IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 3, September 2022, pp. 1143∼1152
ISSN: 2252-8938, DOI: 10.11591/ijai.v11i3.pp1143-1152

A machine learning approach for Bengali handwritten vowel character recognition

Shahrukh Ahsan[1], Shah Tarik Nawaz[1], Talha Bin Sarwar[2], M. Saef Ullah Miah[2], Abhijit Bowmik[1]
[1]Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, Bangladesh
[2]Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pekan, Malaysia

Article Info

Article history:

Received Sep 7, 2021
Revised Jun 17, 2022
Accepted Jun 24, 2022

Keywords:

BanglaLekha-isolated
Bengali handwritten vowel recognition
Handwritten character recognition
Machine learning
Support vector machine

ABSTRACT

Recognition of handwritten characters is complex because of the different shapes and numbers of characters. Many handwritten character recognition strategies have been proposed for both English and other major dialects. Bengali is generally considered the fifth most spoken local language in the world. It is the official and most widely spoken language of Bangladesh and the second most widely spoken among the 22 posted dialects of India. To improve the recognition of handwritten Bengali characters, we developed a different approach in this study using face mapping. It is quite effective in distinguishing different characters. The real highlight is that the recognition results are more efficient than expected with a simple machine learning technique. The proposed method uses the Python library Scikit-Learn, including NumPy, Pandas, Matplotlib, and support vector machine (SVM) classifier. The proposed model uses a dataset derived from the BanglaLekha isolated dataset for the training and testing part. The new approach shows positive results and looks promising. It showed accuracy up to 94% for a particular character and 91% on average for all characters.

Corresponding Author:

Talha Bin Sarwar
Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang
26600, Pekan, Malaysia
Email: talhasarwar40@gmail.com

1. INTRODUCTION

Handwriting recognition has proven to be quite challenging in recent years. Handwritten characters by different people show many complexities as it is not identical and varies in shapes and writing styles [1], [2]. There have been several methods that are introduced for English character recognition. One of the most applicable techniques is by training neural networks for the acknowledgment of characters [3]. At present, Bengali is one of the utmost spoken languages, placed around fifth in the world and second among the South Asian Association for Regional Cooperation (SAARC) countries [4]. In almost all phases of life in Bangladesh and in some parts of India, language is used to communicate. Around 220 million individuals worldwide presently utilize Bengali to talk and compose reason. A proper machine learning system that works efficiently to recognize its characters is long overdue for such a widely used language.

In addition, several works have been done on Bengali character recognition, where it has been challenging to achieve better execution and prediction results due to the natural complexity of most Bengali alphabets. The language has a long and rich scientific heritage of over a thousand years and a history of language evolution. Researchers have presented different types of feature extraction techniques and proposed some new feature extraction techniques for recognizing handwritten Bengali characters. Since Bengali consists of different parts such as the upper part, middle part, lower part, and disjunctive part, many researchers have developed anatomical feature extraction techniques [5], [6]. Since the characters can be divided into different zones, researchers have used a zone-based feature extraction method [7]. Islam *et al.* used a modified syntactic method for recognizing Bengali handwritten characters [8]. The two most popular classifiers used to classify handwritten characters are the support vector machine (SVM) and the hidden Markov model (HMM) [9, 10]. In the SVM, the kernel that works well is the radial basis function (RBF) kernel. It can classify well despite the different ways of writing the characters. The Bengali language has fifty basic characters with numerous comparable signs. To ensure better execution in recognizing the handwritten characters, a method focusing on eleven vowel characters has been proposed in this paper. Figure 1 represents a sample of eleven handwritten Bengali vowel characters.

The proposed method mainly uses the Python library Scikit-learn [11] and an additional SVM classifier on a derived matrix dataset. It has been used to solve pattern recognition problems mostly applied to visual images. Recognition of Bengali handwritten characters is challenging compared to the published forms of characters. This is because the characters put on paper by different individuals are not identical and differ in several features such as size, shape, and orientation of the writing. Scikit-learn is one of the new metrics for image characterization frameworks that legitimately extract visual examples from pixel images with minimal pre-processing. Our proposed method classifies individual characters using adapted Scikit-learn functions, such as classification and regression algorithms, e.g., SVM. SVM is used to classify different shapes, and variants of handwritten characters from the BanglaLekha-Isolated [12] dataset. The proposed method using SVM exhibits high ordering accuracy and outperforms several other approaches.

