FONT RECOGNITION SYSTEM FOR ENGLISH

Geetha S
Professor Grade 2 SCOPE
Vellore Institute of Technology
Chennai, India
geeta.s@vit.ac.in

S.Kowshal
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
kowshal.s2019@vitstudent.ac.in

Utkarsh
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
utkarsh.2019a@vitstudent.ac.in

***Abstract*— Various techniques have been developed to extract text successfully from images, the most famous being OCR. This project on font detection is intended to benefit various creaters that need to identify fonts suggestions from images or posters. Automatic font identification and similar font suggestion from an image or photo has been on the wish list of many designers. We want to take a step further to exploit the usage of OCR to predict the font used in images using deep learning techniques with python.**

Keywords –Font recognition, AdobeVFR, Deep learning, Image processing

1. ABSTRACT

As text style is one of the focuses of plan thoughts, customized literary style ID and practically identical printed style thoughts from pictures or pictures is an element to be checked in the agenda of various creators. We are zeroing in on an awesome yet endeavor, for example, the DeepFont system utilizing the Visual Font Recognition (VFR) and mean to make it more successful. The instruments utilized here are of top quality which is finished by working and creating results on a huge dataset, called AdobeVFR that contains information from both stamped and designed data just as certifiable information. Further, we exhibit the utilization of Convolutional Neural Network(CNN) weakening methodology over the subjects of feasible getting ready and testing information, utilizing a region variety technique dependent on a Stacked Convolutional Auto-Encoder(SCAE) using advantage of a colossal corpus of unlabeled authentic word pictures associated with a particular point in the head. Moreover, we center around a brilliant learning-based model tension methodology, to lessen the size of the DeepFont model without hurting its show. The precision set apart by this learning model raises over an 80% imprint or photo has been on the rundown of things to get off various makers. We focus on the Visual Font Recognition (VFR) issue and advance the top tier astoundingly by encouraging the DeepFont structure. In particular, we foster the most promptly available enormous degree VFR dataset, named AdobeVFR, involving both stamped designed data and somewhat named true data. Then, at that point, to fight the region jumbles between available planning and testing data, we present a Convolutional Neural Network (CNN) crumbling approach, using a region variety strategy reliant upon a Stacked Convolutional Auto-Encoder (SCAE) that exploits a gigantic corpus of unlabeled genuine message pictures got together with produced data preprocessed considering a specific objective. Likewise, we focus on an astute learning-based model strain approach, to diminish the DeepFont model size without relinquishing its show. The DeepFont system achieves an accuracy of higher than 80% (top-5) on our assembled dataset and moreover makes a respectable text-based style comparability measure for text-based style decision and thought. We furthermore achieve on different occasions the tension of the model with close to no clear loss of affirmation precision.

2. INTRODUCTION

We are centered around programming the profound textual style acknowledgment proposed by Adobe on a particular dataset named Adobe VFR and the strategy utilized is a profound learning model to prepare a model dependent on the dataset and test it. We are making a custom dataset dependent on Adobe's VFR dataset.

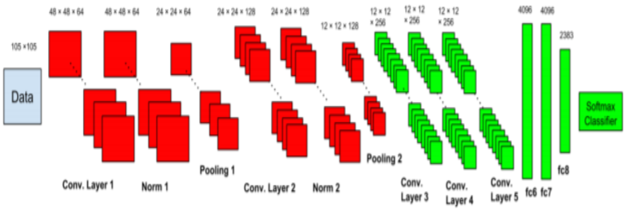
KEY POINTS OF DEEPFONT:

I. Dataset utilized for preparing this model is the immense AdobeVFR dataset comprising of more than 2383 Font Styles. It's prepared on Adobe VFR Dataset which contains 2383 Font Categories
II. It's Domain adapted CNN. Its Domain adapted CNN
III. Use Model Compression technique on learning model. Its Learning is based upon Model Compression

PROCESSING OF INPUT

I. Input: likewise, can be told as assortment of information i.e., distinctive kind of pictures and diverse sort of arrangements.
II. Image processing: once the images are collected the images are processed like noise removal, skew correction, brightness, contrast etc.
III. Document and layout analysis: once the quality of the image is been enhanced it is then analyzed.
IV. Recognition: after the analysis is done the image is subjected for recognition like font style, font size etc.

