtained. We train an one-vs-rest linear support vector machine (SVM) using extracted descriptors corresponding to `initial training set`. We also generate FC-CNN descriptors for rest of the figures and calculate the similarity scores for belonging to each these categories using the trained SVM. The annotation tool displays top 100 figures of a particular class based on their similarity scores. The human annotators un-tick only those figures which are not belonging to the selected class and submit their recommendation. In this way, we annotate more figures and create a `new training set`. We add

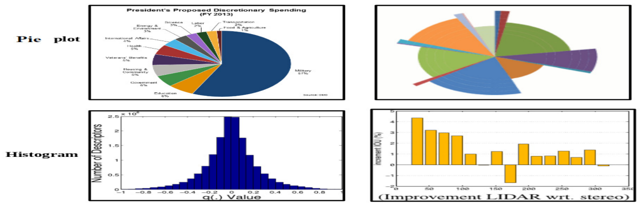


Figure 3: Intra-class dissimilarity in Pie chart and Histogram in DocFigure dataset.

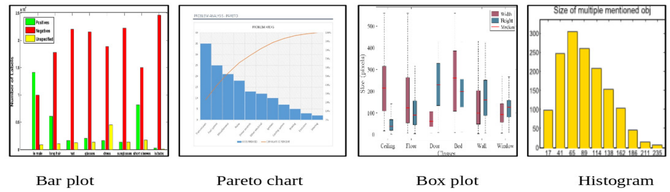


Figure 4: Inter-class similarity among Bar plot, Pareto chart, Box plot and Histogram categories of DocFigure dataset.

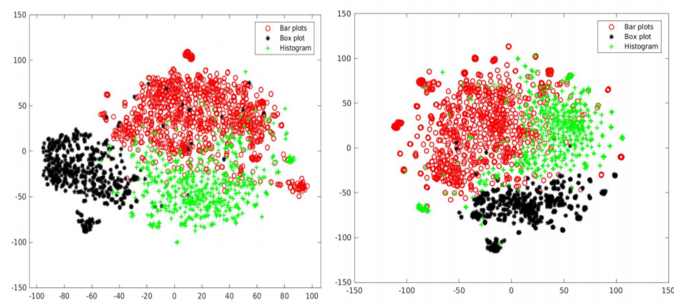
additional examples that have been selected by the annotator and accept their annotations. We again train the SVM using both initial and new training sets and repeat this process until the recommended list has less than 50% images from the selected class.

Although, the proposed annotation approach is efficient however, it is unable to annotate outlier figures in a particular category. The outlier figures are the figures which have less visual similarity (< 0.3) with the commonly occurring figures in a particular class. Those outlier figures make the dataset more complex. Only top similarity scored (> 0.3) figures of a particular class are considered as training samples while outlier figures of the same class are ignored during annotation (stage I). One possible solution to overcome this shortcoming is to include more diverse seed figures however, it is not always practically possible. Here, we propose a stage II annotation approach to include diverse (outlier) figures of a particular category in our dataset. In this approach, we select N figures from a set of figures and extract the FC-CNN descriptors corresponding to these figures [14]. We create a pair wise similarity score of $N \times N$ matrix based on Euclidean distance between FC-CNN descriptors corresponding to N figures. Our tool shows random 10 figures from the N figures ($10 << N$) and the annotator can choose any random outlier figures. The tools displays a set of figures which are similar to the chosen random figure. The annotator can refine the list by un-ticking and choosing the appropriate labels and submit the annotations. This process is repeated until all figures are assigned their corresponding labels. Table II display the statistics of our dataset.

V. Complexity Analysis of DocFigure Dataset

Intra-class dissimilarity and inter-class similarity among various categories make DocFigure dataset complex for classification task. Figure 3 shows the intra-class dissimilarity among Pie chart and Histogram in DocFigure dataset. This figure highlights that both these Pie charts are visually different from each other. Intra-class dissimilarity is also found in Histograms.

