

Student-ID : 2233800

Technique in Machine Learning and Artificial Intelligence(C7082)

Link to my github: [\[https://github.com/mayowa-data/C7082-Assessment\]](https://github.com/mayowa-data/C7082-Assessment)

[https://github.com/mayowa-data/C7082-Assessment]

Abstract

TensorFlow is an open-source machine learning and deep learning framework powered by google, which is suitable and convenient to build the current mainstream deep learning model. (Liang et al 2019)

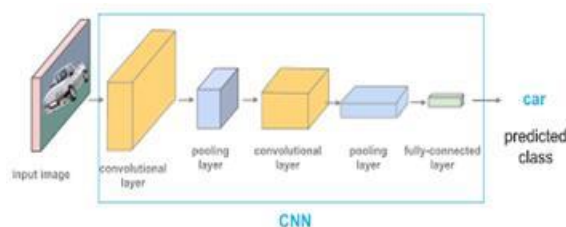
Convolutional neural network (CNN) is a classical model of deep learning, whose advantage is its powerful feature extraction capacity of convolutional blocks. This assessment aims at accurately classifying the images of Agriculture crops. The application of CNN is important to Researchers, farmers, and consumers of various crops. It helps in the identification of various classes of crops, crop diseases, and robotic harvest by farmers while also helping consumers recognize fresh farm crops in stores and supermarkets.

In this assessment, which focuses on the application of CNNs to image classification tasks, a 6-layer CNN model was used, there are four conv2D layers and 2 fully connected layers, and Adam's optimizer to analyze Two Kaggle datasets consisting of 770 images belonging to 30 classes were used in this classification. Assessing the model accuracy base on batch size: (180X180), (90X90), and (Epochs at 50). It discovered that the dataset with 30 classes was burdened by overfitting compared to the dataset with 2 classes.

Background

This essay discusses the implementation of an image classification model for identifying different types of crops. (Asif et al 2021). The goal of using this technology is to improve the accuracy of crop identification systems on farms by farmers, researchers, and consumers in supermarkets. The model is based on convolutional neural networks (CNNs), which have multiple layers that can learn various features. CNNs are commonly used for image classification, object recognition, and speech recognition, and have achieved high levels of accuracy in these tasks. Which forms the interest of this assessment.

A convolutional neural network (CNN or ConvNet) is a type of feed-forward neural network that is commonly used for image analysis. It uses a grid-like topology to process data and is composed of three main layers: a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer is the core of the CNN and performs the bulk of the computation. It applies a set of learnable parameters, known as a kernel, to a restricted portion of the input data, known as the receptive field, using a dot product operation. This allows the CNN to detect and classify objects in an image.



source: google images.

