This is a startup guide for how to set up and run the program in Google Colab.

1. Open the program, which is titled “Copy of SEAP License Plate Detection.ipynb”, in Google Colab.
2. In the top right, there should be a button that says “Connect”. If it has successfully connected, it should show the amount of ram and disk usage.
3. You will need to run the first cell and upload local\_utils.py into it. When you run it, it will ask you to choose a file on your computer. You can download the local\_utils.py file from any of the datasets also within the Google Drive and upload it to the program.
4. Run the second cell – these are the main libraries used in this program. When others are needed later on in the program, they will be in those cells.
5. Run the third cell. Here, you will have to upload wpod-net.h5 and wpod-net.json into the cloud in the same fashion as step 3.
6. To check if you properly loaded the model, run the next cell with the function titled “load\_model”. If the program outputs “loading model successfully”, then you’re good to go.
7. Run the next cell with the function “preprocess\_image”
8. Now, it is important that you mount your Google Drive to the program so that the colab file can access the images of datasets. To do this, on the left hand side, there should be a little file icon. Click it, and then click on the “mount drive button”. A picture has been provided below for convenience with green circles showing where there are.



1. In the next cell, change the path of the dataset to where the pictures are located in your Google Drive. Run this cell, and change the “rows” and “cols” variables as needed to show all of the images.
2. From here, run the cell with the “write\_image” function.
3. Run the next cell. This is where you pick which image you want to apply noise to.
4. Run the next cell. These are necessary imports for image manipulation and noise generation.
5. This next cell adds various noises to the selected image and outputs them. You can change the amount of noise by playing with the parameters of the “random\_noise” function in this block of code. This cell also outputs the images. I’d recommend looking at them within your Google Drive however, where they will be saved. This is because they are too small for proper inspection in this cell.
6. You can skip the next few cells up until the cell with the function “get\_plate”. The cells that we skipped can be used to look at individual images, but is not needed for data collection. Run the cell with the function “get\_plate”. It is the function that grabs the license plate from an image.
7. Run the next cell. This sets up the text-recognition I used using easyocr.
8. The next cell is the most important one. This is was the main block of code that I used for testing. One important thing to note is the variable test\_image. Changing the number next to image\_paths will change which image the noise will be applied to. Once the program finishes executing, you should see a number that shows the total number of license plates that were recognized by the program.
9. Move onto the next cell. This cell is not necessary, but it is helpful. Use it to look at individual license plates within the array we just created and its corresponding text output. Do this by changing “testNum” to the index that you want to look at.
10. Move onto the next cell. This is also not necessary, but helpful. Use it to visualize all of the obtained license plates in the image. The “cols” and “rows” variables may need to be changed depending on how many license plate recognized. For example, if it recognized 100 license plates, change “cols” and “rows” to 10 and 10. If it recognized 30 license plates, change it to 6 and 5 so that you have the best viewing experience.
11. Move onto the next cell. This is also not necessary, but can be used to display the output easyocr provides for every single license plate it obtained.
12. This next cell is very important. This cell filters through the full easy-ocr output to compile license plate numbers/digits and the corresponding confidence number using regular expressions. You should run it so that confidenceList and textList now have data in them.
13. This next cell can be used to see the length of the array and the obtained license plate readings. It also outputs statistical values including mean, median, and standard deviation.
14. The next cell is used to save the data we just generated. Change the first parameter in np.save to change the file name that it saves to. I recommend changing the first number to whatever amount of noise you used.
15. The next cell can load an array you just saved and save it as a .csv file instead. I did this so it was easy to copy and paste numbers to excel.
16. Any cell from here on out is unnecessary for data collection. There are various functions which may be helpful, including one that outlines a green box based on where the program thinks the license plate is, but it is unnecessary for data collection.