






Descriptor: *Multilingual Visual Font Recognition Dataset (MVFR)*

MOSHIUR RAHMAN TONMOY ¹, MD. AKHTARUZZAMAN ADNAN ¹,
ALOE KUMAR SAHA ¹, M. F. MRIDHA ² (SENIOR MEMBER, IEEE),
AND NILANJAN DEY ³ (SENIOR MEMBER, IEEE)

¹Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh

²Department of Computer Science, American International University-Bangladesh, Dhaka 1229, Bangladesh

³Department of Computer Science and Engineering, Techno International New Town, Kolkata 700156, India

CORRESPONDING AUTHOR: M. F. Mridha (e-mail: firoz.mridha@aiub.edu).

ABSTRACT With advancements in deep learning (DL) and computer vision-based applications, the visual font recognition (VFR) field has evolved rapidly. From browser extensions to mobile and web apps, several efficient systems now exist for identifying fonts from images. However, progress in languages other than English has been limited, largely due to insufficient data availability. To address this obstacle, we created the multilingual visual font recognition (MVFR) dataset, an image dataset for the VFR domain encompassing four different languages: Bangla, Hindi, Russian, and Spanish. Our MVFR dataset comprises 50 000 images for recognizing ten distinct font styles for each language, resulting in a substantial corpus of 200 000 VFR images overall by accumulating four languages. Furthermore, we have also provided our developed language-agnostic Python generator script employed to generate the dataset, which can be extended to generate VFR image data for other languages, fueling the advancements of the VFR domain in languages with limited resources.

IEEE SOCIETY/COUNCIL Computational Intelligence Society (CIS)

DATA TYPE/LOCATION Images

DATA DOI/PID 10.17632/cnd2wh65my.1

INDEX TERMS Computer vision, font style recognition, multilingual VFR, pattern recognition, visual font recognition (VFR).

BACKGROUND

Visual font recognition (VFR) involves identifying the font family or families utilized in images that contain text [1]. Despite its numerous practical applications, the VFR domain has largely been overlooked by the vision community. However, with the release of public VFR datasets, researchers are now focusing more on this interesting vision domain, resulting in the production of various effective automated recognition methods. Nonetheless, all these works are based on VFR data from the English language due to the availability of large-scale public English VFR datasets, such as AdobeVFR [2]. Little research has been conducted on other languages such as Chinese VFR [3], Persian VFR [4], and Arabic VFR [5]. We find that the lack of curated public datasets is one of the prime reasons behind the low progress

in VFR of languages other than English, which led to the motivation for the creation of this multilingual VFR dataset comprising visual font styles across four popular languages: Bangla, Hindi, Russian, and Spanish.

The multilingual visual font recognition (MVFR) dataset is the first-ever large open-source VFR dataset on the respective languages. Furthermore, it is a comprehensive synthetic dataset that can be a valuable resource for researchers working on the VFR domain. Researchers and developers can explore the use of deep learning (DL) architectures for effectively recognizing visual font styles by employing this dataset, such as developing tools and applications for VFR of the respective language, e.g., browser extensions and mobile apps. For instance, in one of our recent studies, we proposed an automated lightweight font recognition method deploy-

able in resource-constrained devices, where we utilized the Bangla VFR data from this dataset for training and validation of the proposed model [6].

Apart from the VFR application, this dataset can also be utilized in other computer vision applications, for instance, researchers can experiment with this dataset for optical character recognition (OCR) using diverse font styles. Finally, having the motivation to fuel the advancements of VFR in other less-explored languages, we also released the associated Python generator script with the dataset that researchers can employ to produce synthetic VFR datasets for their other languages.

