ASSESSMENT 2 - Deep Neural Network

Ryan Needham, 19156892

2022

Contents

1 Introduction

The current report is about the development of an organized deep neural network model that is designed to approach and tackle a specific problem pertaining to an element of neural networks. In this report Convoluted neural networks are created and evaluated based on their performance in efficiency, reliability and accuracy. The final output of the artefact is created by using an image classification data set of known plant diseases that includes both a train and test folder. The report itself is organized into sections that detail a description of the image data set that is being used, and the deep learning problem. Another section on the proposed method which is convoluted neural networks, whilst also mentioning the pre-processing section of the machine learning model. The most important section is when dealing with evaluating the final model and effectively demonstrating an accurate interpretation of the results paired with visualizations to reinforce the results.

2 Problem Statement

The problem being tackled in this report is the detection of plant diseases through using image classification and CNN. The proposed challenge will be completed once an effective Convoluted Neural Network is able to accurately categorize plant diseases and the results evaluated effectively. The industrial data set that has been chosen for this project is the “New plant diseases data set”. The data set consists of over 85 thousand RGB images that show either healthy or diseased leaves, and these images have been categorized into 38 different classes. When accessing the data set it’s key to realize the divide in training and validation which is an 80/20 ratio. The data set includes a collection of images in three different directories, train, valid, and test; because this model is built with these three directories in mind it allows for the model to have predictions that are greater at generalizing unseen data The ideal objective is to solve the problem of sorting the images into the 38 classes with a mild degree of high prediction rate, using deep learning techniques. The technique that will be proposed to use is CNN, which is also known as a Convolutional neural network. CNN is a method that relates to deep learning that helps solve problems including analyzing imagery. The main objective of this project is to be able to create a neural network model that helps tackle an industrial data set problem, throughout the project a clear implementation of theoretical concepts that relate to neural networks should be explained and used. It should be mentioned that there are various leaves, that are being used within this data set. The training data will include these leaves that have a variety of different diseases and are shown healthy as well at the same time, because of this the final model would most likely be ineffective at detecting diseases that are not used in the training data set or leaves as well.

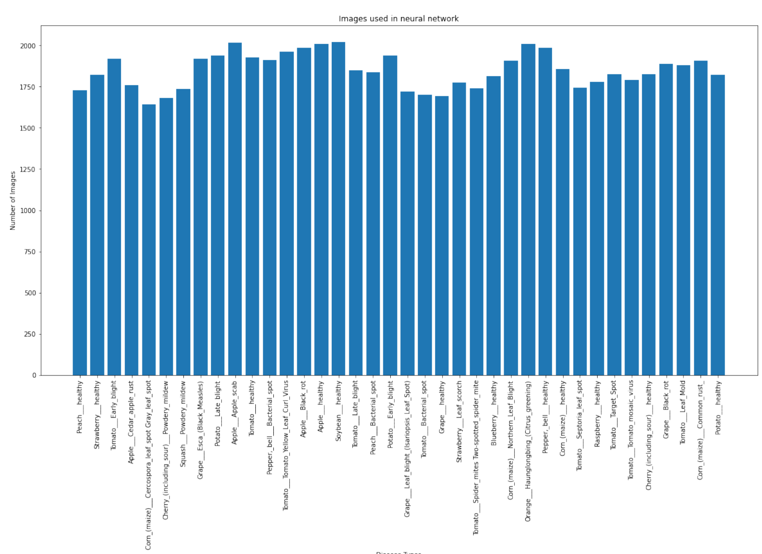

Figure 1: Number of Images Per Class

3 Proposed Method

The main parameters that are going to use in the creation of the neural network are batch size, image height and image width. One of the hardest things to select was the batch size; because throughout the project depending on the size of the batch it would drastically change the training efficiency times. For the project there was a batch size of 100 with an epoch of 5; which was able to be completed within 2 minutes per epoch.

