

Project Report
On
"Deep Learning Approach on Writer Recognition"

(A dissertation submitted in partial fulfilment of the requirements of Bachelor of Technology in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology, West Bengal)

Submitted by

Sandip Pal
Payel Saren
Rajashri Mondal
Subhasis kundu
Biswaraj Majumder
Pragya Das

Under the guidance of

Prof. Jaya Paul
Ass. Prof.,
Dept. of Computer Science and Engineering

**Government College of Engineering and Leather
Technology**

(Affiliated to MAKAUT, West Bengal)

Kolkata - 700106, WB

2022-2023

Certificate of Approval

This is to certify that the project report on "**Deep Learning Approach on Writer Recognition**" is a record of bonafide work, carried out by Sandip Pal, Payel Saren, Rajashri Mondal, Subhasis Kundu, Biswaraj Majumder and Pragya Das under my guidance and supervision.

In my opinion, the report in its present form is in conformity as specified by Government College of Engineering and Leather Technology and as per regulations of the Maulana Abul Kalam Azad University of Technology, West Bengal. To the best of my knowledge the results presented here are original in nature and worthy of incorporation in project report for the B.Tech. Program in Computer Science and Engineering.

Signature of
Supervisor/ Guide

Signature of
Head, Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure, I would like to express my profound gratitude and indebtedness to Prof. Jaya Paul, Department of Computer Science and Engineering, Government College of Engineering and Leather Technology, W.B. for his continuous guidance, valuable advice, and constant encouragement throughout the project work. Her valuable and constructive suggestions at many difficult situations are immensely acknowledged. I am in short of words to express her contribution to this thesis through criticism, suggestions, and discussions.

I would like to take this opportunity to thank Dr. Surjadeep Sarkar, Project Coordinator and Prof. Santanu Halder, HOD, Department of Computer Science & Engineering and Information Technology, Government College of Engineering and Leather Technology.

I would like to express my gratitude to Prof. Jaya Paul for her valuable suggestions and help.

Signatures

1. Sandip Pal – 11200119003 – CSE
2. Payel Saren – 11200119015 – CSE
3. Rajashri Mondal – 11200119023 – CSE
4. Subhasis kundu – 11200119039 – CSE
5. Biswaraj Majumder – 11200119041 – CSE
6. Pragya Das – 11200119045 -CSE

Dedicated to

Ada Lovelace, Alan Turing and John Nash

The pioneers on whose work we expand upon.

OBJECTIVE

Significant amount of research has been done in the field of handwriting recognition, particularly for characters in the Latin-based alphabets (English, French, Spanish, German, etc). However, there is a significant shortage of literature and research on handwriting recognition for Devanagari based languages, such as Hindi, Bangla, Sanskrit, etc. Our project focuses on applying the VGG16 model, which is a convolutional neural network that has been widely used in image recognition tasks, to identify the writer of Bangla script. This is a challenging task, as writing styles can vary significantly from person to person, making it difficult to accurately identify the writer of a particular text. To tackle this challenge, we will train the VGG16 model on a dataset of Bangla text written by different individuals, which will enable the model to learn the unique characteristics of each writer's style. We will then test the accuracy of the model on a separate dataset.s

CONTENTS

CHAPTER 1.	INTRODUCTION	7-9
1.1.	Overview	8
1.2.	Literature Survey	9
CHAPTER 2.	METHODOLOGY	10-16
2.1.	Summary of Present Work	11
2.2.	Dataset	12-13
2.3.	Configure the Model	13-15
2.4.	Hardware/Software	16
CHAPTER 3.	OUTPUT AND RESULTS	17-19
CONCLUSIONS		20
REFERENCES		21

CHAPTER 1. INTRODUCTION

1.1. Overview

Writer recognition involves finding the author of a query document given a reference base with documents of known writers. Writer recognition is generally carried out by calculating a similarity index between the questioned writing and all the writings of known writers and sorting the retrieved results in a hit list with an increasing distance from the query. Writer recognition will also help accelerate the field of forensic analysis, and in turn help the law enforcement authorities. Notes found on crime scenes and related to victims and suspects can be analyzed and such analysis can help us identify perpetrators of a crime.

