Classification of Handwritten Symbols using Deep Convolutional Neural Network

Dipayan Saha, Md Rafid Muttaki, Tashfia Alam
ECE Department, University of Florida, Gainesville, FL-32611
dsaha@ufl.edu, m.muttaki@ufl.edu, tashfia.alam@ufl.edu

***Abstract*—Handwritten character, digit, or symbol classification is one of the most studied research areas in image processing and machine learning. Automatic recognition of handwritten symbols from captured images in adverse practical scenarios is significant. This work also deals with a task where 25 different handwritten symbols are detected. We propose a well-known deep neural network, ResNet-18, to classify these symbols with the help of proper data preprocessing and extensive data augmentation. The proposed methodology reaches 99.71 % classification accuracy in the test set. This work also shows that data augmentation techniques improve the classification accuracy by 9.67%.**

***Index Terms*—Deep Learning, Machine Learning, Image Processing, Symbol Recognition**

I. Introduction

SYMBOL recognition as a research topic has received considerable attention from the early '90s. Numerous techniques have evolved as solutions to this problem. Especially, usage of Deep Neural Networks (DNN) in symbol/letter recognition has become more prominent in recent years due to its robustness towards image blurring, shifting, and scaling distortions. In this project, we have developed a method to classify images of 25 handwritten symbols. Each student from the class contributed to the dataset by creating and recording at least 10 copies of each symbol. The instructor provided the training dataset, which is a small fraction of the merged data submitted by the students. The rest of the data will be used for testing the model and evaluation purposes.

Multiple DNN methodologies have been proposed in pattern recognition applications, identifying symbols, letters, or miscellaneous drawings. Wang [1] proposed a text recognition methodology from images using Convolutional Neural Networks (CNN). They combine CNN with automatic feature extraction, an unsupervised learning algorithm. The learned features are fed into a trained CNN that enables greater accuracy in text recognition. Another CNN-based character and word recognition approach has been introduced in Yuan [2]. It starts with preprocessing the images, followed by segmentation, and applies LeNet-5 model-based CNN for character recognition offline with a 92.2% success rate. Wu [3] proposes a relaxation CNN (R-CNN) and alternately trained relaxation CNN (ATR-CNN) based handwritten character recognition method. It comes with two advantages. R-CNN can quickly learn, and ATR-CNN can be alternately trained on a subset of CNN layers improving the accuracy. It can recognize Chinese handwritten characters with 96.06% accuracy. Bai [4] introduces a character recognition technique using shared hidden layer CNN (AHL-CNN), where the characters share the hidden layers, and the final layer is trained based on the destination language characters. It has been applied on English and Chinese characters and performed better than the conventional CNNs, with an error rate reduced by 16 to 30%. A perspective of Image recognition has been discussed in Liu [5] reviewing the challenges and applications of CNNs. Zhang [6] introduces WAP (Watch, Attend and Parse) approach for mathematical expression recognition based on neural networks. It incorporates a watcher, a CNN encoder to map the images into high-level features, and a parser/decoder (RNN) to translate the features to output sequences. Ptucha [7] proposes a fully convolutional neural network (FCN) for offline handwritten character recognition. The input stream is resampled into a canonical representation processed by the FCN later.

ResNet [8] is one of the most widely adopted neural networks in different applications. Such remarkable success of ResNet in solving very complicated problems inspires us to check the efficacy of this architecture in recognizing handwritten symbols. With the help of effective data preprocessing and augmentation techniques, this work finds out the potential of ResNet-18 to classify symbols captured in different adverse practical scenarios.

The remainder of the report is organized into three sections. We provide the details of the implementation in section II and report the experiments conducted in the following section. Finally, we conclude our reasoning in section IV.

II. Implementation

In the proposed methodology, data preprocessing is applied to the input image before feeding it into a neural network to extract features in an end-to-end process. As proposed network architecture, well-known Resnet-18 is used. Extensive data augmentation is also used during training the network to add new practical observations data and tackle the overfitting problem. Section II-A and Section refda discuss the data preprocessing and data augmentation approaches. Next, Section II-C describes the proposed network architecture and Section II-D narrates the objective function and Section II-E presents the training scheme used in this work.

