Bangla Font Recognition using Transfer Learning Method

Conference Paper · July 2022

DOI: 10.1109/ICICT54344.2022.9850765

5 authors, including:

Md. Sakibul Hasan
Daffodil International University

Golam Rabbi
Daffodil International University

Md. Hasan Imam Bijoy
Daffodil International University

Md. Azizul Hakim
University of Nevada, Reno

Bangla Font Recognition Using Transfer Learning Method

Md. Sakibul Hasan
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
sakibul15-11441@diu.edu.bd

Golam Rabbi
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
golam15-11467@diu.edu.bd

Rejoan Islam
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
rejoan15-11431@diu.edu.bd

Md. Hasan Imam Bijoy
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
hasan15-11743@diu.edu.bd

Md. Azizul Hakim
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
azizul.cse@diu.edu.bd

***Abstract*—Font detection and similar font suggestions are essential in computer vision, document analysis, pattern recognition, web development, and core work for graphics designers and UX-UI engineers. Even though Bangla is one of the most communicated languages globally and the growing popularity of Bangla in online publications and social media platforms, there has been a noticeable demand for Bangla fonts. However, there is no appreciable work for font detection in Bangla, unlike other high resource languages like English, Chinese, Hindi, Arabic, Spanish, and German. In this work, we represent a model to recognize Bangla fonts from images using the transfer learning method. In Image Processing and Classification, resources are insufficient for such work in Bangla, so we needed to build as much raw data as possible to train our model. Therefore, as part of the work, we used 6500 raw images of five different fonts, and with augmentation, we created 26000 image data to train and 2600 images to validate our model. We applied three transfer learning models, which are VGG-16, VGG-19, and Xception. Among them, VGG-16 archives the highest accuracy of 96.23%. This paper is the first publicly available work on Bangla font recognition using the Transfer Learning approach to the best of our knowledge.**

Keywords—Bangla Font Recognitions; Font Recognition; Vgg16; Vgg19; Xception.

I. INTRODUCTION

Bengali, also known as Bangla, is an international language native to Bangladesh and the West Bengal, part of India. Approximately 370 million [1] people speak Bangla as their first or second language. In terms of total population, Bengali is the world's sixth most frequently verbal language [2]. Despite having this large number of speakers, the recent surge for Bangla font has been noticeable. However, there is no remarkable work to detect Bangla fonts automatically. Computerization and automation use computer-composed documents, newspapers, bank checks, invitation cards, birthday cards, wedding cards, visiting cards, testimonials, logo design, and billboards. Moreover, font recognition is essential in the identification, analysis, and reconstruction of documents. Font identification and suggestions from an image can dramatically increase the efficiency of the people working in these sectors. Furthermore, the utilization of various fonts can be used to add style to a website or any kind of document.

Designers can use them to match the tone of the text based on their context. Some fonts affect the readability of a newspaper or website. Therefore, Bangla font recognition and its development could be helpful to designers and web developers along with UI-UX engineers. Bangla font detection is also helpful for the analyst to find out which fonts are popular among the different levels of users having different perceptions. It also helps individuals use their favorites font style and suggests fonts according to their user usage. Bangla font detection is applicable in vast more real-life scenarios like this. Therefore, identifying Bangla font is progressively becoming more essential.

Bangla font style discovery has a broad execution. In typography, Bangla font style discovery can assume an essential part. The expertise of choosing letters and text to make an article understood, neat, and outwardly appealing to the peruse is known as typography. It isn't just barely choosing the engaging font styles. It is an indispensable part of UI plan. A decent typography will lay out a strong visual order, offer the website a practical stability and set the overall tone of the item. It holds an important role in the graphic design sector as graphic designers must have a solid understanding of fonts. In the Advertisement sector, font detection, font suggestion, and font implementation have an immense role since lucrative fonts can easily hold customers' attention. It is helpful for any company for the publicity of their products and services. For brands, font is essential; sometimes, some specific font styles work as a symbol for a brand. For example, Coca-Cola has its font style, which is Coca-Cola which represents the brand's identity. For Newspaper headline font is exigent as some headings use captivating font style to emphasize the importance of certain news. Calligraphy refers to visual art related to word composition. Calligraphy also focuses on beautiful writing by manipulating letters and symbols. For Calligraphers, it is vital to have immense knowledge about different font styles.