While Figure 4 shows inter-class similarity among Bar plot, Pareto chart, Box plot and Histogram in DocFigure dataset. From this figure, it is observed that Bar plot, Pareto chart, Box plot and Histogram are visually very much similar to each other. For better understanding, we visualize the extracted feature vectors (FC-CNN and FV-CNN refer to section VI-A) using tSNE [15] method. From this figure, it is observed that all these three categories are overlapped to each other for both these feature: FC-CNN and FV-CNN spaces. From this figure, it is also observed that the visual similarity among Bar plot and histogram is more than similarity among Bar plot and Box plot and similarity among Box plot and Histogram. From the figure, it is also observed that FC-CNN feature is more effective for discriminating these three classes than the FV-CNN feature.



(a) (b)

Figure 5: (a) and (b) are tSNE visualization of FC-CNN and FV-CNN features of Bar plot, Box plot and Histogram images in DocFigure dataset, respectively.

VI. DeepFigure: Proposed Baseline Approaches

We propose three baselines to validate our generated dataset DocFigure on document figure classification task. Features play an important role in classification. It is already well established that the FC-CNN descriptor which is obtained by extracting features as output of the penultimate Fully-Connected (FC) layer of a CNN, have great success in image classification task [16]–[18]. Cimpoi et al. [14] proposed FV-CNN descriptors which are obtained by Fisher Vector pooling of a CNN filter bank for semantic segmentation task. We consider both these descriptors as feature for figure classification task.

Figure 6 displays basic outline of our three baseline approaches. Each of these approaches consist of basic

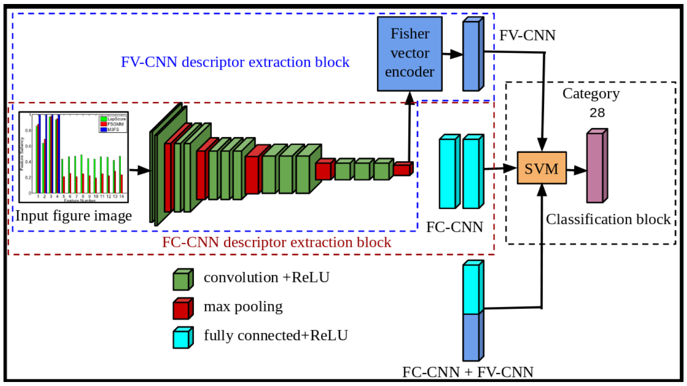


Figure 6: Basic framework for the proposed three baseline approaches. Red dotted rectangle corresponds to FC-CCN features extraction block, Blue dotted rectangle indicates FV-CNN features extraction module and Black dotted rectangle corresponds to classification module (best viewed in color).

two modules: feature extraction and then classification. We consider deep features (FC-CNN) and deep texture features (FV-CNN) extracted from feature extraction module and combination of both these features to represent each figure image. One-vs-rest SVM is chosen as a classifier to assign category label to the figure image in classification module.

A. Feature Extraction Module

It takes a figure image as an input and extracts deep features: FC-CNN as object descriptor and deep texture features: FV-CNN as texture descriptor as output. We consider pre-trained VGG-V [19] to extract these features. Here, we discuss about both these descriptors.

Object Descriptor:* FC-CNN*: The FC-CNN descriptor is obtained by extracting features as the output of the penultimate Fully Connected (FC) layer of Convolutional Neural Network (CNN) which takes figure image as input. Red dotted rectangle in Figure 6 indicates the module for extraction of FC-CNN. This extracted feature can be considered as an object descriptor because the fully connected layers allow FC-CNN to capture the overall shape of the object contained in the region.

