CHAPTER 4

DESIGN AND IMPLEMENTATION OF OCR

In the proposed work, the OCR system for Kannada text is implemented and tested**.** This chapter gives overview of design and implementation process of the system. After a brief amount of research, it was found that Convolutional Neural Networks (CNN) is efficient the method that can be used for OCR. The CNN methodology was adopted to make progress with the project. The general outline of the system as discussed below:

4.1 BLOCK DIAGRAM OF BASIC OCR

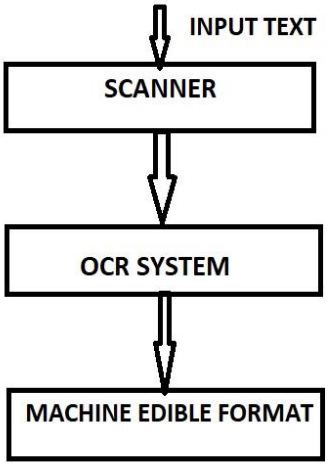

Figure:4.1 Basic OCR System

The Basic block diagram of OCR is shown in the figure 4.1. The printed text document is given to a scanner, which gives out an image. This image, after pre-processing is fed to an OCR system which reads the Character in the image, then recognizes the individual characters and convert the text into a machine edible format (e-book, pdf etc).

4.2 BLOCK DIAGRAM OF OCR

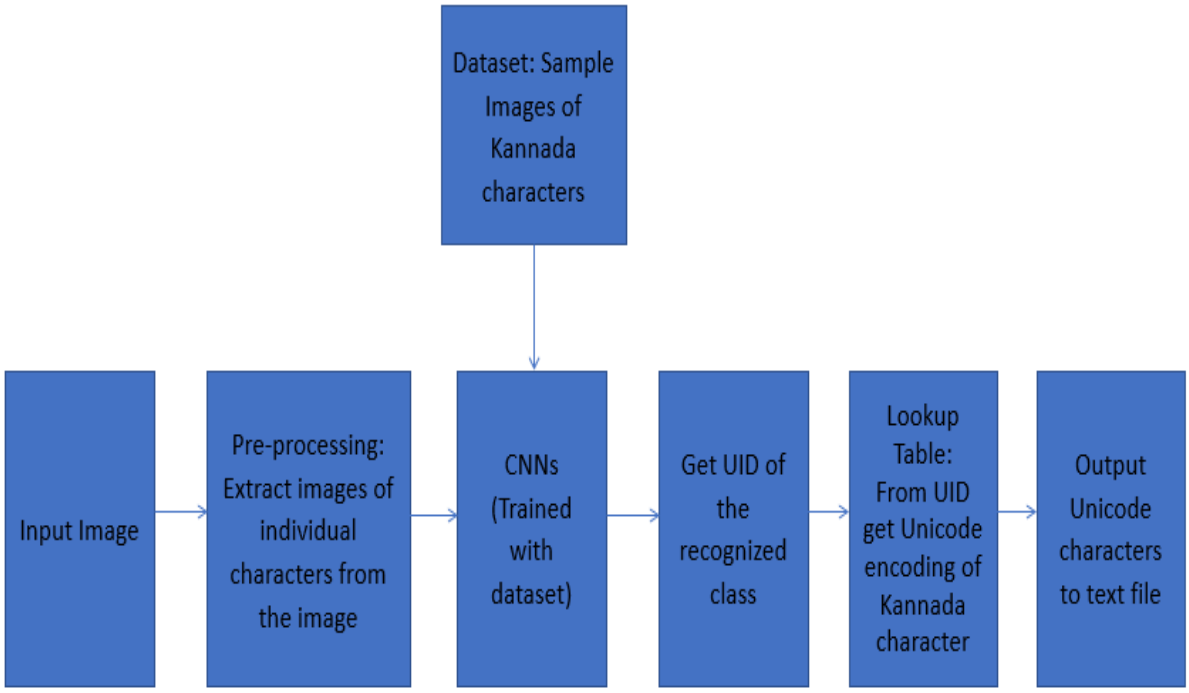

Figure 4.2: System Block Diagram

The simplest form of the system architecture is shown in the figure 4.2. The dataset is divided into Main Aksharas and Vatt Aksharas and separate set of CNNs are trained on them. A lookup table is constructed which consists of the characters, their corresponding UID and their Unicode. The sample image is the printed Kannada text given as the input to the system. The Pre-Processing block helps with the segmentation of words and ultimately characters. These characters are given as input to the appropriate CNN. The CNN predicts the character and using the Lookup Table, a text file is generated.

