Week 1: -

- Here we understand about the python jupyter.
- Understanding and Importing basics libraries (tensorflow, pandas, numpy, keras, glob).

Week 2: -

- How to import the files in jupyter.
- The program imports necessary libraries and modules, including pandas, numpy, glob, and matplotlib.py plot.
- Making the model for image classification using transfer learning with ResNet50 architecture.
- This is a program for image classification using transfer learning with ResNet50 architecture. It uses TensorFlow and Keras libraries for building and training the model.
- The dataset used in this program is located in the directory 'c:/users/lenovo/downloads/archive/dataset/dataset/'. The program uses the ImageDataGenerator class from Keras to preprocess the images and create image generators for training and validation datasets.

- The ResNet50 model is loaded with pre-trained weights on the ImageNet dataset. The last layer of the model is removed, and a new fully connected layer with 512 neurons and ReLU activation function is added. The final layer is a dense layer with 3 neurons and softmax activation function for multiclass classification.

- The model is compiled with the Adam optimizer, categorical cross-entropy loss function, and accuracy metric. The batch size is set to 32, and the number of epochs is set to 10. Finally, the model is trained on the training dataset and evaluated on the validation dataset.

Week 3:-

- Making the model which is using TensorFlow and Keras libraries to build a deep learning model to classify images from the Fashion MNIST dataset.

- The program starts by importing the necessary libraries and loading the Fashion MNIST data. The data is then preprocessed by dividing the pixel values by 255 to scale them between 0 and 1.

- The `build_model()` function defines the architecture of the convolutional neural network (CNN) model with hyperparameters. The function takes in a hyperparameter object `hp` and returns a compiled Keras model. The hyperparameters include the number of filters, kernel size, activation function, dense units, and learning rate.

- The `randomsearch()` function from Keras Tuner library is used to perform a hyperparameter search to find the best model with the highest validation accuracy. The function takes in the `build_model()` function as an argument, along with the search parameters such as the maximum number of trials and the project

directory. The `search()` method is then called to start the search using the training data.

- After the search is complete, the best model is obtained using the `get_best_models()` method. The model summary is printed, and the model is trained on the training data for 10 epochs using the `fit()` method. The `predict()` method is used to make predictions on the test images, and the predicted labels are obtained using `argmax()` method. Finally, the predicted labels are printed along with the actual test labels.

- This program uses hyperparameter tuning to optimize the CNN model's performance on the Fashion MNIST dataset. It uses random search to explore the hyperparameter space and find the best set of hyperparameters that result in the highest validation accuracy.