Unconventional Wisdom: A New Transfer Learning Approach Applied to Bengali Numeral Classification

Conference Paper · August 2018

DOI: 10.1109/ICBSLP.2018.8554435

3 authors:

Hasib Zunair
Concordia University Montreal
10 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Nabeel Mohammed
North South University
40 PUBLICATIONS **213** CITATIONS
SEE PROFILE

Sifat Momen
North South University
37 PUBLICATIONS **146** CITATIONS
SEE PROFILE

Unconventional Wisdom: A New Transfer Learning Approach Applied to Bengali Numeral Classification

Hasib Zunair, Nabeel Mohammed, Sifat Momen
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
zunair.hasib@northsouth.edu.com, nabeel.mohammed@northsouth.edu, sifat.momen@northsouth.edu

***Abstract*—In this modern age, natural language processing (NLP) is evolving due to advances in the field of deep learning and its access to huge amount of data and computation power. Recently a lot of attention has been given to OCR for Bangla, the 5th most widely spoken language in the world. This paper reports on certain rather unconventional transfer learning approaches used to attain 6th place in the Kaggle Numta competition, where the challenge was to classify images of isolated Bangla numerals. The best result reported in this paper is an accuracy of 97.09% on the NumtaDB Bengali handwritten digit datasets test set, which was obtained by freezing intermediate layers. The unconventional approach used in this paper produces better results than conventional transfer learning while taking less epochs and having almost half the number of trainable parameters.**

Keywords—bengali digit classification, deep learning, convolution neural networks, transfer learning, data augmentation, keras, Numtadb

I. INTRODUCTION

Deep learning is widely used in image classification tasks. It is a part of machine learning, where the so called model, with underlying algorithms and optimizations exploit unknown structure when given an input to discover good representations often at multiple levels with high level features. Such kind of work is presented [1] where deep neural networks are used to predict on examples which were not in the training set. The model had to predict with foreign data – which it had never seen before. Another work shows the implementation of model which was a combination of different models combined– resulting in an ensemble. Since, ensembles are computationally expensive, a new approach was to combine the different characteristics of each model into one which showed astounding results on the MNIST data set. As written in [2] these models can be trained very quickly. Also, another work shows the investigation of the effect of convolutional neural networks on large scale image recognition [3]. The findings were based on the ImageNet Challenge for classification of around 1,000 different categories.

Convolutional neural nets have become a component in the state of the art solutions for computer vision problems and it is heavily used for classification problems. These models proved very powerful over the years. Since now, there is access to lots of data and lots of computation power –period. Due to enough training data and more computation power it is still a challenge in the case of mobile devices and big data [4]. One such work is shown, where a convolutional neural network is used for classification of 1.2 million high resolution images which has 1000 different classes [5]. Using many deep neural networks trained on natural images shows a distinct pattern: on the first layer they extract features. These are general features and in that they are applicable to different kinds of datasets and different tasks. Features translate from a generic pattern to a more distinct pattern as the network progresses from layer to layer. Hence, a study on the transferability of the generic features was done [6].

One such model which has been widely used in transfer learning is the VGG16 Deep CNN [7]. This VGG16 architecture achieved the state of the art accuracy in the ImageNet Challenge were a model had to classify around 1,000 different object categories in context. The model was trained with more than a million images which a person can see from day to day activity – mostly. One major drawback of the VGG16 is that it is slow and the model weight itself is large. But still it is used for most image classification tasks. Since, the pre-trained weights of the model are easily accessible; it has been used for different classification tasks using data set which are not in its inputs distribution. Generally, when a pre -trained model is used for transfer learning, its last layer – the classification layer – is discarded and a new classification layer is added. This last layer, quite often a Softmax layer, is then trained while the other layers stay frozen. This process is called fine-tuning and is the initial step of transfer learning. Once the Softmax layer is fine-tuned, if the performance is not satisfactory, as a second step of transfer learning the previous layers can be then also be trained in conjunction with the Softmax layer. In this paper, a VGG16 model which was pre-trained on Imagenet data was used for transfer learning.

There has been some interest recently in the application of deep learning techniques for Bangla numeral classification [10, 11, 12, 13 and 14]. However, these works do not employ transfer learning approaches and do not address the dataset gathered for the Kaggle numta challenge done on Bangla Numerals. The main contributions of the work reported in this paper are in using an Imagenet pre-trained model for an unconventional method of transfer learning applied to the Kaggle Numta challenge for classifying Bangla numerals. The requirements of the Kaggle competition was to classify Bangla hand written digits which is a multi-class classification problem and was to be tested on their challenging test set. In fact, the best results reported in this paper were achieved by freezing intermediate layers in the VGG16 model, thereby

978-1-5386-8207-4/18/$31.00 ©2018 IEEE

halving the number of trainable parameters. This result is particularly interesting because it was done in half the number of epochs as well, compared to the traditional transfer learning routine, while performing well on a very challenging test set.