Also useful in the field of education and academia, handwriting analysis can help us curb plagiarism. Plagiarism checking is a major field of research in academia and handwriting analysis can also help support those efforts. These are only a few of the applications of handwriting recognition that we drove us to choose this topic for our project. Recognition and analysis of handwriting has applications in various fields such as archaeology, criminal detection, academia, education, etc. However, so far handwriting analysis has only been performed by human hands. In modern days, handwriting recognition has mostly only been attempted for languages based on Latin-based alphabet. Literature related to handwriting recognition is limited for Devanagari related languages, such as Hindi and Bangla. Hence, our feeble attempt at remedying that.

1.2. Literature Survey

A survey based on a model related to the project was conducted. VGG 16 was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab [2] at the University of Oxford in 2014 in an article titled "Very Deep Networks for Large Scale Image Recognition." This model achieves 92.7% on the top 5 accuracy tests with the ImageNet dataset of 14 million images belonging to 1000 categories. The model inputs 224 x 224 pixels size images in RGB channels. To understand how handwriting can be treated as images, we looked into articles on handwritten character recognition. We referred to the following works specifically. Hasnat et al.[5] proposed a domain specific OCR which classify machine printed as well as handwritten Bangla characters. For feature extraction they apply Discrete Cosine Transform (DCT) technique over the input image and for classification Hidden Markov Model (HMM) was used [5]. Paul et al. has attempted Bangla character recognition with Mobilenet v1 and Inception v3 [6] and Bangla number recognition with Convolutional Neural Networks [7]. While the works are different from what we are trying achieve, they help us find an appropriate way to approach handwriting.

We also referred to the following works. Christlein et al. [8] attempted author recognition with Convolutional Neural Network. Schlapbach et al. [9] analyzed HMM based handwriting recognition systems and studied the effect of normalization operations. Wu et al. [10] attempted writer identification on English and Chinese languages. Several research studies have been conducted on writer recognition using VGG16. One such study by Farooq et al. (2020) proposed a deep-learning-based mechanism for offline Urdu writer identification using VGG16. The researchers collected a dataset of 840 images from 84 Urdu writers and trained a VGG16 model with transfer learning. The results showed that the proposed method outperformed other state-of-the-art approaches.

CHAPTER 2. METHODOLOGY

2.1. Summary of Present Work

The writer recognition project was an intensive effort that required a multitude of steps to achieve the desired outcome. The Computer Science and Engineering (CSE) department provided us the necessary data for the project. This involved gathering handwriting samples from a diverse group of individuals, which could later be used for analysis and training. Once the data had been collected, it went through a process of pre-processing, where it was segmented into individual images. This process was vital to ensure that the dataset was properly formatted and ready for analysis. The segmentation process resulted in multiple folders containing image files that could be used to train the model. Configuration of the VGG16 model was the next critical step in the process. This involved setting up the model architecture, choosing the appropriate parameters, and ensuring that everything was correctly aligned with the dataset. The successful configuration of the model was essential to ensure accurate results during training. With the model configured and ready to go, the dataset was then used to train the model. This involved feeding the model with the pre-processed image files and allowing it to learn from the data. The training process was time-consuming and required significant computing power, but ultimately resulted in a more accurate and reliable model. Finally, after the model had been trained, it was time to obtain the desired result. This involved testing the model on new data, analyzing the results, and making adjustments as necessary. This final step was crucial to ensure that the model was capable of recognizing handwriting accurately and consistently.

2.2. Dataset

A major criterion of success in working on problems whose solutions depend on machine or deep learning is the presence of large datasets. The larger the dataset, the better and more accurate the model becomes. One of the main challenges of attempting handwriting recognition in a new language is the lack of sufficient datasets. We have been provided the datasets by our CSE Department. During the dataset collection process, we have collected eight attributes which include the name, gender, age, hand, set number, date, time, and native language. In this project, the dataset consists of 100 writers. There are total 5 sets of writers and the ratio of training to testing dataset is 3:2. This means that for every 3 data points in the training set, we have used 2 data points in the testing set. This provided us with the necessary dataset needed for proper model training. For our project, we have used the dataset containing Bangla words only. Datasets are colored.