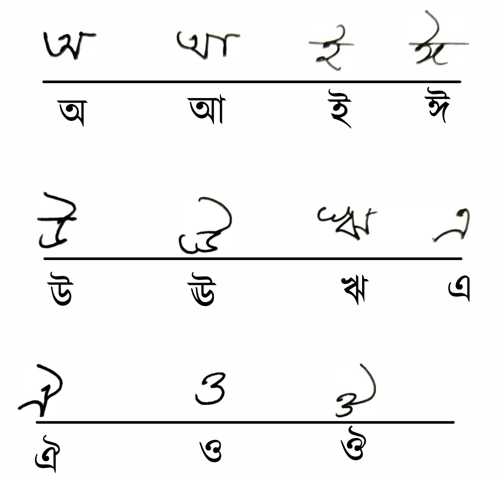


Figure 1. Sample of eleven handwritten vowel characters of Bengali alphabet

2. RELATED STUDY

There are several previous research works on the recognition of Bengali handwritten characters. Most of them use the traditional machine learning approach or neural networks for this task. Convolutional neural networks (ConvNets or CNNs) are deep artificial neural networks used to classify images, group them by proximity, and recognize objects within scenes [13], [14]. Numerous computations can recognize faces, street signs, and numerous pieces of visual information. Hardly less noteworthy work exists for the recognition of Bengali characters. Bhowmik *et al.* proposed a combined classifier utilizing the RBF system and SVM, multilayer perceptron (MLP) [15]. They considered some comparable characters as a single example and prepared the classifier for 45 classes [15]. Another work states that three different component extraction strategies were used in the segmentation phase, but the character samples were divided into 36 classes, combining comparative characters into a single class [16]. On the other hand, fewer works are not based on ConvNets. One of these overlooked things is analyzing the pattern of the image as a matrix and use of the Scikit-learn library, which also specializes in classification and regression algorithms, including SVM. Some studies rely solely on the CNN architecture. Das *et al.* [17] proposed a CNN-based architecture for recognizing handwritten Bangla characters. The authors achieved 93.18% accuracy for handwritten Bangla vowel characters, 99.5% for digits, and 92.25% for consonant characters. Reza *et al*. [18] proposed a transfer learning-based model in combination with CNN to recognize composite characters from basic characters. The authors used a transfer learning approach to recognize compound characters by transferring knowledge from pre-trained basic characters to CNN.

Scikit-learn is an undeniably prominent AI library. It is written in Python and is intended to be productive and straightforward, accessible to non-specialists, and reusable in various contexts [19]. It highlights various clustering, classification, and regression computations, including SVM, k-means, random forests, and DBSCAN, and also works with the scientific and numerical Python libraries called SciPy and NumPy. Few Python libraries have strong execution in most areas of AI computation, and Scikit-Learn is truly outstanding [20]. It is a package that provides effective variants of an enormous number of common algorithms. Scikit-Learn features a new, unified, and streamlined API, as well as equally valuable and complete online documentation [11]. One advantage of this consistency is that once we understand the essential use and language structure of Scikit-Learn for one type of model, switching to another model or algorithm is exceptionally easy [21]. The Scikit-Learn library must be installed prior to use and is essentially based on scientific Python (SciPy). This stack includes NumPy, SciPy, Matpotlib, IPython, Sympy, and Pandas. The library is focused on modeling data. It does not focus on loading, controlling, and sketching data [22].

Now that many systems have already been established for the task of character recognition, many results are compared based on maximum precision [23], [24]. The role that Scikit-learn plays in classifying languages is outstanding. Scikit-learn estimators follow certain principles to make their behaviour increasingly predictive. The precision level of Scikit-learn, which works even better when combined with other techniques such as SVM, can distinguish the complex characters of any language with more ease and gives better results. Some famous compilations of models offered by Scikit-learn include clustering, cross-validation, dimensionality reduction, ensemble methods, feature extraction/selection, parameter tuning, and manifold learning [25], [26].