Several websites and browser extensions, such as the font, font finder, identify font, like font, what font. Nevertheless, those are mainly for English, and their accuracy is not satisfactory for Bangla font detection. Moreover, Bangla has many character sets of 50 characters whose geometric structures are more complex, unlike English. Also, Bangla has been widely ignored by big corporations despite having a large user base.

There has been a noticeable demand for Bangla fonts, and a transfer learning-based font detection can help users identify and use fonts according to their choice. Our research aims to recommend an expert framework to predict font category and identify the font name with the help of the transfer learning method.

II. LITERATURE REVIEW

Various researchers and academics are working on font identification using a variety of methodologies; below are some literature studies.

Islam et al. [3] developed a Convolutional Neural Network (CNN) method for recognizing Bangla literary style based on a space change methodology and a Stacked Convolution auto-encode (SCAE). They used a dataset with seven distinct text styles and 12,828 images, which they eventually enlarged to 77,828, yielding an accuracy of 98.73 percent.

As they foster Bangla OCR, Islam et al. [4] have presented a way to recognize and contain Bangla text from normal scene photos. They suggested the approach in conjunction with MSER and based separation. MSER was used to identify up-and-coming text districts. There were deceptive up-sides in the identification district. To remove misleading up-sides, they used rule-based separation strategies. The approach was tested on 50 different images. They promise that their technique is more precise than others regarding f-measure, accuracy, and review.

Hasan et al. [5] used a CNN-based grouping model and provided a framework for the arrangement of Bangla literary style based on the Deep Convolutional Neural Network (DCNN). They have compiled a massive scope dataset that includes incompletely labeled actual data and marked manufactured data. The CNN has been created to classify images into predetermined textual style classes, each of which has ten distinct PC text styles. They achieved a 96 percent line-level precision.

Wang et al. [6] presented an exchange learning approach to recognize Chinese and English textual styles and the Bangla text style. Their databases contain over 200,000 unlabeled images taken from the internet with an accuracy of 93.97 percent.

Wang et al. [7] presented the Deep Font framework, a CNN-based approach for identifying text styles. They created AdobeVFR, a large-scale VFR collection including both named produced and tagged accurate data. An SCAE-based area adaptation was performed, which helped their prepared model achieve a paramount five precision of more than 80%.

Cheng et al. [8] addressed the wide-ranging subject of visual, textual style recognition (VFR), which attempts to recognize the typeface, weight, and slant of text in an image without any prior knowledge of the content. They created a massive scope dataset with 2,420 textual style classes to solve the VFR issue. They made one picture for each English word for each font class, totaling 2.42 million fabricated images for the dataset.

Tensmeyer et al. [9] presented a straightforward architecture based on the Convolutional Neural Network (CNN). Dataset consists of 40 Arabic computer fonts; the model attained state-of-the-art performance. The authors utilized CNNs to classify fonts and large document images in the script. They got 98.8% of line-level accuracy.

Abuhaiba et al. [10] presented an algorithm for apriori Arabic optical Font Recognition (AFR) to identify the fonts of some often-used Arabic words. They decided tree for detecting Arabic fonts. Forty-eight features trained the tree. However, their decision tree could not detect all Arabic words; it could only recognize the decision tree included words.

Vijayakumar et al. [11] proposed an artificial intelligence-based algorithm to classify font styles. The described font style classification method is combined with the Capsule Network (CapsNet) algorithm to complete the font style classification problem. For a comparative analysis, the presented network structure is compared to the current Naive Bayes Classifier (NBC), Decision Tree Classifier (DTC), and K-Nearest Neighbors (KNN) algorithms.

Ramathan et al. [12] presented a surface-based method for recognizing English written styles. They use the Support Vector Machine (SVM) model to confirm distinct textual style styles. C-SVM is a kind of SVM that was used in this analysis. The suggested model was created using a Gabor channel. Times New Roman (TNR), Arial, Algerian, Courier New, Comic Sans MS, and Tahoma were all combined into four styles: standard, severe, italic, and striking italic. The SVM classifier has a 93.54 percent accuracy rate.

