PART 1:

Q1.1)

Yes, we must need to use TensorFlow (version 1.xx) in the final project as we can see in the file second half of the course, we must need to use tensor flow in all the work like building, training, testing, and in the CNN.

Q1.2)

No, keras was only used in the first half of the course as we can see in the requirements file, and we used for midterm and some assignments.

PART 2:

Here I selected the dataset called LOCBEEF dataset it contains 3268 images, and it has two categories training and the testing, and each subdirectory has rotten and the fresh and it was carried out with the deep learning technique with the convolutional neural network, and it is divided into the 70% of training and the remaining 30% of testing.

And also, this dataset has two subcategories called train and also test so training has 2228 these images we used for the training purpose because the dataset has splited into 70% training images and the testing has the 980 images this images used for the testing purpose.

A screenshot of a computer

Description automatically generated

Download

And this dataset is called unstructured dataset because it has images and different type of data.

Name: LOCBEEF

Website: data.mendeley.com

Number of items: 3268 total

Link to download: https://data.mendeley.com/datasets/nhs6mjg6yy/1

PART 3:

* Firstly, I downloaded the required datafiles from the canvas we can see from the below image.

A screenshot of a computer

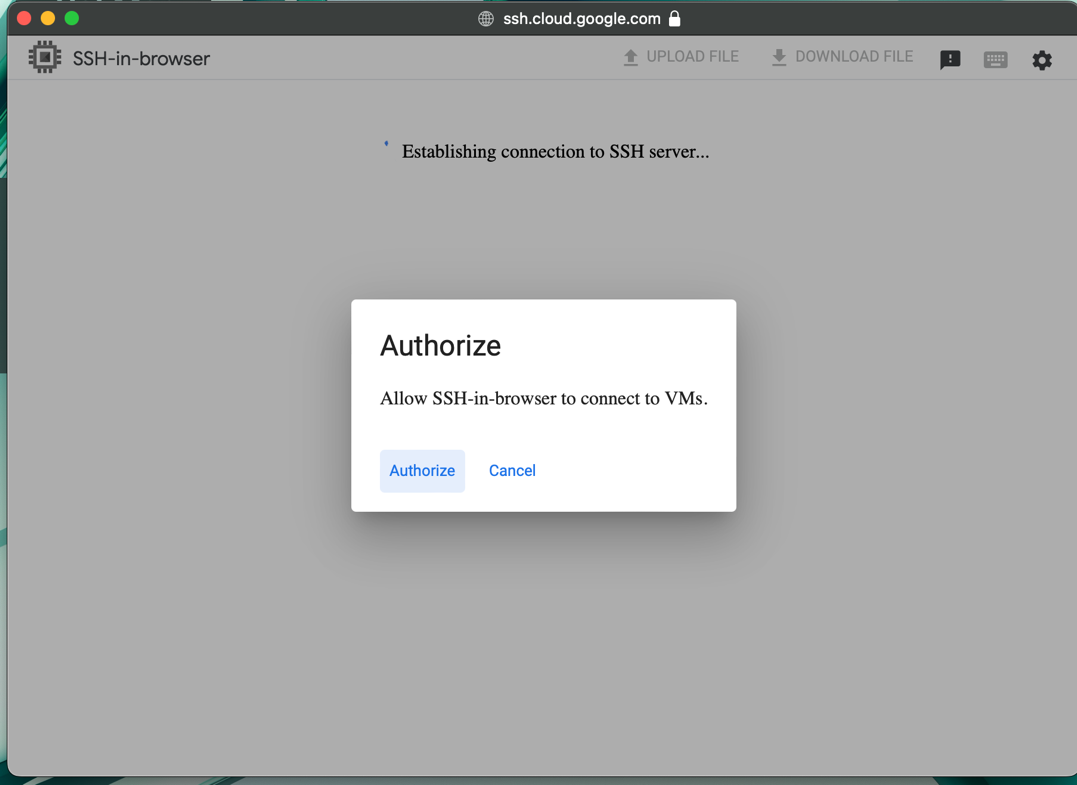
Description automatically generated

Here we can see the data files which I downloaded from the canvas.