A. Data Preprocessing

At first, two preprocessing methods are applied to training and test data. Although it is expected that the input image

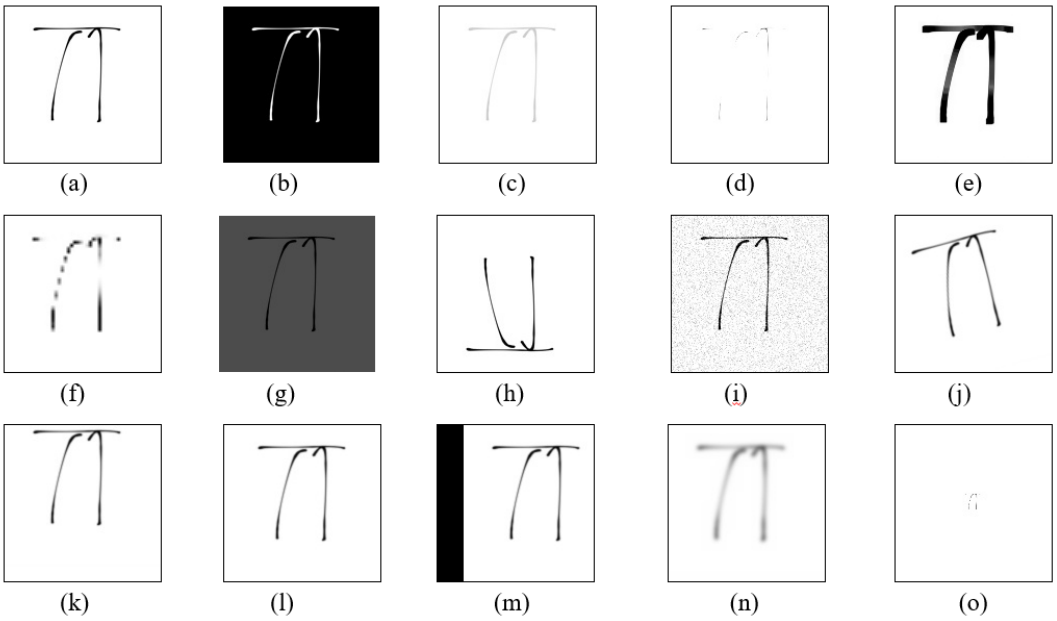

Fig. 1: Different data augmentations used in the work: (a) Original image, (b) color inversion, (c) fading, (d) dilation, (e) erosion, (f) downsampling, (g) scaling, (h) vertical flip, (i) noise addition, (j) rotation, (k) up/down shift, (l) left/right shift, (m) blur, (n) zoom out, (o) double digit masking

will be a grayscale image with the range of pixel values 0 to 255, some samples of the provided dataset are in the range of 0 to 1 or very small values. This results in a very dark image of a handwritten symbol, which is quite unrecognizable. The first preprocessing step mainly solves this specific scaling problem. Based on the image's pixel values, it checks whether this type of scaling problem exists or not. If it exists, it re-scales the image in a range of 0 to 255. Another preprocessing step normalizes the image for faster convergence.

B. Data Augmentation

The primary motivation to apply data augmentation is to add new observations to the training data. In practice, the captured image of the symbol may have different types of irregularities, i.e., noise, illumination problem, shadow, haziness, etc. The training data do not include all of these types of data. This inspires to apply in total 14 different data augmentation techniques: color inversion, fading, dilation, downsampling, erosion, scaling, vertical flip, shearing, rotation, left/right shift, up/downshift, shift with dark pixels, noise addition, blur, zoom out, masking. Figure 1 shows images after applying all these augmentations on a training image sample. Four augmentation methods are randomly applied for each image of the training set.

C. Proposed Network Architecture

In this work, as neural architecture, we propose ResNet [8] with 18 layers. Although ResNet-18 was proposed for RGB images, we use single-channel gray-scale input as input feature maps with input size 150 $\times$ 150. A pre-trained model is not used in this work. The network is initialized with random weights and trained on the training dataset. In short, only knowledge of the structure of ResNet-18 architecture is used.

As the name implies, in ResNet architecture, the previous layer's output is added to the current layer. The motivation of such skip connections is to tackle the vanishing gradient problem. ResNet is different from other plain structures of deep CNNs because these identities skip connections between the layers. Figure 2 shows the structure of ResNet-18. The arrow in the figure indicates the skip connection between the layers. The figure also shows the number of filters, kernel size, stride value used in each layer. Bottleneck basic blocks are used to create a deep neural network. In each block, there exist two convolutional layers. A ReLU and batch normalization layer follows each convolutional layer. Since the network is subjected to complex problems, ReLU is used to bring the non-linearity. Batch normalization is used to enhance the speed of the training and also to bring regularization. The dropout layer is also included in the structure to prevent overfitting. The max-pooling layer with a stride value of 2 is used at the initial part of the network to reduce the dimension of the feature maps. Average pooling is used at the end of the architecture, followed by a fully-connected layer. This network has 25 output features fed into the Softmax function since this is a 25-class classification problem.

