REPORT

Task 1 :

- I completed the course, which u gave upto image processing. And I tried the first task, in the course they mentioned about the canny edge detection. So, I tried that first with different threshold values, but the output was not that clear. Actually, it became even worse than the given image.(I imported the image named "task.jpg", and performed edge detection with the threshold values as [low = int(max(0, 0.7*med)),up = int(min(255,1.3*med))]).
- Then I started searching whether there are any methods, after some good research I got to know about Hough Transformation. And I saw a video in youtube and used that method for the given image, I got some better result and by changing the threshold I got much better output rather than just using Canny Detection. But, there are some lines overlapping over one another. (imported the image, performed edge detection and used that edges as the input for the hough transformation and I have drawn the output lines from hough transformation on the blank image so that the lines can be clearly visible. The output seemed much better than previous output but there are some lines which are overlapping one-another.
- I thought the overlapping is due the noise in the image, so I searched about the denoising the image. I tried with the method mentioned in This documentation. And then the denoised image used for Canny detection and that used to hough transformation.
- I used same above mentioned method in skimage, just to see whether there would be any changes or not, but I didn't see much difference.
- Still there are some overlapping lines, so then I considered the thickness of the each sheet or the equal distance between lines adjacent lines. So I have chosen 2 units as the thickness (added if condition in the hough transformation). So then I finally landed some decent output.

Task 2:

- I completed the remaining course, and started task 2. The primary task is to split the data into train and test. I splitted the given data using os library. I plotted the bar plot of the total given data to see the distribution of data. And created different directories for both training and testing using try except to avoid exceptions. And then I defined the function 'split_data' taking the arguments as Source, training path, testing path, and split_size, to splt into two directories by shuffling them before dividing. And copying the images from source to both the directories. Then I called the function for each class given using split_size as 0.7, and plotted bar graph for training and testing data after splitting.

- I imported ImageDataGenerator to generate different images from the data by manipulating them by increasing size or flipping horizontally, zoom and rescaling the whole data in between 0 and 1 and also I again divided the validation data into training and validation using this ImageDataGenerator, then used this training and validation data for training and validating the model. By using this I generated the images for both train and test data splitted. By defining training_gen and validation_gen, I made a specification whether its training or validation using 'subset' to make the model understand.
- I defined the model containing four set of CONV => RELU => POOL layers. First layer will learn 32 convolution filters, each of which are 5x5 and applied RELU activation function and then I normalized the batches followed by 2x2 max-pooling and other three layers will learn 64 convolution filters, each of which are 5x5 and applied RELU activation function and then I normalized the batches followed by 2x2 max-pooling.
- Then the output from the previous max-pooling is flattened into a single vector and then the fully connected dense layers containing 512 nodes, which then passed through other nonlinear RELU activation, and other dense layer containing 128 nodes and then passed through RELU activation.
- Next 0.4 fraction of total neurons are deactivated. And then to another dense layer with nodes equal to number of classes and this dense layer are fed to softmax classifier.
- And then model is fitted with the train_gen and followed by the plots of accuracies of training and validation data.
- Predicted on testing data and defined the function convert_to_class to convert the predictions into particular class.
- And by comparing actual and predicted classes, confusion matrix is plotted. And it followed the plotting bar graphs for actual and predictions data.

- I designed another model but now I divided the training data as train and validation but testing data which I divided using the function is used as validation because small typo in the code (validation_data_dir = 'Data for DL/Data/testing') , instead of 'testing' there should be training in the path.
- And validation data which was splitted from the train data using ImagDataGenerator is used as the test data for this model (as you can see that in testing_gen I mentioned subset also).
- This model consists of three CONV => RELU => POOL layers and the dropout of 0.5 neurons.

❖ While fitting the data, I got some errors like **cannot identify image file 'Path of that image file'**, for some files even if that file is actually present in the same path, so I deleted some images (7 images) from the original data.