III. Proposed Methodology

In this section, we figured out the best working model for our dataset to recognize the Bangla fonts we have selected, the proposed working methodology is present in Fig. 1.

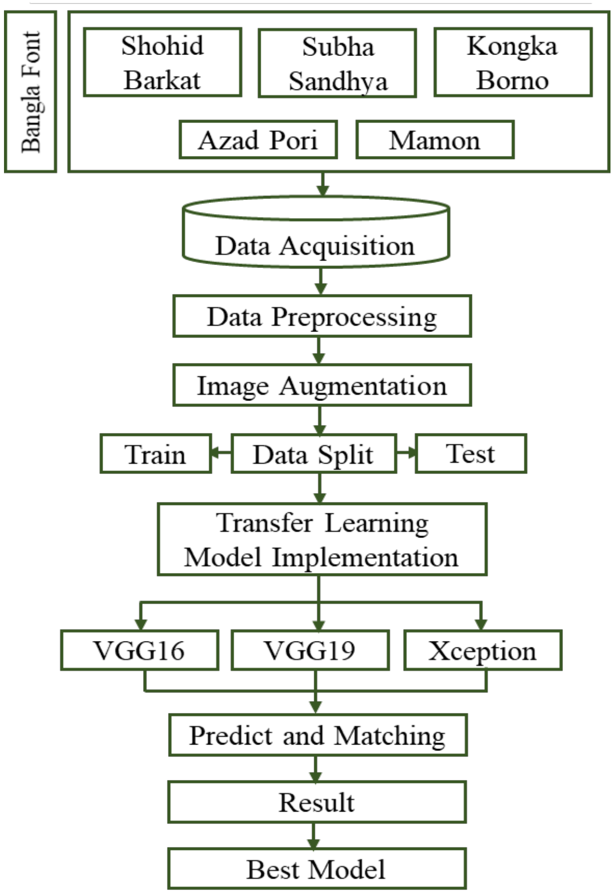

Fig. 1. Workflow diagram of our proposed methodology.

A. Image Acquisition

For a work like this, data is the central aspect for the best outcome. So, we collected every single image manually on our project, and it was challenging work to do. Firstly, we collected some online articles from different news portals, blogs, and social media sites to store good Bengali essays. Then we took pictures with the specific window size of (200 x 100) and font size of 36. Mainly we focused on the words in Bengali, and our primary data are Bengali words and alphabets. By taking a specific (200 x 100) screenshot length, we collected 6500 Bengali words in five categories, 1300 each. The five categories are five different Bengali fonts. For our task, we chose the most downloaded four Bengali fonts from the largest Bengali font publication site Lipighor. So, we selected five fonts, followed by their number of downloads.

These fonts have massive popularity among designers and artists. Therefore, we tried to collect as clean data as possible during the data collection period, which brought us a considerable advantage in image processing.

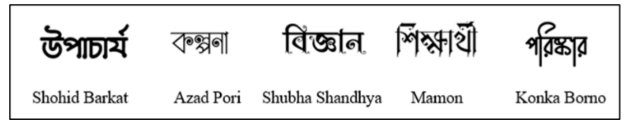

Fig. 2. Five type of Bangla Font Images (Sample Dataset).

B. Data Preprocessing and Data Augmentation

In this step, we convert the images in 200*100 pixels. To reduce overfitting, we use image augmentation in our training dataset. It is also used to get better accuracy in the prediction and data overfitting. In our training data set, we used horizontal flip, rescale, vertical flip, shear and zoom procedures to augment our data. Fig. 3 presents an idea of the picture augmentation approach.

Fig. 3. Sample images after augmentation.

Train datasets are 80% of the total collected data. The rest of the 20% is reserved for testing data. Table I shows the overall training data interpretation and the amount of acquired data after applying five ways to image augmentation.

Similar to train data, we used augmentation for test data also. So, we used 260 images for each class and only used one type of image-generation technique. So, the total test data became 520 data for each class. Table II shows the training and testing data ratio for each class.

TABLE I. Training Data Description

Class Name	Collected Data	Generated Data	AT	Total Data
Shohid Barkat	1040	5200	5	6240
Azad Pori	1040	5200	5	6240
Subha Sandhya	1040	5200	5	6240
Mamon	1040	5200	5	6240
Kongka Borno	1040	5200	5	6240
Total Data				31200

[a] AT: Number of Augmentation Techniques.

