Font-ProtoNet: Prototypical Network based Font Identification of Document Images in Low Data Regime

Nikita Goel, Monika Sharma, Lovekesh Vig
TCS Research, New Delhi, India
Email:{goel.nikita, monika.sharma1, lovekesh.vig}@tcs.com

Abstract

While optical character recognition has attracted considerable interest from researchers in recent times, automating font identification in printed / scanned documents is still not a well explored problem. With the increasing variety of fonts in the open community, identifying the different fonts used in a given document image can often provide important visual cues for document understanding. Font identification is a challenging task owing to smaller inter-class variations and limited availability of labeled image data for a large variety of font images. In the absence of the original true type format (ttf) files, even synthetic data generation is not possible. To this end, we propose to utilize recent few-shot learning techniques like prototypical networks for font identification in scanned / printed document images using character images from different fonts as input for scarce data scenarios and call the proposed method Font-ProtoNet*. This approach uses an initial set of classes to learn an embedding and centroid representations (as class prototypes), that are used to classify novel samples based on euclidean distance. We demonstrate that Font-ProtoNet gives encouraging results by training prototypical networks in few-shot learning settings on a synthetic dataset of* 200 *font classes and using the trained network to identify fonts on a synthetic dataset of* 100 *novel font classes. We have also tested our approach on the real-world Adobe Visual Font Recognition (AdobeVFR) dataset and obtained* 59.86% *and* 71.01% *word-level accuracy of font identification using* 1*-shot and* 5*-shot i.e.,*1 *and* 5 *character images per font class, respectively.*

1. Introduction

In recent decades, automation has taken over every industry, be it inspection industry, banks, insurance and medical field, etc. One particular area where automation is of utmost importance and desirable is digitization of scanned documents like bank application forms, receipts and insurance claim forms [22, 20, 16, 14] to facilitate easy and quick retrieval and management of information for faster processing. There are many associated tasks in the automatic digitization of scanned documents like printed, handwritten text recognition and forming key-value pairs to populate fields of interest like name, address and age etc. Although there have been numerous works in optical character recognition for both printed and handwritten text [11, 25, 12, 5], font identification from document images is still an under-explored field [4]. Font identification in scanned documents can prove to be beneficial in document understanding as there is significant visual structure in the different fonts. Examples include identifying table or section headers, key-value pairs with differing fonts for key and value, italicized or emboldened words for emphasis, as shown in Figure 1. Moreover, identifying fonts automatically is also required in the validation of bank debit / credit card designs. Another area of application is for users that often take a liking to a font in an online document and are unable to identify that font.

Identifying fonts from scanned document images is a challenging task. Different font classes usually have very minute visual differences, difficult even for a human eye to decipher manually. Therefore, an automated process for differentiating fonts from scanned documents would save considerable effort. The main challenge associated with font identification is availability of sufficient labeled data to train the network. Also, the existence of a large variety of fonts makes it difficult to train the network against all these fonts beforehand. Hence, we require a network which is able to train itself quickly with the limited data available and can distinguish even between unknown / novel fonts not seen during training.

In this paper, we propose a network to identify fonts using very few labeled examples with high accuracy. We utilize character images for different font classes to train state-of-the-art distance-metric learning based prototypical networks [17] for few-shot classification settings such as 5-way 1-shot, 5-shot and 10-shot classification and 10-way 1-shot, 5-shot and 10-shot classification. We named the proposed architecture *Font-ProtoNet* which learns embeddings and cluster centroids using a support set during meta-

