KLE Society's

KLE Technological University

Creating Value
Leveraging Knowledge

COMPUTER SCIENCE AND ENGINEERING

Course project report

On

EMNIST: Extended MNIST

Machine Learning (17ECSC306)

Under the guidance of:

Asst. Prof. Sunil V G

Asst. Prof. Uday Kulkarni

Submitted by:

Name	**USN**
Shrenik H	01FE17BCS194
Sourabh J	01FE17BCS214
Sushant M	01FE17BCS222
Sweekar B	01FE17BCS227
Malatesh K	01FE17BCS413
AnilKumar G	01FE16BCS403

Acknowledgement

The success and outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Asst. Prof. Sunil VG, for providing us an opportunity to do the project work as a part of the Machine Learning Course and giving us all support and guidance that made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance.

We owe our deep gratitude to our project guide Asst. Prof. Uday Kulkarni, who took keen interest on our project work and guided us all along, until the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to, fortunate enough to get constant encouragement, support, and guidance from all Teaching staffs of Computer Science Department, which helped us in successfully completing our project. In addition, we would like to extend our sincere esteems to all staff in laboratory for their timely support.

Abstract

This paper summarizes a technique for classification on the EMNIST dataset, which is an extension to the MNIST dataset. The MNIST dataset has set a benchmark in Machine Learning and Computer Vision. Many programmers have evaluated the performance of different deep learning techniques on this MNIST dataset. Over the years the accuracy of prediction over the MNIST dataset is increased to a very high level with different deep learning techniques that the dataset is left with no more challenges. Thus to increase the challenges EMNIST dataset is introduced. Both MNIST and EMNIST datasets are derived from a much larger dataset known as the NIST Special Database 19 which consists of digits, uppercase and lowercase alphabets.

Contents

1 Introduction

MNIST dataset is set of handwritten digits. EMNIST dataset which is Extended MNIST dataset is a set of handwritten digits, uppercase and lowercase alphabets. EMNIST dataset images share the same image structure as that in the MNIST dataset.

The size of the images is (28 pixels * 28 pixels). Each row in the dataset represents an image. There are a total of 785 columns in the dataset, where the first column represents the class label, that is which class does the image belong to. The remaining 784 columns represents the image in a flattened format. The image can be retrieved by layering down 28 pixels one below the other. The images provided in the dataset are actually inverted and rotated by 90 degrees, Thus first step in procedure remains to obtain the image in a correct format.

There are a total of 6 datasets provided for classification over the EMNIST dataset.

1. ByClass dataset : This dataset has uneven number of images per class. There are more digits thanthat of letters. There are a total of 697932 images in the training set and 116323 images in the testing set. The total number of classes in this dataset is 62 which is summed up as 26 uppercase alphabets, 26 lowercase alphabets and 10 digits.

2. ByMerge dataset : This dataset has uneven number of images per class. There are more digitsthan that of letters. There are a total of 697932 images in the training set and 116323 images in the testing set. The total number of classes in this dataset is 47. The number of classes is reduced from 62 to 47 as some of the classes are merged into a single class. This merge operation is done on some alphabets which appear same in both of their lowercase as well as uppercase formats. For example P and p, C and c etc.

3. Letters dataset : This dataset has a balanced set of merged uppercase and lowercase alphabets.There are a total of 88800 images in the training set and 14800 images in the testing set. The total number of classes in this dataset is 37 after merging of few uppercase and lowercase alphabets.

4. Digits dataset : This dataset has a balanced set of digits. There are total of 240000 images in thetraining set and 40000 images in the testing set. The total number of classes is 10.

5. MNIST dataset : This dataset has a balanced set of digits. There are total of 60000 images in thetraining set and 10000 images in the testing set. The total number of classes is 10.

6. Balanced dataset : This dataset is meant to address the balance issue in ByClass and ByMergedatasets. It is derived from ByMerge but with equal number of samples per class. The total number of classes is 47.

Balanced dataset is most applicable dataset for classification over the EMNIST dataset as it has equal number of samples per class and as it is derived from ByMerge dataset, there will be no confusion caused by alphabets which look similar in their uppercase as well as lowercase formats.