V. Output: after all the process is done the output is obtained.



Procedure can be divided into 4 parts:
Dataset: We require a customized dataset far smaller than the AdobeVFR dataset that consists of abundant font styles, which we have generated by utilizing Text Recognition Data Generator
The work is broken into 4 steps:
Dataset: Since Adobe VFR Dataset datalink is huge in size and contains lot of font categories. We created custom dataset based upon required font patches using Text Recognition Data Generator.

I. Our dataset consists of given 5 font styles:

1. Lato: Lato has been used in various physical publications, including information signs and election campaign billboards. It supports all Latin alphabets, along with Cyrillic, Greek, and IPA.

2. Raleway: Raleway is an elegant sans-serif typeface family. Initially designed by Matt McInerney as a single thin weight, it was expanded into a 9-weight family.

3. Sansation: Sansation is a slightly squared sans serif typeface. It was not designed to be a text font.

4. Walkway: This font belongs to sans serif font family that is created by GemFonts, therefore, it is labeled a sans serif font. This font family has 31 varible fonts. Walkway font is sans serif font which is designed by GemFonts. This font is labeled as Sans's serif font. Walkway font family has 31 variants.

5. Roboto: Roboto is a neo-grotesque sans-serif typeface family developed by Google as the system font for its mobile operating system Android.

II. Preprocessing of Dataset: Fonts and objects are not alike which have vast spatial data available to consider their features. Specific preprocessing techniques have to be utilized to recognize very small feature changes. These techniques are: Fonts are not like objects, to have to huge spatial information to classify their features. To identify very minute feature change deep font used certain preprocessing techniques they are:

a) Noise
b) Blur
c) Affine Rotation Perspective Rotation
d) Gradient Fill Shading (Gradient Illumination)
e) Variable Character Spacing
f) Variable Aspect Ratio

III. CNN Architecture: CNN networks use a different classification method that uses 2 sub networks, dissimilar to the other picture classifying techniques, the sub networks are: Unlike other image classification CNN network, they followed a new schema like two sub networks,

a) Low Level Sub-Network: Learning based on combined set of real world and synthetic data. Learned from the composite set of synthetic and real-world data.
b) High Level Sub-Network: Learning based on deep classifying techniques including low level features for a detailed and more clear result.Learns a deep classifier from the low-level features for more details and clarification have a read of their paper.

IV. Framework (Keras): Keras module is used to develop the models for this project in jupyter notebook. As its prototyping, I used Keras to build the entire pipeline. Feel free to prototype in other frameworks.

3. PROPOSED METHDOLOGY

Compose a python content to peruse the dataset and train the model utilizing profound learning, 75% of the dataset is utilized for preparing, and the rest 25% is utilized for testing. Commotion, obscure, and point of view revolution are applied to more readily dissect pictures for textual style acknowledgment. The technique utilized for this venture for textual style acknowledgment has significantly three stages. Stage one is to get an information picture to be handled which is promptly pre handled and afterward sent for other element calculation in second step. The proposed framework for textual style acknowledgment and order framework fundamentally includes three primary stages. In stage 1, an info picture is obtained for handling and continued for pre-handling and further coordinated for highlight calculation in stage two. At last, the highlights processed are characterized through a SVM classifier. Fig 1 portrays the square chart of the proposed framework.

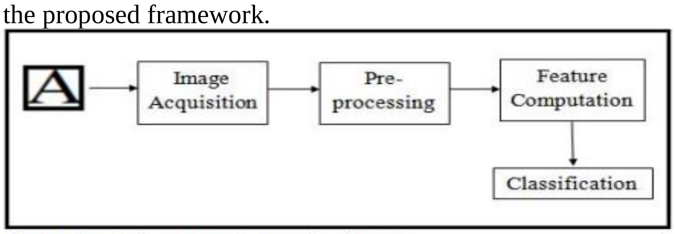


Fig 1: Block diagram of Font classification

4. LEARNING-BASED MODEL COMPRESSION

It is for the most part understood that the significant models are strongly over-characterized and thus those limits can be compacted to reduce limit by researching their plan. For an ordinary CNN, around 90% of the limit is taken up by the thick related layers, which will be our fixation for mode pressure. One technique for getting the quantity of limits is using structure factorization. Given the limit:

$$W \in R(m \times n)$$

We factorize it utilizing solitary worth decay (SVD):

$$W = USV\ T\ (1)$$

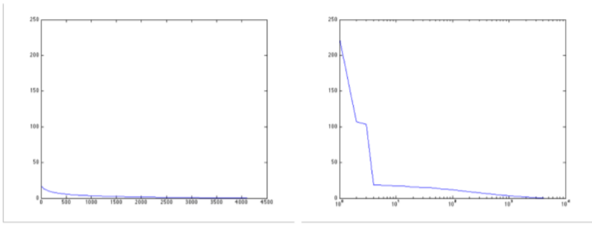
where $U \in R(m \times m)$ and $V \in R(n \times n)$ are two thick symmetrical grids and

$$S \in R(m \times n)$$

is a corner to corner matrix. To restore a harsh W , we can utilize Ue, Ve and Se, which show the submatrices contrasting with the top k single vectors in U and V close by the top k eigenvalue in S:

$$Wf = UeSeVe\ T\ (2)$$

The tension extent given m, n, and k is k(m+n+1)mn
which is outstandingly reassuring when m, n k. In any case, the assessment of SVD is compelled by the decay alongside the eigenvalues in S. For sure, even it is checked in Fig. 7 that the eigenvalue of weight structures conventionally decay fast (the sixth greatest eigenvalue is presently under 10% of the greatest one in degree), the truncation most certainly prompts information adversity, and potential execution corruptions, stood out from the uncompressed model.


(a) Standard scale (b) Logarithm scale

The plots of eigenvalues for the fc6 layer weight matrix in Fig. 5. This densely connected layer takes up 85% of the total model size.

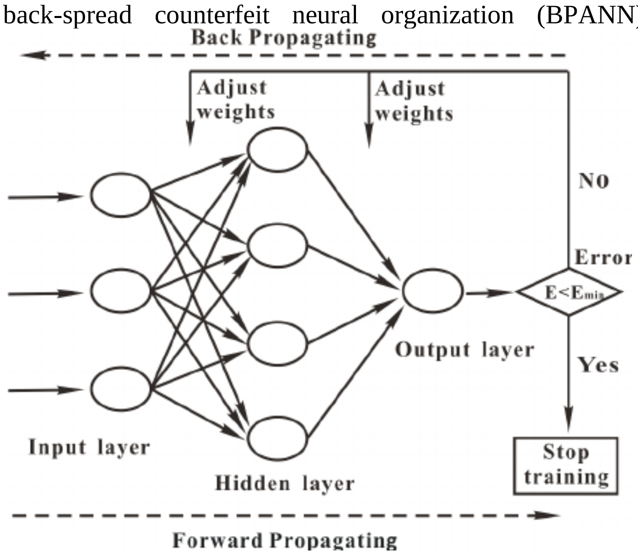
Rather than first setting up a model then lossy-compacting its limits, we propose to directly get comfortable with a lossless compressible model (the articulation "lossless" suggests as there could be no further disaster later a model is ready). Hoping to be the limit lattice W of a particular association layer, we will probably guarantee that its position is overall near somewhat consistent k. To the extent execution, in each cycle, an extra hard thresholding movement is executed on W later it is invigorated by a standard back expansion step:

$$Wk = UTk(S)VT\ (3)$$

where Tk will keep the greatest k eigenvalues in S while setting others to zeros. Wk is awesome to situate k assessment of W, as correspondingly in (2). In any case, not exactly equivalent to (2), the proposed procedure joins low-rank assessment into model getting ready and commonly redesigns them as a general rule, guaranteeing a position k weight network that is fit to be stuffed losslessly by applying (1). Note there are diverse different choices, for instance, vector quantization methods, that have been applied to compacting significant models with drawing in displays. We will inspect utilizing them together to extra pack our model later on.have been applied to compacting significant models with drawing in presentations. We will analyze utilizing them together to extra pack our model later on.

