Accepted for publication in IEEE International Conference on Advances in Information Technology (ICAIT), ICAIT - 2019, Please refer IEEE Explore for final version

Transfer Learning using CNN for Handwritten Devanagari Character Recognition

Nagender Aneja* and Sandhya Aneja *
*Universiti Brunei Darussalam
Brunei Darussalam
{nagender.aneja, sandhya.aneja }@ubd.edu.bn

***Abstract*—This paper presents an analysis of pre-trained models to recognize handwritten Devanagari alphabets using transfer learning for Deep Convolution Neural Network (DCNN). This research implements AlexNet, DenseNet, Vgg, and Inception ConvNet as a fixed feature extractor. We implemented 15 epochs for each of AlexNet, DenseNet 121, DenseNet 201, Vgg 11, Vgg 16, Vgg 19, and Inception V3. Results show that Inception V3 performs better in terms of accuracy achieving 99% accuracy with average epoch time 16.3 minutes while AlexNet performs fastest with 2.2 minutes per epoch and achieving 98% accuracy.**

***Index Terms*—Deep Learning, CNN, Transfer Learning, Pre-trained, handwritten, recognition, Devanagari**

I. Introduction

Handwriting identification is one of the challenging research domains in computer vision. Handwriting character identification systems are useful for bank signatures, postal code recognition, and bank cheques, etc. Many researchers working in this area tend to use the same database and compare different techniques that either perform better concerning the accuracy, time, complexity, etc. However, the handwritten recognition in non-English is difficult due to complex shapes. Many Indian Languages like Hindi, Sanskrit, Nepali, Marathi, Sindhi, and Konkani uses Devanagari script. The alphabets of Devanagari Handwritten Character Dataset (DHCD) [1] is shown in Figure 1

Recognition of Devanagari is particularly challenging due to many groups of similar characters. For example, Figure 2 shows similarity is visible between character_04 and character_07; also between character_05 and character_13.

Several researchers have proposed handcrafted features for recognition of strokes. Building a robust set of features for Devanagari is challenging due to the complex shape of characters.

In recent years, Deep Learning has gained popularity in computer vision and pattern recognition primarily due to Deep Convolution Neural Network (DCNN) [2]. DCNN includes convolutional layers, fully connected layers, pooling layers for feature extraction, image representation, and dimensionality reduction respectively. DCNN also includes sigmoid or rectified linear units for non-linearity. CNN is based on local receptive fields and shared weights Goodfellow et al. [3]. The local receptive fields are regions that provide local information and are spread over the image. Each local receptive field constitutes a neuron and shares weights in the hidden layer.

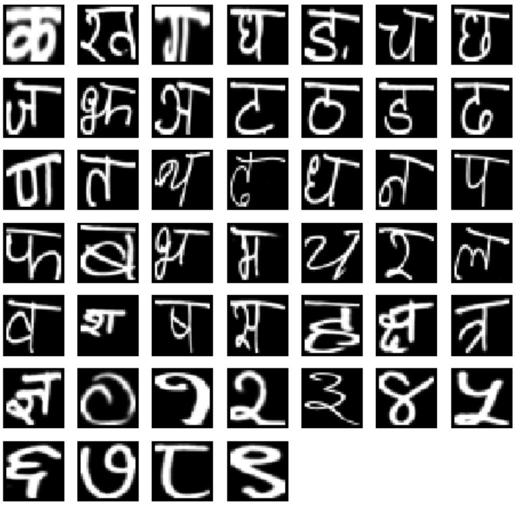

Fig. 1: Devanagari Alphabets

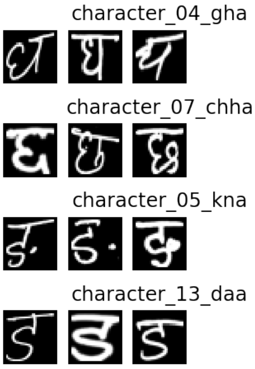

Fig. 2: Similar Characters

This helps to get distinct feature characteristics. Pooling layers that follow convolution layers, down samples and assist in overcoming overfitting by reducing the number of parameters. Additionally, dropout, wherein hidden neurons are randomly removed, also helps to avoid overfitting. Pre-trained CNN models are trained on [4] that carries a large number of data samples and 1000 class labels. The transfer learning techniques can help to apply the existing CNN models to problems where a large dataset is not available.