II. EXPERIMENTAL OVERVIEW

A. *Block Diagram*

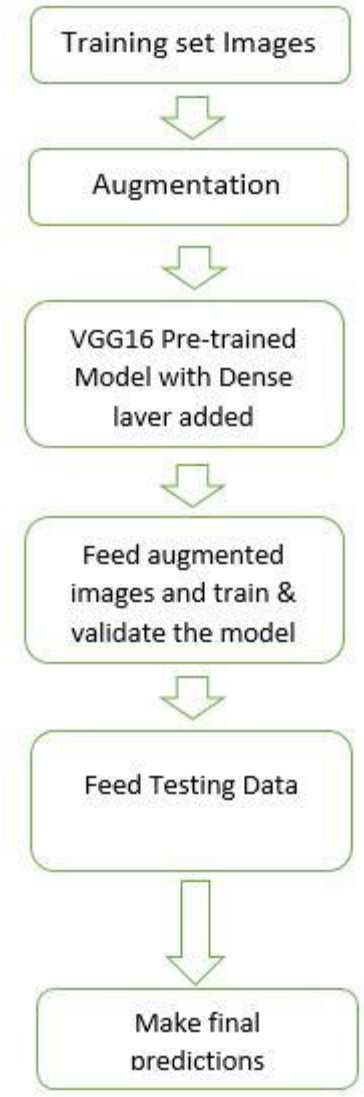

Fig 1: Block Diagram

Fig 1 portrays the whole working flow of the system. Initially, the training images are augmented which will be discussed later in the section, and then it is being fed to the VGG model and validated once training is done. When, the model is trained it is fed the testing set for evaluation the performance of the model. The final output shows the predicted label along with its true label and later a csv file is generated for the final submission of the models output.

B. Model Architecture

The model used here is based on the popular VGG16 architecture. Other model or combination of models (ensemble) could have been used, but due to computation constraints the pre-trained model seemed a good way to go. As the task is to classify the 10 Bangla numerals, the last layer of the VGG16 model was discarded and replaced with a Fully Connected layer of 10 neurons with the Softmax activation function. The model configuration is shown in Fig 2.

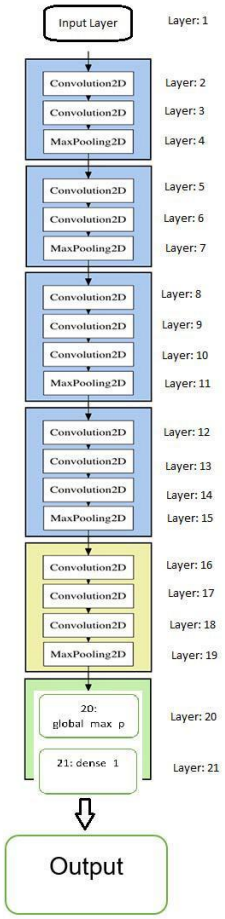

Fig 2: VGG Architecture

Fig 2 portrays the VGG architecture which consists of different convolutional layers with a max pooling layer within each block. The model also has a dense layer. Here, the model uses Adadelta as its optimizer. Different layer of the neural network extract different features from the image. The model is trained using backpropagation and Adadelta is used as the optimizer to maximize the loss function. The loss Categorical Crossentropy is used to compile the model. The equation is given by:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \epsilon C_c}\ log\ p_{model}[y_i \epsilon C_c] \qquad (1)$$

In (1), the double sum is over the observations *i*, whose number is *N*, and the categories *c*, whose number is *C*. The term 1_{y_i} in C_c is the indicator function of the *i*`th observation belonging to the *c*th category. The $p_{model}[y_i \text{ in } C_c]$ is the probability predicted by the model for the *i*th observation to belong to the *c* th category. When there are more than two categories, the neural network outputs a vector of *C* probabilities, each giving the probability that the network input should be classified as belonging to the respective category. When the number of categories is just two, the neural network outputs a single probability with the other one being *1* minus the output.

III. IMPLEMENTATION

This section portrays the final approach taken for the classification problem. Below, briefly discussed, are the key aspects of the whole model.