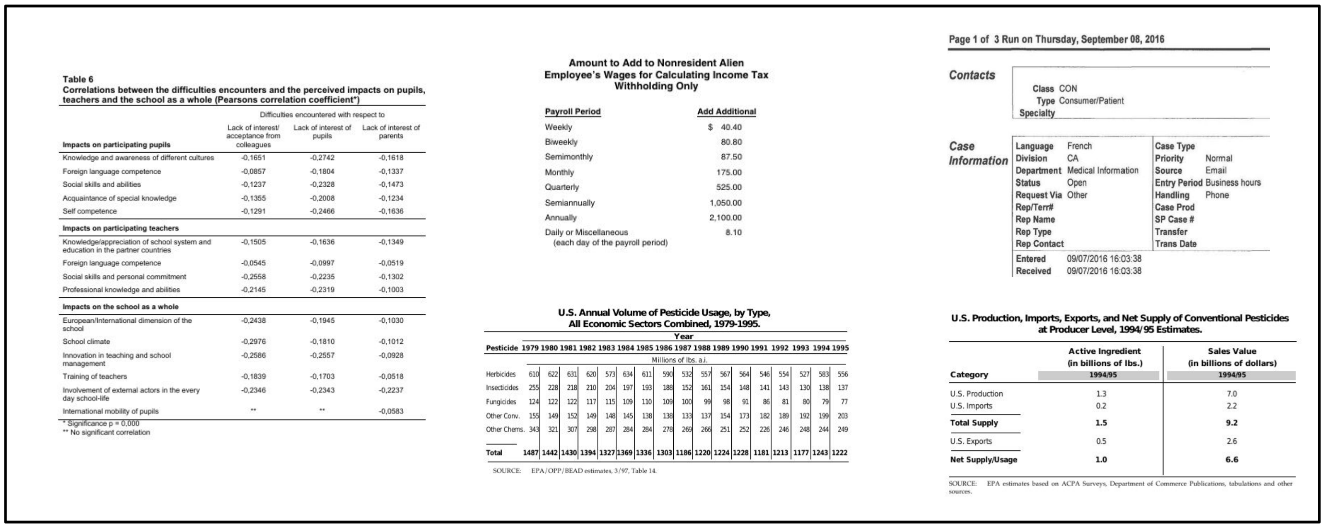

Table 6
Correlations between the difficulties encounters and the perceived impacts on pupils, teachers and the school as a whole (Pearsons correlation coefficient*)

	Difficulties encountered with respect to		
Impacts on participating pupils	Lack of interest/ acceptance from colleagues	Lack of interest of pupils	Lack of interest of parents
Knowledge and awareness of different cultures	-0,1651	-0,2742	-0,1618
Foreign language competence	-0,0857	-0,1804	-0,1337
Social skills and abilities	-0,1237	-0,2328	-0,1473
Acquaintance of special knowledge	-0,1355	-0,2008	-0,1234
Self competence	-0,1291	-0,2466	-0,1636
Impacts on participating teachers			
Knowledge/appreciation of school system and education in the partner countries	-0,1505	-0,1636	-0,1349
Foreign language competence	-0,0545	-0,0997	-0,0519
Social skills and personal commitment	-0,2558	-0,2235	-0,1302
Professional knowledge and abilities	-0,2145	-0,2319	-0,1003
Impacts on the school as a whole			
European/International dimension of the school	-0,2438	-0,1945	-0,1030
School climate	-0,2976	-0,1810	-0,1012
Innovation in teaching and school management	-0,2586	-0,2557	-0,0928
Training of teachers	-0,1839	-0,1703	-0,0518
Involvement of external actors in the every day school-life	-0,2346	-0,2343	-0,2237
International mobility of pupils	**	**	-0,0583

* Significance p = 0,000
** No significant correlation

Amount to Add to Nonresident Alien Employee's Wages for Calculating Income Tax Withholding Only

Payroll Period	Add Additional
Weekly	$ 40.40
Biweekly	80.80
Semimonthly	87.50
Monthly	175.00
Quarterly	525.00
Semiannually	1,050.00
Annually	2,100.00
Daily or Miscellaneous (each day of the payroll period)	8.10

U.S. Annual Volume of Pesticide Usage, by Type, All Economic Sectors Combined, 1979-1995.

	Year																
Pesticide	**1979**	**1980**	**1981**	**1982**	**1983**	**1984**	**1985**	**1986**	**1987**	**1988**	**1989**	**1990**	**1991**	**1992**	**1993**	**1994**	**1995**
	Millions of lbs. a.i.																
Herbicides	610	622	631	620	573	634	611	590	532	557	567	564	546	554	527	583	556
Insecticides	255	228	218	210	204	197	193	188	152	161	154	148	141	143	130	138	137
Fungicides	124	122	122	117	115	109	110	109	100	99	98	91	86	81	80	79	77
Other Conv.	155	149	152	149	148	145	138	138	133	137	154	173	182	189	192	199	203
Other Chems.	343	321	307	298	287	284	284	278	269	266	251	252	226	246	248	244	249
Total	**1487**	**1442**	**1430**	**1394**	**1327**	**1369**	**1336**	**1303**	**1186**	**1220**	**1224**	**1228**	**1181**	**1213**	**1177**	**1243**	**1222**

SOURCE: EPA/OPP/BEAD estimates, 3/97, Table 14.

Page 1 of 3 Run on Thursday, September 08, 2016

Contacts

Class	CON
Type	Consumer/Patient
Specialty	

Case Information

Language	French	**Case Type**	
Division	CA	**Priority**	Normal
Department	Medical Information	**Source**	Email
Status	Open	**Entry Period**	Business hours
Request Via	Other	**Handling**	Phone
Rep/Terr#		**Case Prod**	
Rep Name		**SP Case #**	
Rep Type		**Transfer**	
Rep Contact		**Trans Date**	
Entered	09/07/2016 16:03:38		
Received	09/07/2016 16:03:38		

U.S. Production, Imports, Exports, and Net Supply of Conventional Pesticides at Producer Level, 1994/95 Estimates.