Table I presents a summary of our MVFR dataset. Here, for each language, there are ten different fonts and 5000 common words, resulting in a corpus of 50 000 images containing those words for the respective languages written through ten different fonts. Thus, we obtained a final dataset comprising a total of 200 000 images combining the data from all four languages. Fig. 1 shows sample data for each of the ten fonts of the Bangla languages. Each data sample is generated using a 400×200 white canvas, and a word is printed and centered on top of that using each of the ten distinct font styles. Similarly, Figs. 2, 3, and 4 portray sample data for all ten selected fonts from the Hindi, Russian, and Spanish languages, respectively.

COLLECTION METHODS AND DESIGN

The generation of this synthetic dataset can be presented as a three-step process. It begins with acquiring the necessary raw materials: fonts and word corpus for the respective languages. The next step is to prepare the obtained materials for final production. Finally, a Python script takes the fonts and word corpus and outputs the desired data. Fig. 5 portrays the overall workflow of the data collection and generation design in three steps.

To understand the overall data generation, the three main steps are briefly discussed in the following sections.

Raw Materials Collection

We collected core raw materials such as fonts and word corpora for the respective languages from well-known open-source platforms. The collection of common words was acquired from the popular data science platform Kaggle for all four languages: Bangla [7], Hindi [8], Russian, and Spanish [9]. These word corpora comprised thousands of common words of the respective languages. Next, we randomly acquired ten fonts for each of the languages from multiple open-source font-sharing web platforms: Bangla fonts from Lipighor¹ and FontBD.² Hindi fonts were acquired from IndiaTyping,³ and Spanish fonts were acquired

TABLE I. Summary of the Dataset

Language	Font Name	Number of Images	Total
Bangla	Alinur Phulkuri	5000	50 000
	Alinur Saikat	5000	50 000
	Alinur Sanghoti	5000	50 000
	Fazlay Munnisha	5000	50 000
	Jagat Shonkhoneel	5000	50 000
	Mahfuz Himadri	5000	50 000
	Niladri Russian	5000	50 000
	Patabahar	5000	50 000
	Suborno Jayonti	5000	50 000
	Upohar 56	5000	50 000
Hindi	Aparajita	5000	50 000
	Arya-Regular	5000	50 000
	Kalam-Regular	5000	50 000
	Kokila	5000	50 000
	Rajdhani-Regular	5000	50 000
	RhodiumLibre-Regular	5000	50 000
	RozhaOne-Regular	5000	50 000
	Tillana-Regular	5000	50 000
	Utsaah	5000	50 000
	YatraOne-Regular	5000	50 000
Russian	1 Balmoral LET 1.0	5000	50 000
	1 Cataneo BT	5000	50 000
	1 Fine Hand M2	5000	50 000
	1 Harrington M	5000	50 000
	1 Shelley Volante	5000	50 000
	18VAG Rounded M	5000	50 000
	A920_R	5000	50 000
	Burlak	5000	50 000
	Chocogirl	5000	50 000
	Gotham Pro Black	5000	50 000
Spanish	Angelique Rose	5000	50 000
	Cienfuegos	5000	50 000
	Cronus Round	5000	50 000
	Franciscolumas Briosa	5000	50 000
	Lemonade Stand	5000	50 000
	Sarsaparilla	5000	50 000
	Sevillana	5000	50 000
	Snicker Snack	5000	50 000
	TradingPostNF	5000	50 000
	Yulong	5000	50 000

from 1001Fonts⁴ and FontSpace.⁵ In the end, all the Russian fonts were collected from another open-source font-sharing platform, named Russian Fonts.⁶

Material Preparation

Having the word corpus and selected fonts for each language, we then curated a list of words by opting for character lengths between 9 and 10 to maintain word-length

¹<http://www.lipighor.com>

²<http://www.fontbd.com>

³<http://www.indiatyping.com/index.php/download/top-50-hindi-unicode-fonts-free>

⁴<http://www.1001fonts.com/spanish-fonts.html>

⁵<http://www.fontspace.com/category/spanish>

⁶<http://www.russianfonts.org>



FIG. 1. Sample VFR data in the Bangla language. (a) Alinur Phulkuri. (b) Alinur Saika. (c) Alinur Sanghoti. (d) Fazlay Munnisha. (e) Jagat Shonkhoneel. (f) Mahfuz Himadri. (g) Niladri Russian. (h) Patabahar. (i) Suborno Jayonti. (j) Upohar 56.