4.3 PRE-PROCESSING

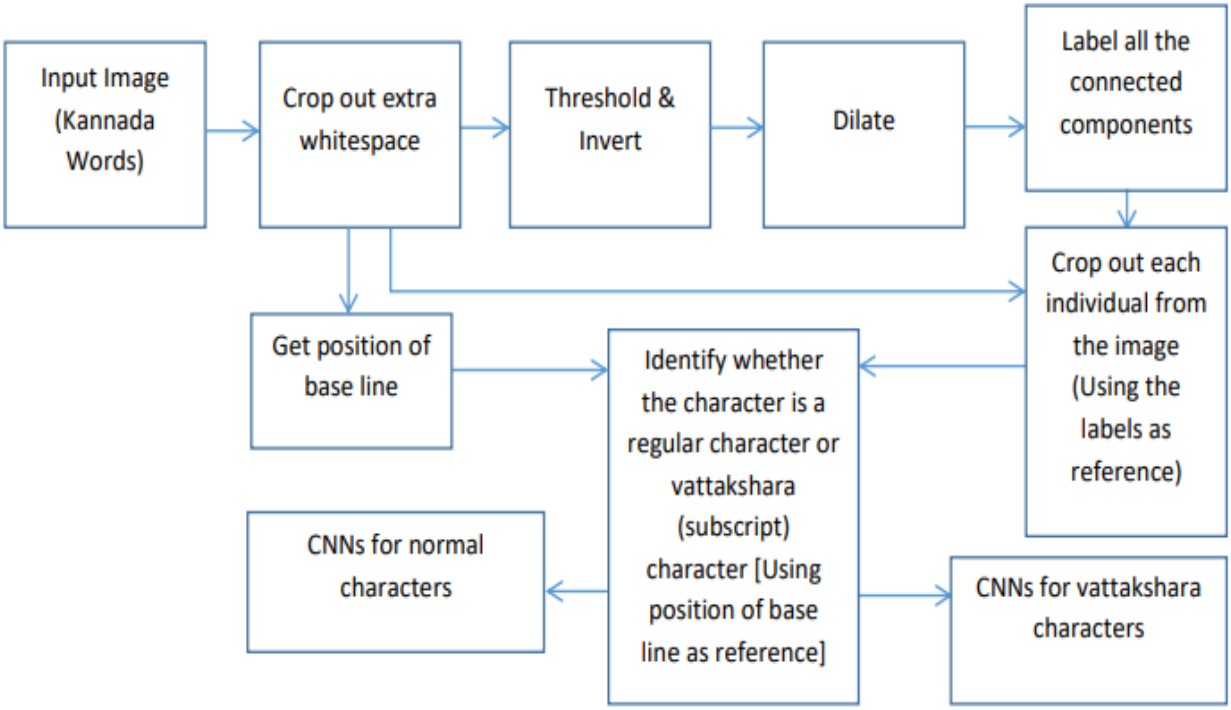

Figure :4.3 Pre-processing Block Diagram

The pre-processing block consists of segmentation of words into characters. The input is an image containing a Kannada word. Initially, we trim out the extra image pixels around the word as shown in the figure 4.5. Next, we identify the position of the base line. Then we perform thresholding to differentiate between foreground pixels and background pixels. The image is then inverted to ensure that the foreground is white and the background is black. It is a convention followed in Image Processing. The image is then dilated to identify all the connected components, i.e. essentially to identify the individual characters. The characters are cropped out and using the position of the baseline, they are identified as either a Main Akshara or a Vatt Akshara and sent to the appropriate CNN for identification. We then crop out our characters. We identify them as regular or vattakshara characters using the position of the base line, then we feed them to the appropriate CNN for identification.