2 Literature Survey

CNN are always considered to be one of the best option for image processing tasks. CNN is generally called as Convolutional Neural Network and has gained a lot of popularity in recent times. The most important feature of CNN is that it uses multi layer perceptrons to do computational works. As compared to other classification algorithm, CNN use relatively little pre-processing. Since, CNN use various different filters to train the model,hence the amount of pre-processing on the image is reduced.

Basically a CNN classifies the given input image and assigns a category to it. All the images given as input are converted into grey scale thus retaining only a single channel. An image is considered as a 2-D array of pixels, which ranges from 0-255 where 0 is a complete dark pixel and 255 is a white pixel. Sequential model is the API used to create this CNN where an instance of Sequential class is created and layers are added to it.Hence, this project uses CNN to classify the data.

Keras is a python library which is used in building CNN. Keras consists of numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and different tools to work with image and text data and to simplify the coding necessary for writing Deep Neural Network code. Thus, this project uses Keras to implement CNN.

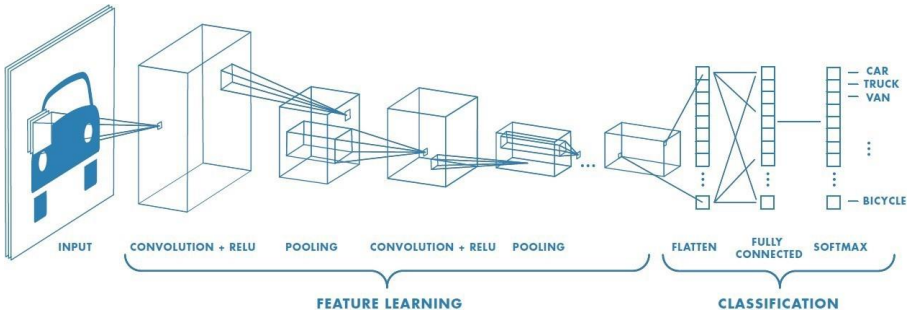

Figure 1: CNN Working

3 Methodology

The dataset given had a very little pre-processing.The images were first reshaped to (28*28). 10 percent of the train data is splitted to act as validation data. All the images in the dataset were rotated by 90 degree and hence are brought back to correct form by rotating the same image by 270 degrees, All the output classes are converted to categorical data i.e all the output classes are processed by one hot encoding. Hence the data is pre-processed and is ready to act as input to the CNN.

In this project, the procedure of CNN model follows in a way where all the training images go through a set of filters which are also called as kernels. These filter's sizes may change depending upon the dataset. The filters perform few mathematical operations on the images and preserve their features. These operations are performed by multiple number of convolutional layers. Layers are activated by the activation functions. There are several activation functions like 'relu','tanh'.'sigmoid','softmax'. After each convolutional layer, pooling is done where the best feature is selected from each kernel's calculation. Pooling is used to reduce the number of features but still considering the important ones. There are several different techniques of pooling like max pooling, average pooling, sum pooling. After a satisficatory number of convolutional layers are added, next a flattened layer is added so that it can perform operations on the further dense layers. Dense layers are fully connected neural networks. Dropouts are added so that the model does not

overfits. The last layer in the CNN is activated with 'softmax' activation function. 'Softmax' provides a probability for all the classes. The image belongs to the class with highest probability.'Adam' optimizer is the optimizer used to find the adaptive learning rates for each parameter.In this model,