Activation Function 'ReLu' is known as a rectified linear activation function which allows for a linear function that will output the input directly if it is positive, throughout the project, this is mainly used activation function alongside the 'SoftMax' activation function. The 'SoftMax' function is mostly used to help with the sparse categorical cross-entropy in the compile stage, but it ultimately allows for testing reliability of the model using a loss function. The difference between this activation function and something like 'Sigmoid' which was not used in the project is that 'Sigmoid' is [primarily used for binary classification for example when using the famous 'MNist Dataset'

The loss function is a way to measure the neural network of its error through each epoch when training the testing the data. This method of measuring the error allows for the model to make the necessary changes to greater lower the error. For the neural network, we will be using a cross-entropy loss function which is typically used in neural networks. For this specific project, we will be using 'sparse categorical cross-entropy which is a loss function that is typically used when dealing with a multi-class classification class instead of a binary one, the multiple classes that are used represent the multitude of plant diseases that have been labeled and will be predicted.

Auto Encoders Convolution autoencoding is an alternate method that could be included in the final model. In essence auto encoding is the idea of dimensional/noise reduction that compresses and reconstructs the input of an image within a neural network. There is a slight negative in auto encoding that due to the reconstruction of images there is a possibility of losing information so it's mostly used when trying to de-noise certain images as there's not that much information to lose but to gain.

Architecture In this section, there will be a discussion of the important layers being implemented into the CNN to help explain the method. A Pooling layer is another layer of concept that will be used in the creation of the neural network, a pooling layer is used to help decrease the size of the image. This layer will assist in the speed of the computation, and the detection features are slightly more robust. The layer itself also uses kernel image processing. Essentially the pooling layer applies similar effects to that of PCA. Flattening is another layer that is used in the CNN model which helps convert the 2-dimensional or 3-dimensional arrays into a single vector, to then process the input layer and build the neural network and help pass from every neuron. In this CNN we use the Dropout function to help generalize, instead of summarizing each example by skipping or dropping some of the weights or connections of the layers. Within this module, the rate within the hyper parameter is set at 0.5. One note that should be mentioned is that neural networks rely heavily on computer memory; because of this 'Google Colab' was the main piece of software used and allowed access to a feature known as 'GPU hardware acceleration'; because of this feature the allocated time to train the neural network was drastically increased from one hour to two minutes per epoch.

Data Augmentation was a feature that was left untouched throughout this project but it should be noted that it's greatly important for combatting overfitting. Data augmentation is a tool that allows for adjustments to be made to the images being inserted within a neural network; one such example of this is enlarging the dataset by rotating or scaling the size of different images and inserting them back into the dataset to better equip the neural network for training. Kernal Processing One of the key things to look at in CNN is Kernal image processing within the convolution layer. This particular process will involve a kernel of size typically 3x3 scanning through an image and carrying out the convolutional operation. The objective of this image processing using a kernel is to extract features such as edges, from the input image or by adding more convolutional layers and even more in-depth features such as texture, which will be key when detecting diseases in plants. In essence, the convolutional layer transforms the input image to extract features. The dot product of the kernel and the image is what creates the convolved feature. When using multiple channels or colored images, the image and matrix will gain an adding depth of 3, creating a matrix and image size of something like 3x3x3. The channels in the kernel must be the same size as that of the image. It should also be noted that CNN is a linear operation, which would require the use of a linear activation function, typically these are either SoftMax or ReLU. The activation function will allow for the output to have increased non-linearity.