D. Objective Function

Categorical cross-entropy loss is chosen as objective function. This objective function L is given by

$$L = -\sum_{c=1}^{N} (l_c \times log(\widehat{l_c})) \tag{1}$$

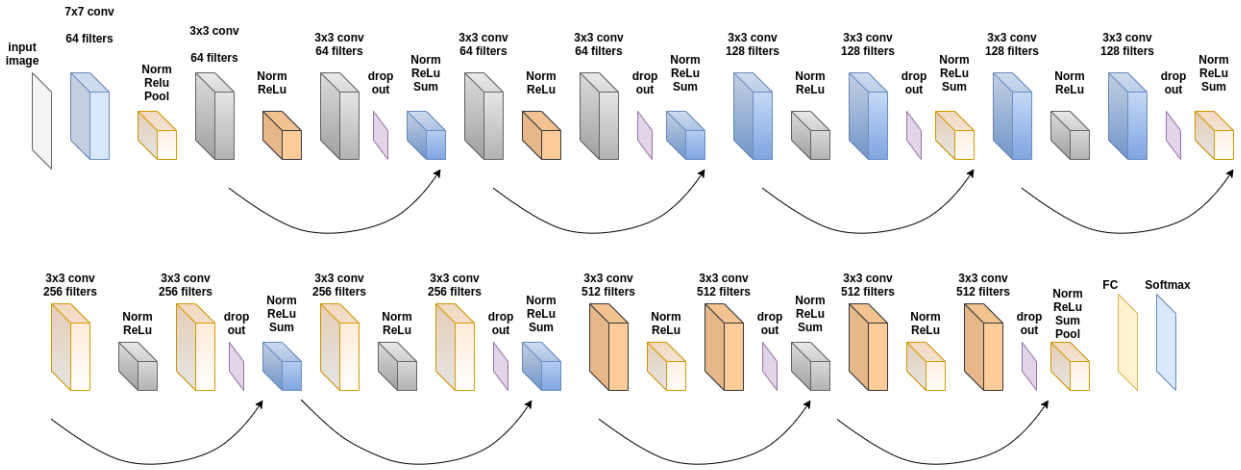

Fig. 2: Network architecture of ResNet18 [8]

here, l_c and $\widehat{l_c}$ denote the ground truth and probability of the class prediction for an image. N indicates the number of class.

E. Training Scheme

The mini-batch optimization technique is utilized for training, with the batch size set to 32. For the optimization purpose, Adam Stochastic Optimization algorithm [9] is chosen, with learning rate, decay values for first and second moments are set to 10^{-4}, 0.9, and 0.999, respectively. The NVidia RTX GPU is used for training purposes.

III. Experiments

Exhaustive experiments are performed to certain the potential of the proposed methodology in handwritten symbol classification. Section II-E describes training setup and parameter settings. Section III-A, Section III-B and Section III-C respectively provide detail descriptions of performance metrics, dataset and other comparing methods used in this work. Finally, Section III-D provides the results of experiments and also analyzes them.

A. Evaluation Metrics

The proposed method's performance and comparing approaches are evaluated in terms of accuracy in this study. Accuracy, in particular, highlights instances of true positives and true negatives. The classification performance is interpreted through visualization of the confusion matrix. In order to evaluate and compare the performance of different networks, the computational complexity of the model is also calculated. The total number of trainable and non-trainable parameters is considered the network architecture's computational complexity.

B. Dataset

After data collection and curation were done by the course EEL5840, at first, in total, 29920 samples were provided for training purposes. Later, a more curated dataset of around 25000 samples was provided. We first merge these two datasets in this work, excluding the common data samples. In this way, a dataset of 31295 data samples is obtained. Later, we manually separate these data into two categories: clean and unclean data. The unclean data includes different challenges: noise, scaling problem, fading problem, discontinuity, haziness, shadow, rotation, etc. We emulate these types of practical challenges with the help of data augmentation techniques described in Section II-B. We apply these augmentation approaches only on the clean dataset. In this way, we increase the size of the dataset by multiple times and new types of observations to the initially provided small dataset. We also added other additional data collected by our group. We keep a portion of this additional data to the validation data. In this way, we create an augmented training dataset of size 140317 samples. Since the main test dataset is not released, we consider this validation dataset as our test data in this experiment. The number of samples in this test set is 3075.

C. Comparing Methods

The performance of the proposed network is compared to the other network architectures. As comparing neural networks, well-known models e.g., VGG11, and VGG13 are used for their previous success in image classification tasks. Resnet architectures with other different depths are also tested for performance comparison. Besides these existing network architectures, shallow CNN networks built from scratch are used for comparison. We named these networks CNN-5, were respectively five convolutional layers are used, followed by two fully connected layers and Softmax.