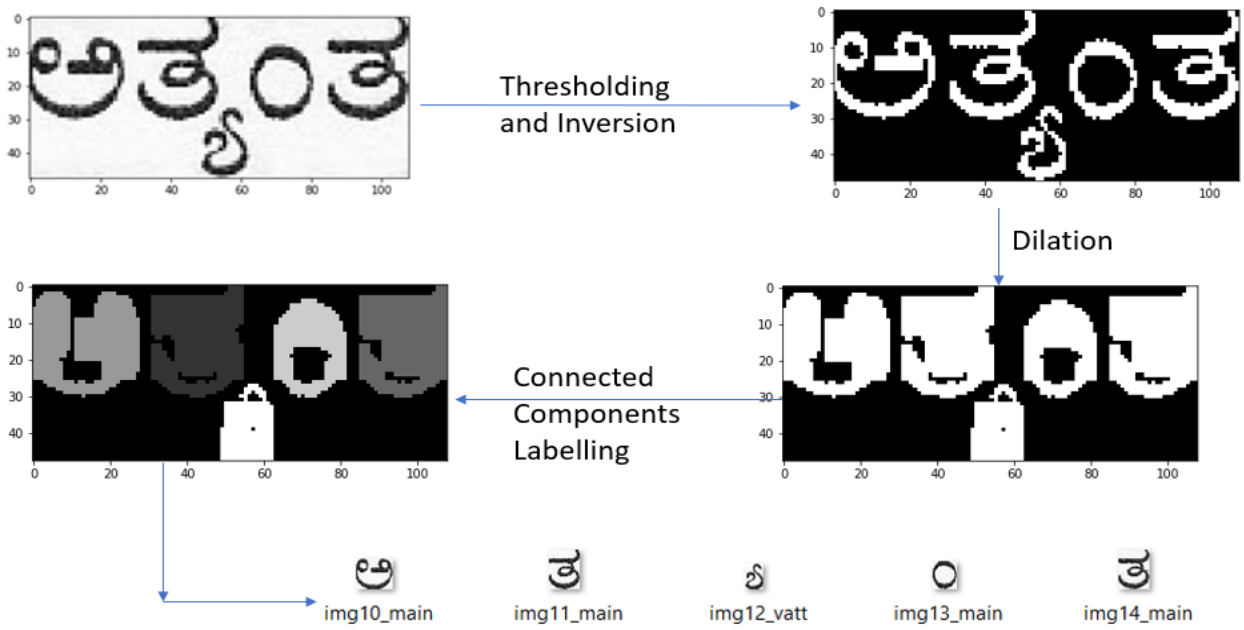

Figure:4.4 Illustration of Character Segmentation

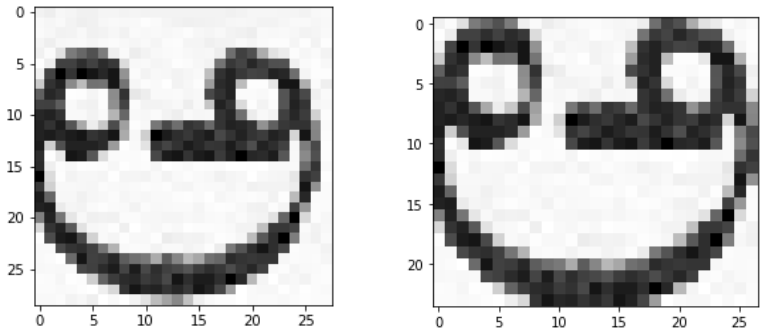

Figure:4.5 Illustration of trimming extra white space around a character

4.4 DETERMINATION OF BASELINE POSITION

To find the position of the base line, we first take an input image of a word and then apply thresholding to differentiate between foreground and background pixels. This is followed by Sobel Edge Detection. Sobel operator performs a 2-D spatial gradient measurement on an image and emphasises regions of high spatial frequency that correspond to the edges. HPP is determined for

the resultant image and when plotted on a graph, two maximas are obtained. The position of the lower maxima is the required position of the baseline.

