Analysis and Report on Model Performance

Beltrán Hilda, A01251916, Tecnológico de Monterrey Campus Querétaro

Abstract – This report presents the implementation and analysis of the performance of a Convolutional Neural Network. As well as some modifications to improve the model's performance, using Regularization and Hyperparameter Tuning Techniques.

I. INTRODUCTION

Plant diseases are conditions caused by various microorganisms that can have detrimental effects on the health and growth of plants. These diseases can manifest in various ways, including leaf spots, discoloration, and even plant death. In this case, we aim to detect different diseases affecting tomato leaves.

Early detection of these diseases can greatly benefit the agricultural industry by minimizing the economic impact they pose, as well as by preserving ecosystems, as animals can also be affected when consuming these plants.

Utilizing Deep Learning algorithms is an effective approach for timely disease detection. This allows farmers to take prompt action to maintain food security and reduce the reliance on pesticides.

II. DATASET

The dataset used for this analysis is from the Kaggle platform, there's a train and a validation folder. Each folder contains images for each category of disease for the tomato leaves. The dataset contains the following categories to generate predictions:

- Tomato_mosaic_virus
- Target_Spot
- Bacterial_spot
- Tomato_Yellow_Leaf_Curl_Virus
- Late_blight
- Leaf_Mold
- Early_blight
- Spider_mites
- Two-spotted_spider_mite
- Tomato___healthy
- Septoria_leaf_spot

III. MODEL PROPOSAL

Convolutional Neural Networks (CNNs) have changed the field of computer vision and are used to classify images, object detection and image segmentation. A CNN was used to train a model for image classification for diseases of tomato leaves.

The model is constructed as a sequence of layers, starting with a Convolutional 2D layer, which has 32 convolutional filters, in 3x3 grids that slide over the image to extract

local patterns. The padding used is 'same' which gives an output with the same spatial dimensions of the input data. The activation function is ReLU, Rectified Linear Unit, which is helpful for the model to learn complex features.

Next, we have a Max Pooling of size 2x2, which reduces the dimensions of the feature maps, only leaving the most relevant information and reduces the resolution in half. Since the objective is to reduce the computational complexity of the training process.

A Flatten layer is included since we want to process the data in a 1D vector, and it was stored in a 2D feature map. This is useful when fully connected layers are used.

There's three dense layers, the first one has 128 neurons, which helps the model learn intricate patterns with a ReLU activation function. The next layer has 64 neurons, which helps with processing the learned features. The last layer has 10 neurons and a softmax activation function, which produces the classification probabilities.

The optimizer used in this case is Adam, thanks to its gradient-based optimization, the loss function is categorical cross-entropy since we are dealing with a multi-class classification task. Accuracy is chosen as the metric for the training process.

The model is then trained using a generator that feeds batches of training data. It undergoes 30 epochs, which means it iterates through the entire training dataset 30 times. Each epoch refines the model's parameters, gradually improving its ability to make accurate predictions.