In this paper, we present Transfer learning, which is a deep learning technique that uses a pre-developed model for a task as a starting point for another task. Thus, it is a learning optimization technique that improves the performance of the second task. However, it only works if the model features learned from the first task are general. Transfer learning can be used either by (i) Develop Model Approach or (ii) Pre-trained Approach. In the develop model approach, the model is designed for one task and reused for the second task, however, in the pre-trained approach, one of the models available from different research organization is chosen as a starting point to tune it for the second problem.

This research compares various pre-trained models including [5, 6, 7, 8] in predicting Devanagari handwriting recognition. Following outlines are the main contributions of this paper:

a) Inception model representations outperform in comparison to others regarding accurate results for our dataset.
b) Vgg representations are also equally suitable for accurate results; however computationally expensive in comparison to Inception.
c) DenseNet representations did not show better accurate results but would be interesting to study over other datasets.
d) AlexNet is fastest and also provides reasonably good accuracy

II. Related Work

CNN based Deep Learning approaches have performed successfully in image processing; however, due to small data-size of Devanagari characters, the CNN models may overfit. The problem of training based on a small dataset has been solved by transfer learning.

Ng et al. [9] classified human emotions in images extracted from movies. The authors implemented transfer learning from models AlexNet and Vgg and reported 48.5% accuracy on the validation set.

Chakraborty et al. [10] analyzed the accuracy of the deep neural network for handwritten Devanagari characters. The authors implemented CNN of five different depths. Two Bidirectional Long Short Term Memory (BLSTM) layers were added between the convolutional layers and the fully connected part of the five CNN networks. The authors reported the accuracy of 96.09% with seven convolution layers and two fully connected layers.

Jangid and Srivastava [11] developed a deep convolutional neural network (DCNN) and adaptive gradient methods to identify handwritten Devanagari characters. The authors implemented six network architectures with different optimizers. The authors reported the accuracy of 96.02% using NA-6 architecture and RMSProp optimizer.

Boufenar et al. [12] addressed Handwritten Arabic Character Recognition (HACR) using AlexNet [13] transfer learning approach in Matlab. The authors used a batch size of 100 images and 200 epochs and achieved 86.96% accuracy. The authors obtained 99.33% accuracy by fine-tuning the model.

Akçay et al. [14] used pre-trained AlexNet [13] and GoogLeNet [15] to classify images for X-ray baggage security and achieved 98.92% accuracy.

III. Experimental Setup

Handwriting recognition is an image classification problem, and this paper proposes the transfer learning and convolution neural network for this task [16, 17]. In particular, instead of designing a new deep convolutional neural network with the random initialization, we exploited existing trained DCNN on 1000 objects as a starting point to identify Devanagari alphabets. We implemented AlexNet [5]; DenseNet 121 and DenseNet 201 [6]; Vgg 11, Vgg 16, and Vgg 19 [7]; and Inception V3 [8]. Training an entire DCNN with random initialization is highly computationally intensive and may not be practically feasible due to a limited dataset. Transfer Learning allows using the pretrained ConvNet either as initialization or fixed feature extractor. The pre-trained models are generally trained on ImageNet that has around 1.2 million images with 1000 different categories.

Following are mainly two techniques that are used in transfer learning:

1) Finetuning: It includes instead of random initialization of weights of different layers, the network is initialized with a pretrained network, and the rest of the training takes place, as usual, using the new dataset
2) Fixed Feature Extractor: In this approach, the weights of different convolutional layers are frozen that acts as feature extractor while the weights of the fully connected layer are updated. Thus in this approach only fully connected layer is trained.