5. RELATED WORKS

Ramanathan et al. introduced a technique for text style acknowledgment based help vector machines (SVMs) utilized in characterization or relapse difficulties (directed AI algo) and Gabor channel utilized for surface investigation highlights. The technique showed 93.54% exactness in the English language.
One more review was performed by Jaiem et al. on 10 Arabic text styles [2]. The creators involved a steerable pyramid in include extraction and for acknowledgment the creators utilized k closest neighbors, a basic, managed AI calculation turns out to be delayed with information size increment and a back-spread counterfeit neural organization (BPANN).



Overall, BPANN shows better recognition accuracy than k-nearest neighbors.
The system developed by Bharath and Rani [3] has three steps; in the first step, the system reads a character image and preprocesses it. In the second step, distance profile features are computed. In total, 74 features per character image are generated. In the last step, a support vector machine (SVM) and k-nearest neighbors (KNN) are used for classification. Five different font styles are tested, and the system achieves approximately 80% average accuracy for SVM and 75% for **KNN**.
Jaiem et al. presented a font recognition system based on steerable pyramids called (AFR/SP) Arabic font recognition steerable pyramids [4]. The system uses three levels of text entity analysis: word, line and text block. In feature extraction, steerable pyramids vision (The Steerable Pyramid is a linear multi-scale, multi-orientation image decomposition that provides a useful front-end for image-processing and computer) and two statistical variables (standard deviation and mean) are used. This study used two databases, APTID/MF and APTI with different resolutions. In the classification phase, the BPANN is used. The experimental results for high-resolution text block samples show high recognition rates of approximately 99% and 93% for low-resolution.
Tensmeyer et al. presented a system that classifies font at two levels: text lines and text pages by using the convolutional

neural networks (CNNs) A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance framework [5]. At the line level, the King Fahd University Arabic Font Database (KAFD) was used to recognize 40 Arabic fonts, while at the page level, two datasets were used: the KAFD and Latin Medieval Manuscripts (CLaMM) databases, which classify 12 English scripts. The author split the dataset into three sets, namely: training, validation and testing sets; which helps in choosing the most accurate model. For comparison, the authors used two CNN architectures: the AlexNet architecture, which has five convolution layers, and the state-of-the-art ResNet-50. In the training, two models for each the ResNet and AlexNet were created from the KAFD database on the line level and page level, and in the CLaMM database, one model was created for each of architecture. In the validation, the ResNet architecture had better performance for both datasets.

6. PROBLEM STATEMENT

The issue is to perceive textual style in pictures of different organizations. This is a requirement for different authors, banner producers and other substance makers like promoters, and so forth We really want to build exactness of model by utilizing profound learning strategies.

7. CONCLUSION

In the paper, we encourage the DeepFont structure to incredibly drive the forefront in the VFR task. A gigantic arrangement of stamped certifiable data similarly as a huge corpus of unlabeled genuine pictures is assembled for both planning and testing, which is the first of its sort and will be made openly open soon. While relying upon the learning furthest reaches of CNN, we need to fight the dumbfound between available planning and testing data. The introduction of SCAE-based space adaption helps our pre-arranged model with achieving a higher than 80% top-5 accuracy. A unique lossless model tension is also applied to propel the model accumulating efficiency. The DeepFont system not only is convincing for text style affirmation, but can moreover convey a text style similarity measure for text style assurance and thought.

8. REFERENCES

[1] 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 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala, India, pp. 766- 769. IEEE, 2009.

[2] F. K. Jaiem, S. Kanoun, and V. Eglin, "Arabic font recognition based on a texture analysis." 2014 14th International Conference on Frontiers in Handwriting Recognition. IEEE, 2014.

[3] V. Bharath and N. Shobha Rani, "A font style classification system for English OCR." In 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, pp. 1-5. IEEE, 2017.

[4] F.K. Jaiem, F. Slimane, and M. Kherallah, "Arabic font recognition system applied to different text entity level analysis." 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C). IEEE, 2017.

[5] C. Tensmeyer, D. Saunders, and T. Martinez, "Convolutional neural networks for font classification." 2017 14th IAPR international conference on document analysis and recognition (ICDAR). vol. 1, IEEE, 2017.