A. Dataset Description

The dataset, which is named NumtaDB, is a combination of multiple datasets including numerals from the previously published BanglaLekha-Isolated Dataset[9] . The data set is partitioned into a set of training images and also a set of testing images. The training images are divided into 5 different directories (A – E), each one sourced from a different existing dataset. The testing set is composed of 8 different directories. 6 of these are from existing resources (A-F) while the other two are heavily augmented versions of the existing test set A and test set C (Aug-A, and Aug-C). These last two additions make the classification task more challenging as the training set does not have any images representative of these types of augmentations. According to the NumtaDB dataset [8] the augmentations applied included changes in Brightness, Contrast, Saturation, Hue, Shift, Noise as well as addition of Occlusions and Superimposed images.

Few instances of the data set, both testing and training, is shown below.

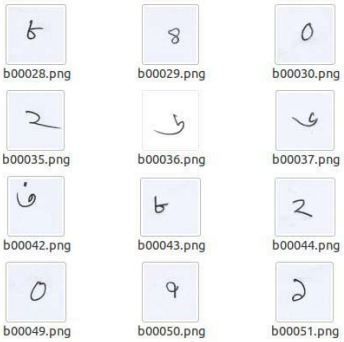

Fig 3: Samples from training set

Fig 3 shows an instance of the training set B. Here are samples of digits with minor variations.

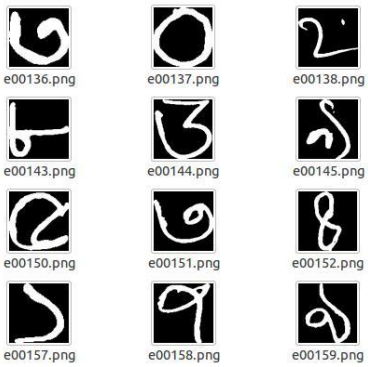

Fig 4: Samples from test set E

Fig 4 shows samples from the test set which is color inverted. And, also majority of the digits outer surface touch the boundary of the whole image space.

Fig 5: Samples from augmented test set

Fig 5 shows the testing data set which consists of heavy augmentations such as rotation, occlusions, blur and noise. There were two set like this which made the competition more challenging. Thus, concluding that the training set was much easier than the testing set in terms of recognition.

B. Data Preprocessing

As the data set images have variable size, they were all resized to 48 X 48 pixels. However, before the resize operation, the images were padded to make them square, this was to ensure that the character contained within the image retained its original aspect ratio.

C. Data Augmentation

Although the training set had close to 85,000 images, they were not representative of the test set. This is especially true for Test set. As the aim is to maximize test performance, certain image augmentation techniques were applied to the training data. These are summarized in Table 1. Change of color space and Gaussian Blur was performed with a probability of 0.5 and the rest were performed in randomized order on each image as portrayed below.

Table 1: Augmentation Parameters

Augmentation Type	Parameters
Contrast Normalization	(0.5, 1.5) increase/decrease in range
Crop	(0, 0.2) 0-20% of their height/width
Rotate	(-25, 25) degrees in range
Translate percent	x: (-0.2, 0.2) , y: (-0.2, 0.2) move by this range
Shear	(-25, 25)
Color Space	HSV to RGB
Gaussian Noise	(0.0, 0.05*255) Sample noise once per pixel

After augmentation, the initial training set of 85,000 images was increased to over 450,000. Fig 8 shows some examples of augmented images.

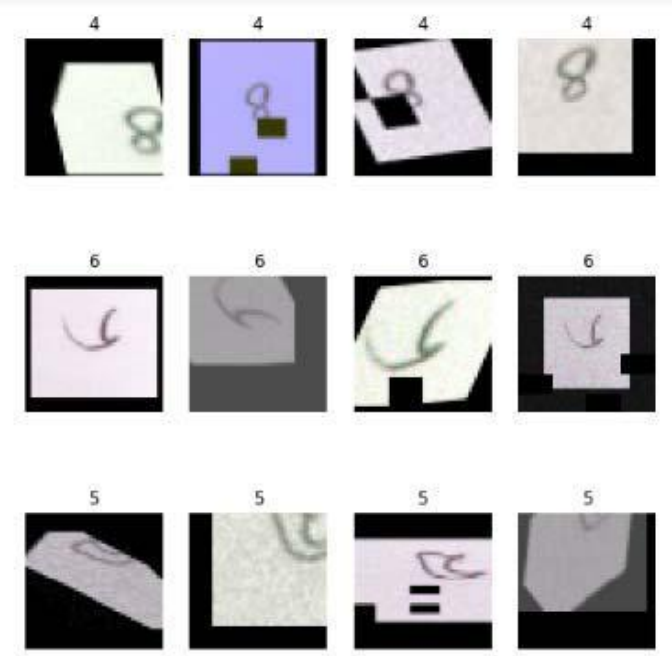

Fig 6: Augmentation applied on training set

D. Training and Validation

The augmented training set is split up into training and validation set in a 80:20 ratio. As access to test data labels was not granted, it was not possible to get test performance information during training.