Most of the studies mentioned above use neural network architectures, deep neural network architectures, and a combination of SVM and MLP. Very few papers use SVM alone. Moreover, the studies that have used SVM have also used MLP in conjunction with it. The works that used neural networks proposed a very complex architecture and a resource-intensive system. So, it is high time to investigate a simple approach and compare the results with the existing approaches. With this background, this study proposes a very simple architecture for recognizing the vowel signs of the Bangla alphabet.

3. PROPOSED METHOD

The proposed method is divided into two segments for easy and understandable implementation. The first segment is about the preparation of the training dataset, which is unique for Bengali character recognition and comes from the ”BanglaLekha-Isolated” dataset. The second segment is the prediction part which can be the core of the system. In this segment, the derived dataset is trained with a SVM machine learning algorithm, and a prediction model is built. The derived dataset used in this study is divided into two parts, one of which is used for training the system and the other for testing. The proposed method uses a very simple architecture for Bangla vowel character classification. Common image processing tasks such as edge detection and matrix generation are used, and a simple machine learning model like a linear support vector machine is used. Although the architecture is simple, the performance in classifying vowel characters is remarkable compared to other works.

In SVM, all points in a support vector problem are considered as one vector with one magnitude and one direction. The vector of the point x is projected onto another vector $\vec{w}$ that is perpendicular to the median line to classify it as positive or negative. If this projected value is greater than constant c, then it is a positive sample. Otherwise, it is a negative one. The SVM is defined as the dot product of two vectors, where x is the input and w is the perpendicular vector of the median line, expressed by (1).

$$\vec{w} \cdot \vec{x} >= c \tag{1}$$

3.1. Data preparation

This section describes the steps required to prepare the data for training the machine learning model. Several steps are required to prepare the dataset, namely collecting raw images, deriving pixel output, and creating a machine-readable file with comma-separated values (CSV). Figure 2 gives an overview of the data preparation steps.

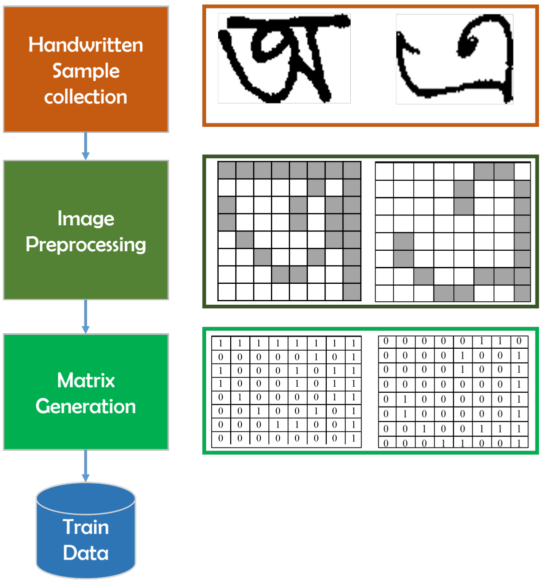


Figure 2. Data preparation steps for the proposed method

3.1.1. Collecting handwritten samples

There are very few Bengali datasets available online. "BanglaLekha-Isolated" is chosen to take the first handwritten samples so that we can train our dataset. The entire set contained a significant number of samples ranging from simple to modern handwriting and cursive. Two hundred samples are taken for individual characters for implementation. These selected samples are later converted into the individual matrix of characters. For the pre-processing of the examples, a median filter is used to remove the noise and keep the visual edges comparatively sharp.

3.1.2. Deriving pixel output

The Python imaging library (PIL) is used for image processing-related tasks in this thesis. For example, for manipulation and I/O operation of various image file formats. Some of the file types processed in this system are joint photographic experts group (JPEG), portable network graphics (PNG), tagged image format (TIF), graphics interchange format (GIF), and portable document format (PDF). The included modules contain definitions for a predefined set of filters that allow the use of color strings, image sharpeners, and a high-quality downsampling filter. Sample files input via our custom Python module are processed, and output is a pixel matrix for individual alphabets.