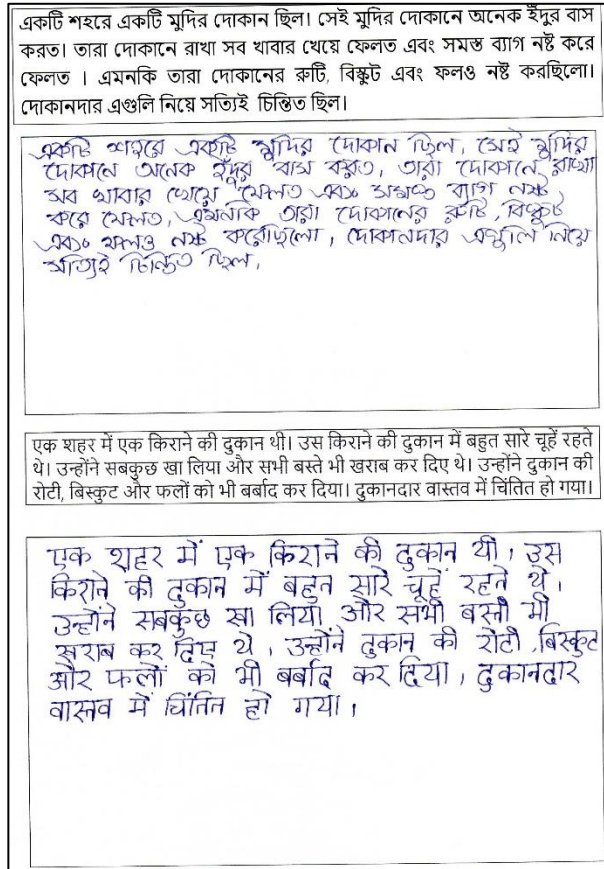


Fig: Dataset

Preprocessing: The handwritten passages were scanned into image. The scanned images passages were processed to eliminate noise and then segmented into word-sized images. These word sized images are then arranged into several folders in the following method.

- The images are distinguished according to pairs of authors and for each author, there are five (5) sets of data.
- Each of these 5 sets are divided into two subsets; train containing three sets and test containing two sets.

- The images are in Tag Image File Format (TIFF). They are named in the following format:
<Author Code>_<Set Number>_<Image Number>. For example, the first image of the first set of the first author (Author Code 0) is 0000_01_0.tiff.
- There are five thusly organized datasets.

These images are now ready to be fed as inputs for the model.

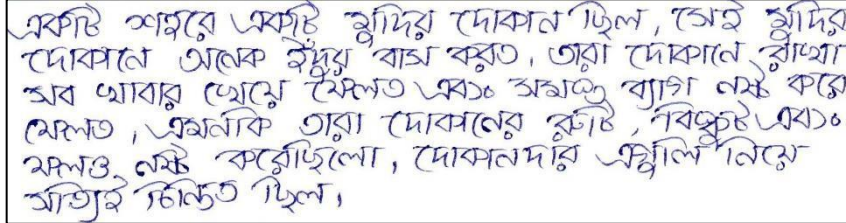


Fig.3. Segmented Paragraph of Writer 1

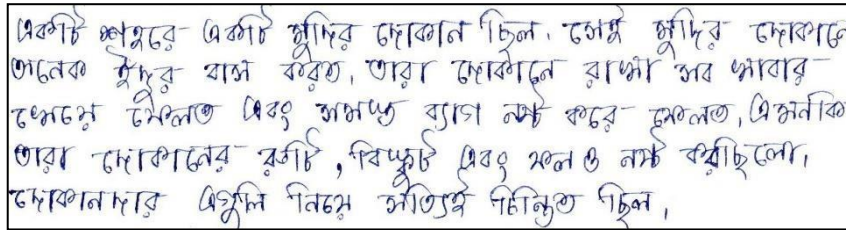


Fig.4. Segmented Paragraph of Writer 2



Fig.5. Word Segmented Image

2.3. Configure the Model

Architecture of VGG16 Model:

The input to the network is image of dimensions (224, 224, 3). The first two layers have 64 channels of 3*3 filter size and same padding. Then after a max pool layer of stride (2, 2), two layers which have convolution layers of 256 filter size and filter size (3, 3). This followed by a max pooling layer of stride (2, 2) which is same as previous layer. Then there are 2 convolution layers of filter size (3, 3) and 256 filter. After that there are 2 sets of 3 convolution layer and a max pool layer. Each have 512 filters of (3, 3) size with same padding. This image is then passed to the stack of two convolution layers. In these convolution and max pooling layers, the filters we use is of the size 3*3 instead of 11*11 in Alex Net and 7*7 in ZF-Net. In some of the layers, it also uses 1*1 pixel which is used to manipulate the number of input channels. There is a padding of 1-pixel (same padding) done after each convolution layer to prevent the spatial feature of the image [13].

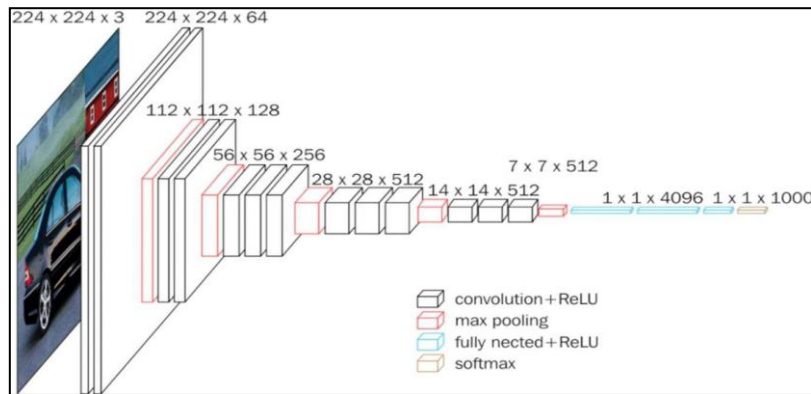


Fig.6. Architecture of VGG16 Model ([Source](#))

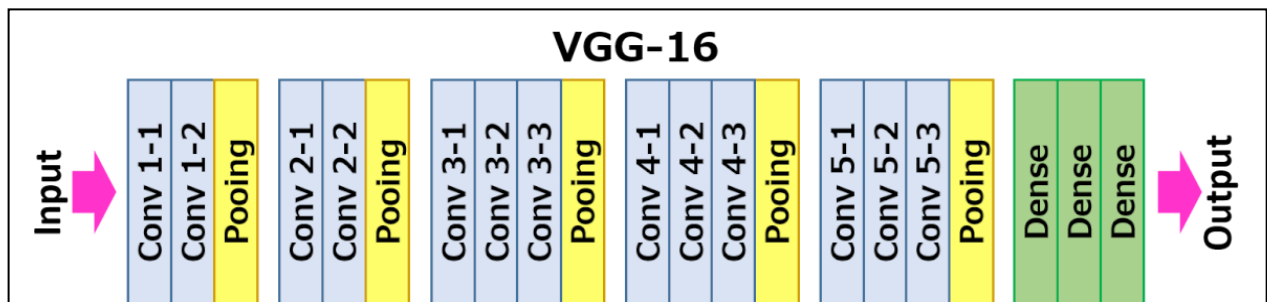


Fig.7. Flowchart of VGG16 Model ([Source](#))

After the stack of convolution and max-pooling layer, we got a (7, 7, 512) feature map. We flatten this output to make it a (1, 25088) feature vector. After this there are 3 fully connected layer, the first layer takes input from the last feature vector and outputs a (1, 4096) vector, second layer also outputs a vector of size (1, 4096) but the third layer output 1000 channels for 1000 classes of ILSVRC challenge, then after the output of 3rd fully connected layer is passed to SoftMax layer in order to normalize the classification vector. After the output of classification vector top-5 categories for evaluation. All the hidden layers use ReLU as its activation function. ReLU is more

computationally efficient because it results in faster learning and it also decreases the likelihood of vanishing gradient problem. This is how the pretrained VGG16 model works [13].

