# Bengali Handwritten Character Recognition using Transfer Learning

Raihan Tanvir
raihantanvir@outlook.com

Susmoy Kar
susmoy.sun16@gmail.com

## 1 Introduction

### 1.1 Definition

We propose a solution that uses state-of-the-art techniques in Deep Learning to tackle the problem of Bengali Handwritten Character Recognition (HCR). Our method uses lesser iterations to train than most other comparable methods. We employ Transfer Learning [1] on ResNet-50 [2], a state-of-the-art deep Convolutional Neural Network Model, pretrained on ImageNet dataset. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones. We also use other techniques like 'One Cycle Policy' [3] to ensure that our training occurs fast. This cyclic learning rate policy is meant to be applied over one entire learning cycle: e. After each cycle, the learning rate finder is reapplied to find new good values, and then fit another cycle, until no more training occurs; hence the name. We use the Ekush Dataset [4] for the evaluation of our technique.

### 1.2 Motivation

Bangla is one of the most spoken languages, ranked fifth in the world. It is also a significant language with a rich heritage; February 21st is announced as the International Mother Language Day by UNESCO to respect the language martyrs for the language in Bangladesh in 1952. Bangla is the second most spoken language in India and it's the first language of Bangladesh. More than 200 million people all over the world speak this language and it is the sixth most popular language in the world. Thus, proper recognition of Handwritten Bengali Characters is an important problem that has many noble applications like Handwritten Character Recognition (HCR), Optical Character Recognition (OCR), Word Recognition, etc.

### 1.3 Challenges

However, Bengali is also much more difficult to tackle in this regard than English. This is because apart from the basic set of characters, i.e. vowel and consonants, in Bengali script, there are conjunct-consonant characters as well which is formed by joining two or more basic characters. Many characters in Bengali resemble each other very closely, being differentiated only by a period or small line. Because of such morphological complexities and variance in the handwriting style, the performance of Bengali handwritten character recognition is comparatively quite lower than its English counterpart.

# 2 Related Works

## 2.1 Important Prior Works

In recent years, remarkable progress has been made in the domain of deep learning and its application. Because of this advancement, deep learning-based transfer learning approaches have become popular for character recognition tasks.

D. C. Ciresan et. al. [5] proposed a transfer learning approach for Latin and Chinese character recognition with convolutional neural networks. In this study, they trained CNN on Latin digits and transferred it for recognizing uppercase Latin letters. They conducted a similar analysis taking Chinese characters as source data and Latin characters as target data; also, vice-versa. This study found that using the transfer learning approach a target character classification task can be performed perfectly with minimal retraining.

In another similar research, A. K. Tushar et. al. [6] introduced a convolutional neural network-based transfer learning technique for Bengali, Hindi, and Urdu numeric characters. Here, they considered numerals of one language as the source data and numerals of another language as the target data. This research outcome depicts that a model pretrained on one numeric character type can perform well on another type of numeric characters with minimal retraining.

Akm Shahariar Azad Rabby et. al. [7] proposed a lightweight CNN model for classifying Bangla Handwriting Character, which contains 50 basic Bangla characters (11 vowels and 39 consonants). The model is a 13-layer convolutional neural network with 2 sub-layers. It uses the ADAM optimizer. To make the optimizer converge faster and closer to the global minimum of the loss function an automatic learning rate reduction method is used. For character recognition, the proposed BornoNet model gets 98%, 96.81%, 95.71%, and 96.40% validation accuracy respectively for CMATERdb, ISI, BanglaLekha-Isolated [8] dataset and mixed dataset.

## 2.2 Improvisation in the Proposed System

Although a lot of previous work has been done on this, improvements can still be achieved using recent advancements both in Deep Learning and also in model training procedures like hyperparameter tuning as, data augmentation, etc. To make improvements in the BHWCR task we used data augmentation to increase the variation in the dataset.

To tune the learning rate for training the network we employ cyclic learning [3]. Cyclic learning is a learning rate scheduling technique for (1) faster training of a network and (2) a finer understanding of the optimal learning rate. Cyclic learning rates have an effect on the model training process known somewhat fancifully as ”super convergence”.