	Active Ingredient (in billions of lbs.)	Sales Value (in billions of dollars)
Category	**1994/95**	**1994/95**
U.S. Production	1.3	7.0
U.S. Imports	0.2	2.2
Total Supply	**1.5**	**9.2**
U.S. Exports	0.5	2.6
Net Supply/Usage	**1.0**	**6.6**

SOURCE: EPA estimates based on ACPA Surveys, Department of Commerce Publications, tabulations and other sources.

Figure 1. Figure showing sample examples where font-identification can help in document understanding by providing visual cues present in different fonts.

training stage and subsequently, obtains cluster representations by passing examples in the support set at the meta-test stage through the learned prototypical network and finally maps the examples in the query set using a distance measure such as Euclidean distance. As there is no publicly available train / test dataset, we created our own synthetic dataset of character images of 200 different fonts to train the prototypical networks for meta-training. During meta-test stage, we demonstrate the effectiveness of the proposed network on this self-created synthetic dataset of characters and words of novel different 100 font classes. The dataset is generated using the true-type format of fonts with each font consisting of 62 characters (a, b,...,z, A, B,..., Z, 0,...,9). We name the synthetic dataset *FewShot-FontID Dataset* and plan to make it available online for evaluation by community. Further, we show the efficacy of our proposed network on a real-world test set of word-images of different font classes named AdobeVFR [24] for 1-shot and 5-shot learning using only 1 and 5 character images per class, respectively. To summarize, we make following contributions in the paper:

- We formulate the font identification problem as a few-shot classification problem using prototypical network which is able to identify any new / unseen fonts accurately with very limited data. The proposed approach is named *Font-ProtoNet*.
- Due to the non-availability of public datasets for font identification, we created a synthetic dataset of character images belonging to 200 different font classes for meta-training and 100 new font classes for meta-testing. We also generated word-images of 100 new font classes for meta-test. We plan to publish this dataset called *FewShot-FontID Dataset* online for the benefit of research community.
- We performed 5-way 1-shot, 5-shot and 10-shot, and 10-way 1-shot, 5-shot and 10-shot experiments on *FewShot-FontID dataset* using the proposed network and present the results in Section 4.3.
- In addition, we evaluate Font-ProtoNet on test-set of AdobeVFR [24] for font classes containing more than one word image and obtain word-level accuracy of 59.86% and 71.01% using only 1 and 5 character images per font class, respectively.
- We also demonstrate the robustness of Font-ProtoNet by providing t-SNE [21] visualization of learned embeddings.
- We compare the performance of Font-ProtoNet against a transfer learning based pre-training baseline and present the results in Table 2 which demonstrate that Font-ProtoNet gives better identification accuracy.

We have organized the paper into following sections. Section 2 will describe the work previously done in the field of font identification and how our approach is better than prior literature. Next, we provide description of our proposed Font-ProtoNet network for font identification in Section 3 and give details of prototypical network and the architecture of convolutional network used. Subsequently, Section 4 provide details of the experiments conducted, their results and discussions on them. In the end, Section 5 will conclude the work with future avenues in this field.

2. Related Work

Font recognition from images has been studied in past and there exists some online portals which allow users to upload any font image and recognize fonts using similarity with a collection of available fonts. All of these online interfaces such as FontMatcherator [7], Identifont [10], WhatTheFont [3], Font Identifier [18] search fonts by exhaustively traversing the list of available fonts in the corpus and cannot identify a novel font that it has not seen before.

Most of the earlier works using data-driven machine learning techniques for font identification assume that fonts are known prior. Authors in [1] proposed global texture analysis to perform font recognition for very few font types in document images that employs sliding window approach to obtain features of the document using fourth and third order moments. Another work [9] used Gabor filter based classifier to classify the scripts, font-faces and font-styles (bold, italic, normal, etc.) in spanish documents where the classes are known beforehand. Similar work of font identification using Gabor filters to extract document features and subsequently, classifying the obtained feature vectors via support vector machines was proposed in [15]. To eliminate the need of prior information on font types, Hung Ming [19] proposed to automatically extract representative stroke templates from a text image, which contained characters of the same typeface and eventually, it used Bayes decision rule to determine which font entrant in the database provides the best matching to the unknown font. Tui Bui *et al.* [2] proposed an approach which made use of hierarchical bag of visual words representation and subsequent classification via logistic regression yielding per-character probabilities which were combined across the string to decide the posterior for each font.