Some of the Challenges of VGG16 are ---

- It is very slow to train (the original VGG model was trained on NVidia Titan GPU for 2-3 weeks).
- The size of VGG16 trained ImageNet weights is 528 MB. So, it takes quite a lot of disk space and bandwidth that makes it inefficient.

Model: “VGG16”

Layer (type)	Output Shape	Param #
Input_1(InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool(MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None,128)	73856
block2_conv2 (Conv2D)	(None, None, None,128)	147584
block2_pool(MaxPooling2D)	(None, None, None,128)	0
block3_conv1 (Conv2D)	(None, None, None,256)	295168
block3_conv2 (Conv2D)	(None, None, None,256)	590080
block3_conv3 (Conv2D)	(None, None, None,256)	590080
block3_pool(MaxPooling2D)	(None, None, None,256)	0
block4_conv1 (Conv2D)	(None, None, None,512)	1180160
block4_conv2 (Conv2D)	(None, None, None,512)	2359808
block4_conv3 (Conv2D)	(None, None, None,512)	2359808
block4_pool(MaxPooling2D)	(None, None, None,512)	0
block5_conv1 (Conv2D)	(None, None, None,512)	2359808
block5_conv2 (Conv2D)	(None, None, None,512)	2359808
block5_conv3 (Conv2D)	(None, None, None,512)	2359808
block5_pool(MaxPooling2D)	(None, None, None,512)	0

The total number of parameters in this model is 14,714,688. Out of these, all 14,714,688 parameters are trainable. There are no non-trainable parameters. The input image size for the model is (244,244,3), while the output size is (1,244,244,3).

2.4. Hardware / Software Used

Software:

Primary language used for programming is **Python 3.8**.

Packages used in Python are **TensorFlow v2.0** (a deep learning framework by Google, Inc.), **Keras** (to help with integrating TensorFlow with Python), **NumPy** (NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays), **Pandas** (software library written for the Python programming language for data manipulation and analysis), **Matplotlib** (used for plotting data and graphs) and **Seaborn** (Python data visualization library based on matplotlib; It provides a high-level interface for drawing attractive and informative statistical graphics).

Hardware:

The primary hardware used comprise the personal laptop computers belonging to the team members; HP Pavilion with 2.7GHz Quad-Core Intel i5, Integrated Graphics Card, 8GB RAM; Dell G3 with 2.6GHz Hexa-Core Intel i7 Processor, Integrated Graphics Card, 8GB RAM and Asus Vivo book 2GHz Quad-Core AMD Ryzen 5 Processor, Integrated Graphics Card, 8GB RAM.

CHAPTER 3. OUTPUT AND RESULTS

The results obtained from 50 writers' dataset:

Writer No.	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
Writer1	58.71%	68.4%	54.07%	62.77%
Writer2	70.02%	60.2%	57.36%	75.80%
Writer3	69.25%	55.5%	62.34%	80.56%
Writer4	72.03%	69.8%	71.08%	55.5%
Writer5	62.55%	70.3%	60.2%	77.03%
Writer6	75.98%	58.3%	52.03%	75.80%
Writer7	81.36%	52.6%	71.08%	57.36%
Writer8	60.74%	72.4%	57.36%	60.25%
Writer9	82.34%	57.9%	52.03%	80.56%
Writer10	69.25%	55.5%	70.3%	80.56%
Writer11	72.03%	69.8%	71.08%	75.80%
Writer12	82.34%	60.2%	54.07%	69.8%
Writer13	75.98%	52.6%	52.03%	77.03%
Writer14	81.36%	58.3%	62.34%	80.56%
Writer15	68.71%	57.9%	60.25%	75.80%
Writer16	62.55%	70.3%	52.03%	55.5%
Writer17	69.25%	72.4%	71.08%	62.77%
Writer18	60.74%	58.3%	57.36%	70.3%
Writer19	70.02%	72.4%	62.34%	80.56%
Writer20	82.34%	52.6%	55.5%	77.03%
Writer21	62.55%	69.8%	50.02%	75.80%
Writer22	72.03%	55.5%	69.8%	62.77%
Writer23	81.36%	70.3%	50.02%	60.25%
Writer24	69.25%	60.2%	71.08%	70.3%
Writer25	82.34%	72.4%	70.3%	69.8%
Writer26	75.98%	58.3%	62.34%	77.03%
Writer27	60.74%	57.9%	60.25%	80.56%
Writer28	72.03%	69.8%	54.07%	57.36%
Writer29	68.71%	72.4%	50.02%	70.3%
Writer30	81.36%	52.6%	71.08%	62.77%
Writer31	62.55%	58.3%	55.5%	57.36%
Writer32	82.34%	57.9%	50.02%	60.25%
Writer33	75.98%	69.8%	69.8%	77.03%
Writer34	82.34%	60.2%	54.07%	75.80%
Writer35	69.25%	72.4%	62.34%	57.36%
Writer36	81.36%	52.6%	70.3%	69.8%
Writer37	72.03%	72.4%	71.08%	62.77%
Writer38	62.55%	58.3%	57.36%	77.03%
Writer39	82.34%	57.9%	60.25%	80.56%
Writer40	68.71%	58.3%	50.02%	55.5%
Writer41	81.36%	72.4%	62.34%	77.03%
Writer42	75.98%	55.5%	54.07%	70.3%
Writer43	82.34%	57.9%	71.08%	57.36%
Writer44	70.02%	69.8%	55.5%	75.80%
Writer45	69.25%	72.4%	50.02%	69.8%
Writer46	72.03%	52.6%	71.08%	62.77%
Writer47	68.71%	69.8%	57.36%	80.56%
Writer48	60.74%	58.3%	62.34%	75.80%