Layer 1 is a Convolution layer which has 256 filters, with padding 'same' (which pads the input in such a way that output matches with the original image size). The kernel size is (3*3). The activation function used is 'relu'. Layer 1 being the input layer, it converts the image into grey scale image.
Layer 2 is a Maxpooling layer with pool size (2*2) and strides (2*2).
Layer 3 is a Batch normalization layer.
Layer 4 is a Convolutional layer which has 128 filters, with padding 'same'. The kernel size is (3*3). The activation function used is 'relu'.
Layer 5 is a Maxpooling with pool size (2*2) and strides (2*2).
Layer 6 is a Batch normalization Layer.
Layer 7 is a Convolutional layer which has 64 filters, with padding 'same'. The kernel size is (3*3). The activation function used is 'relu'.
Layer 8 is a Maxpooling with pool size (2*2) and strides (2*2). Batch normalization is applied.
Layer 9 is a flattened layer to form a fully connected neural network or dense layer.
Layer 10 is a dense layer which has 64 units and uses activation function 'relu'.
Layer 11 is a dropout of 0.4 is added to avoid overfitting.
Layer 12 is a dense layer which has 64 units and uses activation function 'relu'.
Layer 13 is a dropout of 0.4 is added to avoid overfitting
Layer 14 is a dense layer and also the final output layer which has 47 (Total number of classes) units and uses the activation function 'softmax'.

4 Results and discussions

This model is able to predict the output for a given image. It predicts which letter or digit the image is. It can also predict one or more letters at a time in the same image provided the spacing between each letter in the image is same and the model should know the number of letters in the image.

The model achieved an accuracy of 88.62 and the plot for Model accuracy is as follows

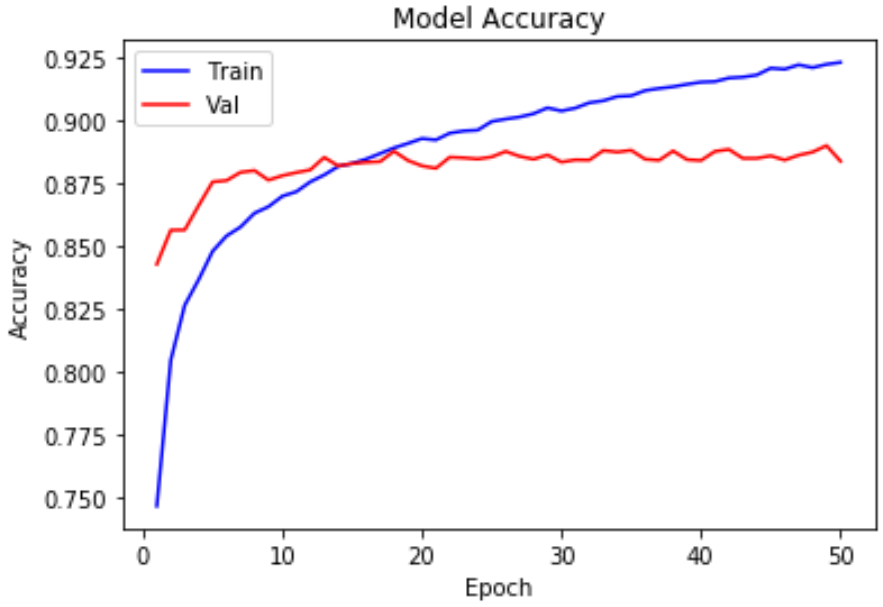

Figure 2. Model Accuracy

The model mis-classifies some of the letters. This can be viewed with the help of confusion matrix as shown below.

