**Spring 2024: CS5720 Neural Networks & Deep Learning - ICP-8**

**Bhanu Chandrika Lakkimsetti (700747439)**

Image Classification with CNN

GitHub Link: [https://github.com/bhanuchandrika99/NNDL\_ICP\_8](https://github.com/bhanuchandrika99/NNDL_ICP_8%206)

Use Case Description:

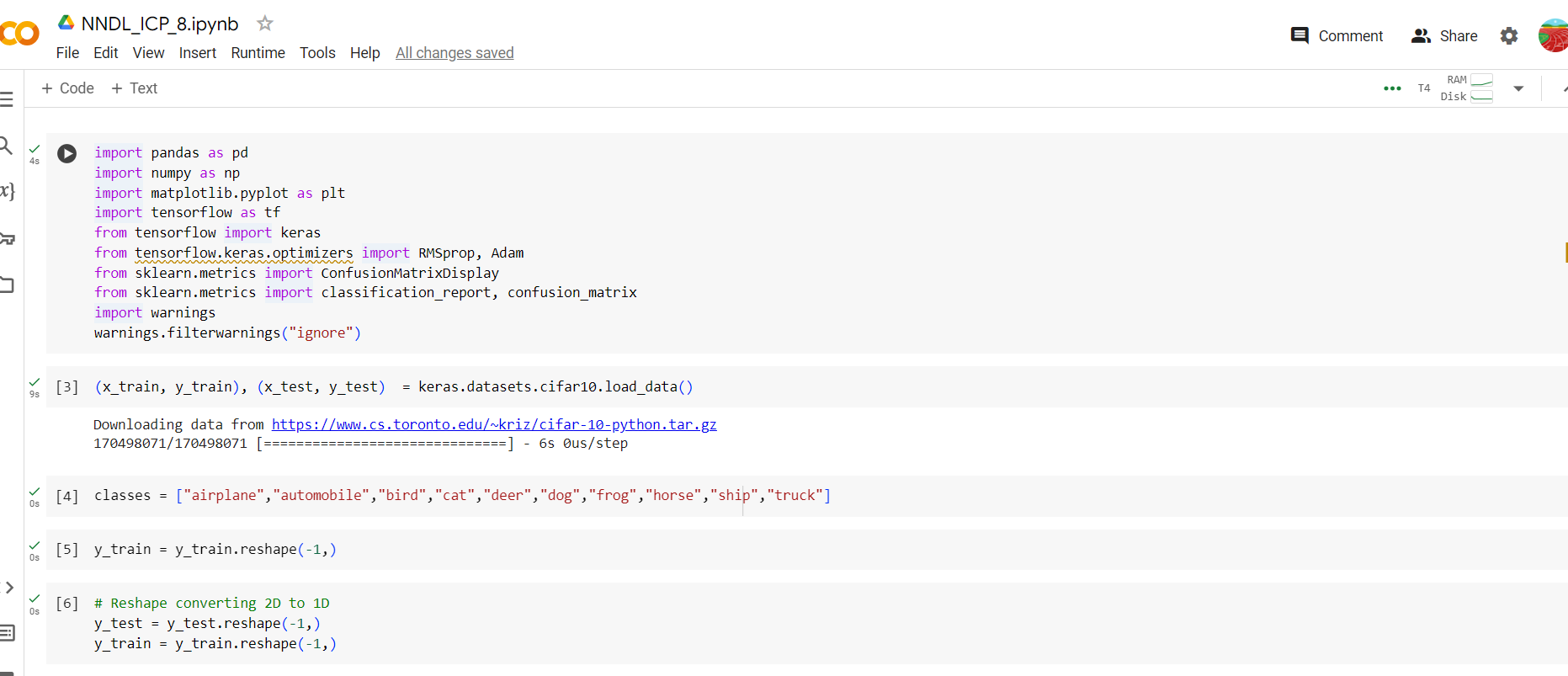
LeNet5, AlexNet, Vgg16, Vgg19   
1. Training the model   
2. Evaluating the model

Programming elements:   
1. About CNN   
2. Hyperparameters of CNN   
3. Image classification with CNN

**In class programming:**   
1. Tune hyperparameter and make necessary addition to the baseline model to improve validation accuracy and reduce validation loss.   
2. Provide logical description of which steps lead to improved response and what was its impact on architecture behavior.   
3. Create at least two more visualizations using matplotlib (Other than provided in the source file)   
4. Use dataset of your own choice and implement baseline models provided.   
5. Apply modified architecture to your own selected dataset and train it.   
6. Evaluate your model on testing set.   
7. Save the improved model and use it for prediction on testing data   
8. Provide plot of confusion matric   
9. Provide Training and testing Loss and accuracy plots in one plot using subplot command and history object.   
10. Provide at least two more visualizations reflecting your solution.   
11. Provide logical description of which steps lead to improved response for new dataset when compared with baseline model and enhance architecture and what was its impact on architecture behavior.

kvn

Lenet



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph of a curve

Description automatically generated with medium confidence A graph of a graph

Description automatically generated

A screenshot of a computer code

Description automatically generated

A computer code on a white background

Description automatically generated

A graph of confusion matrix

Description automatically generated

A screenshot of a computer

Description automatically generated

AlexNet

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer code on a white background

Description automatically generated

A graph of a line

Description automatically generated with medium confidence A graph of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

A chart of a graph

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A close-up of several images

Description automatically generated

Vgg16

A screenshot of a computer

Description automatically generated

A blurry image of a truck

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of a person's face

Description automatically generated

Vgg19

A screenshot of a computer

Description automatically generated

A screenshot of a web page

Description automatically generated

Training, Testing and splitting data

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

A white background with black text

Description automatically generated

A graph of a curve

Description automatically generated A graph of a loss

Description automatically generated

Using the code above, the CIFAR-10 dataset—a well-known collection of photographs divided into 10 classes—is loaded. The data is used to construct testing and training sets. The input images are converted from integers to floats and normalized between 0 and 1. The output labels are encoded using a one-hot process.

Next, the dimensions of the input data are changed to match the expected input shape of the convolutional neural network (CNN).

The CNN model is then defined using the Keras Sequential API. It consists of many convolutional layers with ReLU activation, dropout layers to prevent overfitting, and dense layers with ReLU activation. Sort the pictures into the ten groups. Sort the pictures into the ten groups. Put the picture into each of the ten classes. The optimizer in the model is stochastic gradient descent (SGD), the loss function is categorical cross-entropy, and the performance measure for the training phase is accuracy.For each of the 25 epochs of model training, 32 batches are employed. model's accuracy is printed once it has been assessed on the test set.

Next, the model is used to forecast the images in the test set. The projected labels and the actual labels are compared to see if the model has predicted correctly. Finally, the accuracy and loss are demonstrated using the history object that the fit() method returned.