* Secondly, the deep learning virtual machine was turned on as we can see from below image and opened SSH in the new window browser.

A screenshot of a computer

Description automatically generated



* Here after opening the SSH in new window, it’s asking for an authorization to access the and connect to the virtual machine and after clicking on the authorize it will check for the google account and reconnect to the vm.

A screenshot of a computer

Description automatically generated

* This is the home page of the virtual machine and here is the home directory and to check the existing files we can provide the command called ls -1

A screenshot of a computer

Description automatically generated

* We can see there are some previous files, and we need to make a new directory called CIFAR\_10\_DATA in the below image we can see that.

A screenshot of a computer

Description automatically generated

* And here we uploaded the files to the virtual machine which we downloaded from the canvas.

A screenshot of a computer

Description automatically generated

* Here we uploaded successfully.

A screen shot of a computer

Description automatically generated

* Finally, these are the files were uploaded in the CIFAR\_10\_DATA sub folder.

PART: 4

BUILDING, TRAINING, TESTING on CIFAR DATASET

Here we can see the Jupiter notebook file from the canvas uploaded successfully to the remote server.

A screenshot of a computer

Description automatically generated

Design of the CNN:

* Conv2D Layer: The initial convolutional layer, has 32 filters and a 3x3 kernel size. It has a 3x3 kernel size and 32 filters.
* Although your code does not explicitly specify the input shape, it should resemble the shape of the CIFAR-10 images.
* Conv2D Layer: This convolutional layer is the second one, with a kernel size of 3 by 3 and 64 filters. Once more, the activation function is relu.
* Layer of Flattening: This layer collapses the input's spatial dimensions into the channel dimension, hence flattening the input. In doing so, the result is ready for use in the Dense layers.
* Dense Layer 64 units: This layer has 64 units and is completely connected. Relu is the activation function that it employs.
* Dense Layer 10 units: This network's output layer. Ten units make up the CIFAR-10 dataset, which comprises ten classes. Although your code does not provide activation function used recently for multi-class categorization issues.

The performance of the convolutional neural network report:

Firstly, the CNN model was trained and tested over 5000 steps as we can see from the Jupiter file at step 0 we got the results like this 0.5949 accuracy initially the accuracy has been enhanced a so much here are the step 0 results.

A screenshot of a computer

Description automatically generated

And after lot of steps up to 5000 the model improved and provided good accuracy in the final step 4900 = 0.6714 and performance.

A screenshot of a computer

Description automatically generated

Analysis:

Learning Rate: Based on the consistent accuracy rise from 0.5445 to 0.7102 in the first 1200 steps, we can see the model was gaining some performance.

Overfitting: The accuracy of the model begins to vary and eventually decline after 1200 steps, which may indicate overfitting.

Model Complexity: The accuracy of the model, which is approximately 0.71, indicates that it can reasonably anticipate the proper labels. The lack of a discernible increase in accuracy with time, however, may suggest that the model's complexity is insufficient to fully capture more subtle patterns in the data.

Evaluation Frequency: The model's performance is examined in detail as it is assessed every 100 steps.

Final Performance: The model's moderate predictive performance is indicated by its final accuracy of roughly 0.67.

To conclude even if the model produces encouraging results, there may be space for development. The use of more representative and diverse training data, the application of regularization techniques to reduce overfitting, or an experiment with a more sophisticated model are possible next steps.

PART: 5

Compare CCN performance of CIFAR\_10\_DATA with CNN performance with MNIST dataset.

Dataset: MNIST

* The data collected in the numbers called as MNIST data recently used to test and train other pictures and modifying programs.
* 10,000 test pictures and 60,000 training pictures The accuracy of the CNN utilized on the MNIST dataset increases steadily, rising from 0.12 at step 0 to 0.94 at step 2900.