In some complex tasks, few upper convolutional layers are also trained in addition to fully connected layer. In this research, we used ConvNet as a fixed feature extractor.

A. Data Preparation

The dataset [1] of Handwritten Devanagari characters has 46 classes with 2000 images of each class. We partitioned the dataset of 92,000 images into a training set of 78,200 images (0.85) and a testing set of 13,800 (0.15). The images have a resolution of 32 by 32 in the grayscale png format with the original character in the center. A padding of 2 pixels along sides was used in the images.

The dataset generated by Acharya et al. [1] had the handwriting of different people in wide variation that can be seen in Figure 3. The authors scanned handwritten documents

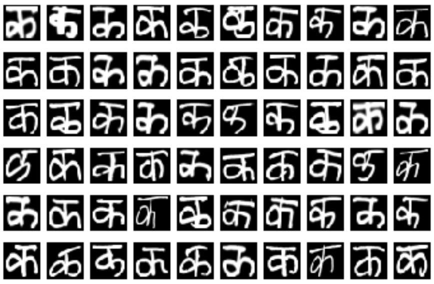

Fig. 3: Sample images to demonstrate variation of Handwriting

and cropped each character manually to create this dataset of handwritten characters.

We load the dataset into PyTorch tensors by resizing the images in size 224 by 224 and a normalizing using mean of (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225). We used PyTorch data generators to generate data in the batch size of 32.

B. *Architectures*

We used our base models on modern CNN architectures AlexNet, DenseNet, Vgg, and Inception. We first analyze the architecture of pre-trained models and their properties.

1) AlexNet: Krizhevsky [5] has five convolution layers, three pooling layers, and two fully-connected layers. It was ILSVRC 2012 winner [4] with a DCNN that outperformed other handcrafted features resulting in a top-5 error rate of 16.4%. This research extracted the features of convolution layers and trained the fully connected layers for Devanagari alphabets identification.

2) DenseNet: With a large number of layers, gradient washes out; however, accuracy is better with a large number of layers. Each layer features are input to subsequent layers so to preserve the information extraction. CNN architecture proposed by Huang et al. [6] is interesting and useful for many datasets.

3) Vgg: Simonyan and Zisserman [7] showed that the small size convolutions with a preferred 3x3 filter size for large depth CNN networks come up significantly accurate over CNN architectures proposed by Huang et al. [6] and Krizhevsky [5]. Whereas a large number of parameters due to small convolutions incurs a high computational cost in comparison to prior-art configurations.

4) Inception: Szegedy et al. [8] extended the Vgg model by improving the filter blocks of size 3x3, 3x1,1x3 and 1x1 to reduce the number of parameters consequently a high reduction in computational cost.

An overview of the comparison of the models used in this study is explained in Table I.

C. *Dataset and training*

Training a CNN model with a limited amount of data without overfitting is a challenging task. However, the proposed approach of transfer learning using pre-trained CNN tackled the problem with significant improvement in performance. We trained the model using PyTorch by freezing the parameters of the convolutional layers of the pre-trained model while training the fully connected layer called classifier with the number of classes 46. We retained the original structure of the pre-trained model. Table II shows hyperparameters used in the training and machine configuration.

IV. Results and Discussion

Table III summarizes the results. The results are explained regarding accuracy achieved in the first epoch, the best accuracy achieved in 15 epochs, epoch number in which we achieved the best accuracy, the total time taken by 15 epochs, average epoch time, and the number of features required by the model architecture.