Texture Descriptor:* FV-CNN*: The FV-CNN descriptor introduced by Cimpoi et al. [14] based on texture descriptor [20] using Fisher Vector (FV) encoding technique. We perform FV encoding on the output of last convolutional layer of the convolutional neural network. Since, the fully connected layer of the network is not involved in the FV-CNN feature generation, images with various size can be used to generate the FV-CNN features. Different from FC-CNN, FV pools local features densely within the regions by removing global spatial information. Therefore, this feature describes textures rather than objects. FV is computed on the output of last convolution layer of CNN. Blue dotted region in Figure 6 indicates FV-CNN feature extraction module.

Combination of* FC-CNN *and* FV-CNN*: We also concatenate both the features FC-CNN and FV-CNN to represent figure image.

B. Classification Module

We use one-vs-rest linear support vector machine (SVM) [21] to assign category labels to the figure images. Black dotted rectangle in Figure 6 specify the classification module. SVM is trained with each of three different representations (FC-CNN, FV-CNN and FC-CNN+FV-CNN) corresponding to training set containing figure images. Finally, trained SVM is used to assign category labels to figure images of test set.

VII. Experiments

A. Implementation details

The FC-CNN and FV-CNN features corresponding to document figure images are extracted using pre-trained VGG-V model [19]. This network architecture produces the FC-CNN feature with 4096 dimension and FV-CNN with 512 dimension. In the FC encoding, 16 Gaussian components are used on the 512 dimensional convolution features, resulting in 16K dimensional FV-CNN feature. In order to accommodate the different image scales, we calculate the FV-CNN feature after re-scaling the image by factors 2^s, $s = -3, -2.5, ..., 1.5$ (for efficient calculation, we select the scale for which the number of image pixels in the range 30 to 1024^2).

Learning details.: The descriptors FC-CNN, FV-CNN and FC-CNN+FV-CNN corresponding to figure images are classified using one-vs-rest SVM classifier. We normalize each descriptor using L_2 before classification and set miss classification weight $C = 1$. After normalization, C has minimal effect on SVM performance.

Furthermore, to improve SVM performance, we recalibrate the SVM score after training, by scaling the weight vector and bias such that the median scores of the negative and positive training samples for each class are mapped to -1 and 1, respectively.

B. Quantitative Results Analysis

Results obtained using our proposed three baseline approaches are summarised on Table III. We obtained best classification accuracy while both FC-CNN and FV-CNN descriptors are concatenated to represent the figure images. The best obtained results are indicated by bold values in Table III. We observed that FV-CNN is more effective than FC-CNN for the document figure classification task (except 3D object, Algorithm, Bar plot, Box plot, Flow chart, Heat map, Histogram, Medical image, Pie chart and Polar plot) as it represents texture rather than object shape. Use of FV-CNN descriptor improved average classification accuracy by 1.84% over FC-CNN descriptor. While combination of

Labels	**FC-CNN**	**FV-CNN**	**FV-CNN +FC-CNN**
3D objects	98.24%	94.73%	**98.53%**
Algorithm	**93.81%**	91.75%	**93.81%**
Bar plots	**93.97%**	91.97%	93.64%
Box plot	91.39%	88.07%	**92.05%**
Flow chart	92.53%	91.04%	**97.01%**
Heat map	99.25%	95.89%	**99.62%**
Histogram	**94.89%**	88.26%	**94.89%**
Medical images	97.87%	92.55%	**98.93%**
Pie chart	91.66%	89.81%	**94.44%**
Polar plot	**85.71%**	78.57%	**85.71%**
Area chart	84.61%	91.02%	**92.30%**
Block diagram	97.26%	97.65%	**98.43%**
Bubble Chart	80.95%	**91.66%**	90.47%
Confusion matrix	85.22%	89.65%	**93.10%**
Contour plot	59.34%	**74.72%**	72.52%
Geographic map	88.59%	**95.81%**	95.43%
Graph plots	98.49%	98.84%	**99.33%**
Mask	**99.23%**	**99.23%**	**99.23%**
Natural images	98.04%	98.25%	**99.23%**
Pareto charts	87.17%	96.15%	**97.43%**
Radar chart	78.94%	**86.84%**	85.52%
Scatter plot	90.14%	91.19%	**93.66%**
Sketches	95.65%	96.37%	**98.18%**
Surface plot	76.76%	**89.89%**	88.88%
Tables	97.25%	**98.73%**	97.67%
Tree Diagram	67.04%	68.18%	**70.45%**
Vector plot	79.86%	81.94%	**86.80%**
Venn Diagram	87.03%	**93.51%**	93.05%
Average	88.96%	90.80%	**92.90%**