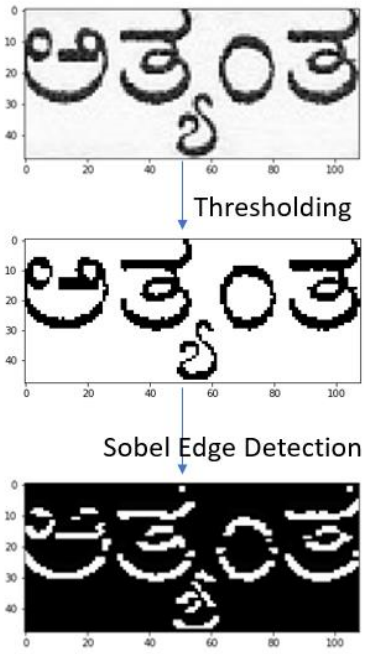

Figure: 4.6 Illustration of Baseline Detection

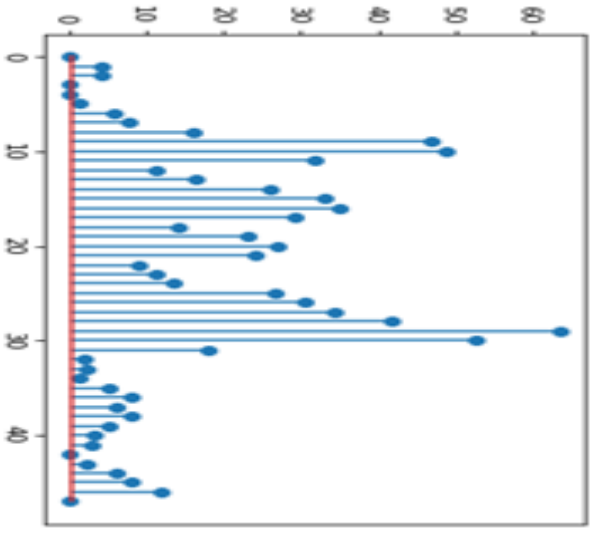

Figure: 4.7 HPP Plot

4.5 NEURAL NETWORKS

Artificial Neural Networks or ANNs are computing systems modelled after and the biological neural networks in animal brains and designed to mimic them. They consist of an input layer and an output layer with multiple hidden layers in between. Each layer consists of a number of processing units called as neurons.

A neuron, the basic unit of a neural network, takes multiple inputs and returns a single output. In a feedforward neural network, the outputs of the neurons in one layer are fed as inputs to the neurons in the next layer. The input layer is the first layer in this chain which takes its input signals from the user and the output layer is the last layer in the chain which returns its outputs to the user. A neural network may have multiple inputs and multiple outputs.

4.5.1 Convolutional Neural Networks (CNNs)

The convolutional layers serve as feature extractors and thus they learn the feature representations of their input image. The neurons in convolutional layers are arranged into feature maps. Each neuron in a feature map has a receptive field which is connected to a neighbourhood of neurons in the previous layer via a set of trainable weights also referred to as filter banks. Inputs are convolved with the learned weight in order to compute a new feature map and the convolved results are sent through a nonlinear activation function. The weights or the filter values, which in the domain of computer vision and image processing were set for various purposes like vertical edge detection, horizontal edge detection, etc, are automatically updated in CNN filters using back propagation algorithm. For the first epoch the weights are usually [111;111;111].

Using, a CNN thereby features are extracted and stacked against each class and is to be used in the future for testing. The process of convolution using filter on selected section and complete convolution is shown in Fig. 4.8 and Fig. 4.9 respectively

The purpose of **the pooling layer** is to reduce the spatial resolution of the feature maps and thus achieve spatial invariance to input distortion and translations. Traditionally average pooling layers were used to propagate the average of feature map sections fed on to it. But max pooling selects the largest elements within its receptive field and is shown in Fig. 4.10 for a completely convolved section