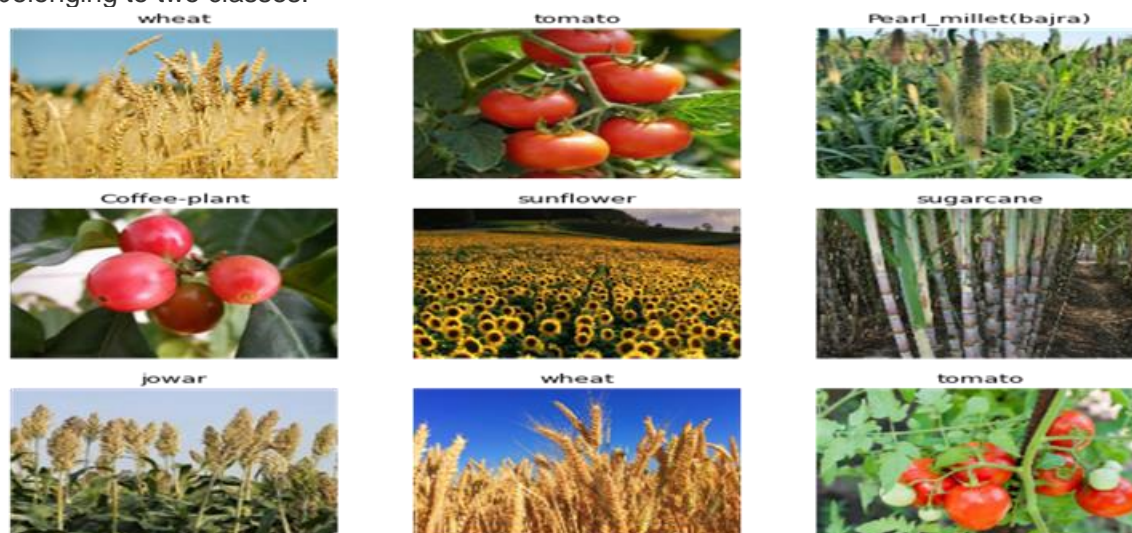
Method

This section describes the methodology and the dataset used for this assessment. The convolutional neural network was used for this assessment as stated earlier. The Model was used in training a dataset of 770 images across thirty categories and achieved an accuracy of 99% training accuracy but the validation accuracy stalled at 25%. It was also used in training another fruit dataset with 297 images across two categories with 86% training accuracy and 71% validation accuracy. This assessment will focus on evaluating the effect of image size and class size on the training and validation accuracy of the training model.

- o The training and validation accuracy when the image size is set at 180X180, batch size = 32
- o The training and validation accuracy when the image size is set at 90X90, batch size = 32
- o While also comparing the outcome with testing the model on another dataset with 297 images and 2 classes (Apples and Tomatoes)

Dataset

The dataset used in this essay is sourced from Kaggle and contains images of 30 different classes of crops, including cherry, coffee-plant, cucumber, fox_nut (Makhana), lemon, olive-tree, pearl_millet (bajra), tobacco-plant, almond, banana, cardamom, chilli, clove, coconut, cotton, gram, jowar, jute, maize, mustard-oil, papaya, pineapple, rice, soyabean, sugarcane, sunflower, tea, tomato, vigna-radiati (Mung), and wheat. The images in the dataset are RGB, with three color channels (red, green, and blue), A total of 770 images belonging to 30 classes, split into 60% training and 40% validation. The size of the images was set to 180 X 180 and a batch size of 32. Nine randomly selected images from the dataset are shown in the assessment and another dataset of 297 images belonging to two classes.