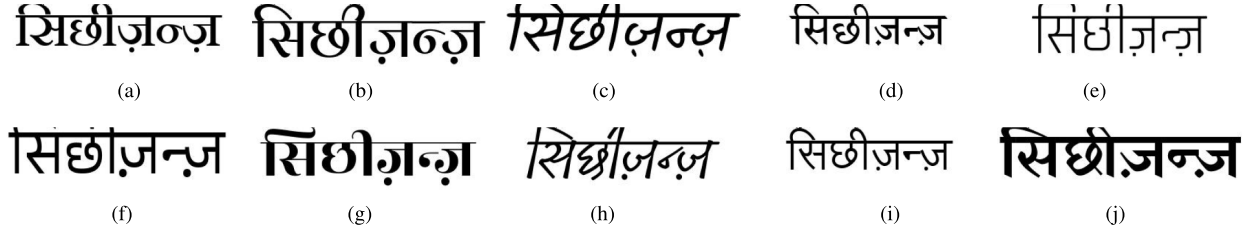


FIG. 2. Sample VFR data in the Hindi language. (a) Aparajita. (b) Arya-Regular. (c) Kalam-Regular. (d) Kokila. (e) Rajdhani-Regular. (f) RhodiumLibre-Regular. (g) RozhaOne-Regular. (h) Tillana-Regular. (i) Utsaah. (j) YatraOne-Regular.



FIG. 3. Sample VFR data in the Russian language. (a) 1 Balmoral LET 1.0. (b) 1 Cataneo BT. (c) 1 Fine Hand M2. (d) 1 Harrington M. (e) 1 Shelley Volante. (f) 18VAG Rounded M. (g) A920_R. (h) Burlak. (i) Chocogirl. (j) Gotham Pro Black.

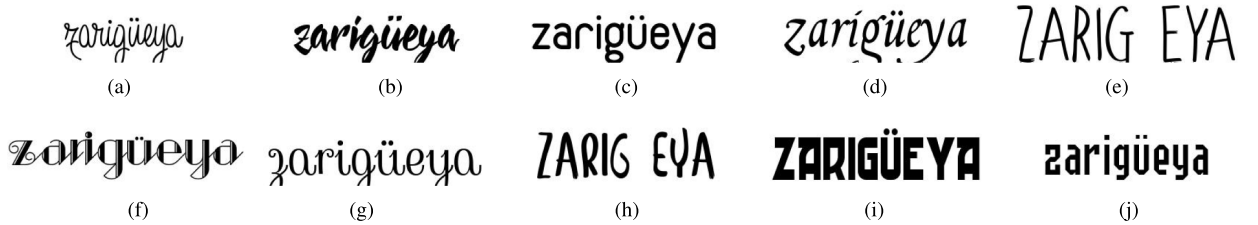


FIG. 4. Sample VFR data in the Spanish language. (a) Angelique Rose. (b) Cienfuegos. (c) Cronus Round. (d) Franciscolucas Brios. (e) Lemonade Stand. (f) Sarsaparilla. (g) Sevillana. (h) Snicker Snack. (i) TradingPostNF. (j) Yulong.

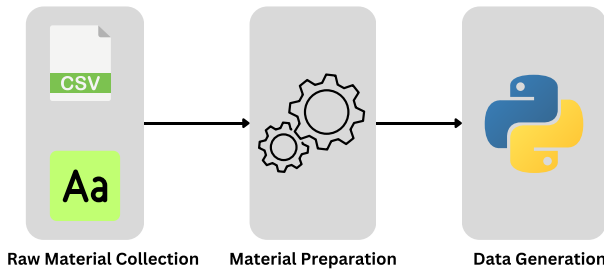


FIG. 5. Overview of data generation process.

uniformity. Then, we randomly sampled 5000 words from the curated list using Python's random module for the final production.