Recently, deep learning based approach has also been applied for font identification purposes. One such approach was proposed in [24] which utilized convolutional neural networks and domain adaptation techniques to identify a variety of fonts. However, their proposed approach relied on huge amount of labeled data to train the network. [24] have also made available a real world test dataset namely, AdobeVFR for font text images. We have used this dataset also to evaluate our proposed network and obtained satisfactory results. Another deep learning based approach which utilizes a siamese like architecture to find if two fonts are similar or not, has been proposed by Daichi *et al.* [8]. To our knowledge, we have not found any system for font identification which can identify new fonts by quick adaptation of neural networks with very few annotated samples per font. This motivates us to explore recently discovered few-shot based meta-learning techniques [6, 13, 23, 17]. We specifically use distance metric based meta-learning techniques known as Prototypical network [17] which allows us to learn class prototypes using a support set containing very few examples per font and then assigns labels to the test-set samples using euclidean distance measure. This allows us to identify any unseen / novel fonts quickly and with much less annotated data.

3. Proposed Method: Font-ProtoNet

The objective of the paper is to propose a method for identifying fonts from scanned document images in scenarios where annotated data is limited. The low availability of labeled data poses the problem of not having good generalization of neural networks across novel font classes as these networks tend to overfit when trained on very few data samples. Moreover, training a deep neural network repeatedly for all the classes whenever a new font class is to be identified, is a very time consuming process. Therefore, font identification network should also have the capability of quickly adapting to novel classes with much less annotated data. This motivates us to explore state-of-the-art few-shot learning based approaches like prototypical networks [17] for font identification from text images. Therefore, we formulate the problem of font identification as few-shot classification problem by training a prototypical network on single character images of most commonly occurring fonts to learn font-embeddings in low data scenarios. Subsequent to this, the trained network is adapted to new font classes quickly whenever a new font image comes, by computing embeddings based on support set and assigning labels to the query image using Euclidean distance measure.

The architecture of proposed Font-ProtoNet, as shown in Figure 2, consists of two stages namely, Meta-training stage and Meta-testing stage. The meta-training stage involves training the prototypical networks to learn representations of known / most common fonts for which annotated data is available. This stage obtains cluster representations using embeddings of support set examples. The input to the network is images of size 64×64 having only single character from (a, b,...,z, A, B,..., Z, 0,...,9) of a particular font. At meta-test stage, the support set images of the given font class are passed through the trained network to obtain embeddings and subsequently, cluster representation is calculated. Finally, the test image is mapped to the label of the font class having the smallest euclidean distance. The details of the prototypical networks and the network architecture of the convolutional network used are given in following subsections.

3.1. Prototypical Networks

Prototypical network [17] is a meta-learning based few shot approach that works on the assumption that there exists an embedding in latent space where samples from each class cluster around a single prototypical representation which is obtained using the mean of the embedding of the individual samples. This assumption forms the basis of classification