Results show that Inception outperformed for our dataset with 99% accuracy in the first epoch with the average time 16.3 minutes due to regularization imposed by highest number of layers and smaller convolution filter sizes. However, the computational cost of Inception is lower than second best model Vgg11 which shows 99% accuracy in 45.6 minutes (time taken by eight epochs with average 5.7 minutes per epoch) as compared to one epoch of Inception in 16.3 minutes. The DenseNet model performed worst due to its architectural structure wherein the model feeds the connection between each layer to its subsequent layer. Since our dataset size is much smaller in comparison redundancy required for subsequent layers. AlexNet outperformed concerning computational cost with 98% accuracy in 6.6 minutes with three epochs.

References

[1] S. Acharya, A. K. Pant, and P. K. Gyawali, “Deep learning based large scale handwritten devanagari character recognition,” in *Software, Knowledge, Information Management and Applications (SKIMA), 2015 9th International Conference on.* IEEE, 2015, pp. 1–6.

[2] S. Aneja, N. Aneja, and M. S. Islam, “IoT Device Fingerprint using Deep Learning,” in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, Nov 2018, pp. 174–179.

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[4] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, “Imagenet large scale visual recognition competition (ilsvrc),” *See net.org/challenges/LSVRC*, 2012.

[5] A. Krizhevsky, “One weird trick for parallelizing convolutional neural networks,” *CoRR*, vol. abs/1404.5997, 2014. [Online]. Available: http://arxiv.org/abs/1404.5997

TABLE I: Comparison of Models

Model	No of Parameters (in million)	Top-1 Error	Top-5 Error	Depth	No of crops
AlexNet	60	37.5%	17.0%	8	nil
DenseNet 121	25	25.02%	7.71%	100	nil
DenseNet 201	20	22.58%	6.12%	190	nil
Vgg 11	134	24.7%	7.5%	11	1
Vgg 16	138	24.4%	7.1%	16	1
Vgg 19	144	23.7%	6.8%	19	1
Inception V3	$\leq$25	21.2%	5.6%	100	1

TABLE II: Parameters

Parameter Name	Values
Hyper Parameters	Batch Size = 32, Learning rate = 0.001, Momentum = 0.9, Step size=7, Gamma=0.1, Epochs = 15
Machine	i9 CPU, 64 GB RAM, Nvidia 1080 GPU, 1TB HDD
Operating System	Ubuntu 18
Implementation	PyTorch and Torchvision
Dataset	Training Files: 78200 and Testing Files: 13800
Image Size	32x32
Image Type	PNG

TABLE III: Comparison of Best Accuracy

Model	Valid Accuracy (in 1st epoch)	Best Accuracy (in 15 epochs)	Best Accuracy acheiveed in # epochs	Total Time (15 epochs)	Average Training Time per Epoch	No of in features
AlexNet	95	98	3	33m 8sec	2.2m	9216
DenseNet 121	73	89	7	80 m 3 s	5.3m	1024
DenseNet 201	74	90	6	113m 22s	7.6m	1920
Vgg 11	97	99	8	86m 6s	5.7m	4096
Vgg 16	97	98	3	132m 12s	8.8m	25088
Vgg 19	96	98	3	148m 57s	9.9m	25088
Inception V3	99	99	1	244m 36s	16.3m	2048

[6] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[9] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, “Deep learning for emotion recognition on small datasets using transfer learning,” in *Proceedings of the 2015 ACM on international conference on multimodal interaction*. ACM, 2015, pp. 443–449.

[10] B. Chakraborty, B. Shaw, J. Aich, U. Bhattacharya, and S. K. Parui, “Does deeper network lead to better accuracy: A case study on handwritten devanagari characters,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 411–416.

[11] M. Jangid and S. Srivastava, “Handwritten devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods,” *Journal of Imaging*, vol. 4, no. 2, p. 41, 2018.

[12] C. Boufenar, A. Kerboua, and M. Batouche, “Investigation on deep learning for off-line handwritten arabic character recognition,” *Cognitive Systems Research*, vol. 50, pp. 180–195, 2018.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] S. Akçay, M. E. Kundegorski, M. Devereux, and T. P. Breckon, “Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery.” IEEE, 2016.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[16] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.