Writer49	70.02%	55.5%	60.20%	70.3%
Writer50	58.65%	67.9%	53.34%	62.79%

We got the average accuracy of the model after training on 50 writers' dataset. The model was trained for 10 epochs with a batch size of 20, taking 3.5 hours to complete. The average training loss was 68.71%, and the average training accuracy was 58.04%. The model's average validation loss was 64.76%, and the average validation accuracy was 54.09%. After obtaining the final accuracy, we found it to be 62.75%.

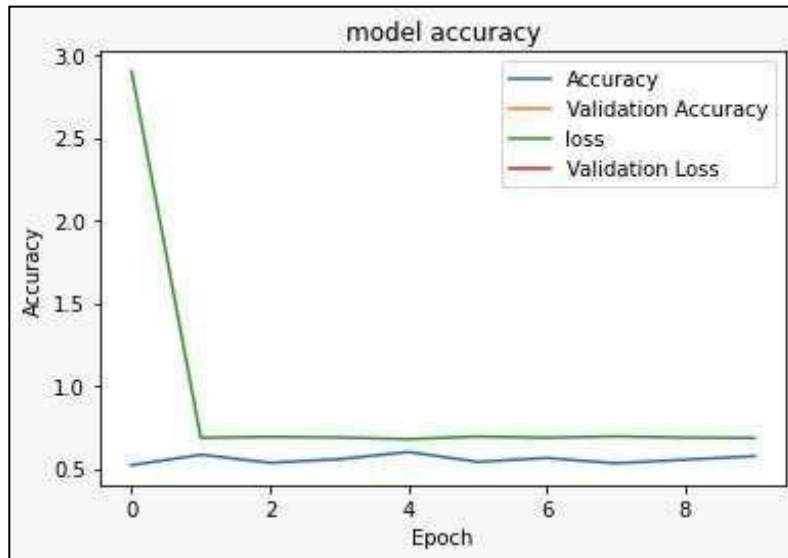


Fig. Graph depicting accuracy, validation accuracy, loss and validation loss over multiple epochs