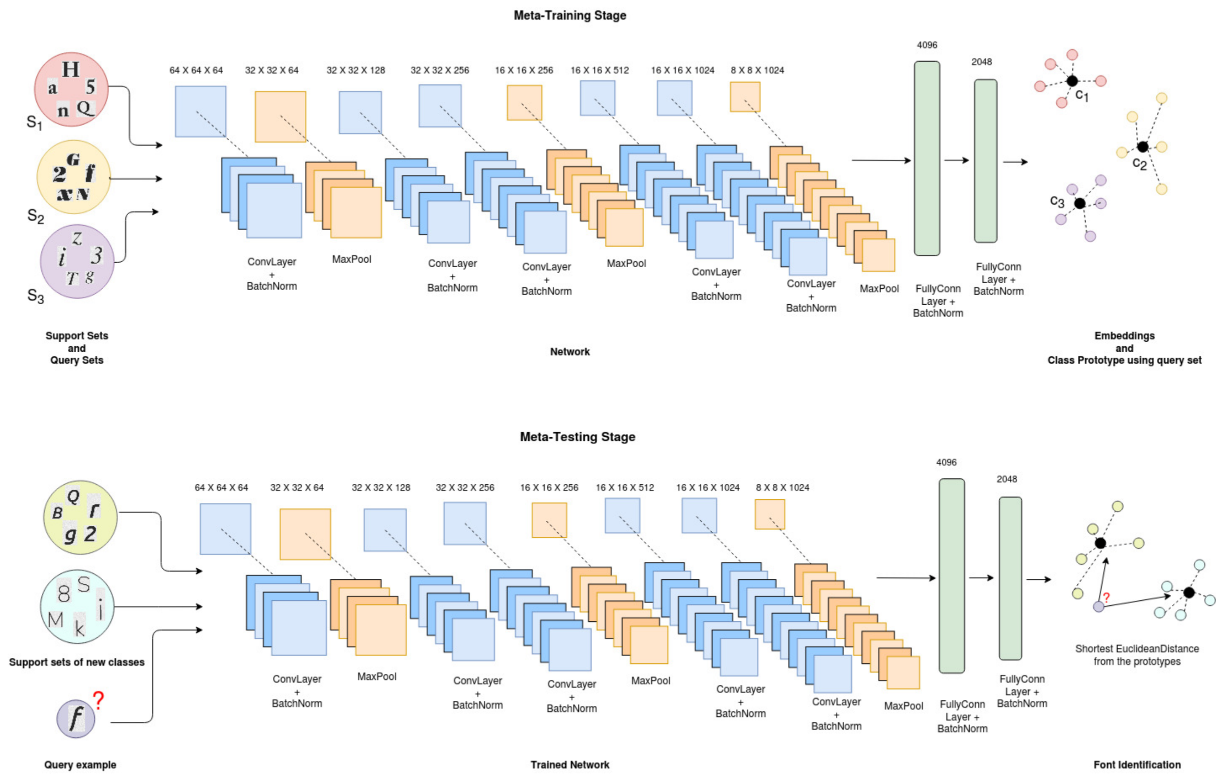

Figure 2. Figure showing proposed architecture of Font-ProtoNet for font identification in data-scarce scenarios. The proposed pipeline consists of two stages: *Meta-Training* stage in which the network is trained to learn font embeddings and compute cluster centroids and *Meta-Testing* stage in which given a query image having new and unseen font, the support set is used to find cluster representation and then font class is assigned using Euclidean distance measure.

which is simply performed by finding the euclidean distance of the given sample to all the class prototypes and assigning the label of the closest class prototype to the given sample.

Mathematically, for a set $A = \{(x_1,y_1),(x_2,y_2)...(x_n,y_n)\}$, where x_i denotes i^{th} input with y_i class label and total n number of examples. Let A_k denotes the set of all examples with label k. A subset is randomly chosen from set of all training classes from which two sets, support (S_k) and query (Q_k) are formed by sampling from each class of the subset. This process is known as a training *episode*. Next, for each $x_i \in \mathcal{R}^D$, prototypical network computes an embedding using a function, $f_\theta : \mathcal{R}^D \rightarrow \mathcal{R}^m$ that maps D-dimensional feature space to m-dimensional vector with learnable parameters θ, such that representative $r_k \in \mathcal{R}^m$ of class k. The prototype / representative r_k of a class with label k is calculated as the average of embedded support set examples belonging to class k using equation as follows:

$$r_k = \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} f_\theta(x_i) \tag{1}$$

After class prototypes are computed, prototypical network computes the distribution over predicted class labels for query instance x with a distance function d by using softmax over distance to the representative of a class in the embedding space as follows:

$$p_\theta(y = k|x) = \frac{e^{-d(f_\Theta(x), r_k)}}{\sum_{k'} e^{-d(f_\Theta(x), r_{k'})}} \tag{2}$$

Parameters θ are updated by minimizing the negative log of probability so as to improve the probability for the query class through stochastic gradient descent. Loss for training episodes, Z_θ is calculated by,

$$Z_\theta = -log\, p_\theta(y = k|x) \tag{3}$$

3.2. Network Architecture

We have used a 5-layer convolution neural network consisting of $32, 64, 128, 512$ and 1024 filters of size 3×3 each, as shown in Figure 2. Each layer is followed by batch normalization and relu activation. We have applied a 2×2 max-pooling layer after $1st$, $3rd$ and $5th$ layer. The last convolution layer is followed by two fully-connected layers of size 4096 and 2048, respectively. Next, the prototype of a class is computed by calculating mean of the embedded

Training Dataset	Testing Dataset	Number of font classes		Number of examples	
		Train	Test	Train	Test
FewShot-FontID	FewShot-FontID	200	100	12,400	6200
FewShot-FontID	AdobeVFR Real Test	300	100	18,600	500

Table 1. Details of the datasets used for performance evaluation of Font-ProtoNet.

support set examples of each class. During inference, the given test sample is assigned the label of the closest class prototype having the minimum euclidean distance.