3.1.3. Creating training CSV file

The proposed method requires a CSV file containing all pixel matrices of handwritten sample characters as input to the machine learning algorithm. This has not yet been implemented for Bengali digits and characters to the best of the authors' knowledge. After getting the matrix as output from the pixel generation of each raw image, it is stored in a single line for each output in the CSV file. This line contains a label that is the first index or character in that line. We separate the label followed by the matrix with a comma. This CSV file is needed for the next phase to distinguish the two.

3.2. Machine learning model training and prediction

In this step, a machine learning model, namely SVM, is trained with the prepared training dataset and later the prediction is executed in the trained model. A custom Python module is developed to read the dataset from the CSV file as individual matrices. All image pixel matrices from separate rows and their corresponding labels are trained serially and the trained data is stored in a decision tree classifier as arrays. The implemented algorithm takes the row of input image pixel values and matches them with the trained images from the decision tree classifier. The value of the input image pixels and the trained image pixels are matched, and the label to which the data most closely matches is returned from the trained images and printed as a prediction of the input image. A black and white output image of the 28x28 matrix is displayed in a new and different window.

Figure 3 represents the overview of the training and prediction process of the system. In this study, SVM was used for the classification task because the SVM algorithm is simple and fast. Compared to other algorithms such as artificial neural network (ANN), CNN, random forest (RF), the structure of SVM is very simple to implement and faster than the mentioned approaches considering the performance of the model [27, 28].

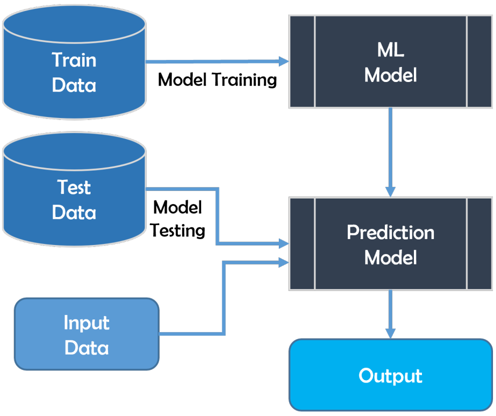


Figure 3. System training and prediction steps of the proposed method

3.2.1. Implementation of SVM model

Python is an incredibly useful programming language on its own, but with the help of some mainstream libraries it becomes an amazing domain. Running SVM with Scikit-Learn in this project requires importing libraries like NumPy [29], Pandas [30], and Matplotlib [31] alongside the dataset. NumPy stores estimates of similar information types in a multidimensional array. Scikit-Learn has the SVM library, which has worked in classes for various SVM algorithms. The support vector classifier (SVC) class, which is included in the SVM library as SVC, performs the classification task [32]. This class requires one important parameter, which is the kernel type. Given a simple SVM, we essentially set this parameter to ”linear” since a simple SVM can only characterize linearly distinguishable information. The fit technique for the class SVC is invoked to prepare the algorithm for the training data that is passed as a parameter to the fit strategy. To make predictions, the prediction technique for class SVC is used [19].

3.3. Evaluation metrics

Precision, recall, and F1 are evaluation metrics for machine learning classification models [33]. However, they are different methods to measure the accuracy of a model from different angles. True positives (TP) and true negatives (TN), and false positives (FP) and false negatives (FN) are values that indicate how often a model correctly or incorrectly predicts a particular class. For instance, a classification model predicts words A and B. If the model avoids most errors in predicting both words A and B, then the model has high precision. If the model makes no errors in predicting A as B, then it has high recall. However, what if the model excels at predicting one class but fails at the other? Here it would be misleading to consider precision or recall in isolation. This is where F1 comes in, which balances and considers both precision and recall. The metrics are calculated using (2) to (4).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

3.4. Experimental setup

All experiment codes are written in Python programming language, version 3.6.5. The Scikit-learn library is used for system training, data processing, and image processing-related tasks. A laptop with an AMD Ryzen 4800H processor, a GTX 1650 GPU, and 24 GB RAM is used as training and testing hardware with the Windows 10 Professional Edition operating system.