# 3 Project Objectives

To solve our problem of Bengali handwritten character recognition we choose apply Transfer Learning [1] with Ekush dataset [4]. Traditional learning is isolated and occurs purely based on specific tasks, datasets and training separate isolated models on them. No knowledge is retained which can be transferred from one model to another. In transfer learning, you can leverage knowledge (features, weights etc.)

from previously trained models for training newer models and even tackle problems like having less data for the newer task.

To accomplish our goal the required tasks are enlisted below:

- **Data Augmentation:** Since we are using Ekush dataset [4], we don't need to perform any preprocessing. Yet, to increase variations in samples and to reduce the probability of overfitting we choose to apply data augmentation techniques such as rotation, flip random cropping etc.

- **Determining Appropriate Hypermeters:** Hyper-parameters such network architecture, batch size, number of epochs, learning rate etc. should be determine for optimal training of the network.

- **Initial Training:** To employ transfer learning only the fully connected layer at the top are kept unfrozen. We need to train that layer to learn the initial weights with our dataset.

- **Finding the Learning Rate:** An appropriate learning rate need to be determined from the previous training.

- **Retrain the Network:** Unfreeze all the layers of the network and retrain with whole network

- **Compute Accuracy Metrics:** Appropriate accuracy metrics need to be chose to measure the performance of the trained network.

**Sample Input and Output:** The input of the system is an $28 * 28$ image of a Bengali handwritten character and the output is the prediction of the class that the character belongs to 1. Some sample characters with their assigned class is shown in table: 1

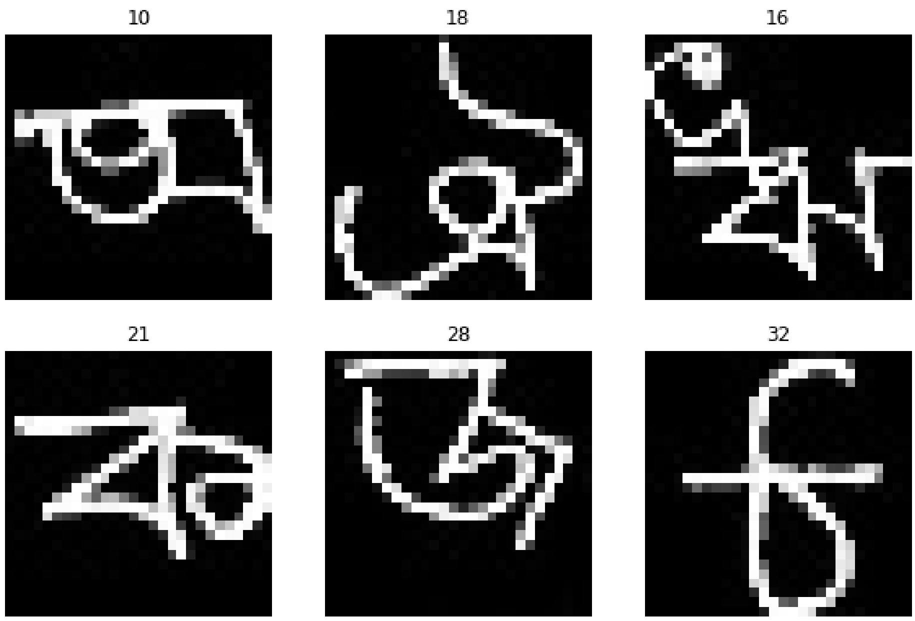


Figure 1: Sample input and output where input is a image of handwritten character and output is the class label

| Label | Char Name | Label | Char Name |
|---|---|---|---|
| 0 | া | 1 | ি |
| 10 | অ | 11 | আ |
| 12 | ই | 21 | ক |
| 22 | খ | 112 | ০ |
| 113 | ১ | 114 | ২ |

Table 1: Sample characters with assigned label

# 4 Methodologies

Transfer learning using a convolutional neural network (CNN) is a powerful approach to perform an image classification task. In CNN, the first few layers extract some generic feature of input data and the features extracted from the last few layers become more specific with the particular task. Therefore, in case of a new classification task, it is quite reliable to reuse the starting layers and train only the finishing layers.