CONCLUSIONS

The project is achieved through the conventional neural network. This algorithm will provide both the efficiency and effective result for the recognition. The project gives stratify outcome for the text which has less noise. The outcome is completely depending on the dataset. Although some progress was made, our goal remains unfulfilled. To achieve better results, we propose exploring innovative methods such as increasing the quantity of training data, optimizing algorithms utilized in the project, and fine-tuning parameters to enhance accuracy and efficiency. The future scope of a project on writer recognition using deep learning is quite promising, with several potential avenues for exploration and advancement. Here are some areas of future development in this field: Improved Accuracy, Multi-lingual Recognition, Historical Document Analysis, Forgery and Fraud Detection, Online Authenticity Verification, Forensic Applications, Personalized Content Generation, Real-time Recognition etc.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep Convolutional Neural Networks”, *Communications of the ACM*, Vol. 60 Issue 6, pp. 84–90, Jun 2017, doi: 10.1145/3065386.
- [2] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, in *Proc. International Conference on Learning Representations (ICLR 2015)*, 2015 [Online], Available: <http://arxiv.org/abs/1409.1556>.
- [3] A. Rehman, S. Naz, M. I. Razzak and I. A. Hameed, "Automatic Visual Features for Writer Identification: A Deep Learning Approach," *IEEE Access*, Vol. 7, 2019, pp. 17149-17157, Jan. 21, 2019, doi: 10.1109/ACCESS.2018.2890810.
- [4] J. John, Pramod K. V. and K. Balakrishnan, “Handwritten Character Recognition of South Indian Scripts: A Review”, in *Proc. National Conference on Indian Language Computing*, Feb. 19-20, 2011.
- [5] M. A. Hasnat, S. M. Habib, M. Khan, Eds., “A High-Performance Domain Specific Ocr For Bangla Script”, *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*, Dordrecht: Springer, 2008, doi: 10.1007/978-1-4020-8737-0_31.
- [6] J. Paul, A. Roy and A. Sarkar, “Bangla character recognition based on Mobilenet v1 and Inception v3”, in *Proc. International Conference on Emerging Technologies for Sustainable Development (ICETSD '19)*, Mar. 5-6, 2019, pp. 511-514.
- [7] J. Paul, A. Dattachaudhuri and A. Sarkar, “CNN implementation based on Bangla numeral character recognition”, in *Proc. International Conference on Emerging Technologies for Sustainable Development (ICETSD '19)*, Mar. 5-6, 2019, pp. 520-523.
- [8] V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, “Offline writer identification using convolutional neural network activation features,” In *Proc. German Conf. Pattern Recognition*, 2015, pp. 540–552. DOI:10.1007/978-3-319-24947-6_45.
- [9] A. Schlapbach and H. Bunke, “Writer identification using an HMM-based handwriting recognition system: To normalize the input or not,” in *Proc. Conf. IGS, 2005*, pp. 138–142.
- [10] X. Wu, Y. Tang and W. Bu, "Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform," in *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 526- 536, March 2014, DOI: 10.1109/TIFS.2014.2301274.
- [11] C. Adak, B. B. Chaudhuri and M. Blumenstein, “An Empirical Study on Writer Identification and Verification from Intra-Variable Individual Handwriting”, *IEEE Access*, Vol. 7, 2021, pp. 24738-24758, Feb 18, 2019, doi: 10.1109/ACCESS.2019.2899908.
- [12] T. Mondal, S. A. Hossain, S. Mondal, R. Afroz and A. Hossain, “Preprocess the handwritten document image for preparing writer recognition”, Government College of Engineering and Leather Technology, Kolkata, India, Project Report, June 2020.
- [13] R. Thakur, “Step by step VGG16 implementation in Keras for beginners”, *towardsdatascience.com*, Aug. 6, 2019. [Online]. Available: <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c> [Accessed Mar. 3, 2021].
- [14] Kaggle, Inc., “Dogs vs. Cats: Create an algorithm to distinguish dogs from cats”, Kaggle, Inc., Available: <https://www.kaggle.com/c/dogs-vs-cats/data>.
- [15] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition”, in *Proc. IEEE conference on computer vision and pattern recognition 2016*, 2016, pp. 770-778, [Online], Available: <https://arxiv.org/abs/1512.03385>
- [16] D. Gershgorn, “The data that transformed AI research—and possibly the world”, *The Quartz*, para. 38, July 26, 2017, [Online], Available: <https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/>, [Accessed June, 7, 2021].
- [17] Stanford Vision Lab, Stanford University, Princeton University, “Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)”, *Stanford Vision Lab, Stanford University, Princeton University*, [Online], Available: <https://image-net.org/challenges/LSVRC/2012/results.html>