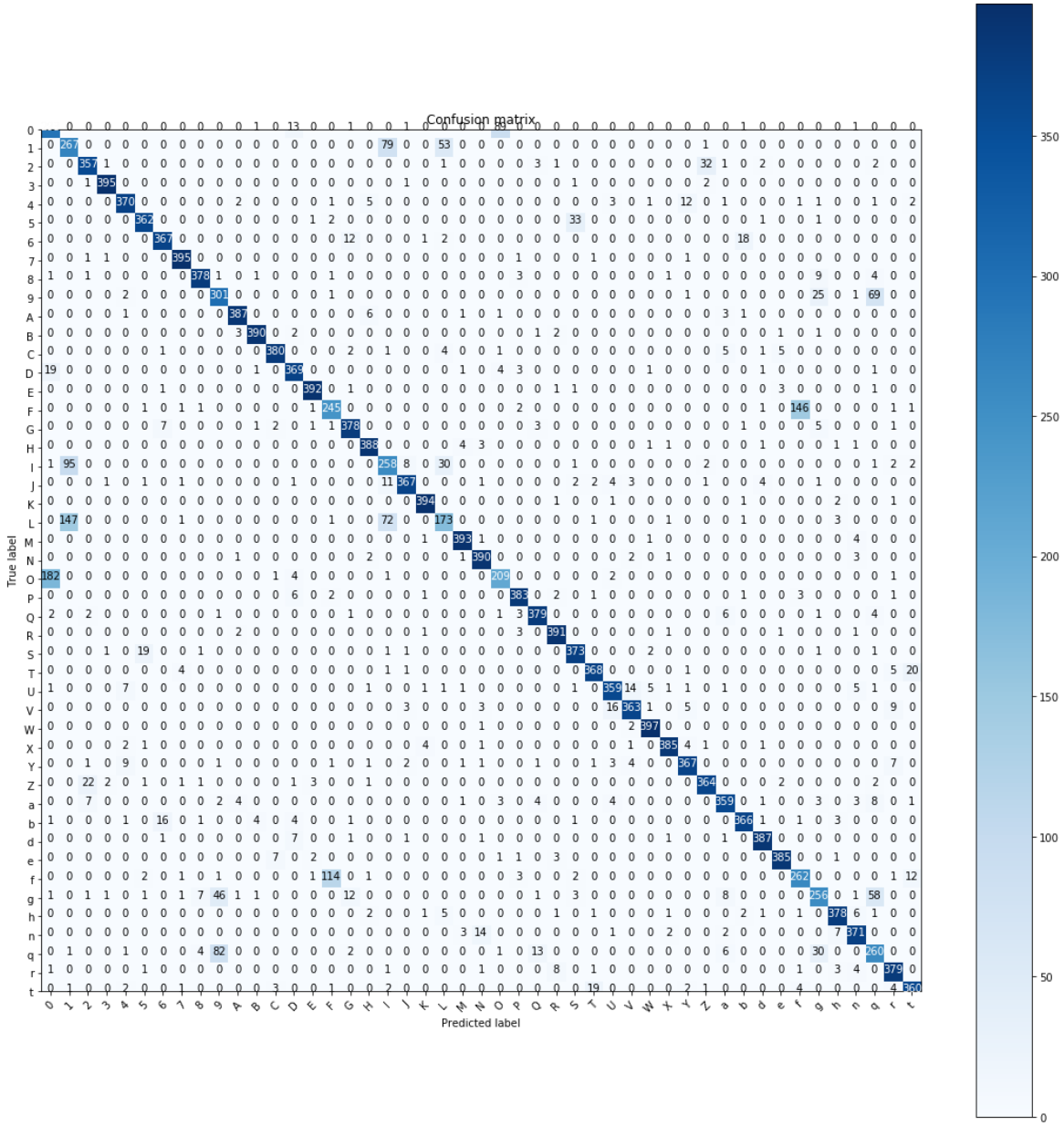

Figure 3. Confusion Matrix

5 Conclusion and Future Scope

This project predicts letter/digit from a image. Hence, this can be used as a document reader. This can further be extended to text to speech recogniser so that even blind people can listen it. New text document can be feed into braille printers so that it can be further used as a braille script.

6 References

[1] Baldominos, Alejandro S´aez, Yago Isasi, Pedro. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. Applied Sciences. 2019. 3169. 10.3390/app9153169.

[2] Eijaz Allibhai, 'Building a Convolutional Neural Network (CNN) in Keras',2018. [Online].Available:https://towardsd a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5. [Accessed: 24- Nov- 2019].

[3] Prabhu, 'Understanding of Convolutional Neural Network (CNN) — Deep Learning', 2018. [Online].Available:https:
of-convolutional-neural-network-cnn-deep-learning-99760835f148 [Accessed: 10- Oct- 2013].

[4] P. Grother, "NIST special database 19 handprinted forms and characters database," National Institute of Standards and Technology, Tech. Rep., 1995.

[5] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik, "EMNIST: an extension of MNIST to handwritten letters",2017.