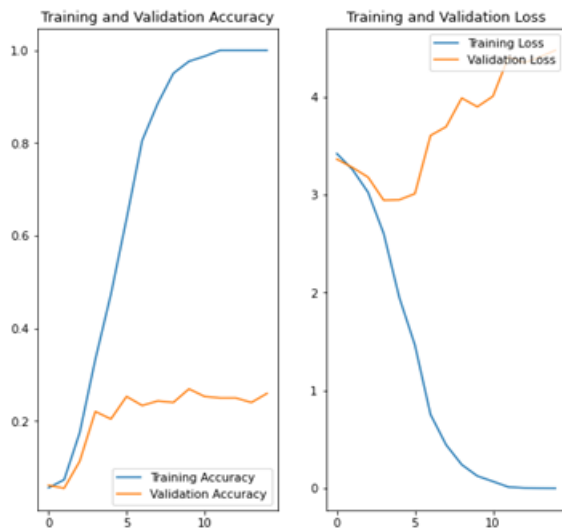
Result and Discussion

Experiment 1: Image size of (180 X180) and trained on 15 epochs

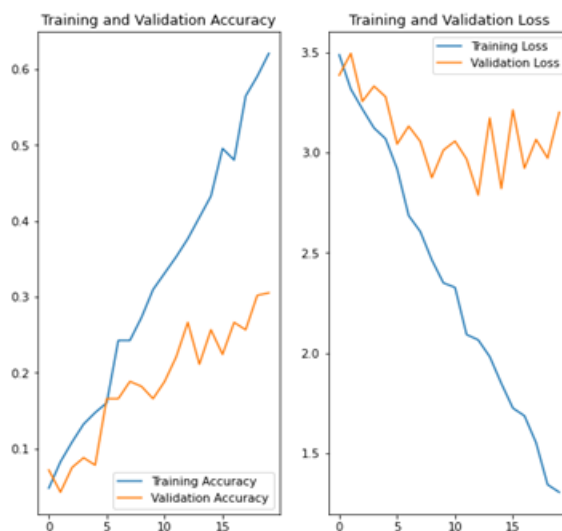
The Keras Sequential model consists of three convolution blocks (`tf.keras.layers.Conv2D`) with a max pooling layer (`tf.keras.layers.MaxPooling2D`) in each of them. There's a fully-connected layer (`tf.keras.layers.Dense`) with 128 units on top of it that is activated by a ReLU activation function (`'relu'`). (tensorflow) training a dataset image size is 90x90 and the total number of parameters is

over 1 million, and batch size of 32 at 15 epochs. With this combination of training, we got a training accuracy of 99% and a validation accuracy of 25%. This suggests overfitting due to a large discrepancy between the training accuracy and the validation accuracy. The dataset was improved using data argumentation and dropout before retraining with 20 epochs to reduce the overfitting. The training accuracy dropped to 57% and the validation accuracy increased to 30%. The graph below shows the position of the dataset after augmentation.

As illustrated in the graph below;