4. RESULTS ANALYSIS AND DISCUSSION

The SVM is a widely used machine learning technique with significant results on high-dimensional datasets. SVMs have been particularly studied and evaluated as pixel-based image classifiers [34]. Sample images are used to train the SVM, and the final image classification results are satisfactory in the experiment conducted. The results of this experiment are based on the derived outputs. The prediction is achieved by the number of correct predictions of the test images. After processing and training our datasets with SVM in Python, the result is extracted. A correct prediction is considered only if the predicted value matches the actual label of the image. To measure accuracy, we determine how many of the test images are actually predicted correctly. If the predicted value matches the actual label of the image, the image is predicted correctly. Therefore, a variable count is incremented for each correctly predicted image based on the actual label. Once the final count value is determined, it is divided by the number of input images entered and then multiplied by 100 to obtain the percentage accuracy of the system. The result is a prediction accuracy between 87% and 94% and an average accuracy of 91% for all vowel signs. However, the work mentioned in [17] achieved 93.18% accuracy in classifying vowel characters, but it used a very complex deep neural network model, whereas the model we proposed uses a very simple structure and the difference in accuracy is very small. Table 1 shows the experimental results obtained for all vowel characters.

Table 1. Obtained result of all vowel characters with accuracy, precision, recall and F1 scores from the experiment

Character	Accuracy	Precision	Recall	F1
অ	94	61.97	88	72.73
আ	92.36	55.56	80	65.57
ই	92.36	55.13	86	67.19
ঈ	92.73	58.33	70	63.64
উ	92.27	60	78	67.83
ঊ	90.73	49.23	64	55.65
ঋ	87.82	38.03	54	44.63
এ	92	53.57	90	67.16
ঐ	88.54	43.3	84	57.14
ও	87.64	40.63	78	53.42
ঔ	90.18	47.5	76	58.46

It can be observed from Table 1 that the letters which possess similar patterns to others (আ, ঊ, ঐ, ঔ) has a comparatively lower recognition accuracy. Even in real-world scenarios, different handwriting types lead to confusion, even to the human eye at times. With the exclusion of some misclassifications, machine learning does reduce this, given that significantly more training is done. A visual representation of the outcome of the experiment is presented in Figure 4, which shows the comparative scores of accuracy, precision, recall, and F1 scores obtained by the experiment for all eleven Bengali handwritten vowel characters. Furthermore, the training and testing accuracies of the model for 50 epochs are depicted in Figure 5.

Therefore, from the approach, it can also be deduced that Scikit-learn is a good library for image classification and clustering. It fully transforms the input data for the machine learning algorithm and compares the parameters. It is used in our feature extraction and normalization method for better prediction of the result. Overall, the SVM classification approach has shown great promise for recognizing vowel characters in Bengali handwriting. The efficiency of SVM in reading a dataset like ours shows excellent results with only a small amount of training required. It promises even better results with more supervised training. SVM implementation in Python is comparatively fast and makes the recognition task efficient. Although a very large dataset was not trained for this study, the first step with SVM on a character matrix dataset showed impressive results. Table 2 shows the comparative results with some modern classification techniques.

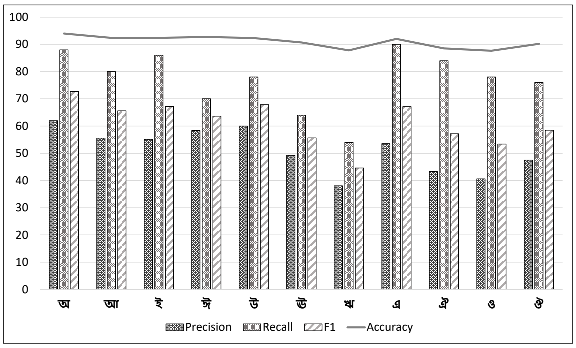


Figure 4. Comparative scores of accuracy, precision, recall and F1 metrics for eleven handwritten bengali vowel characters, from left to right, each bar represents precision, recall and F1 scores, the blue line represents the accuracy score