E. Model Hyperparameters

Different model configurations were created by selecting different layers to be trainable. Each configuration was trained on the augmented dataset according to the following procedure.

Table 2: Parameter types and values

Parameter Name	Type/ Value
Epochs	50
Learning Rate	0.0001
Loss	Categorical cross entropy
Input Size	48 x 48
Pooling	Average
Batch Size	16
Verbose	2

Keeping the above parameters in the model, a new softmax layer is added.

IV. MODEL CONFIGURATION

This section describes the different model configurations trained on the augmented training set. Each configuration embodies a different type of transfer learning, with the first two being the traditional approach. The main contributions documented in this paper are found from the last three configurations.

Table 3: Configuration settings

Configuration Name	Notes	Layers Frozen(refer to fig 2)	#of trainable parameters
CFG-A		All layers except the softmax layer Layer frozen: 1 to 20	5,130
CFG-B	Used the model trained after CFG-A	None	14,719,818
CFG-C		Only softmax layer Layers frozen: 21	14,714,688
CFG-D		Softmax and previous 5 layers Layers frozen: 16,17,18,19,20,21	9,995,072
CFG-E		Only layers 16 – 20	7,640,394

The experiments are performed by changing some aspects in the VGG16 architecture. Each part provides an

understanding, as to what aspects were changed, along with their respective results.

CFG-A demonstrates the conventional transfer learning approach. A Softmax layer is added then the model is trained on layer 21 as shown in Table 3. In CFG-B, the previous model is taken and all the layers are trained from scratch using the training data on another 50 epochs. The model had significant improvement here. A new VGG16 model is taken and a Softmax layer is added in CFG-C. Here, the softmax layer is frozen and the model trains on all the remaining layers from 1 to 20. CFG-D is made by training only the last five layers, other layers frozen, of the VGG network – layer 16, 17, 18, 19, 20 as shown in Table 3. The softmax layer is also frozen. Lastly, the last five layers of the vgg network are frozen and training is done on all remaining layers including the Softmax layer, as shown in CFG-E in Table 3.

A. Evaluation Metric

The evaluation metric used by the Numta Kaggle Competition is Unweighted Average Accuracy (UAA), which is the average of the accuracies a model achieves over each of the 8 test data sets (A to F, Aug-A, Aug-C). This is shown in Table 4.

V. RESULTS AND DISCUSSION

Table 4: Evaluation with UAA and each test directory

	CFG- A	CFG-B	CFG-C	CFG-D	CFG-E
AVG (%)	64.53	96.88	96.96	96.83	97.09
A (%)	78.76	99.74	99.68	99.62	99.79
B (%)	66.66	100	100	100	100
C (%)	77.10	99.68	99.63	99.58	99.58
D (%)	76.79	99.43	99.48	99.69	99.38
E (%)	73.63	97.13	97.17	97.03	96.80
F (%)	44.44	92.12	92.12	91.91	93.33
AUG_A (%)	48.80	93.95	94.28	94.04	94.32
AUG_C (%)	50.61	92.27	93.30	92.73	93.54

Table 5: Performance of different configurations

Tuning on VGG16	**Epochs(steps)**	**Training Parameters**	**Accuracy (%)**
CFG-A	50	5,130	64.53
CFG-B	50 (total 100 as we are using the trained CFG-A model)	14,719,818	96.88
CFG-C	50	14,714,688	96.96
CFG-D	50	9,995,072	96.83
CFG-E	50	7,640,394	97.09

From Table 4 & 5 it is evident that the best performing model is CFG-E, closely followed by CFG-C. These are both very surprising results. In the case of CFG-C, the VGG16 model was trained while the Softmax layer, whose weights were randomly initialized, remained frozen and was not trained. This still led the model to achieve better accuracy compared to the traditional transfer learning models of CFG-A and CFG-B, where the Softmax layer was trained. In the case of CFG-B, as it was a continuation of CFG-A, it in reality got trained for 100 epochs, whereas CFG-C was only trained for 50 epochs.