Before Augmentation



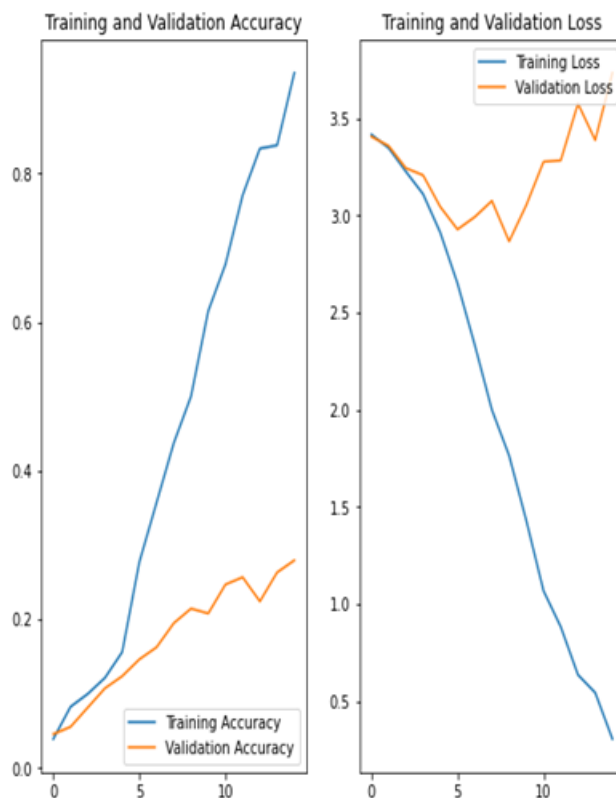
After Augmentation

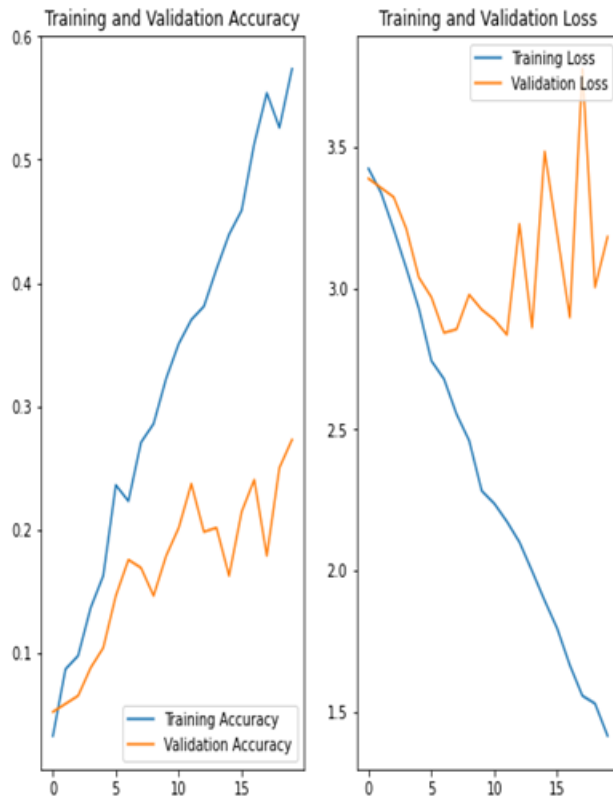
Graphs showing the training/validation accuracy and training and validation loss 180X180

Experiment 2: Image size reduced by half (90 X90) and trained on 15 epochs

The first training outcome was not encouraging due to the effect of overfitting and low training accuracy after augmentation was applied. Therefore, the image size was reduced by half (90X90) to see if there will be an improvement or possibly fixed the overfitting.

The same model as above was used to train a dataset image size is 90x90 and the total number of parameters is over 1 million, and batch size of 32 at 20 epochs. With this combination of training, we got a training accuracy of 94% and a validation accuracy of 28%. This suggests overfitting due to a large discrepancy between the training accuracy and the validation accuracy. The graph below shows the result of the training on the dataset. The dataset was improved using data argumentation and dropout before retraining with 20 epochs to reduce the overfitting. The training accuracy dropped to 62% and the validation accuracy increased to 28%. The graph below shows the position after augmentation





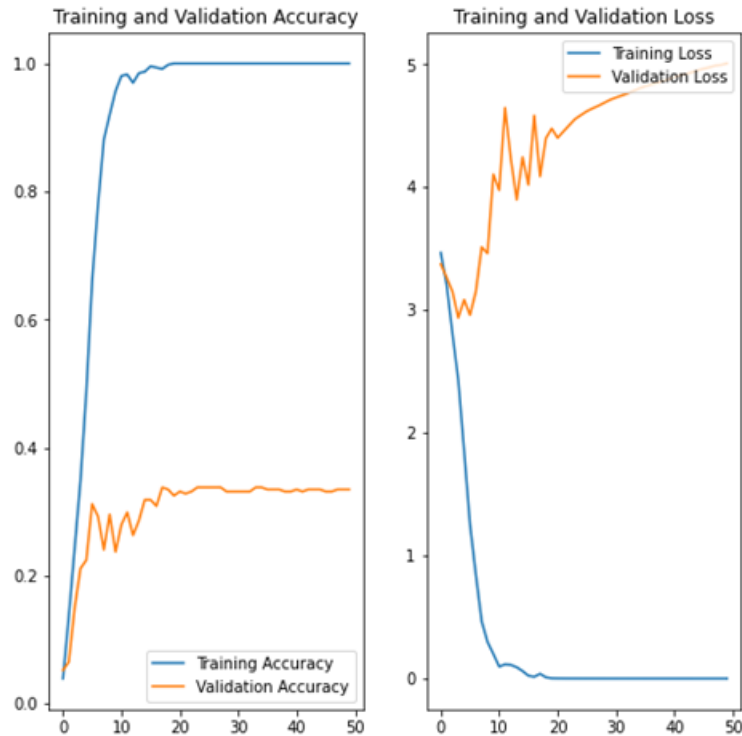
Before Augmentation

After Augmentation

Graphs showing the training/validation accuracy and training and validation loss at 90X90 image size

Experiment 3: Increased Epochs at Epochs= 50

The overfitting in the two experiments above is a great concern and so the epochs time was increased to see if it can help improve the overfitting. The graph below shows the result. The training was repeated with a batch size of 32 in 50 epochs. With this combination, we got a training accuracy of 100% and a validation accuracy of 23%. The model still overfitting.