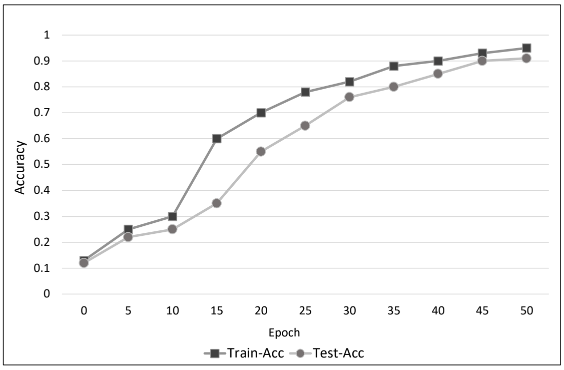


Figure 5. Training and testing accuracy of the proposed model

Table 2. Comparison of the proposed approach with some existing studies

Study	Classification method	Accuracy
Rahman *et al.* [35]	CNN	85.96%
Das *et al.* [36]	MLP	85.40%
Bhowmik *et al.* [15]	SVM	89.22%
Das *et al.* [2]	MLP	79.25%
Roy *et al.* [37]	DCNN	90.33%
Rahman *et al.* [38]	BWS + FWS+TMS +MLP+MPC	88.38%
Proposed approach	SVM with matrix mapping	91%

5. CONCLUSION

This paper describes an approach to implement SVM in Python through minimal training for handwritten character recognition tasks. This system provides a model for recognizing Bengali vowels that can be used efficiently for any other language. The new strategy has provided excellent results and seems promising. It has an accuracy of up to 94% for a single character and an average of 91% for all characters. It is expected that the research will provide positive insight into the few concepts involved and lead to advances in the field. This is an area of great interest at present, and a large number of researchers are already working on it. This method provides good results compared to existing methods of Bengali handwriting recognition and is more efficient. In the future, we aim to improve further and develop new perspectives. Our current research is limited to vowel characters, but this method can be improved to recognize overwritten or compound characters, words, sentences, and even entire documents. It has been shown that selecting appropriate feature extraction and classification methods plays a crucial role in the performance of similar systems. We plan to become more efficient in generating results and showing our model's compatibility in handwriting recognition. In the future, we will try to make this system more precise to achieve higher accuracy.

REFERENCES

[1] B. Sarma, K. Mehrotra, R. Krishna Naik, S. R. M. Prasanna, S. Belhe, and C. Mahanta, “Handwritten assamese numeral recognizer using hmm amp; svm classifiers,” in *2013 National Conference on Communications (NCC)*, 2013, pp. 1–5, doi: 10.1109/NCC.2013.6488009.

[2] N. Das, B. Das, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, “Handwritten bangla basic and compound character recognition using MLP and SVM classifier,” *CoRR*, 2010. [Online]. Available: http://arxiv.org/abs/1002.4040.

[3] Y. Perwej and A. Chaturvedi, “Neural networks for handwritten english alphabet recognition,” *CoRR*, 2012. [Online]. Available: http://arxiv.org/abs/1205.3966.

[4] A. Majumdar and B. Chaudhuri, “Curvelet-based multi svm recognizer for offline handwritten bangla: A major indian script,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, pp. 491–495, doi: 10.1109/ICDAR.2007.4378758.

[5] T. I. Aziz, A. S. Rubel, M. S. Salekin, and R. Kushol, “Bangla handwritten numeral character recognition using directional pattern,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–5, doi: 10.1109/ICCITECHN.2017.8281820

[6] P. Das, T. Dasgupta, and S. Bhattacharya, “A bengali handwritten vowels recognition scheme based on the detection of structural anatomy of the characters,” in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications* 2018, pp. 245–252, doi: 10.1007/978-981-10-3373-5_24.

[7] R. Ghosh and P. P. Roy, “Study of two zone-based features for online bengali and devanagari character recognition,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 401–405, doi: 10.1109/ICDAR.2015.7333792.

[8] M. B. Islam, M. M. B. Azadi, M. A. Rahman, and M. M. A. Hashem, “Bengali handwritten character recognition using modified syntactic method,” in *2nd National Conference on Computer Processing of Bangla (NCCPB-2005)*, 2005.

[9] V. L. Sahu and B. Kubde, “Offline handwritten character recognition techniques using neural network: A review,” *International journal of science and Research (IJSR)*, vol. 2, no. 1, pp. 87–94, 2013.