TABLE II. Data Distribution for each Class.

Class Name	Train	Test	Total
Shohid Barkat	6240	520	6760
Azad Pori	6240	520	6760
Subha Sandhya	6240	520	6760
Mamon	6240	520	6760
Kongka Borno	6240	520	6760

C. Model Implementation

At this phase, we are trying to train a precise model for the data we have gathered. VGG16, Vgg19, and Xception are three dominant transfer learning models we used in this study. The following is the model's fundamental theory.

VGG16: VGG16 is a widely used Convolutional Neural Network (CNN) [13] model used in ImageNet; a massive data set project administered by the WorldNet order for visual object recognition programming research. In ImageNet, the model achieves 92.7 percent top-5 test precision on a dataset with over 14 million images and 1000 classifications [14]. Thus, it is one of the most model structures for visual recognition. The contribution to the cov1 layer of VGG16 is a 224 by 224 RGB image with a fixed size. Instead of hyper-parameter complexity, VGG16 prioritized (3 x 3) filter convolution layers with stride one, as well as equivalent padding and maxpool layer of (2 x 2) filter of stride two. This model constantly follows the convolution and maxpool layer sequence all through the structure. Finally, it has two entirely coupled layers, followed by a yield softmax. This model needs significant preparation and is a massive organization with 138 million borders.

VGG19: VGG19 is a VGG16 variant. It has 19 layers rather than the usual 16 layers (1 SoftMax layer, three completely associated layers, 16 convolution layers, and 5 MaxPool layers). As a gift to this organization, a fixed-size (224 x 224) RGB photograph was delivered. Throughout the whole preparation set, pre-processing comprised removing the average RGB value from each of the pixel. The full notion of the picture is covered by (3 x 3) pieces with a step size of 1 pixel. The spatial purpose of the image is protected by spatial cushioning. Step 2 involved max-pooling north of a (2 x 2) pixel window. This was followed by a Rectified straight unit (ReLu) to better understand non-linearity and work on computing time. Execution of three entirely related layers, the first two of which were 4096 bytes each, followed by a layer with 1000 channels for a 1000-way ILSVRC order, and finally, a softmax operation.

Xception: Xception stands for extreme inception. It is a vast improvement over beginning v3. It takes the opposite approach to the beginning, first applying the channels to each depth guide and then packing the information space with 1x1 convolution applied across the depth. The existence or absence of a non-linearity following the initial activity differs between Inception and Xception. A ReLU non-linearity follows the two activities in the Inception model. However, there is no non-linearity in Xception.

Sequentially, the block diagram of VGG-16, VGG19, and Xception for implementation transfer learning to identify the Bangla font is present in Fig. 4, Fig. 5, and Fig. 6.

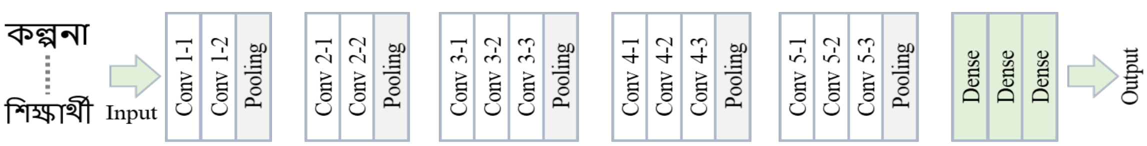

Fig. 4. VGG-16 architecture diagram.

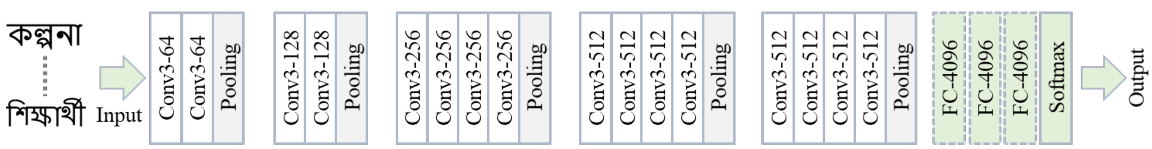

Fig. 5. VGG-19 architecture diagram.