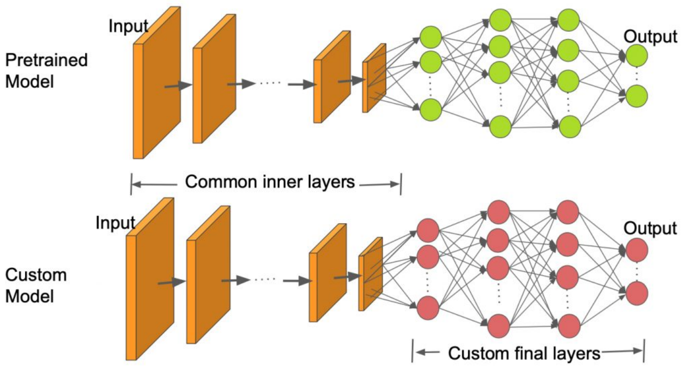


Figure 2: Transfer Learning

For accomplishing our goal of Bengali handwritten character recognition using transfer learning, we choose to use residual network architecture with 50 layers, pretrained on ImageNet dataset [9]. Residual Network [2] is arguably the most groundbreaking work in the computer vision/deep learning community in the last few years. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance. The core idea of ResNet is introducing a so-called “identity shortcut connection” that skips one or more layers 3. This ensures that the deeper model should not produce a training error higher than its shallower counterparts.

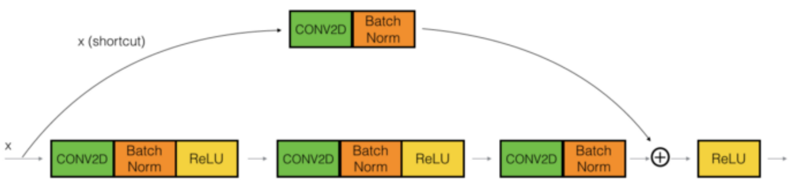


Figure 3: Skip connection between layers

After the final layer is trained, the other layers are unfrozen. To find the perfect learning rates for training the whole network learning rate finder method is used. Then we retrain the whole network. Finally, we evaluate the performance of our network using accuracy as metrics. We also determine the confusion metrics.

# 5 Experiments

## 5.1 Dataset

To employ our techniques, we use Ekush dataset [4]. Ekush the largest dataset of handwritten Bangla characters for research on handwritten Bangla character recognition. Dataset is separated according to the gender of the data provider. We use only the version formed with samples provided by male person since the whole dataset size is huge and it would take a quite long time to train the network. The statistics of the Ekush dataset are shown in 2:

| Basic | Compound | Modifiers | Numerals | Total | Total Class | Samples per Class |
|---|---|---|---|---|---|---|
| 154,824 | 150,840 | 30,667 | 30,687 | 367,018 | 122 | 1507 |

Table 2: Statistics of Ekush Dataset

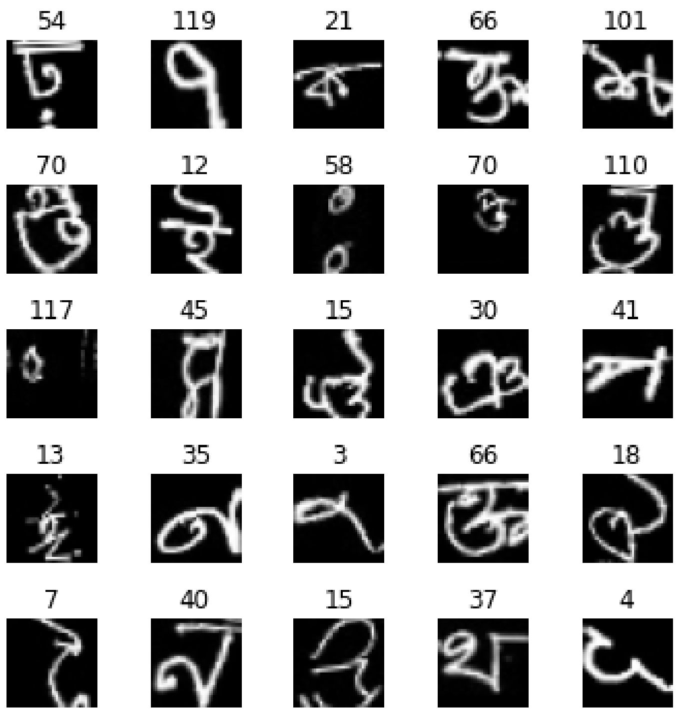


Figure 4: Samples from Dataset with Classes

The dataset is splitted into 0.8: 0.2 ratio. Eighty percent of the dataset is used to train the network and rest is used as dev set. The train set has 144232 images and test set has 36057 images.