D. Results

The performance of the proposed approach is compared to that of other comparing networks. Table 1 provides the comparison in terms of accuracy and the computational complexity of the network. From the table, it can be noted that the highest accuracy is found for ResNet-18 and ResNet-50. But ResNet-50 has around two times more parameters than ResNet-18. The

TABLE I: Comparison of performance with other comparing methods in terms of accuracy and computational complexity

Methods	Accuracy (%)	Computational Complexity
CNN-5	98.46	2.76M
VGG-11	99.61	128.67M
VGG-13	99.67	129.05M
ResNet-34	99.64	21.30M
ResNet-50	99.71	23.55M
ResNet-18 (Proposed)	99.71	11.18M

TABLE II: Impact of data preprocessing and augmention for proposed method

Preprocessing	Augmentation	Accuracy (%)
Present	Present	99.71
Present	Absent	90.04
Absent	Absent	87.77

accuracy of VGG-13 is also very close to the proposed ResNet-18, but in terms of computational cost, it has more than 10 times the parameters. CNN with five convolutional layers has the lowest number of trainable and non-trainable parameters since the depth of this architecture is very low compared to the other deep neural networks. But it has come at the cost of reducing classification accuracy. Considering both accuracy and the complexity of the networks, the ResNet-18 is proposed as the deep neural architecture in this work.

Figure 3 shows the confusion matrix obtained on the test set for the proposed methodology. Since the accuracy is very close to 100%, only very few images have been misclassified for the trained model. That is why bright colors cells are visualized along the diagonal of the matrix.

In order to evaluate the impact of the data preprocessing and the data augmentation, separate experimentation is performed. In this case, the proposed architecture is trained for additional two scenarios. In one case, the model is trained on non-augmented training data. In another case, augmentation is used, but data is not preprocessed. The results of these experiments are mentioned in Table II. The table shows that when augmentation is not used, the classification accuracy of the proposed network drops by around 10%. It shows the strong positive impact of the data augmentation on the model's performance. It happens because extensive data augmentation helps the model learn the unseen and additional data, which will ultimately help recognize symbols in difficult practical scenarios. The table also suggests that when data preprocessing is kept off, the model's performance further degrades by around 2%. Such results explain the reason for the inclusion of data preprocessing and augmentation in the proposed methodology.

IV. Conclusion

Classification of handwritten symbols or characters has always focused on image processing and machine learning researchers. In this work, we successfully classify 25 different handwritten symbols from images. The dataset is a collective effort of the participants of course EEL5840. Our proposed methodology includes data preprocessing and data augmentation before applying deep neural networks. We applied 14

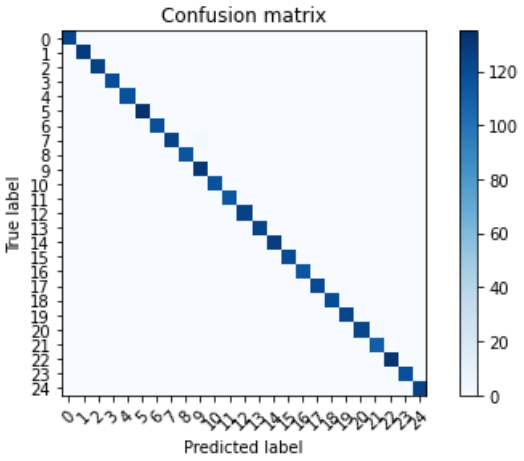

Fig. 3: Confusion matrix for the proposed method

different data augmentations to include new observations in the training dataset. We used ResNet-18 to train our augmented and preprocessed data. On our test set, it achieves 99.71% classification accuracy. The performance of the proposed methodology outperforms other comparing methods. Our detailed experiments show how data preprocessing, and augmentation significantly improve the proposed methodology's performance. We plan to implement more explainable neural networks and other advanced image processing techniques in the future.

References

[1] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 3304–3308.

[2] A. Yuan, G. Bai, P. Yang, Y. Guo, and X. Zhao, "Handwritten english word recognition based on convolutional neural networks," in *2012 international conference on frontiers in handwriting recognition*. IEEE, 2012, pp. 207–212.

[3] C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, "Handwritten character recognition by alternately trained relaxation convolutional neural network," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 291–296.

[4] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 2560–2564.

[5] Q. Liu, N. Zhang, W. Yang, S. Wang, Z. Cui, X. Chen, and L. Chen, "A review of image recognition with deep convolutional neural network," in *Intelligent Computing Theories and Application*, D.-S. Huang, V. Bevilacqua, P. Premaratne, and P. Gupta, Eds. Cham: Springer International Publishing, 2017, pp. 69–80.

[6] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[7] R. Ptucha, F. Petroski Such, S. Pillai, F. Brockler, V. Singh, and P. Hutkowski, "Intelligent character recognition using fully convolutional neural networks," *Pattern Recognition*, vol. 88, pp. 604–613, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320318304370

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.