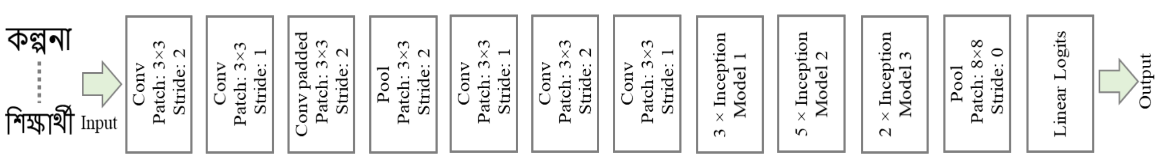

Fig. 6. Xception architecture diagram.

IV. Result and Discussions

In our research, we have used a total number of 31200 of five classes to train our model. we have also used a total of 2600 images to test our model. To measure the accuracy, we need to compare the confusion matrix of the different algorithms. Our research used VGG16, VGG19, and Xception models in image recognition and classification. In our work, we will analyze and compare the results of each transfer learning model to get the overall idea. We tried 40 epochs for each of the models.

In a classification task for language recognition [15], each of the appropriate models' resulting confusion matrix (True Label and Predicted Label) is presented in Fig 6 for the VGG-16 model, Fig. 7 for the VGG-19 model, and Fig. 8. for the Xception model. The confusion matrix [16] generates true label value and predicted label for each class in our study.

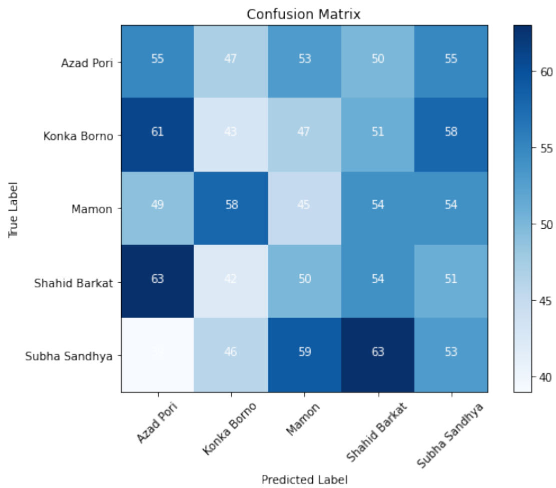

Fig. 7. Confusion matrix of VGG-16

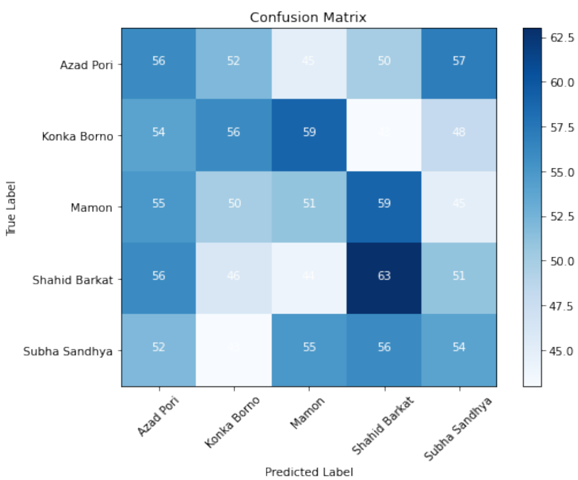

Fig. 8. Confusion matrix of VGG-19.

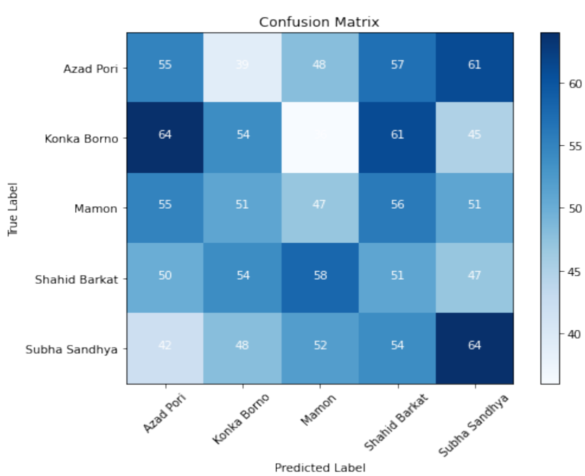

Fig. 9. Confusion matrix of Xception.