## 5.2 Evaluation Metrics

We choose accuracy as evaluation metrics for our network. Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$Accuracy = TP + TN/TP + FP + FN + TN$$

We also calculate the confusion matrix. A confusion matrix displays counts of the True Positives, False Positives, True Negatives, and False Negatives produced by a model. Using a confusion matrix, we can get the values needed to compute the accuracy of a model. Since for model with too many classes confusion matrix yields more confusion, we also determine top losses or samples for which model is most confused.

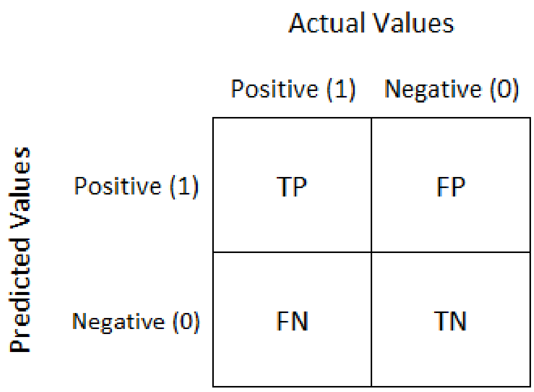


Figure 5: Confusion Matrix

## 5.3 Result

The model was trained initially for 30 epochs keeping the inner layers frozen. The training started with a loss of 3.925877 on the training set and 0.237929 accuracy was obtained. As the number of epochs increases the loss tends to decreases. And after the end of 30 epochs, the training loss reduced to 0.447053 and accuracy increased 0.901240. Losses and accuracies are shown in table:3

| epoch | train_loss | valid_loss | accuracy |
|---|---|---|---|
| 0 | 3.925877 | 3.331836 | 0.237929 |
| 1 | 3.086453 | 2.695943 | 0.339130 |
| 2 | 2.320579 | 1.913326 | 0.494273 |
| — | — | — | — |
| 27 | 0.438727 | 0.352028 | 0.900935 |
| 28 | 0.433550 | 0.348528 | 0.901878 |
| 29 | 0.447053 | 0.348920 | 0.901240 |

Table 3: :Loss and Accuracy for initial training

From the figure: 6 we can see that as the model observers more samples, it becomes more competent in classifying the characters. Initially, training loss is higher than validation loss. But as the number of iteration increases training loss tends to touch the validation loss and thus the training stops.

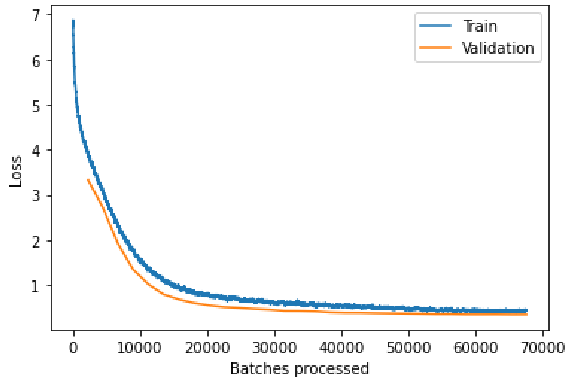


Figure 6: Loss graph for initial training

The appropriate learning rate for training the whole network is obtained using a learning rate finder method. The suitable learning rate is shown in the graph 7. This graph shows how the learning rate can affect the model's accuracy. We can see as the learning rate increases, so does the loss of our model. Here we are searching for the point with the steepest downward slope that still has a high value, which in this case is between 1e-6 and 1e-5.

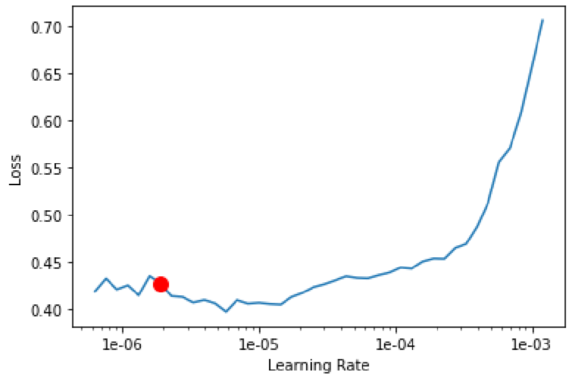


Figure 7: Finding appropriate learning rate

The whole network is trained with the newly found learning rate for another 20 epochs. We have now increased our accuracy to 91.3%.It's not perfect, but it's pretty good for the amount of work we put in. Losses and accuracies are shown in table: 4