[10] A. Pal, “Bengali handwritten numeric character recognition using denoising autoencoders,” in *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, 2015, pp. 1–6, doi: 10.1109/ICETECH.2015.7275002.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, “Scikit-learn: machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[12] M. Biswas, R. Islam, G. K. Shom, M. Shopon, N. Mohammed, S. Momen, and A. Abedin, “Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters,” *Data in brief*, vol. 12, pp. 103–107, 2017, doi: 10.1016/j.dib.2017.03.035.

[13] K. Maladkar, “6 types of artificial neural networks currently being used in machine learning,” Jan. 2018. [Online]. Available: https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/.

[14] D. S. Maitra, U. Bhattacharya, and S. K. Parui, “CNN based common approach to handwritten character recognition of multiple scripts,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1021–1025, doi: 10.1109/ICDAR.2015.7333916.

[15] T. K. Bhowmik, P. Ghanty, A. Roy, and S. K. Parui, “Svm-based hierarchical architectures for handwritten bangla character recognition,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 2, pp. 97–108, 2009, doi: 10.1007/s10032-009-0084-x.

[16] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, “A hierarchical approach to recognition of handwritten Bangla characters,” *Pattern Recognition*, vol. 42, no. 7, pp. 1467–1484, 2009, doi: 10.1016/j.patcog.2009.01.008.

[17] T. R. Das, S. Hasan, M. R. Jani, F. Tabassum, and M. I. Islam, “Bangla handwritten character recognition using extended convolutional neural network,” *Journal of Computer and Communications*, vol. 9, no. 3, pp. 158–171, 2021, doi: 10.4236/jcc.2021.93012.

[18] S. Reza, O. B. Amin, and M. Hashem, “Basic to compound: A novel transfer learning approach for bengali handwritten character recognition,” in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2019, pp. 1–5, doi: 10.1109/ICBSLP47725.2019.201522.

[19] L. Buitinck, *et al.*, “API design for machine learning software: experiences from the scikit-learn project,” *CoRR*, 2013. [Online]. Available: http://arxiv.org/abs/1309.0238.

[20] J. Briggs, “How to setup Python for machine learning,” Sep. 2021. [Online]. Available: https://towardsdatascience.com/how-to-setup-python-for-machine-learning-173cb25f0206.

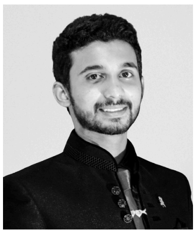
[21] J. VanderPlas, “Introducing Scikit-Learn,” *Python Data Science Handbook*, 2016.

[22] M. A. Karim, *Technical challenges and design issues in bangla language processing*, 2013.

[23] R. Ghosh, C. Vamshi, and P. Kumar, “RNN based online handwritten word recognition in devanagari and bengali

scripts using horizontal zoning," *Pattern Recognition*, vol. 92, pp. 203–218, 2019, doi: 10.1016/j.patcog.2019.03.030.
[24] B. Purkaystha, T. Datta, and M. S. Islam, "Bengali handwritten character recognition using deep convolutional neural network," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–5, doi: 10.1109/ICCITECHN.2017.8281853.
[25] J. Brownlee, "A gentle introduction to scikit-learn: A python machine learning library," Apr. 2014. [Online]. Available: https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/.
[26] A. Bhowmik and A. E. Chowdhury, "Genre of bangla music: a machine classification learning approach," *AIUB Journal of Science and Engineering (AJSE)*, vol. 18, no. 2, pp. 66–72, 2019, doi: 10.53799/ajse.v18i2.42.
[27] G. D. Luca, "Advantages and disadvantages of neural networks against SVMs," 2021. [Online]. Available: https://www.baeldung.com/cs/ml-ann-vs-svm.
[28] J. Cardoso-Fernandes, A. Teodoro, A. Lima, and E. Roda-Robles, "Evaluating the performance of support vector machines (svms) and random forest (rf) in li-pegmatite mapping: Preliminary results," in *Proceedings Volume 2889, High-Power Lasers: Solid State, Gas, Excimer, and Other Advanced Lasers*, 2019, p. 111560Q, doi: 10.1117/12.2532577.
[29] C. R. Harris, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.
[30] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, 2010, pp. 51–56, doi: 10.25080/Majora-92bf1922-00a.
[31] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
[32] X. Zhou, J. Li, C. Yang, and J. Hao, "Study on handwritten digit recognition using support vector machine," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 452, no. 4, p. 042194, doi: 10.1088/1757-899X/452/4/042194.
[33] K. P. Shung, "Accuracy, precision, recall or f1," 2018. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9.
[34] R. Azim, W. Rahman, and M. F. Karim, "Bangla hand-written character recognition using support vector machine," *International Journal of Engineering Works*, vol. 3, no. 6, pp. 36–46, 2016, doi: 10.5281/zenodo.60329.
[35] M. M. Rahman, M. Akhand, S. Islam, P. C. Shill, and M. H. Rahman, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 8, pp. 42–49, 2015, doi: 10.5815/ijigsp.2015.08.05.
[36] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "An improved feature descriptor for recognition of handwritten bangla alphabet," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1501.05497.
[37] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated bangla compound character recognition: A new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, pp. 15–21, 2017, doi: 10.1016/j.patrec.2017.03.004.
[38] A. F. R. Rahman, R. Rahman, and M. C. Fairhurst, "Recognition of handwritten bengali characters: a novel multistage approach," *Pattern Recognition*, vol. 35, no. 5, pp. 997–1006, 2002, doi: 10.1016/S0031-3203(01)00089-9.