Data Generation

To generate data utilizing the acquired fonts and curated word corpus, we employed Python and one of its popular packages, Pillow, to develop the generator script. Algorithm 1 presents the simplified algorithm employed for developing our synthetic data generator. The generator takes the word corpus, fonts, output image size, text font size, and output directory as input. Next, it prints each of the words from the word list on the center of a new blank white image each time and saves them in the output directory. The process continues for all the fonts in the font directory as input. Thus, it produces a collection of images containing words in distinct fonts for the respective languages.

Algorithm 1: MVFR Data Generation

Input: *WORDS*, *FONT_DIR*, *IMG_SIZE*, *BG_COLOR*, *F_SIZE*, *OUT_DIR*
Output: Generated images saved in *OUT_DIR*

```

1: for each font in FONT_DIR do
2:   for each word in WORDS do
3:     Create an image  $\leftarrow$  IMG_SIZE, BG_COLOR
4:     Define center coordinates  $\rightarrow$  (X, Y)
5:     Print word  $\leftarrow$  image, font, (X, Y), F_SIZE
6:     Save the image  $\rightarrow$  OUT_DIR
7:   end for
8: end for

```

In our MVFR dataset, we utilized white colored 400×200 canvas and chose a fixed word length between 9 and 10 along with an optimal font size and centered coordinates for printing the words. Additionally, the words are printed as they were in the source. Nonetheless, the generator script can be customized according to further requirements. For example, the choice of background color can be simply white to complex combinations of colors along with effects such as Gaussian blurs and noise. Font sizes and text coordinates can also be modified for the creation of a large-scale complex and diverse dataset.

VALIDATION AND QUALITY

Our MVFR dataset offers fresh and curated VFR data across four popular languages. To validate the suitability of our MVFR dataset for DL-based applications, we fine-tuned and evaluated two popular pretrained CNN models, namely VGG16 and MobileNetV2. They were implemented utilizing the Keras API over the dataset, and the experimental phase was carried out using the free computational resources provided by Google Collaboratory. The data from each of the four languages were trained and evaluated separately. Both models were trained for up to ten epochs, with a learning rate of $1e-3$, batch size of 64, and a 20% validation split.

With this evaluation, we find that MobileNetV2 tends to yield better performance than VGG16 in all four languages, attaining an accuracy score in the range of 86%–96%. Fig. 6 presents a visualization of the attained F1-score comparison of the employed DL models over the MVFR dataset for each language. Table II represents the evaluation metrics of the employed models in both the training and validation sets across each language.

RECORDS AND STORAGE

The MVFR dataset is stored in a zipped folder that contains subsequent four zipped folders holding the data of the four languages. The base folder is named *MVFR Dataset*, and the subsequent four data folders are named according to their respective languages. Each of these language folders contains ten subfolders that hold 5000 image data instances in JPG format. These subfolders are named by their respective font names, which can serve as class names in classification or

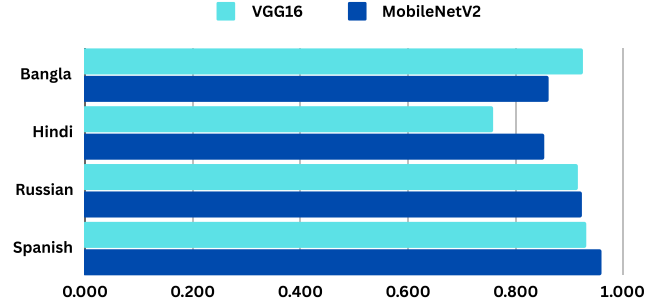


FIG. 6. Comparison of F1-scores attained by VGG16 and MobileNetV2 for each language.