Now, we compute the accuracy and loss of each model. The performance of each model is presented in Table III.

TABLE III. MODEL PERFORMANCE.

No	Model Name	Model Accuracy	Model Loss
1	VGG16	96.23%	17.80%
2	VGG19	95.25%	16.30%
3	Xception	92.46%	31.38%

The accuracy of the VGG16 model was 96.23 percent, while the loss was 17.80 percent. The accuracy of the VGG19 model was 95.62 percent. In this case, the model has lost 16.30 percent of its value. We acquired a 92.46 percent accuracy in the Xception model. In this case, the model has lost 31.38 percent of its value. This table shows that model VGG16 has the best accuracy among the other models, followed by VGG19. ROC curve [17] for VGG-16 is present in Fig. 10.

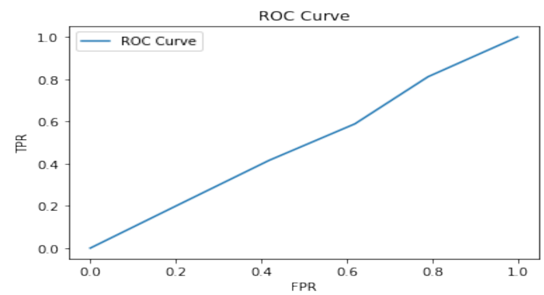

Fig. 10. ROC curve of VGG-16 model.

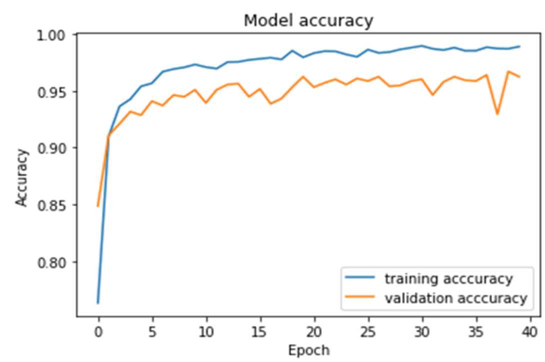

Fig. 11. Graph of Model accuracy for best model VGG-16.

Fig. 12. Graph of Model loss for best model VGG-16.

The proportion of classifications a method correctly predicts divided by the total number of predictions is known as model accuracy. A loss is a numerical value that indicates how inaccurate the model's forecast was for a particular case. The best model, VGG-16 model performance, and model loss is present in Fig. 11 and Fig. 12, respectively. After model performance gathering, we compared our work with other work. The comparative analysis is present in Table IV. Compared to the previous study, few pieces studies have been done with several models. Since this work is the first work with a promising Bangla font dataset with advanced deep learning, the CNN-based transfer learning model achieves superior accuracy in our dataset and beats all previous work with 96.23% accuracy.

TABLE IV. COMPARISON AMONG RELATED WORKS.

Authors	Method	Data	Language	Accuracy
Abuhaiba et al.	Apriori	6000	Arabic	90.8%
Hasan et al.	DCNN	6000	Bengali	96%
Wang et al.	Transfer Learning	200000	Chinese & English	93.97%
Ramanathan et al.	SVM	216	English	93.54%
This Work	**VGG16**	**33800**	**Bangla**	**96.23%**

V. CONCLUSION AND FUTURE WORK

An efficient font recognition algorithm has been presented and discussed in this work. In addition, a huge set of data is collected for both training and testing. We applied five different transfer learning models that have been trained to classify images into predefined font classes containing 5 different Bangla fonts. Among the Transfer Learning Models (TLMs) that we have applied, VGG16 performs better and obtains an accuracy of 96.23%.

The design of the proposed transfer learning-based system is shown in the Fig. 13. Firstly, the user needs to take a picture or screenshot of the font. Then the picture needs to be sent to a system where the font will be analyzed, and it will compare the font, which is a database, and if it finds any similarity with the input, it will provide an output with the name of the font and the user will get the font source link where it could be downloaded or could get some previews.