BIOGRAPHIES OF AUTHORS



Shahrukh Ahsan has completed his Bachelor of Science in Computer Science and Engineering at American International University Bangladesh (AIUB), Dhaka, Bangladesh. He is currently working as Operations Executive at Battery Low Interactive, Dhaka, Bangladesh. His research interest includes Computer Vision, Image Processing, Data Mining, Natural Language Processing and Machine Learning. He can be contacted at email: shahrukh.ahsan2009@gmail.com.

Shah Tarik Nawaz has completed his Bachelor of Science in Computer Science and Engineering at American International University Bangladesh (AIUB), Dhaka, Bangladesh. He is currently working as Software Quality Assurance Engineer at EchoLogyx Ltd, Dhaka, Bangladesh. His research interest includes Computer Vision, Image Processing, Data Mining, Natural Language Processing, Machine Learning, and Human Computer Interaction. He can be contacted at email: tariknz17@gmail.com.

Talha Bin Sarwar received his Master of Science in Computer Science (MScCS) from American International University - Bangladesh (AIUB), Dhaka, Bangladesh. He likewise received his Bachelor of Science in Computer Science and Engineering (BSc in CSE) from American International University - Bangladesh (AIUB), Dhaka, Bangladesh with academic distinguished honor Magna Cum Laude. He was mentioned in the Dean's List honor as well. After the consummation of MScCS, he joined as a lecturer in the Department of Computer Science, American International University – Bangladesh (AIUB). His research interest includes Natural Language Processing, Data Mining, and Machine Learning. He can be contacted at email: talhasarwar40@gmail.com.

M. Saef Ullah Miah is a PhD candidate at University Malaysia Pahang (UMP) and is currently working as a Graduate Research Assistant at the Faculty of Computing, UMP. He was an assistant professor in the Department of Computer Science, American International University-Bangladesh (AIUB). He is currently engaged in research and teaching activities and has practical experience in software development and project management. He earned his Master of Science and Bachelor of Science degrees from AIUB. In addition to his professional activities, he is passionate about working on various open source projects. His main research interests are data and text mining, natural language processing, machine learning, material informatics and blockchain applications.He can be contacted at email: md.saefullah@gmail.com.

Abhijit Bhowmik Completed his B.Sc. in Computer Science & Engineering in 2009 and M.Sc. in Computer Science in 2011 from the American International University– Bangladesh (AIUB). Currently he is pursuing his PHD degree from University Malaysia Pahang in NLP and Machine Learning. He is working as Associate Professor and Special Assistant, Office of Student Affairs (OSA) in the Department of Computer Science, AIUB. His research interests include NLP, Machine Learning, wireless sensor networks, video on demand, software engineering, mobile & multimedia communication, and data mining. Mr. Bhowmik can be contacted at abhijit@aiub.edu.