| epoch | train_loss | valid_loss | accuracy |
|---|---|---|---|
| 0 | 0.442622 | 0.350906 | 0.901711 |
| 1 | 0.441207 | 0.345438 | 0.902626 |
| —— | —— | —— | —— |
| 17 | 0.318165 | 0.306372 | 0.914164 |
| 18 | 0.362462 | 0.303969 | 0.914469 |
| 19 | 0.323438 | 0.305791 | 0.913859 |

Table 4: :Loss and Accuracy for retraining

The confusion matrix for model with a lot of classes is hard to interpret. Thats why we determined most confused samples. From the figure 8 we get too see the samples for which loss is maximum. This allows us to examine the images our model was most confident it predicted correctly, but the model was wrong. This can happen due to the resemblance in the samples, mislabeling the samples.

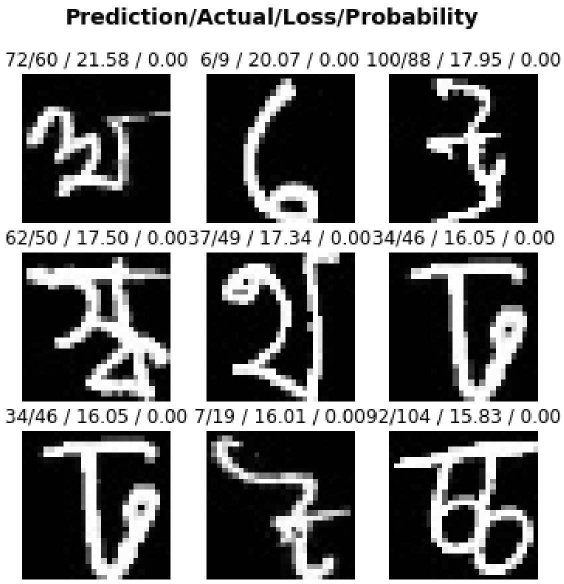


Figure 8: Top losses / Most confused prediction

# 6 Conclusion

The model has obtained an accuracy of 91.3% with just 50 epochs. Still, there is a lot of room from improvements, yet the outcome is quite satisfactory considering the amount of available resources. This performance shows the brilliance of transfer learning and how it can be used to solve many major problems in the domain of computer vision with ease.

# References

[1] Lisa Torrey and Jude Shavlik. Transfer learning.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *CoRR*, abs/1803.09820, 2018.

[4] AKM Shahariar Azad Rabby, Sadeka Haque, Md. Sanzidul Islam, Sheikh Abujar, and Syed Akhter Hossain. Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters. In K. C. Santosh and Ravindra S. Hegadi, editors, *Recent Trends in Image Processing and Pattern Recognition*, pages 149–158, Singapore, 2019. Springer Singapore.

[5] D. C. Cireşan, U. Meier, and J. Schmidhuber. Transfer learning for latin and chinese characters with deep neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2012.

[6] Abdul Kawsar Tushar, Akm Ashiquzzaman, Afia Afrin, and Md. Rashedul Islam. A novel transfer learning approach upon hindi, arabic, and bangla numerals using convolutional neural networks. In D. Jude Hemanth and S. Smys, editors, *Computational Vision and Bio Inspired Computing*, pages 972–981, Cham, 2018. Springer International Publishing.

[7] Akm Shahariar Azad Rabby, Sadeka Haque, Sanzidul Islam, Sheikh Abujar, and Syed Akhter Hossain. Bornonet: Bangla handwritten characters recognition using convolutional neural network. *Procedia Computer Science*, 143:528 – 535, 2018. 8th International Conference on Advances in Computing & Communications (ICACC-2018).

[8] Mithun Biswas, Rafiqul Islam, Gautam Kumar Shom, Md. Shopon, Nabeel Mohammed, Sifat Momen, and Anowarul Abedin. Banglalekha-isolated: A multipurpose comprehensive dataset of handwritten bangla isolated characters. *Data in Brief*, 12:103 – 107, 2017.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.