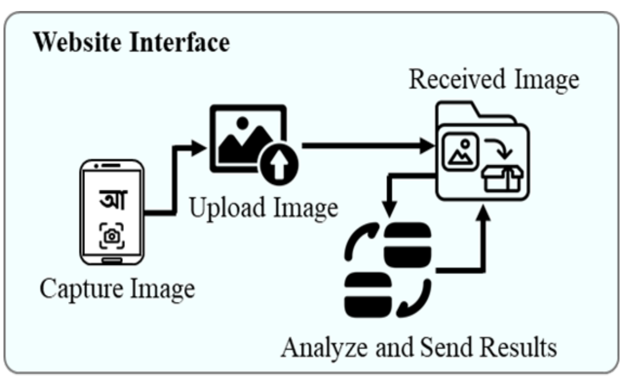

Fig. 13. Proposed Architecture of Font Recognition System in Website.

REFERENCES

[1] Bengali language - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Bengali_language. [Accessed: 28- Mar- 2022]

[2] Bijoy, Md Hasan Imam, Mehedi Hasan, Abdur Nur Tusher, Md Mahbubur Rahman, Md Jueal Mia, and Masud Rabbani. "An Automated Approach for Bangla Sentence Classification Using Supervised Algorithms." In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6. IEEE, 2021.

[3] Islam, Md, A. K. M. Rabby, Nazmul Hasan, Jebun Nahar, and Fuad Rahman. "A Novel Bangla Font Recognition Approach Using Deep Learning." In *Emerging Technologies in Data Mining and Information Security*, pp. 745-754. Springer, Singapore, 2021.

[4] Islam, Rashedul, Md Rafiqul Islam, and Kamrul HasanTalukder. "Rule Based Filtering Approach for Detection and Localization of Bangla Text from Scene Images." In *2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*, pp. 139-144. IEEE, 2017.

[5] Hasan, Zahid, Tanzila Rahman, Rokeya Islam Riya, K. M. Hasan, and Nusrat Zahan. "A CNN-based classification model for recognizing visual bengali font." In *Proceedings of International Joint Conference on Computational Intelligence*, pp. 471-482. Springer, Singapore, 2020.

[6] Wang, Yizhi, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. "Font recognition in natural images via transfer learning." In *International conference on multimedia modeling*, pp. 229-240. Springer, Cham, 2018.

[7] Wang, Zhangyang, Jianchao Yang, Hailin Jin, Jonathan Brandt, Eli Shechtman, Aseem Agarwala, Zhaowen Wang et al. "Deepfont: A system for font recognition and similarity." In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 813-814. 2015.

[8] Chen, Guang, Jianchao Yang, Hailin Jin, Jonathan Brandt, Eli Shechtman, Aseem Agarwala, and Tony X. Han. "Large-scale visual font recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3598-3605. 2014.

[9] Tensmeyer, Chris, Daniel Saunders, and Tony Martinez. "Convolutional neural networks for font classification." In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, vol. 1, pp. 985-990. IEEE, 2017.

[10] Abuhaiba, Ibrahim SI. "Arabic font recognition using decision trees built from common words." *Journal of computing and information technology* 13, no. 3 (2005): 211-224.

[11] Vijayakumar, T., and R. Vinothkanna. "Capsule network on font style classification." *Journal of Artificial Intelligence* 2, no. 02 (2020): 64-76.

[12] Ramanathan, R., 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*, pp. 766-769. IEEE, 2009.

[13] A. F. Al Haque, R. Hafiz, M. A. Hakim, and G. R. Islam, "A computer vision system for guava disease detection and recommend curative solution using deep learning approach," in *2019 22nd International Conference on Computer and Information Technology (ICCIT),* pp. 1–6, IEEE, 2019

[14] Manoharan, J. Samuel. "Capsule Network Algorithm for Performance Optimization of Text Classification." *Journal of Soft Computing Paradigm (JSCP)* 3, no. 01 (2021): 1-9.

[15] Adam, Edriss Eisa Babikir. "Deep learning based NLP techniques in text to speech synthesis for communication recognition." *Journal of Soft Computing Paradigm (JSCP)* 2, no. 04 (2020): 209-215.

[16] J. Mia, H. I. Bijoy, S. Uddin and D. M. Raza, "Real-Time Herb Leaves Localization and Classification Using YOLO," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021, pp. 1-7, doi: 10.1109/ICCCNT51525.2021.9579718.

[17] Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27, no. 8 (2006): 861-874.