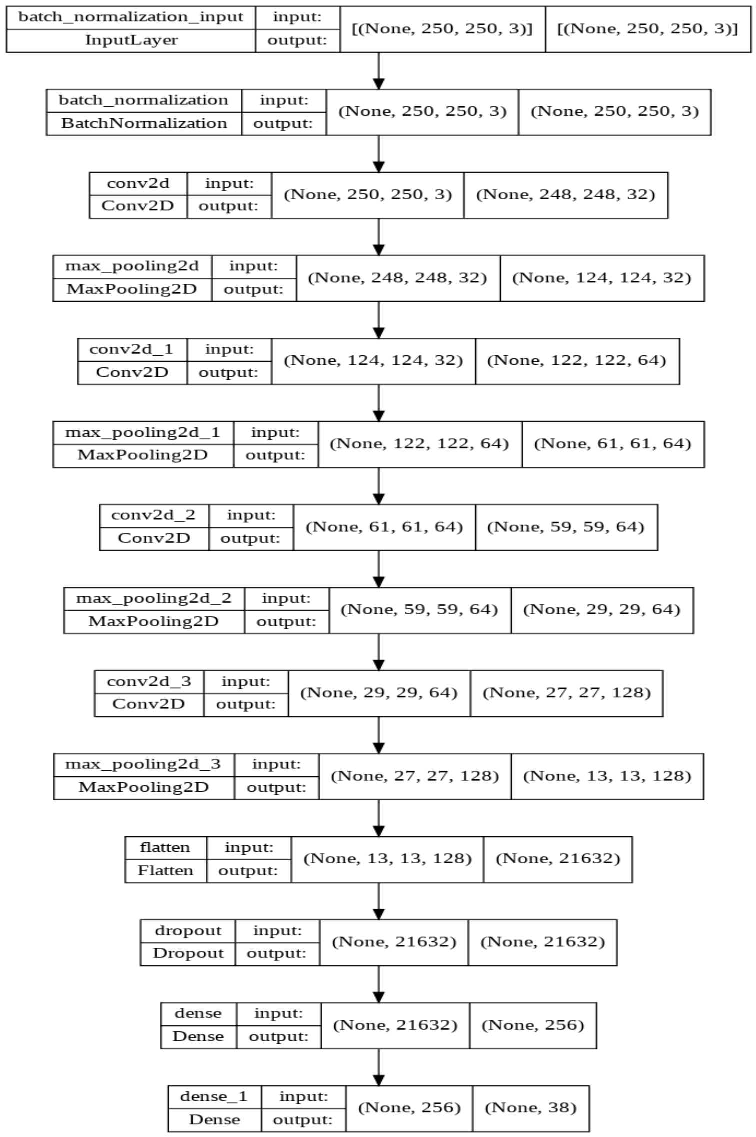

Figure 2: Architecture of CNN Model

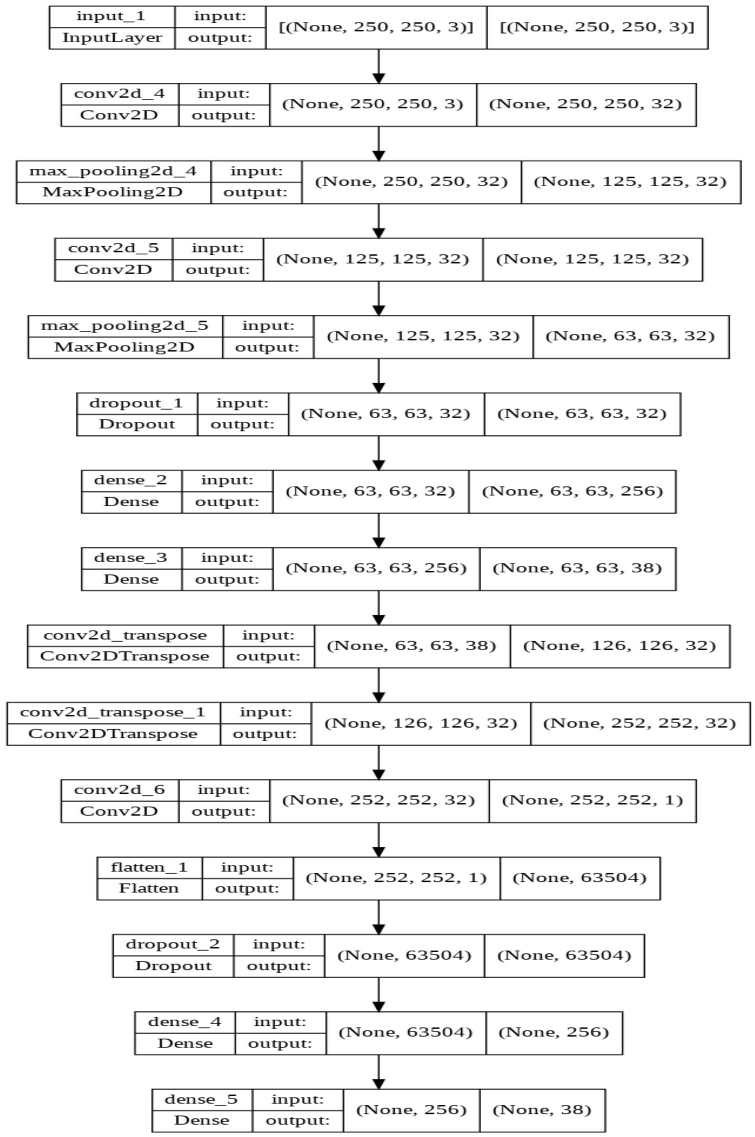

Figure 3: Architecture of Convoluted Auto Encoder

4 Experimental Results

The first thing that should be mentioned is the parameter settings that were used in the final model of Convoluted Neural Network. The batch size was at 100, this would allow for the GPU acceleration to be most efficient and allow for quicker training times of the training dataset; a lower batch size was attempted but this ultimately lead to too long of a batch time. Image Height and Width were both set at 250, a larger image size was chosen in order for the kernel to be able to identify the smaller disease symptoms within the leaves, it should be noted that the height and width of the image was minimal in the deterioration or improvement of the model. The model includes four different convolutional layers and max pooling in hopes of reducing the feature maps which ultimately make the model more efficient

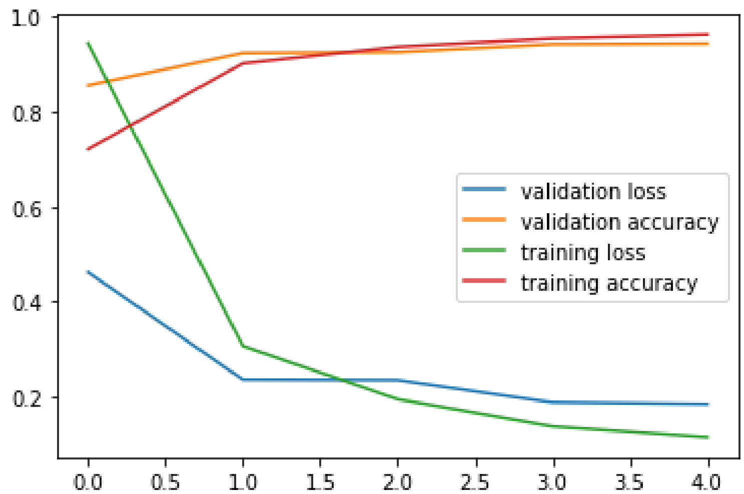

Figure 4: Training and Validation: Loss/Accuracy

In order to most effectively analyze the performance of the model it is best to visualize it into a graph of the validation and training metrics which include the loss and accuracy. The training loss is there to help indicate how well the model is fitted to the training data while the validation