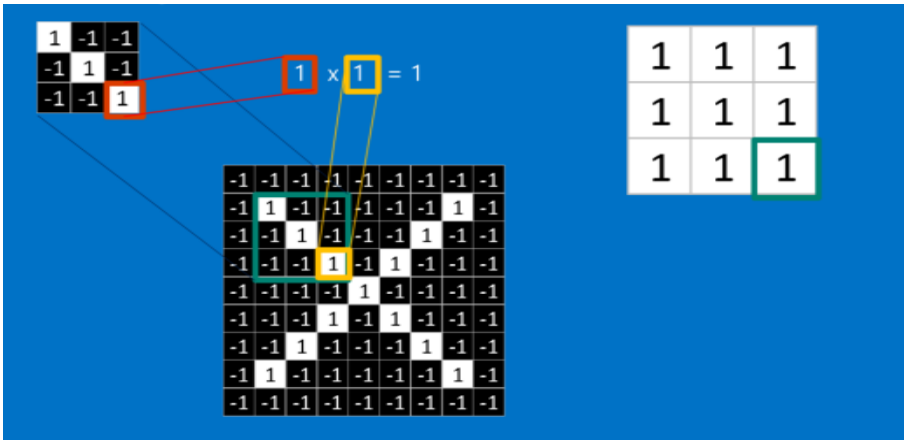

Figure: 4.8 Depiction of Convolution for A Selected Section

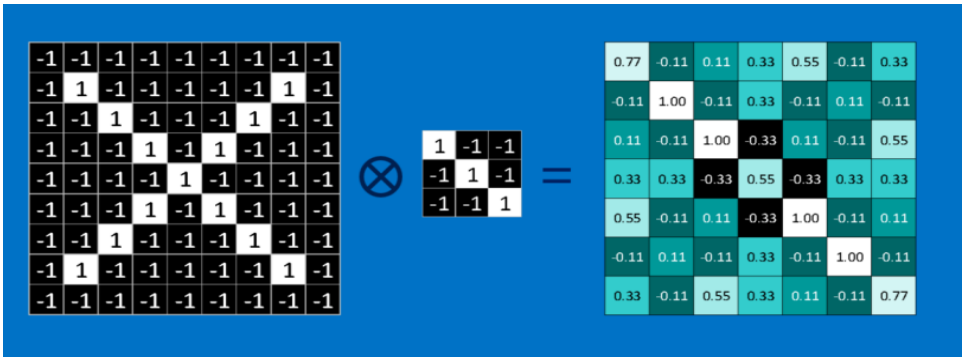

Figure: 4.9 Complete Convolution with A Given Filter

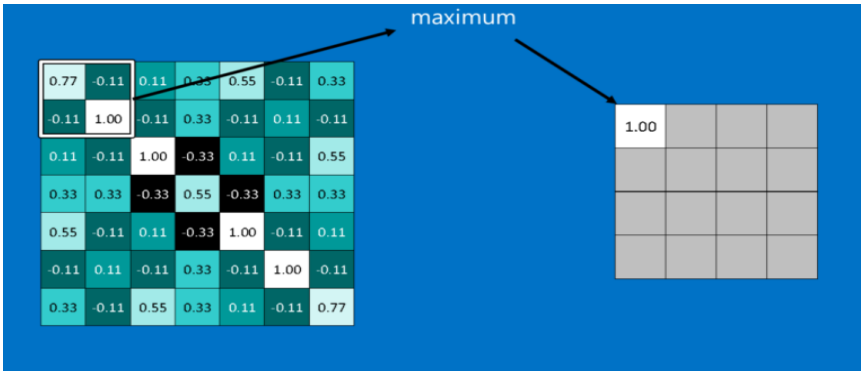

Figure 4.10 Max Pooling Operation

4.6 CNN IMPLEMENTATION

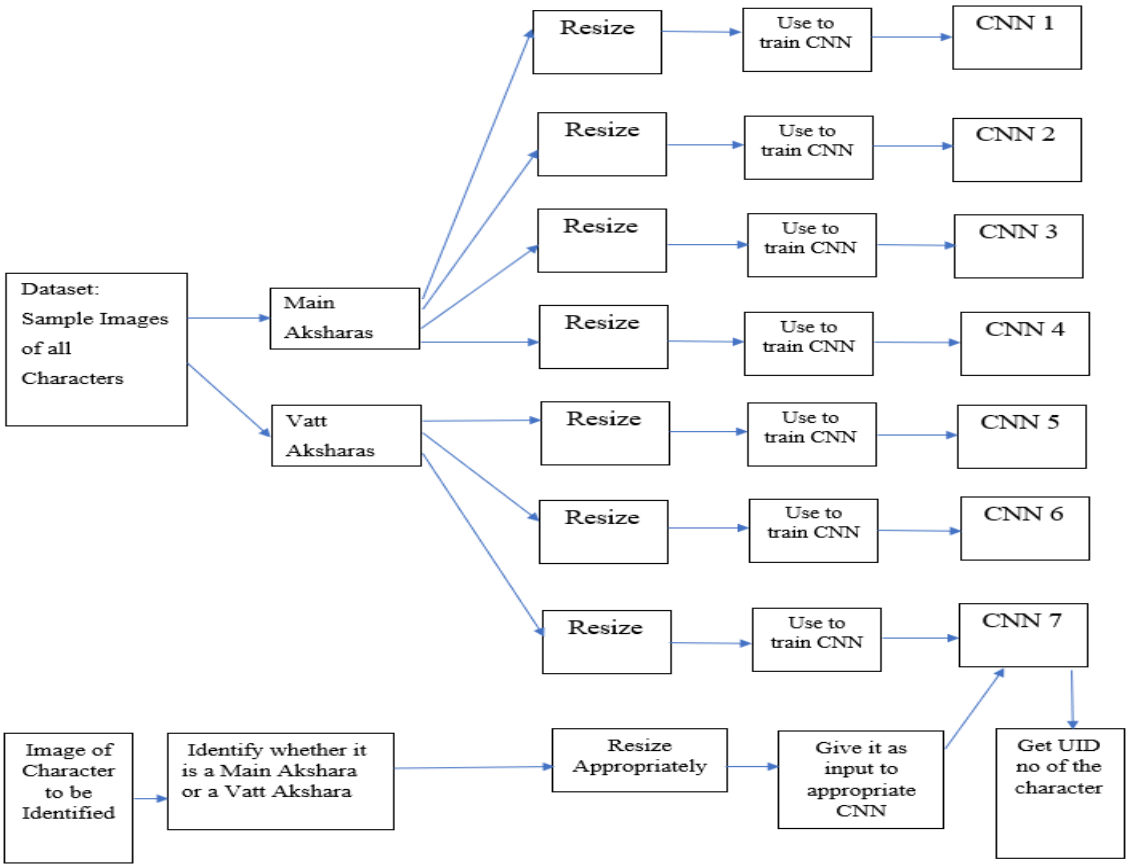

Figure 4.11 Block Diagram Depicting CNN training

For the Main Aksharas dataset, all the images are separately resized to dimensions 15x20, 25x20, 30x20 and 40x20 respectively, thus creating four different datasets which contain the same images but differ in their dimensions. These datasets are separately used to train four different CNN models, namely CNN1, CNN2, CNN3 and CNN4. For the Vatt Aksharas dataset, all the images are separately resized to dimensions 15x15, 20x15, 30x15 respectively. These datasets are again separately used to train three different CNN models, namely CNN5, CNN6 and CNN7. The character to be predicted is determined whether it is a Main Akshara or a Vatt Akshara. In case it is a Main Akshara, it is resized to height twenty keeping the width constant. The Aspect Ratio is multiplied by a factor of twenty. It is assigned to the appropriate CNN by implementing the below mentioned decision rules:

- If the width is less than or equal to twenty, is is assigned to CNN1
- If the width is greater than twenty but less than or equal to twenty-five, it is assigned to CNN2
- If the width is greater than twenty-five but less than or equal to thirty-five, it is assigned to CNN3
- If the width is greater than thirty-five, it is assigned to CNN4

In case it is a Vatt Akshara, the image is resized to height fifteen keeping the width constant. The Aspect Ratio is multiplied by a factor of fifteen. It is assigned to the appropriate CNN by implementing the below mentioned decision rules:

- If the width is less than or equal to twenty, is assigned to CNN5
- If the width is greater than twenty but less than or equal to twenty-five, it is assigned to CNN6
- If the width is greater than twenty-five, it is assigned to CNN7

The idea behind using separate CNNs to predict a Main Akshara and a Vatt Akshara is due to the fact that in Kannada script, several Main Aksharas and the corresponding Vatt Aksharas look similar and if only one CNN is used for prediction, there is a fair chance that it may mis predict. The idea behind using more than one CNN for predicting either a Main Akshara or a Vatt Akshara is due to the fact that a CNN cannot take in inputs with variable dimensions. Using a single CNN for all characters, ignoring the Aspect Ratios will again result in the model mis predicting. Hence, the Aspect Ratios of the characters are considered, and then the characters are resized and fed to the appropriate CNN.

