**Part 1 Analysis:**

Train set -> trainPart1

Step 1: importing libraries/ modules

Step 2: Data Pre-processing : Since the data is an image, we need to use ImageDataGenerator from tensorflow.keras.preprocessing.images to get some meaningful insights [ in form of pixels ].

Step 3: Reshaping the train images are important. It should be reshaped basing on the test image sets.

Step 4: Structuring the Model.

This is a type of a Sequential Stacking Model.

A randomly initialized neural network [ANN] won’t give much accuracy and will consume more space and memory. So, using CNN [Convolutional Layers] before the neural networks will increase the model accuracy and reduce the space and time complexity.

CNNs are more efficient because they use filters to reduce the matrix dimensions with each Conv layer and reduces it to a feature map. After the Conv layer there is a Pooling layer. There are 2 types of Pooling. Max Pooling and Average Pooling.

In the model, I have used Max Pooling Layer.

There are various Activation Functions involved in different layers of the Model.

I have used Relu as activation function for all the Conv Layers and Hidden Layers.

Output layer Activation function is softmax and not Sigmoid as it has more classifications.

Flatten Layer is for coverting 2D matrixes to 1D.

Then for compiling the model, optimizers are used.

SGD(Stochastic Gradient Descent) can be used for optimization but I have used RMSprop as it is more efficient in case of images.

Then saving the model.

Atlast, clearing the session and deleting the model is a good practice.