

Assignment Report

Prepared For:



By:

Kanchi Sharma

Table of Contents

Contents	Page No.
Final approach for projects	3
Challenges Faced	4
Results	5
References	6

Final Approach Used For Projects

Project 1:

- I started with loading libraries and images path on colab. I tried all the adjustments on a single image first to understand the working and to eliminate errors while running my final code.
- I have use python function for every process in my code as it is convenient to apply those on a bunch of images altogether.
- I started by reading the images and masks using `read_images_and_masks()`. The image masks had several white dots in the background so I applied preprocessing to fill them by using `preprocess_masks()`.
- After that my next step was to create the background of car using wall and floor and replace it from the actual one. `create_background()` and `replace_background()` does this process smoothly by assigning 65% of background to wall and the rest to floor. And then by using car masks the background is replaced keeping the car intact.
- I am still figuring out a way to map shadow masks to the correct position in the image. Also, I am trying to resize the cars and adjust their locations on floor. The approach I am trying: I am trying to set a threshold for resizing the cars, so that all the cars can be of same size. I am also trying to set a threshold for car's location on floor so that they all appear on floor and not in air.
- Lastly `process_images()` runs all the functions and output is saved.

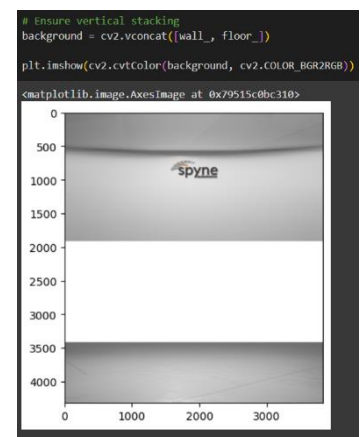
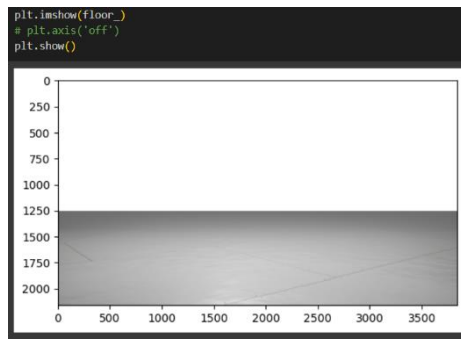
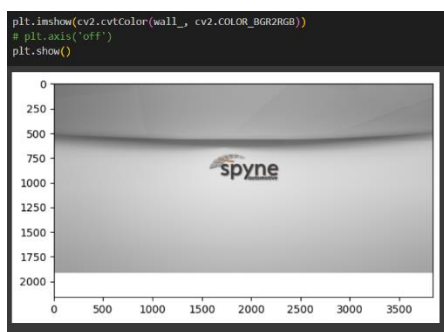
Project 2:

- I used pytorch in my overall code. I started by loading pytorch frameworks an I have followed methods and functions used in pytorch official documentation.
- I resized the image to 224*224 and applied transformations before training them. I used resnet18 to classify these images. Average train accuracy is 92% and val accuracy 94%.
- Currently I was only able to run the model for 3 epochs as the data is very large.
- I have also implemented the fastapi method for implementing it in .ipynb form.

Challenges Faced

Project 1:

- The main challenge I faced was to stack wall and floor. I suppose there was some padding in both the images which was not visible before I loaded it using `imread()`. I spent hours figuring out the mistake while stacking them. I tried different methods to load and stack images until I realized the images has some discrepancies. I ended up cropping the space to perfectly stack wall and floor. Some visuals of my hardships:



- Another challenging task for me is to align shadow and adjust car size and location.

Project 2:

- Object classification using pytorch is my strong area so I didn't face any challenge in that part.

Results

Project 1:

Output for car1:



References

- docs.opencv.org
- https://github.com/fiyero/Opencv_replace_bg/blob/master/OpenCV_replace_bg_git.ipynb