

INTRODUCTION

Computer vision is a field of artificial intelligence that trains and works with computers so they can interpret and understand the visual world. It is impacting the future and changing the way we live. Curious about its influence and what it can achieve, I decided that I want to learn about computer vision during the summer. In a span of a week, I focused on learning about image detection and recognition. I also had the intention of creating several projects using the information that I found. Computer vision was very new to me, so while the week was quite stressful, I was able to obtain a lot of information and work on some projects.

One of the projects that I worked on, which will be the main focus of this github, was creating a custom image detector. This project was part of a bigger project that was originally for a hackathon that I was going to attend, but due to the limited time and complexity of the project, I ended up working on it for my own pleasure.

OBJECT

The original goal of my custom detector was for it to be able to detect the objects one would find on the street. However, since there are so many things that can be found, I decided to pick out the most common ones. So, the main focus became that my detector was to detect five different categories: animal, bike, car, person, and pole. An object would be determined as an animal if it was a dog or a cat. A bike would be either a bicycle or motorcycle. The other three are self-explanatory.

I used Anaconda, OpenCV, TensorFlow, and Terminal on MacOS. My main language was python.

CHALLENGES

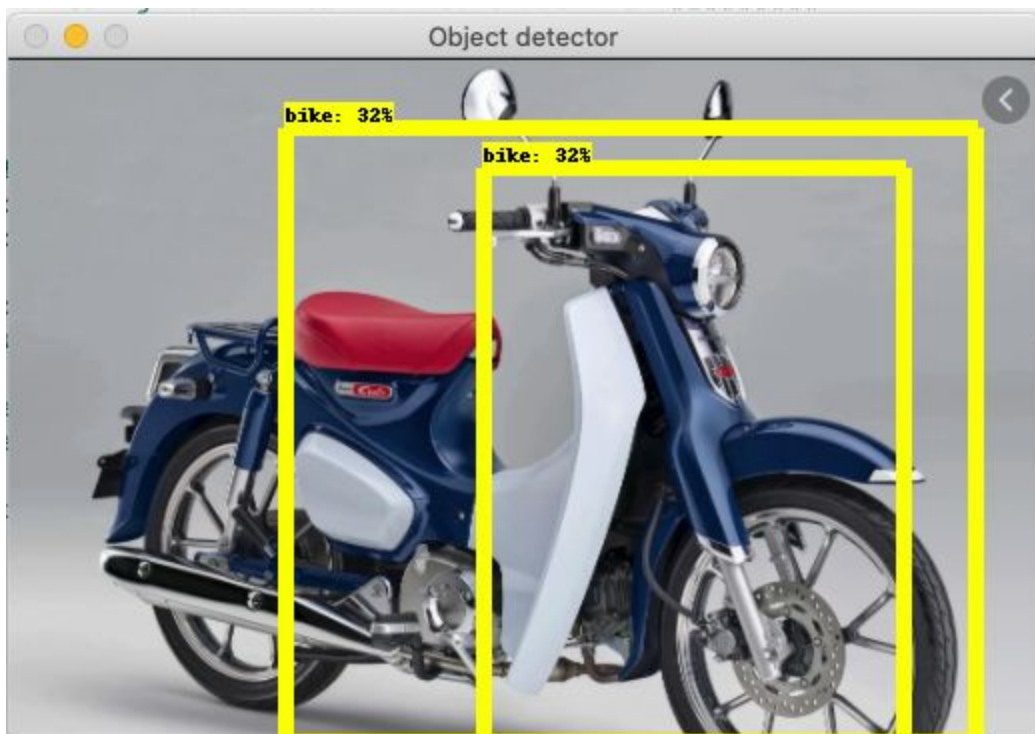
As I mentioned earlier, this was my first time learning image recognition. From the languages to the programs I used, I had to learn and understand everything while I worked on this project. Setting up environments was quite difficult for me, too, because I never had to set them up for my previous projects. I ran into a lot of errors and mostly spent my time trying to debug. Getting familiar with everything was my main challenge.

STEPS FOR TRAINING

1. Set up the working environment: [tutorial](#)
2. Data Preparation: [tutorial](#)
3. Train and generate the model: [tutorial](#)
4. Test the trained model

CONCLUSION

Here is a result that the object detection gave through my trained model:



As you can see, the model detected two bikes with a detected score of 32%. The detected score is noticeably low, and it accidentally detected two bikes when there is only one. This is all because of two reasons: the model received too little data when it was training, and the categories were too broad.

If you have checked out the steps, you will see that I used about 40 images for each category. This amount is too little for the model to work with, causing the detected scores to be low.

Furthermore, I had made the categories too broad for the model to be certain of what it was detecting.

WHAT'S NEXT

From this project, I learned that I need to make the data specific and large. So, for future projects, I will focus on creating distinct categories and widening the amount of images in order to raise the accuracy of the detector. I will also work on pairing this with a product, such as an app or a hardware device, to finish off the project.