4. Experimental Results and Discussions

4.1. Dataset

Font identification is a very challenging task due to the presence of numerous possible fonts having very subtle and character-dependent differences among them such as letters slopes, weights and endings etc. Moreover, machine learning / deep learning networks are data-hungry approaches requiring a large collection of data for their training. Collection and annotation of real world font images is quite an expensive, error-prone and time consuming task. Therefore, there does not exist datasets for font identification. Very recently, DeepFont [24] collected and synthesized a huge dataset of fonts called *AdobeVFR*, however they have not released the entire dataset publicly, except for test-dataset of fonts. So, we use only the test set of AdobeVFR for evaluating performance of our proposed network. For training purposes, we resort to utilizing our own synthetically generated dataset *FewShot-FontID*. The details of both the datasets used in this paper are given as follows:

- **FewShot-FontID Dataset**: We utilize the Text Recognition Data Generator (trdg)[1] to generate character images of 300 distinct commonly used fonts. Truetype formats (ttf) of these fonts are used to obtain 62 character images of size 64×64 per font class with each image having only one character from the list of possible characters, i.e., (a, b,...,z, A, B,..., Z, 0,...,9). Basically, the FewShot-FontID dataset contains only one image per character in one font type. The sample images of different fonts in FewShot-FontID dataset are shown in Figure 3(a). Out of 300 total generated fonts, we use 200 for training and remaining 100 fonts for testing. So, there are a total of 12400 and 6200 images in the train and test sets, respectively, as tabulated in Table 1. We also created word-images having 100 different fonts for the test-set to evaluate the performance of our proposed network.

- **AdobeVFR Dataset**: We used only the test set of AdobeVFR [24] dataset which consists of real-world images of words having different fonts along with annotations. In total, it has 612 types of fonts available, with each font class having one or more word images. We discard font classes for which only one image is present because for evaluating the performance of our proposed Font-ProtoNet network, we require atleast one word image of a given font in the train set at the meta-test stage to perform 1-shot and 5-shot character classification for the remaining images of the same font in the test set for the meta-test stage. For the meta-train stage, we use a synthetic dataset of 300 fonts available in the FewShot-FontID dataset to train the prototypical network.

[1]Text Recognition Data Generator: https://pypi.org/project/trdg/

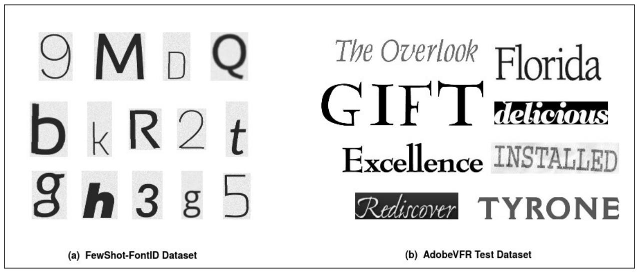

Figure 3. Figure showing some sample images from (a) synthetic FewShot-FontID dataset and (b) AdobeVFR [24] real-world test dataset.

4.2. Training Details

Prototypical Network: We used 200 and 300 font classes from the FewShot-FontID dataset to generate training episodes for n-way classification while performing evaluation experiments for FewShot-FontID and AdobeVFR test-sets, respectively. The prototypical networks were trained using train shots as $1, 5, 10$ for n-way classification tasks, i.e, we sampled $1, 5, 10$ images per class per episode from the n classes for training Font-ProtoNet on these images. We performed 5-way and 10-way classification and trained the network using the Adam optimizer with a learning rate of 0.0001 through stochastic gradient descent. The input to the network is the character image of size 64×64. During the meta-test stage, we generated the episodes similar to the training such as 1-shot, 5-shot and 10-shot learning for 5-way and 10-way classification. In the test set, we have a meta-test set having 100 fonts for both, synthetic FewShot-FontID and AdobeVFR datasets. In case of the

Dataset	n-way	k-shot	Pre-trained Model	Font-ProtoNet	
			Character-Level	Character-Level	Word-Level
FewShot-FontID Dataset	5-way	1-shot	40.01	69.18	77.1
		5-shot	47.96	75.26	87.48
		10-shot	55.23	**77.51**	**88.01**
	10-way	1-shot	29.12	58.65	69.77
		5-shot	34.01	62.12	71.5
		10-shot	40.57	**71.5**	**80.04**
AdobeVFR Dataset	5-way	1-shot	37.7	50.07	59.86
		5-shot	48.1	**62.01**	**71.01**

Table 2. Performance comparison (average accuracy in %) of Font-ProtoNet against a pre-trained network baseline for Font Identification on the FewShot-FontID and AdobeVFR datasets.

FewShot-FontID dataset during the testing stage, we sampled n classes randomly from the meta-test set to form an n-way classification task and 1, 5 and 10 samples are selected from each font class for creating a support set corresponding to 1-shot, 5-shot and 10-shot learning and the remaining samples in the meta-test set are used as a query set. For the test-set of FewShot-FontID, we also created a set of images having single words belonging to the 100 font classes of the meta-test set.