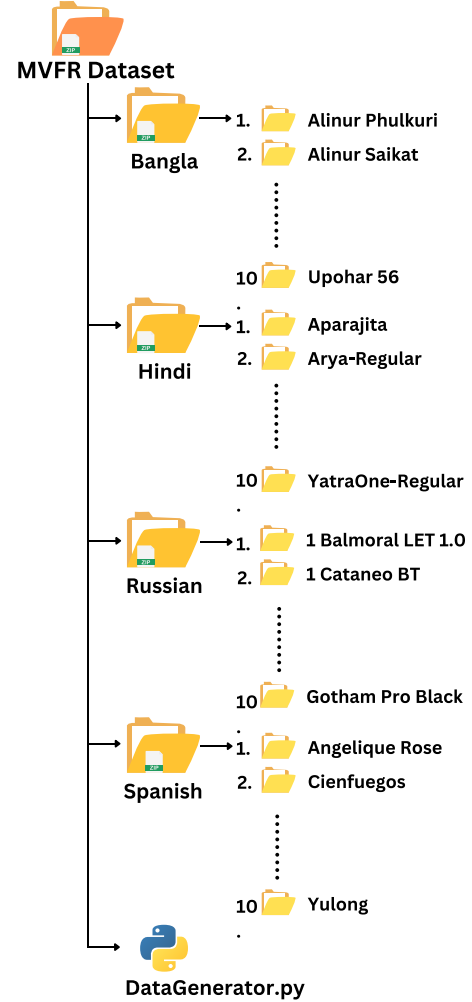


FIG. 7. Outline of the dataset storage directory structure.

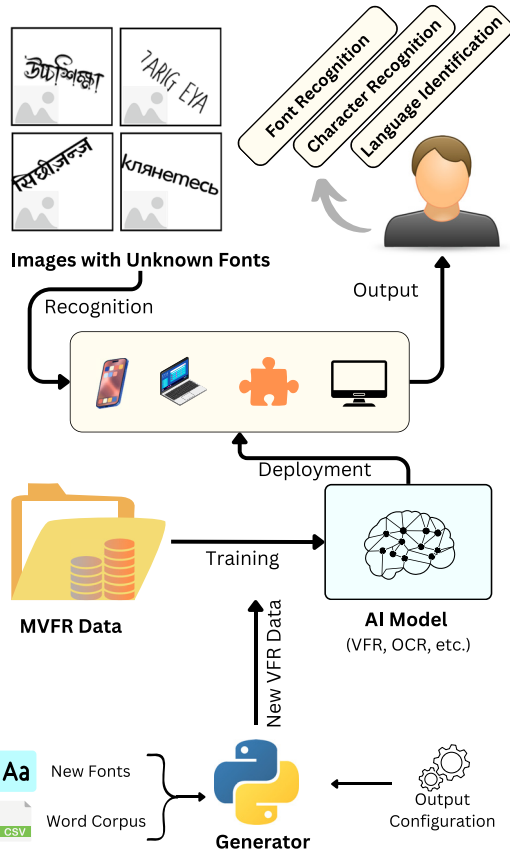
recognition tasks. In addition, the generator python script, named *DataGenerator.py*, is also shared within the base folder. Fig. 7 illustrates the overall folder structure of the MVFR dataset.

