A

Project Report on

HANDWRITTEN CHARACTER RECOGNITION

SUBMITTED IN PARTIAL FULFILLMENT TO

THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

IN THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science Engineering)

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

Submitted To:

Dr. SUKHVIR KAUR

(INT246)

Submitted By:

Samarth Gupta (11911223)

Abhishek Jha (11911129)

Nikhilesh Kumar Verma (11904133)

Contents

1. Introduction

Handwriting recognition is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen devices etc. The main aim of this project is to design expert system for, "Handwritten Character Recognition using Neural Network" that can effectively recognize a particular character of type format using the Convolutional Neural Network (CNN) approach.

Handwriting recognition has been one of the active and challenging research areas in the field of image processing and pattern recognition. It has numerous applications which include, reading aid for blind, bank cheques and conversion of any handwritten document into structural text form.

2. Technologies Used

This project is based on neural networks and for this purpose Python is used. The libraries used in this project are: TensorFlow and Keras, OpenCV, NumPy, Pandas, Matplotlib.

2.1 TensorFlow and Keras

TensorFlow and Keras are deep learning APIs written in Python. For performing experiments on neural networks, these libraries are required.

2.2 OpenCV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. In this project, we used OpenCV to read, show and pre-process the Images.

2.3 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

2.4 Pandas

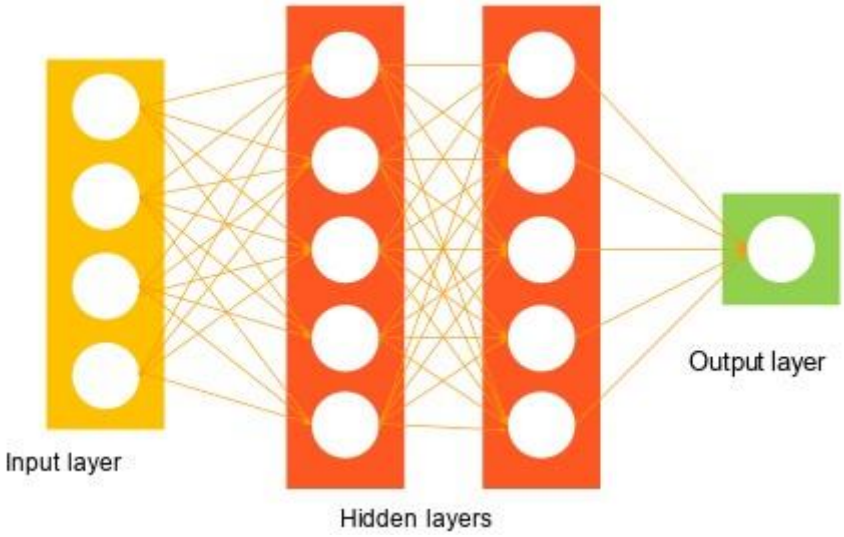
Pandas is an open-source python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

3. Neural Networks

Neural networks are computing systems with interconnected nodes that work much like neurons in the human brain. Using algorithms, they can recognize hidden patterns and correlations in raw data, cluster and classify it, and – over time – continuously learn and improve.

A neural network has many layers. Each layer performs a specific function, and the complex the network is, the more the layers are. That's why a neural network is also called a multi-layer perceptron.

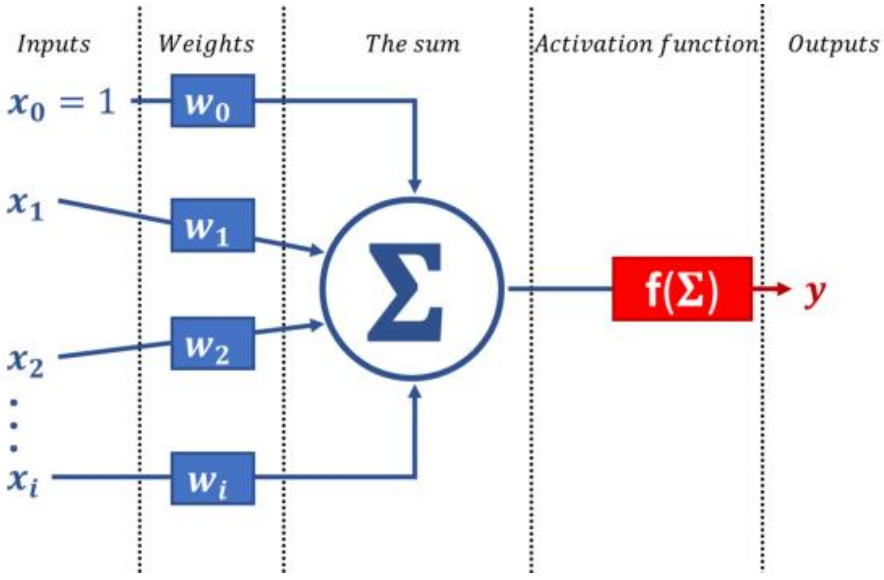
The purest form of a neural network has three layers:



1. The input layer: The input layer picks up the input signals and transfers them to the next layer. It gathers the data from the outside world.
2. The hidden layer: There can be multiple hidden layers in a neural network according to the requirements. The hidden layer performs all the back-end tasks of calculation. A network can even have zero hidden layers. However, a neural network has at least one hidden layer.
3. The output layer: The output layer transmits the final result of the hidden layer's calculation. There can be single or multiple nodes in the output layer. If we have a binary classification problem the output node is 1 but in the case of multi-class classification, the output nodes can be more than 1.

3.1 How a neuron work

A neuron takes input, performs weighted sum of all inputs, applies activation function to it and gives the output. As shown in following figure.



3.2 Activation Functions

An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network. The choice of activation function has a large impact on the capability and performance of the neural network, and different activation functions may be used in different parts of the model.

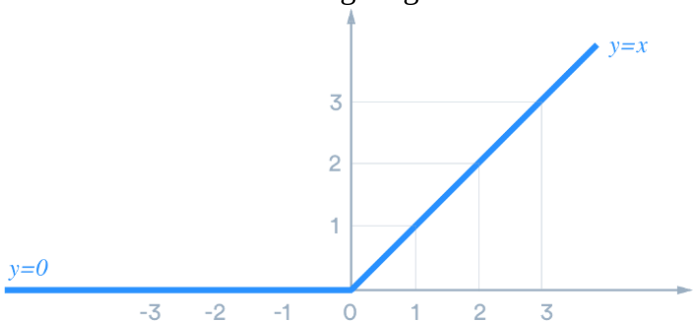
In this project work, two activation functions namely, ReLU and Softmax are used which can be described as:

1. ReLU:

   ReLU stands for rectified linear activation unit and is considered one of the few milestones in the deep learning revolution. ReLU is a non-linear activation function that produces identity output for positive sums. For negative values, its output is zero. If x is weighted sum of neuron then ReLU activation function can be described as –

   $$f(x) = \max(0, x)$$

   Thus it gives an output that has a range from 0 to infinity. Graphical representation of this activation function looks like following diagram.

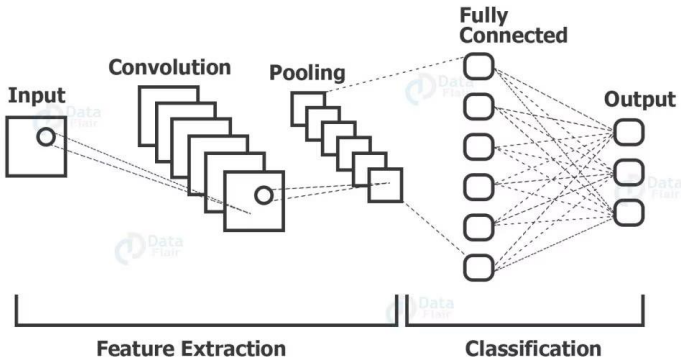


2. SoftMax:

   SoftMax is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. Softmax is used as the activation funtion for multi-class classification problems where class membership is required on more than two class labels. The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1.

4. CNN

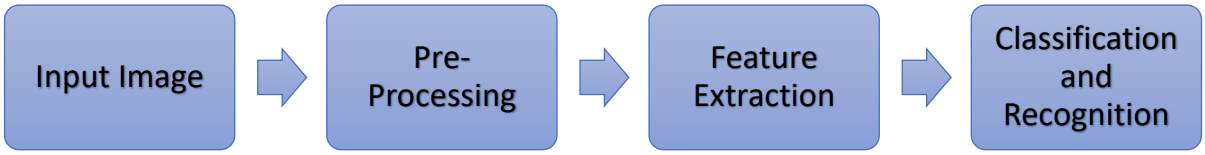
CNN stands for Convolutional Neural Networks that are used to extract the features of the images using several layers of filters.



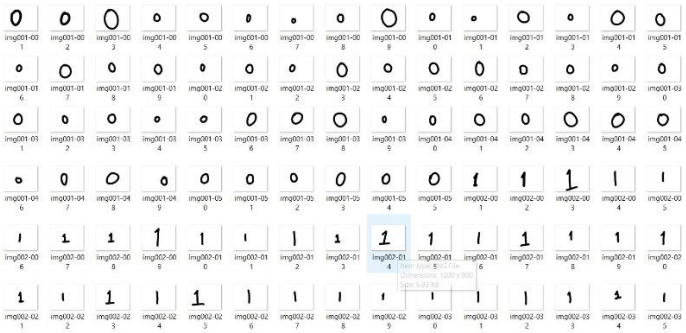
The convolution layers are generally followed by maxpool layers that are used to reduce the number of features extracted and ultimately the output of the maxpool and layers and convolution layers are flattened into a vector of single dimension and are given as an input to the Dense layer (The fully connected network).