For evaluation on AdobeVFR dataset, as mentioned earlier in Section 4.1 we chose only those font classes where we have more than one word-image. This was done so as to take characters of first word for creating support-set of meta-test stage for a given font class and the remaining word images of that particular font class were chosen as query-set. We report average accuracy for font identification at both character-level and word-level.

Pre-trained Network as baseline: For creating a baseline for font identification, we used transfer learning where we pre-trained the convolutional network (similar to Font-ProtoNet described in Section 3.2) on the entire dataset of all the training font classes. In a subsequent step, this pre-trained network is fine-tuned and evaluated on the testing dataset. Fine-tuning of the pre-trained network is done by creating classification tasks. Each task involves 5-way and 10-way classification by sampling of k $(1, 5, 10)$ images per class for 1-shot, 5-shot and 10-shot learning from the testing classes. Finally, the fine-tuned network is evaluated on the remaining test samples. The results of the fine-tuned network are compared and reported for performance analysis in Table 2.

4.3. Results

We report comparative results of transfer learning based pre-trained network versus meta-learning based prototypical networks in this section. We compute character-level and word-level font identification accuracy. Word-level accuracy is computed by first finding character level font predictions of a given word image using the Font-ProtoNet for all the characters present in the image and then taking the majority vote of the font predictions to assign the font class to the word. As it is evident from Table 2 that Font-ProtoNet surpasses the pre-trained network's identification accuracy. For example, in case of FewShot-FontID dataset, we obtain an average accuracy of 77.51% as compared to 55.23% with a pre-trained baseline for the 5-way 10-shot font identification experiment. We observe similar performance in the case of 10-way identification experiments. We also compare these two approaches on AdobeVFR dataset where we observe similar boost in identification accuracy on replacing pre-training baseline by Font-ProtoNet. As shown in Table 2 that Font-ProtoNet gives character level accuracy of 62.01% as compared to 48.1% with pre-trained network for 5-way 5-shot experiment on AdobeVFR dataset. The poor performance of the pre-trained baseline is due to the fact that the fine-tuning uses 1, 5 and 10 sample images per class for performing 1-shot, 5-shot and 10-shot classification which makes the network overfit on train classes and this network does not generalize well on test set containing new fonts. On the other hand, prototypical network performs remarkably well on novel / unseen fonts with very few labeled samples.

In Table 2, we first provide results of the experiments conducted for 5-way classification using a pre-trained baseline and proposed Font-ProtoNet using 1-shot, 5-shot and 10-shot. We observe that accuracy increases on increasing the number of shots, i.e., 10-shot gives best results compared to 1-shot and 5-shot. For example, at character-level, Font-ProtoNet gives an average accuracy of font-identification of 77.51% with 10-shot as compared to 69.18% and 75.26% for 1-shot and 5-shot, respectively on FewShot-FontID dataset. Similar trend is observed for 10-way classification on this dataset. Next, we present results of 5-way 1-shot and 5-shot font identification on AdobeVFR test-set and obtain an average accuracy of 50.07% and 62.01% at character-level for 1-shot and 5-shot respectively. The network yields word level accuracy of

59.86% and 71.01% for 1-shot and 5-shot font identification, respectively on AdobeVFR dataset. We could not perform 10-shot experiment on AdobeVFR test-set due to the non-availability of enough examples for creating support-set.

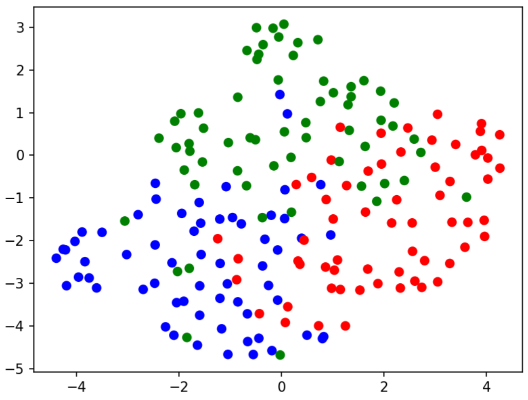

Figure 4. Figure showing t-SNE [21] plot of embeddings of text-images having 3 unknown / novel-font characters. Embeddings are computed from the trained Font-ProtoNet network. Different colors represent different fonts and plot illustrates that prototypical networks are able to learn robust visual embeddings of character images of different fonts.

