Centre For Innovation
Office of Dean Students
IIT Madras
Chennai - 600 036

iitmadras

Centre for Innovation (CFI)

Indian Institute of Technology Madras

PDF REPORT FOR MINI PROJECT

Name:	V ARAVIND CHOWDARY	Nick:	Avi
Roll:	CE19B099	**CGPA:**	8.06
Phone no:	7893691531	**Email:**	aravindrocks369@gmail.com

1) Apply Image processing to make it easy to identify the number of sheets, in the image.

- Firstly I've imported the required libraries and then stored the image in img variable.
- Then I've defined as function namely **show_img** which will used to display the final image as my required size.
- Then I've blurred the image using a **bilateral filter**.
- Then I've applied **Adaptive Threshold** method to convert the blurred image into black and white image.
- Then I've applied **erode function** with a kernel size of 5 and 1 iteration to erode some of the data to make the lines visible clear.
- Then I've applied a **Sobel gradient**(X gradient) to make the eroded image much more clear.
- Then I finally displayed the image using **show_img** function.

2) Developing a Deep Learning based image classifier to classify images into 6 distinct categories

- Firstly I've wrote a code to store all the directories where the data is located and where created folders where the data is to be copied into a list.
- The path to the **original data** is in **org_dir** and the path to the **6 subfolders inside** that are stored in a list named **ins_org_dir**.
- The path to the new directory where the **images are to be copied** is in **new_dir** and the path to the **train, validation and test** folders is in **ins_new_dir**.
- I've copied the images to the folders train, validation and test folders in the percentage **50%, 25%, 25%.**
- For doing the above I first copied all the file names in the subfolders into 6 lists and then chose the set which I want in train, validation and test.

- So inside each train, validation and test folder there will be **6 subfolders** and the path to those are in the **ins_ins_new_dir[i][j]** where I corresponds to train ,validation and split folder and j corresponds to the 6 sub folders.
- After that I imported the **Pre trained convet VGG16.**
- Then I **used the conv base** of that model.
- Then I manually have written the densely connected layers along with layer flatten and drop out.
- As our data is a **Multiclass, Single label classification** and as we have 6 labels for our data I gave the output to be 6, and the last activation to be **softmax**.
- Then I **froze the weights** of conv base from modifying.
- Then I used **Image Data Generator** to **resize** all the images and perform **augmentation** on the train data.
- I used the **class mode** to be **categorical** as we have multiple labels.
- I compiled the model loss function to be **categorical cross entropy** and metrics to be **accuracy**.
- I took 20 epochs and some other values.
- Now I **unfroze the layers Conv block** 5 in the pertained convet to do fine tuning.
- Then I compiled the model once again as above and took 20 epochs.
- Now I evaluated the model for test images and got test accuracy.
- Then I plotted the learning curves that are **Accuracy vs epoch** and **Loss vs epoch** for training as well as validation data.
- Then I plotted another **smoother curve** for the above 2 for an easier understanding.
- And I've made a **confusion matrix** at the last.
- Now I kept on reducing the size of network and adding drop out layers to prevent over fitting and kept on checking with the values.

OBSERVATIONS AND MODIFICATIONS MADE BASED ON THE GRPAHS (OVER FITTING)

- Firstly when I run the data by freezing the weights of conv_base and via fine tuning when I unfreeze the block5_conv1 then I found that the train accuracy is very less even though I used a very dense network.
- I kept increasing the network but the train accuracy didn't improve much,
- So I unfroze the weights of conv_base and trained both at the same time.
- When I built a denser network I found the model to be over fitting.
- So I made changes accordingly by reducing the size of network and adding dropout layers and adjusting the epochs.
- As I unfreeze the training weights there is no need for fine tuning I guess.
- The final thing which I made is what I felt as better with a moderate over fitting and an average test accuracy.
- I've kept the parts which I don't want to run but checked that in #s