INSIGHTS AND NOTES

The MVFR dataset is motivated by the inception and advancements of the VFR domain in less-focused yet popular languages through eradicating public data scarcity. During the generation, we incorporated black-and-white combinations for generating images, whereas real-world data can

TABLE II. Performance of the Pretrained CNN Models Over the MVFR Dataset

Model	Languages	Train Set				Validation Set			
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
VGG16	Bangla	92.71%	92.75%	92.71%	0.9271	92.48%	92.52%	92.48%	0.9248
	Hindi	75.40%	77.97%	75.40%	0.7524	75.97%	78.37%	75.97%	0.7579
	Russian	92.11%	92.45%	92.11%	0.9213	91.52%	91.90%	91.52%	0.9154
	Spanish	93.03%	93.38%	93.03%	0.9309	93.04%	93.38%	93.04%	0.9310
MobileNetV2	Bangla	86.86%	88.39%	86.86%	0.8690	86.10%	87.62%	86.10%	0.8611
	Hindi	86.17%	86.83%	86.17%	0.8614	85.28%	85.96%	85.28%	0.8526
	Russian	92.67%	93.11%	92.67%	0.9265	92.32%	92.75%	92.32%	0.9229
	Spanish	96.42%	96.43%	96.42%	0.9642	95.94%	95.95%	95.94%	0.9593

**FIG. 8.** Usage of the MVFR dataset and the generator script.

be composed of various combinations including grayscale to RGB. Furthermore, we only employed ten fonts for each language, while there are enormous fonts out there for the respective languages, and fonts that share common styles might cause false positives for recognition models. In addition, we generated high-quality data while real-world data can be blurry, noisy, and poor in overall quality. Furthermore, our dataset is curated and only contains words printed in uniform size and color using respective fonts, while real-world data can be messy and complex. For instance, characters might be in heterogeneous forms and comprise a variety of colors. Nonetheless, our developed language-agnostic Python generator script can be further extended to generate complex and more realistic data in the future,

incorporating the above concerns for other languages. In Fig. 8, we illustrate the multipurpose usage of the proposed MVFR dataset and the generator script. In need of VFR data in any new language, the generator can be employed with user-defined configurations and materials to generate diverse data. In addition, the dataset can also be utilized for other applications apart from VFR as presented in the figure and discussed in “Background” section.

SOURCE CODE AND SCRIPTS

The MVFR dataset is stored and publicly available in the open-source cloud-based data-sharing platform, Mendeley Data, under the name of *MVFR: Multilingual Visual Font Recognition Synthetic Dataset* [10]. The dataset can be directly accessed using the following URL: <https://doi.org/10.17632/cnd2wh65my.1>.

ACKNOWLEDGMENT

The authors have no conflicts of interest.

REFERENCES

- [1] G. Chen et al., “Large-scale visual font recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3598–3605.
- [2] Z. Wang et al., “DeepFont: Identify your font from an image,” in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, New York, NY, USA: ACM, 2015, pp. 451–459. [Online]. Available: <https://doi.org/10.1145/2733373.2806219>
- [3] X. Li, J. Wang, H. Zhang, Y. Huang, and H. Huang, “SwordNet: Chinese character font style recognition network,” *IEEE Access*, vol. 10, pp. 8388–8398, 2022.
- [4] M. Mohammadian, N. Maleki, T. Olsson, and F. Ahlgren, “Persis: A Persian font recognition pipeline using convolutional neural networks,” in *Proc. 12th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, 2022, pp. 196–204.
- [5] C. Tensmeyer, D. Saunders, and T. Martinez, “Convolutional neural networks for font classification,” in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 01, 2017, pp. 985–990.
- [6] M. R. Tonmoy et al., “A lightweight visual font style recognition with quantized convolutional autoencoder,” *IEEE Open J. Comput. Soc.*, vol. 5, pp. 120–130, 2024.
- [7] M. M. Hasan, “80k Bangla words list (Bangla Dictionary),” 2022, Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/mahadivai/spelling-checker-v1>
- [8] I. Kukreti, “Common Hindi words,” 2023, Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/ishantkukreti/common-hindi-words>
- [9] J. Pardyak, “Lists of words in 30 European languages,” 2021, Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/jacekpardyak/languages-of-europe>
- [10] M. R. Tonmoy, “MVFR: Multilingual visual font recognition synthetic dataset,” Jan. 2024. [Online]. Available: <https://doi.org/10.17632/cnd2wh65my.1>