We also performed an experiment to show the robustness of the embeddings learned by Font-ProtoNet. For this, we chose 3 new fonts which Font-ProtoNet had not seen during its training. We obtained the embeddings of the character images of these 3 font classes by passing the images through trained Font-ProtoNet. Subsequently, we plotted these embeddings using t-SNE [21], as shown in Figure 4, with different colors representing different fonts. It is clearly visible in Figure 4 that different font embeddings form different and almost visibly separable clusters containing all examples belonging to a particular font class. This illustrates that Font-ProtoNet is capable of learning embeddings that can adapt to new fonts quickly and easily.

5. Conclusion and Future Work

We proposed to use few-shot based Prototypical networks for identifying fonts from scanned document images when sufficient amount of annotated data is not available. We named the network Font-ProtoNet and demonstrated that the network is able to learn embeddings which can quickly adapt to newer font classes easily with very few data samples. We also created a synthetic dataset of text-images having varying fonts called FewShot-FontID for font identification purposes and benchmarked the dataset using our proposed network. We would also release this dataset for evaluation by the research community. In future, we would like to extend the use of prototypical networks to identify different language texts present in a scanned document having multi-lingual texts which would be very helpful in improving multi-lingual OCR.

References

[1] Carlos Avilés-Cruz, Risto Rangel-Kuoppa, Mario Reyes-Ayala, A. Andrade-Gonzalez, and Rafael Escarela-Perez. High-order statistical texture analysis–font recognition applied. *Pattern Recogn. Lett.*, 26(2):135–145, Jan. 2005.

[2] T. Bui and J. Collomosse. Font finder: Visual recognition of typeface in printed documents. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3926–3930, 2015.

[3] MyFonts by Monotype. Whatthefont, Last accessed March, 2020.

[4] Guang Chen, Jianchao Yang, Hailin Jin, Jonathan Brandt, Eli Shechtman, Aseem Agarwala, and Tony X. Han. Large-scale visual font recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3598–3605, 2014.

[5] Arindam Chowdhury and Lovekesh Vig. An efficient end-to-end neural model for handwritten text recognition. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 202, 2018.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[7] FontSpring. Font matcherator, Last accessed March, 2020.

[8] Daichi Haraguchi, Shota Harada, Brian Kenji Iwana, Yuto Shinahara, and Seiichi Uchida. Character-independent font identification. *ArXiv*, abs/2001.08893, 2020.

[9] Huanfeng Ma and D. Doermann. Gabor filter based multi-class classifier for scanned document images. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 968–972, 2003.

[10] Identifont. Identifont, November.

[11] Anthony Kay. Tesseract: An open-source optical character recognition engine. *Linux J.*, 2007(159):2, July 2007.

[12] P. S. Mukherjee, B. Chakraborty, U. Bhattacharya, and S. K. Parui. A hybrid model for end to end online handwriting recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 658–663, Nov 2017.

[13] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[14] Rohit Rahul, Arindam Chowdhury, Animesh, Samarth Mittal, and Lovekesh Vig. Reading industrial inspection sheets by inferring visual relations. In *ACCV Workshops*, 2018.

[15] R. Ramanathan, K. P. Soman, L. Thaneshwaran, V. Viknesh, T. Arunkumar, and P. Yuvaraj. A novel technique for english font recognition using support vector machines. In *Proceedings of the 2009 International Conference on Advances*

in Recent Technologies in Communication and Computing, ARTCOM '09, page 766–769, USA, 2009. IEEE Computer Society.

[16] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1162–1167, Nov 2017.

[17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[18] Font Squirrel. Font identifier, Last accessed March, 2020.

[19] Hung-Ming Sun. Multi-linguistic optical font recognition using stroke templates. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ICPR '06, page 889–892, USA, 2006. IEEE Computer Society.

[20] Vishal Sunder, Ashwin Srinivasan, Lovekesh Vig, Gautam M. Shroff, and Rohit Rahul. One-shot information extraction from document images using neuro-deductive program synthesis. *CoRR*, abs/1906.02427, 2019.

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[22] D Vishwanath, Rohit Rahul, Gunjan Sehgal, Swati, Arindam Chowdhury, Monika Sharma, Lovekesh Vig, Gautam Shroff, and Ashwin Srinivasan. Deep reader: Information extraction from document images via relation extraction and natural language. *ArXiv*, abs/1812.04377, 2018.

[23] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Neural Information Processing Systems*, 2019.

[24] Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S. Huang. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 451–459, New York, NY, USA, 2015. Association for Computing Machinery.

[25] Christoph Wick, Christian Reul, and Frank Puppe. Calamari - A high-performance tensorflow-based deep learning package for optical character recognition. *CoRR*, abs/1807.02004, 2018.