4.7 MODEL ARCHITECTURE

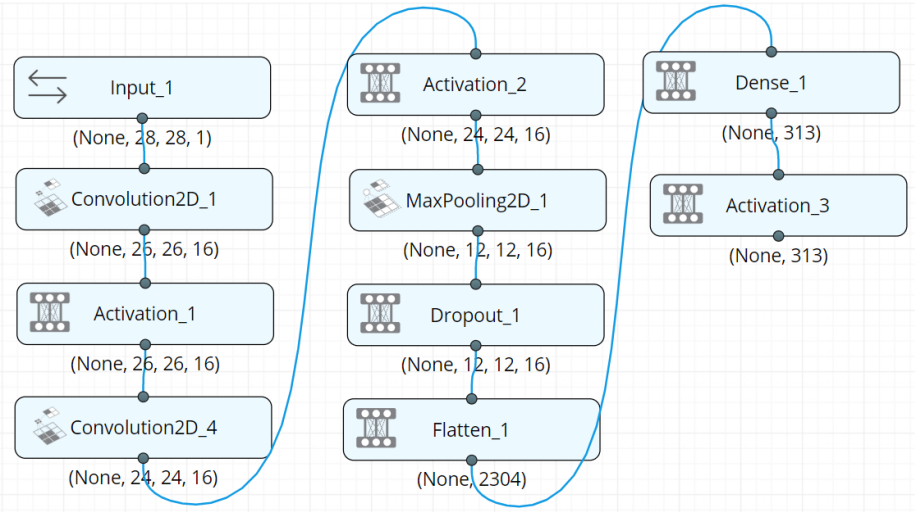

Figure: 4.12 Model Architecture

The CNN model consists of four layers, namely two Convolution layers, one Max Pooling layer and a Fully Connected layer wherein every neuron is connected with every neuron in the previous layer. The Convolution layers perform feature extraction, Max Pooling layer reduces the dimensions of feature maps, thereby reducing the number of parameters to learn and the amount of computation performed by the network. The Fully Connected layer performs the classification task. The two Convolution layers are followed by ReLu activation. It helps to alleviate vanishing gradient problem and introduces an amount of non-linearity in the network. The final Fully Connected layer is followed by Softmax activation. This ensures that the output of all the neurons in the final layer is in the range [0,1]. The output indicates the probability that the input belongs to that particular class. Dropout randomly ignores a certain fraction of the total neurons. It helps to prevent over-fitting and hence helps the model to generalize better.

Hyperparameters:

Loss Function: Categorical cross-entropy

Optimizer: RMS Prop

Number of epochs: 15

Batch size: 32

Activation function: SoftMax

Each model has been trained for fifteen epochs.

The loss refers to the prediction error of the neural network. Loss function is a method to calculate the loss. Batch size refers to the number of training samples utilized before updating the weights. Optimizer is an algorithm used to update the weights of the neural network in order to reduce the loss. All these are defined as hyperparameters since they are initialized at the beginning and cannot be changed during training.

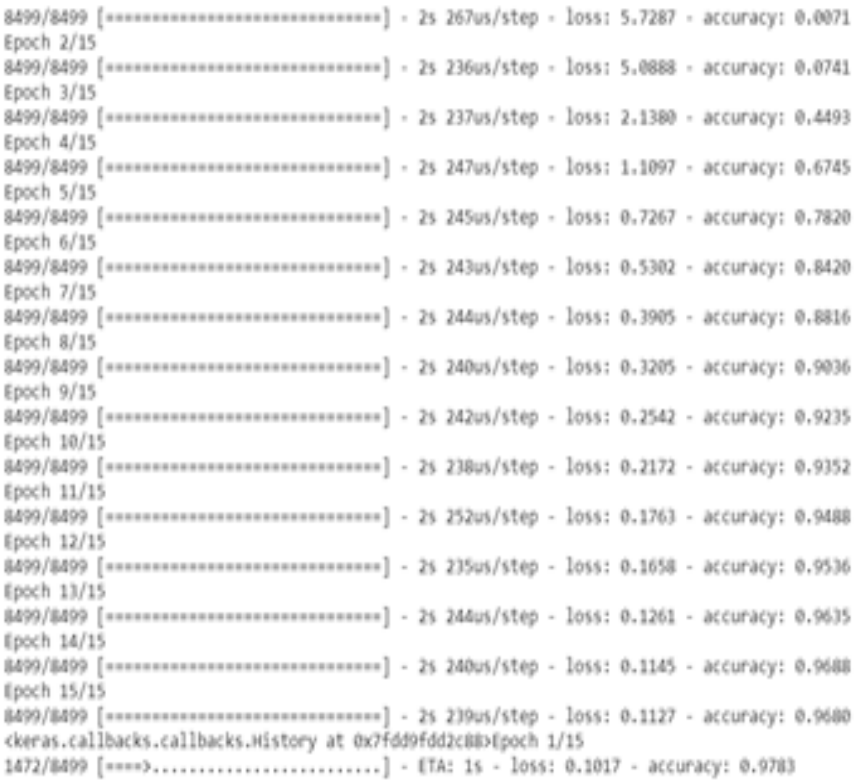

Fig. 4.13 CNN Training