In CFG-E, layers 16 -20 was frozen and the rest were trained, leading to a network divided into two trainable parts, the Softmax layer and the first 15 VGG16 layers. This halved the number of trainable parameters, but still led to the best overall accuracy within 50 epochs of training.

VI. CONCLUSION

This paper reported on the unconventional transfer learning approach used in the Kaggle Numta competition to classify Bangla Numerals eventually achieved 6th place among the 49 teams that participated. The highest accuracy achieved overall was 99.35% and the lowest being 51.1%. The model used the first position was an ensemble of different models combined which was much larger than the model reported here –which differs only by 2% in test accuracy compared to megabytes of difference in the model size.

Here, two results in particular are of interest. The first one where a randomly initialized softmax layer is added to a pre-trained VGG16 model, but is left frozen, i.e. the random weights do not change. The VGG16 model was then trained. This approach is almost the antithesis of traditional understanding of transfer learning, but led the better results than traditional transfer learning. An even more surprising result was found when layers 16-20 were frozen and the softmax layer, along with the first layers of the VGG16 model was trained jointly. This configuration led to the best average accuracy result among all the configurations attempted, and achieved better performance than traditional transfer learning,

while only requiring half the number of trainable parameters and half the number of epoch compared to CFG-B.

In future work, these results will be further analyzed to gain a better understanding of the reasons behind the results and similar configurations will be applied to other standard image classification tasks.

REFERENCES

[1] B. Yoshua, "Deep Learning of Representations for Unsupervised and Transfer Learning," JMLR: Workshop and Conference Proceedings 27:17–37, 2012.

[2] H. Geoffrey, V. Oriol, D. Jeff, "Distilling the Knowledge in a Neural Network," https://arxiv.org/pdf/1503.02531.pdf , 2015.

[3] S. Karen, Z. Andrew "VERY DEEP CONVOLUTION NETWORKS FOR LARGE SCALE IMAGE RECOGNITION," https://arxiv.org/pdf/1409.1556.pdf, 2015.

[4] S. Christian, V. Vincent, I. Sergey, S. Jon, "Rethinking the Inception Architecture for Computer Vision," JMLR: Workshop and Conference Proceedings 27:17–37, 2012.

[5] K. Alex, S. Ilya, H. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, 2012.

[6] Y. Jason, C. Jeff. B. Yoshua, L. Hod "How Transferable are Features in Deep neural networks http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf, 2012.

[7] K. Gopalakrishnan & Khaitan, S.K. & Choudhary, Alok & Agrawal, Ankit. (2017). Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. Construction and Building Materials. 157. 322-330. 10.1016/j.conbuildmat.2017.09.110.

[8] S. Alam, R. Tahsin, D. Mohammad, Rashed & Humayun, Ahmed. (2018). NumtaDB-Assembled Bengali Handwritten Digits. 10.13140/RG.2.2.33418.36800.

[9] B. Mithun, K. Gautam Shom, I. Rafiqul, Md. Shopon, M. Nabeel, M. Sifat, A. Anowarul, "BanglaLekha-Isolated: A multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters," *Data in Brief*, 2017

[10] Sharif, S. M. A., Mohammed, N., Mansoor, N., & Momen, S. (2016, December). "A hybrid deep model with HOG features for Bangla handwritten numeral classification,"2016 9th International Conference on (pp. 463-466). IEEE

[11] Shopon, M., Mohammed, N., & Abedin, M. A. (2016, December). "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network". In Computational Intelligence (IWCI), International Workshop on (pp. 64-68). IEEE.

[12] Sharif, S. M. A., Mohammed, N., Momen, S., & Mansoor, N. (2018). "Classification of Bangla Compound Characters Using a HOG-CNN Hybrid Model". In Proceedings of the International Conference on Computing and Communication Systems (pp. 403-411). Springer, Singapore

[13] Saha, Sourajit, and Nisha Saha. "A Lightning fast approach to classify Bangla Handwritten Characters and Numerals using newly structured Deep Neural Network." Procedia Computer Science 132 (2018): 1760-1770.

[14] Shopon, M., Mohammed, N., & Abedin, M. A. (2017, February). "Image augmentation by blocky artifact in Deep Convolutional Neural Network for handwritten digit recognition". In Imaging, Vision & Pattern Recognition (icIVPR), 2017 IEEE International Conference on (pp. 1-6). IEEE.