indicates how well the model has fitted to the new data. Seeing as the validation and training loss are modelled quite close together it is able to show that the dataset is predicting both at a similar rate and the final model represents both sets quite well. It seems evident that from the visualizations the model is currently able to avoid any high variance or overfitting; as the model is able to adapt to new incoming data and accurately predict it. The final accuracy of the model of the validation data was 95

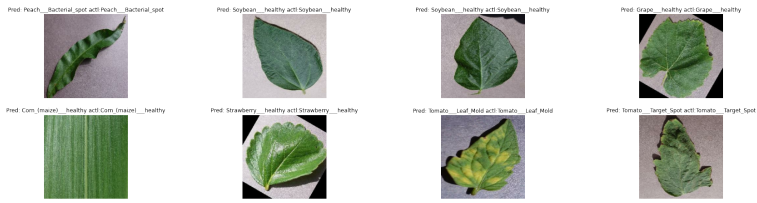

Figure 5: Prediction of Plant Diseases

5 Summary

The main findings of this report show that the a convoluted neural network is capable of tackling a large dataset that revolves around an image classification task. The development of problem also showed how the parameters of the CNN directly impact the training speed of the model and also it's efficiency. A Convoluted Auto Encoder was also looked into and while it didn't harm the accuracy on more noisy images it would have the potential to denoise them and go back to predicting the correct plant diseases. For future improvements of an auto encoder it would be greater to noise the images beforehand and enter them into the autoencoder. The project showed great efficiency in avoiding high variance or bias which could possibly be due to the dropout layer within the convoluted neural network.