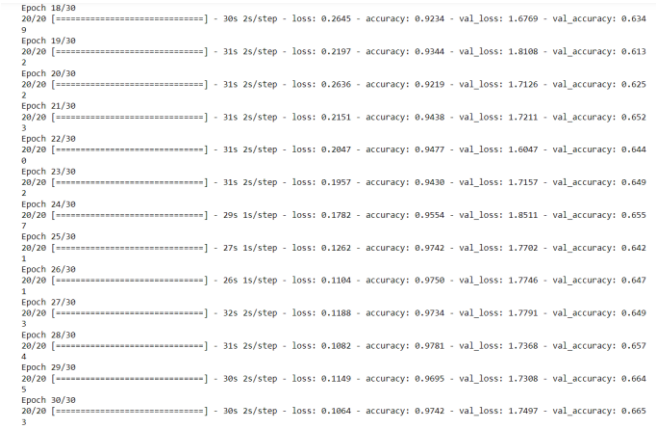

Fig 1. Training and Validation Accuracy

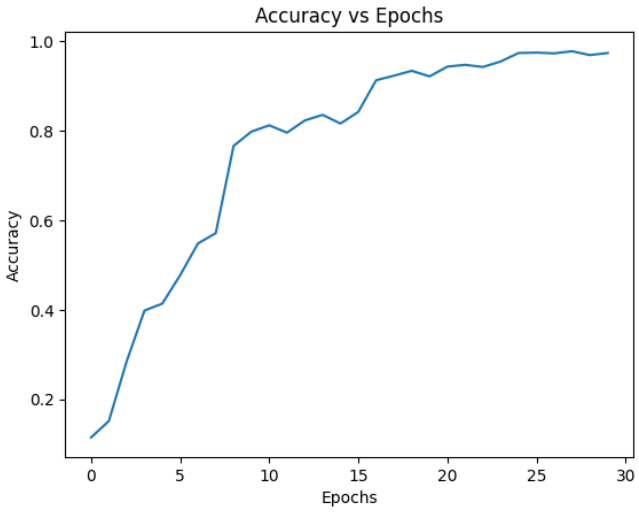

Fig 2. Accuracy of the Model

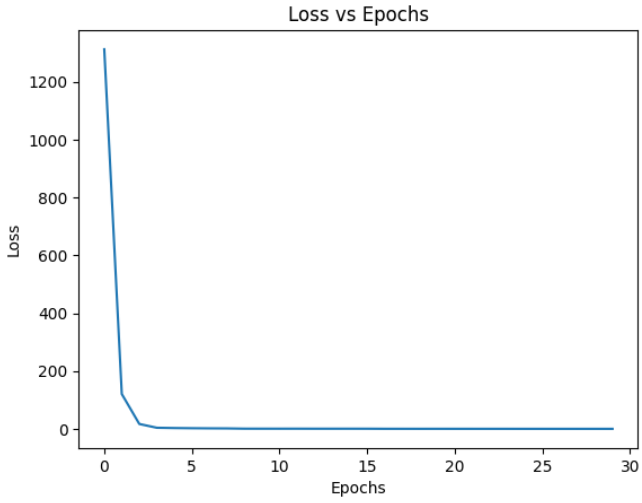

Fig 3. Loss of the Model

We can see there's Over Fitting in the model's training since the Training's Accuracy is significantly higher than the Validation's Accuracy. Meaning the model's configuration is not optimal, and there needs to be an adjustment in regularization techniques and hyperparameter tuning.

IV. MODEL IMPROVEMENT

When evaluating the last model, there was a detection of overfitting, since the training accuracy significantly exceeded the validation accuracy, signaling that the model was memorizing data instead of learning patterns. Making it hard to generalize to unseen data, since its performance was in fact no good.

To combat overfitting, there is an introduction to regularization techniques to the model architecture. Dropout is a powerful regularization method which randomly deactivates a fraction of neurons during training.

Compared to the last model, a Dropout layer was added after the first dense layer, with a dropout rate of 0.5. Which introduces a controlled level of randomness when training, diversifying the learning process.

Additionally, an extra Conv2D layer with 16 filters and a ReLU activation function was inserted. This facilitated the extraction of deeper and more complex features from the input images.

The transition also involved switching from Adam to RMSprop. This change was made after considering the unique characteristics of the problem at hand and the desirable properties of RMSprop for certain scenarios.

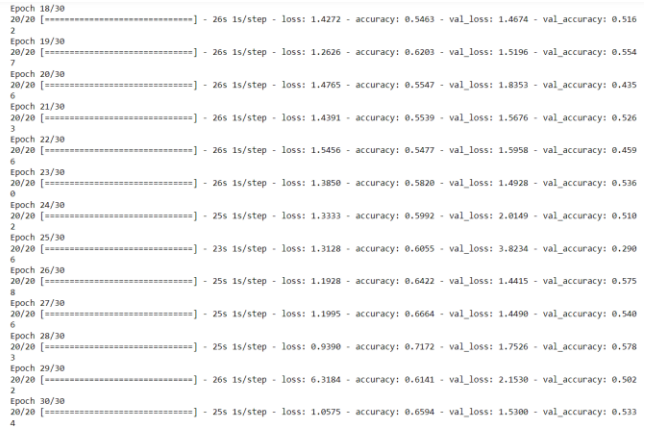

Fig 4. Training and Validation Accuracy

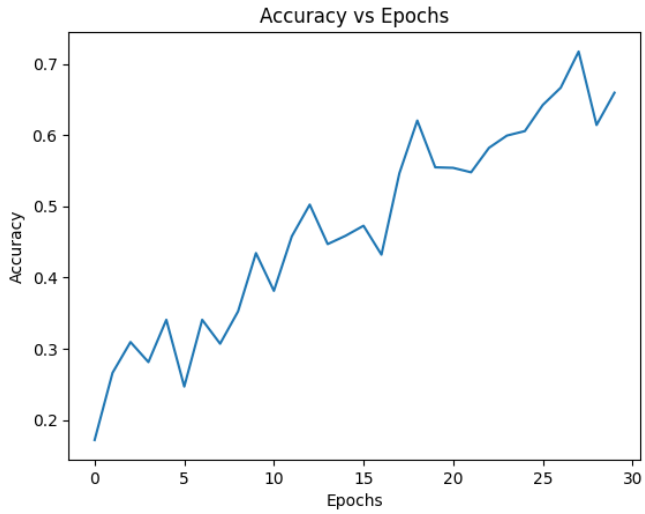

Fig 5. Accuracy of the Model

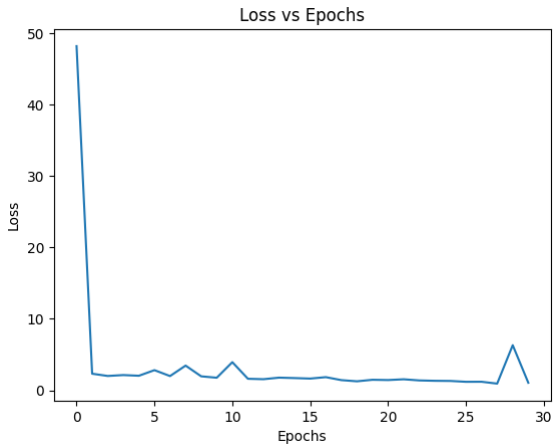

Fig 6. Loss of the Model

V. CONCLUSIONS

After modifying the hyperparameters, the performance of the model was significantly better, since the model is not memorizing data, noise and it's generalizing. Concluding that the second model is a better implementation for a classification model, working best for the kind of classification problem we want to solve in this case. Even though the results were good, it's not enough to predict the probability of classifying the disease. That's why the model must be complemented with other hyperparameters, data cleaning and transformation to use the features more relevant to the situation.

VI. DATASET RETRIEVAL

The dataset was taken from Kaggle's Tomato leaf disease detection Dataset, and it can be found in this link:

https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf

VII. DATA

https://drive.google.com/drive/folders/1BPDyo_2PpnQw3MjVPz-gaXx8VnUw54TI?usp=sharing