**Handwriting Recognition using Deep Learning**

Name: Aafaq Saleh Al Shidhani

**Abstract**

This project implements a handwriting recognition system using deep learning. Handwriting is unique to each individual. So the handwriting is differed from one person to another person. Two datasets are used for this handwriting recognition process, the MNIST data set is and it has 70000 handwritten digits, and the EMNIST data set is and it has 131600 handwritten letters and digits. There are many Machine Learning Algorithms that can be used for this digit classification. This project performs the recognition system using Convolutional Neural Networks (CNN). The proposed solution performs the recognition with overall accuracy is 98.4% for the MNIST and 88.7% for the EMNIST.

**Introduction**

**problem definition**

Handwritten Recognition is a challenging task in the field of pattern recognition. Because every person in this world has unique handwriting. Recognizing handwritings is a complex problem; because there is some level of difference and uniqueness in every type of writing like the way of holding the pen, the spaces, type of pen used in the writing and the amount of pressure put on paper.

**motivation**

The conversion of handwritten text into digital format is very important task because it's widely used in many applications, such as verification of signatures in banks, recognizing ZIP codes, number plate recognition, historical documents, forensic evidence, etc. Therefore assigning this task to computers can be very valuable, helpful and time saving.

**objective  of the work**

This project aims to classify an individual handwritten digits and characters so that handwritten text can be translated to a digital version.

problem Input: images (28×28 pixels) as raw pixel values

problem Output: Digit (0-9) / Character (A-Z)

**Method**

To implement this recognition I will be using Convolutional Neural Network (CNN) with many other Machine Learning techniques to build a model that can accurately classify digits and characters.

**Deep Learning**

Deep learning has been widely used to recognize handwriting. Deep learning algorithms are applied to unsupervised learning tasks in machine learning. This is important because un labeled data are more abundant than the labelled data.

**Convolutional Neural Network (CNN)**

In deep learning, Convolutional Neural Network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery. It has been used in many applications such as pattern recognition, sentence classification, speech recognition, face recognition, text categorization, document analysis, and handwritten digit recognition.

A convolutional neural network is a combination of input and an output layer, and multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers.

In this problem Neural networks are better than classical methods, because there is no need for features extraction. In CNN, parameters are learnt during the training process, it uses convolution of image and filters to generate features which are passed on to the next layer. This makes deep learning methods more resilient to changes in handwriting styles, and reduce the challenges in feature extraction in classical methods. Also CNN performs very well with big dataset. Another feature is that convolutional networks are flexible and work well on image data. Diagram

Description automatically generated

CNN architecture

**Proposed Method**

1. **Data Preprocessing**
   1. Load data
   2. Normalization

This can reduce the complexity and improve the efficiency of the model.

* 1. Reshape

The images are of a 2D shape, and to apply CNN with Keras we need to reshape them to 3D matrices. Size of images are (28px x 28px), we reshape them to (28x28x1).

* 1. Label encoding

We need to encode the lables to one hot vectors to be either 0 or 1.

* 1. Split training and validation set

1. **CNN**

I will be using Keras with TensorFlow backend to build the Neural Network model.

* 1. Create the model
     + 2 convolution layers with 32 filters followed by two fully connected layers with 64 filters. The used activation function is RelU.
     + Each layer is  followed by a max-pool layer
     + Dropout some neurons, used for regularization.
     + Flatten layer to convert the final feature maps into a one single 1D vector.
  2. Compile the model

With adam optimizer and accuracy metric.

* 1. Data Augmentation

To avoid overfitting, we will apply some transformations on the testing set to reproduce variations of each sample, The applied transformations are :

* + Randomly rotate some images
  + Randomly Zoom images
  + Randomly shift images horizontally
  + Randomly shift images vertically
  1. Fit the model

1. **Model Evaluation**

Performance metric:

* + 1. Accuracy
    2. confusion matrix

**Dataset**

For this project I will be using two datasets:

1. **MNIST dataset**

Modified National Institute of Standards and Technology (MNIST) is a large set of computer vision dataset which is used for training and testing different systems. It was created from the two special datasets of National Institute of Standards and Technology (NIST) which holds binary images of handwritten digits. The training set contains handwritten digits from 250 people, among them 50% training dataset was employees from the Census Bureau and the rest of it was from high school students. The MNIST dataset consists of 60,000 training examples and 10,000 examples in the test set.There are 28×28 pixel images of handwritten single digits between 0 and 9.

1. **EMNIST dataset**

The EMNIST Dataset is an extension to the original MNIST dataset to also include letters. The dataset consists of 112,800 training examples and 18,800 examples in the test set, and 47. There images are 28×28 pixel of handwritten letters from A to Z and digits from 0 to 9.

EMNIST Classes\*: {0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8: '8', 9: '9', 10: 'A', 11: 'B', 12: 'C', 13: 'D', 14: 'E', 15: 'F', 16: 'G', 17: 'H', 18: 'I', 19: 'J', 20: 'K', 21: 'L', 22: 'M', 23: 'N', 24: 'O', 25: 'P', 26: 'Q', 27: 'R', 28: 'S', 29: 'T', 30: 'U', 31: 'V', 32: 'W', 33: 'X', 34: 'Y', 35: 'Z', 36: 'a', 37: 'b', 38: 'd', 39: 'e', 40: 'f', 41: 'g', 42: 'h', 43: 'n', 44: 'q', 45: 'r', 46: 't'}

\* The letters are represented by numbers from 11 to 46. The data set contain a file that maps the letters with their corresponding number. Many letters have problems in character recognition that the upper and lower case variants are very similar. This causes problems in trying to classify these letters. To counteract this they have merged the letters they thought this was a problem for. The merged classes, are for the letters C, I, J, K, L, M, O, P, S, U, V, W, X, Y and Z.

A picture containing text

Description automatically generatedText

Description automatically generated with low confidence

Figure 1 MNIST Dataset Sample

Figure 2 EMNIST Dataset Sample

Chart, line chart

Description automatically generated**Experimentation and discussion**

**MNIST Dataset**

Accuracy Metric:

Training:

acc: 0.9839

loss: 0.0544

Validation:

acc: 0.9931

loss: 0.0373

**EMNIST Dataset**

Accuracy Metric:

Figure 3 MNIST

**Chart

Description automatically generated with medium confidence**Training:

acc: 0.8441

loss: 0.4545

Validation:

acc: 0.8874

loss: 0.3086

Figure 4 EMNIST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | **Training** | | **Validation** | |
| acc | loss | acc | loss |
| **MNIST** | 98.39 | 5.44 | 99.31 | 3.73 |
| **EMNIST** | 84.41 | 45.45 | 88.74 | 30.86 |

Overall the performance of the model was better in the MNIST data set. The performance on the EMNIST could be improved.

**Conclusion**

The result of the project shows that the maximum accuracy 99.39% was obtained in MNIST and 88.41% in the EMNIST dataset using the Convolutional neural network technique (CNN). Further work should be focused in improving the model performance on the EMNIST dataset. Also this project focused on recognizing separated letters and digits, though real handwritings can be Cursive, implementing a recognition system based on complete words would be more useful and practical.

**References:**

* <https://en.wikipedia.org/wiki/MNIST_database>
* <https://medium.com/datadriveninvestor/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8>
* <https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64>
* <https://arxiv.org/pdf/1702.05373v1.pdf>
* MNIST Dataset Link: http://yann.lecun.com/exdb/mnist/
* EMNIST Dataset Link: https://arxiv.org/abs/1702.05373