5. Project Definition and Method

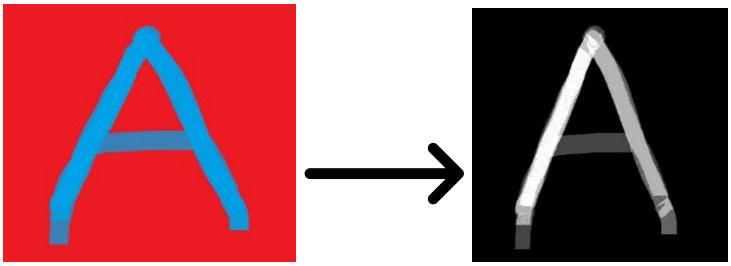
In this project, handwritten characters i.e., digits (0-9) and English alphabets (A-Z, a-z) are recognized. This we are going to achieve by modelling a neural network as shown in fig, that will have to be trained over a dataset containing images of digits and English alphabets.



a. Input image:
Images of handwritten characters are taken as input. The dataset is collected from Kaggle.



b. Pre-processing
Pre-processing takes input image to perform cleaning task. In this task, 3-channel image is converted to single channel.



c. Feature Extraction

Features are extracted from pre-processed image using CNN with ReLU activation function. CNN works on each character image to form a matrix of reduced size using convolution and pooling. Finally, the reduced matrix is compacted to a vector form using the ReLU function. This vector is regarded as feature vector.

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 48, 48, 32)	320
conv2d_3 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_1 (MaxPooling 2D)	(None, 23, 23, 64)	0
dropout_2 (Dropout)	(None, 23, 23, 64)	0
flatten_1 (Flatten)	(None, 33856)	0
dense_2 (Dense)	(None, 128)	4333696
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 62)	7998

Total params: 4,360,510
Trainable params: 4,360,510
Non-trainable params: 0

d. Classification and recognition

The derived feature vector is used as individual input to formulate corresponding class. During the training phase, the parameters, biases, and weights are calculated. The calculated parameters, biases, and weights are used in the testing phase for classification and recognition purposes.

6. Result

Figures show the that with increase in number of epochs, accuracy of trained model increases and value of loss decreased.

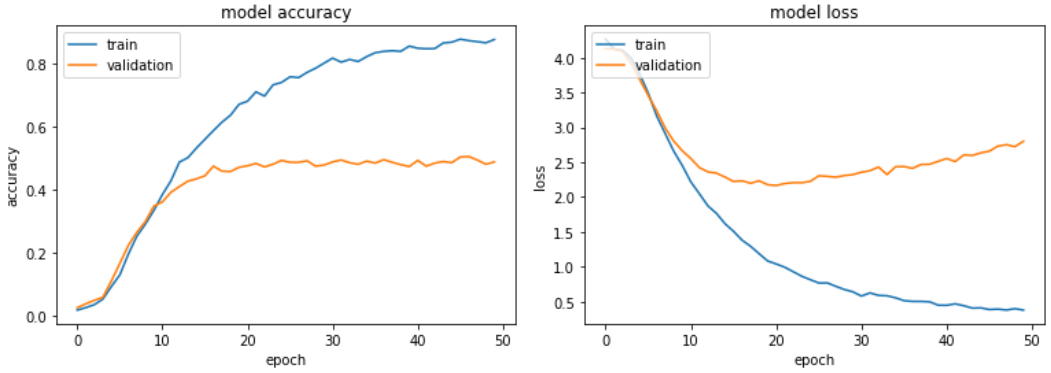


Table shows the result obtained on test dataset. The model achieved accuracy up to 50 percent.

Accuracy	0.4985337257385254
Loss	2.5552594661712646
Precision	0.5784499049186707
`Recall	0.4486803412437439

The results obtained represents that handwriting character recognition performance needs to be improved.

7. Summary and Conclusion

This project is based on Handwritten Character Recognition. For this purpose, a deep neural model is trained on characters (digits & alphabets) dataset. We trained model for 50 epochs and training accuracy reaches up to 80%.

From this project we can conclude that CNN can be used to perform handwritten character recognition and further modifications can make result much better.

Bibliography

1. https://www.pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-with-opencv-keras-and-tensorflow/
2. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
3. https://www.kaggle.com/dhruvildave/english-handwritten-characters-dataset/version/3 [For Kaggle dataset]
4. https://www.tensorflow.org/