Dataset: CIFAR\_10\_DATA

* They are 60,000 pictures and separated in to 10 classed and holds 6000 picture in one class in cifar dataset
* The accuracy of the CNN applied to the CIFAR-10 dataset increases from 0.5445 at step 0 to 0.714 at step 4900.

COMPARISON:

It is displays from a comparison of the two CNNs' performances that the CNN on mnist and cifar dataset. There are various possible causes for this:

Complexity of the Data:

With color images of ten distinct classes, the CIFAR-10 dataset is more complex than the MNIST dataset, which only includes grayscale images of numbers. This might make classification task more difficult.

Size of the Dataset:

A CNN's performance can be greatly affected by the size of the training set. In this instance, the quantity of photos in both databases is comparable.

CNN Architecture: The CNN's architecture, which includes the number and types of layers, can have an effect on how well it performs.

Overfitting: A CNN that was train on cifar dataset may perform poorly on test data, resulting in lower accuracy levels, if it is overfit to the training data.

CONCLUSION:

In summary, the CNN train on mnist performs better than the CNN train on cifar dataset, even though both CNNs demonstrate an increase in accuracy with time. Analysing CNN topologies, examining validation loss to check for overfitting, and so on are a few examples.

PART: 6

IMPROVED CNN PERFORMANCE

Ten steps were used to train and test the Convolutional Neural Network (CNN) model, with accuracy being tallied at the conclusion of each step.

Improved Results :

* Following the first epoch, the validation accuracy was 0.5593 and the training accuracy was 0.4369.

A screenshot of a computer

Description automatically generated

And ends up with the 0.7081 test accuracy by looking at the results we can say that the model performance was enhanced.

Analysis:

* The accuracy of more than 79% on the training set, the model performs well.
* The test accuracy, however, is only about 70.81%, suggesting that the model may be overfitting the training set.
* The model enhanced from so much about the information and sound in the training data that it effects the model's efficiency on new data we can call this overfitting
* This shows that model finds a sound and also the different oscillations on train section.

We could think about techniques like applying regularization, employing more training data, or adopting a simpler model in order to enhance the model's performance.

Furthermore, methods like dropout and data augmentation can be applied to lessen overfitting in CNN models.

Conclusion :

To summarize, an enhanced test accuracy of 70.81% was obtained by implementing the suggested modifications to the network or network training procedure. This implies that the modifications were benefits and enhanced the CNN model's performance using the CIFAR-10 dataset.

PART: 7

Introduction:

Convolutional neural networks CNN were designed, implemented, and evaluated specifically for image classification tasks in this study. The main dataset utilized was the cifar 10 dataset The project was broken up into three sections, PART IV, V, and VI.

Description :

In PART 4, I used Python and the TensorFlow framework to build, train, and test a CNN model. You gained knowledge on how to assess a model's performance, the significance of data pre-processing, and the function of various layers in a CNN. and came to the conclusion that the first model worked moreover discussed about the CNN design and results.

In PART 5 I compared about the CNN with MNIST dataset with the CNN CIFAR\_10\_DATA dataset to compare the results of both the results of MNIST dataset are from the previous assignment and concluded at last observed the difference from the results.

In PART 6 as we can see I applied the modifications and reassessed the model in PART 6. To discovered that an important component of the machine learning workflow is putting changes into practice and assessing their effects critically. And came to the conclusion that the revised final model outperformed the original model and had a greater test accuracy.

Conclusion:

This project gave me useful experience with CNNs for picture classification applications. learnt how to use TensorFlow and Python to construct, train, and test a CNN suggested and executed modifications to enhance the model's functionality; and critically assessed the outcomes. The research demonstrated how machine learning initiatives are iterative, meaning that models are continuously improved and assessed to get the best results.