Table III: The class wise accuracy of 28 classes in our proposed dataset `DocFigure` using shape feature (FC-CNN), texture feature (FV-CNN) and combination of both (FC-CNN+FV-CNN). The labels written in italics are more discriminative in shape feature than texture feature.

both FC-CNN and FV-CNN improves the classification accuracy over individual feature FC-CNN and FV-CNN for all categories except Bar plots, Bubble chart, Contour plot, Geographic map, Radar chart, Surface plot, Table and Venn diagram. We also noticed that use of concatenation of FC-CNN and FV-CNN improved the average classification accuracy by 3.94% and 2.10% over individual use of FV-CNN and FC-CNN, respectively. With this experiment, we concluded that the deep texture descriptor (FV-CNN) is better than the shape descriptor (FC-CNN) for few classes.

VIII. Conclusions

In this article, we introduced a dataset `DocFigure` with 33K figure images of 28 different categories present in the scientific articles. Our generated dataset is dedicated for document figure classification task. Here, we also designed an efficient web based annotation tool to annotate 33K images with minimum efforts of human annotators. We also proposed three baseline approaches based on deep feature (FC-CNN), deep texture feature (FV-CNN) and concatenation of both these features to validate our generated dataset on document figure classification task. Experimentally, we concluded that concatenation of deep feature and deep texture feature is more effective for figure classification task.

Acknowledgements

Jobin K.V. is supported by Visvesvaraya Ph.D. fellowship.

References

[1] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "Revision: Automated classification, analysis and redesign of chart images," in *ACM-UIST*, 2011.

[2] B. Tang, X. Liu, J. Lei, M. Song, D. Tao, S. Sun, and F. Dong, "Deepchart: Combining deep convolutional networks and deep belief networks in chart classification," *Signal Processing*, 2016.

[3] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, "Figureseer: parsing result−figures in research papers," in *ECCV*, 2016.

[4] Y. P. Zhou and C. L. Tan, "Hough technique for bar charts detection and recognition in document images," in *ICIP*, 2000.

[5] ——, "Bar charts recognition using hough based syntactic segmentation," in *ICTAD*, 2000.

[6] Y. Zhou and C. L. Tan, "Learning-based scientific chart recognition," in *IWGR*, 2001.

[7] V. S. N. Prasad, B. Siddiquie, J. Golbeck, and L. S. Davis, "Classifying computer generated charts," in *CBMI*, 2007.

[8] V. Karthikeyani and S. Nagarajan, "Machine learning classification algorithms to recognize chart types in portable document format (pdf) files," *IJCA*, 2012.

[9] W. Huang and C. L. Tan, "A system for understanding imaged infographics and its applications," in *ACM-SDE*, 2007.

[10] C. Clark and S. Divvala, "Pdffigures 2.0: Mining figures from research papers," in *Digital Libraries (JCDL)*, 2016.

[11] J. C. Schlimmer and D. Fisher, "A case study of incremental concept induction," in *AAAI*, 1986.

[12] Y. Liu, X. Lu, Y. Qin, Z. Tang, and J. Xu, "Review of chart recognition in document images," in *VDA*, 2013.

[13] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv preprint arXiv:1804.06236*, 2018.

[14] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *CVPR*, 2015.

[15] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, 2008.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[19] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[20] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, 1995.