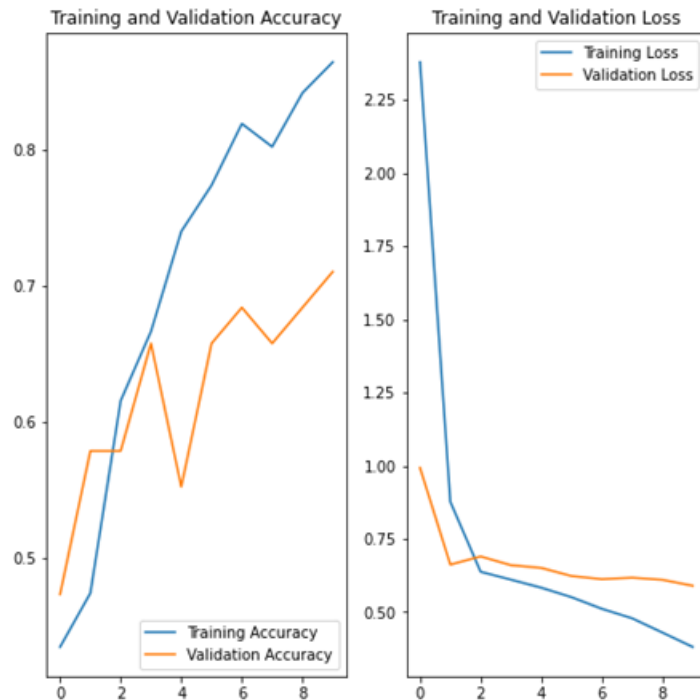
Graphs showing the training/validation accuracy and training and validation loss at 50 epochs

Experiment 4: Training another Dataset with fewer images and also reduced to two classes. ('apples', 'Tomatoes')

The model was used to train another dataset (fruit images) which has 297 images across two classes. Image size 180 X 180, batch size of 100 (deliberate because of the small number of dataset) in 10 epochs. With this combination of training, we got a training accuracy of 86% and a validation accuracy of 71%. No overfitting occurred. As illustrated in the graphs below:



Random cross-section of the images in the fruit dataset



Graphs showing the training/validation accuracy and training and validation fruit dataset with two classes.

Conclusion

The assessment on evaluating the effect of image size and class size on training and the validation accuracy of the training model shows that the model during training of the dataset with 30 classes struggles with overfitting which means the model cannot generalize and it's fitting too closely to the training dataset and the varying of the image sizes from 180X180 to 90X90 and increased epochs could not correct the overfitting. The performance of the model in the other dataset however was the opposite despite its lower data sample.

This assessment will therefore conclude that the performance of a model will generally be adversely affected when there is a higher number of classes of images in a dataset to be trained. However, the entire spectrum of Convolutional Neural Networks was not covered in this assessment.

My github repository link attached above has all the files containing my code and each experiment.

Arranged in this order:

Assessment for C7082- ipnb For Experiment 1 (180X180)

Assessment for C7082- ipnb_2ipnb For experiment 2 (90X90)

C7082 Assessment - ipnb increased epochs Experiment 3 (Increased Epochs = 50)

Fruit_classification_ipnb – For experiment 4 Fruit dataset classification

Reference

Ahmed, M. Israk & Mamun, Shahriyar & Asif, Asif. (2021). DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study. 235-243. 10.1109/ICCCSP52374.2021.9465499.

Tensorflowlite Creative Common Attribution 4.0 license and code samples are licensed under the Apache 2.0 license.

Liang Yu¹, Binbin Li¹ and Bin Jiao¹

Published under licence by IOP Publishing Ltd

IOP Conference Series: Material Science and Engineering, Volume 490, issue 4

Citation Liang Yu *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **490** 042022 **DOI** 10.1088/1757-899X/490/4/042022