Speed Limit Reader

Jay Yadav, Thomas Carr, Kyle Ward, Payne Miller & Makaila Vang

GITLAB:

https://cci-git.uncc.edu/itcs-4152-5152/spring-2022/project-11

Research

Business/Customers

- What problem will your Computer Vision solution solve, and for whom?
    - Our project will solve the problem of resolving the speed limit of roads. There are two main customers for our project: Autonomous Vehicles and Map/GPS designers. Autonomous vehicles need the ability to know what speed it is allowed to do without relying on external data. Map/GPS designers, namely Google, can use street view images in order to get the speed limit of roads.

Academic Literature Review

- What academic work is relevant to your project topic? Pick 3 papers, ask us for help if you need it.
    - A Speed limit Sign Recognition System Using Artificial Neural Network
        - K. A. Ishak, M. M. Sani, N. M. Tahir, S. A. Samad and A. Hussain, "A Speed limit Sign Recognition System Using Artificial Neural Network," 2006 4th Student Conference on Research and Development, 2006, pp. 127-131, doi: 10.1109/SCORED.2006.4339324.
        - https://ieeexplore.ieee.org/abstract/document/4339324
    - Automatic Recognition of Speed Limits on Speed-Limit SIgns by Using Machine Learning
        - Miyata, Shigeharu. 2017. "Automatic Recognition of Speed Limits on Speed-Limit Signs by Using Machine Learning" Journal of Imaging 3, no. 3: 25. https://doi.org/10.3390/jimaging3030025
        - https://www.mdpi.com/2313-433X/3/3/25/htm

    - Speed sign detection and recognition by convolutional neural networks
        - Download Limit Exceeded - Pennsylvania State University. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.704.4183&rep=rep1&type=pdf.
        - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.704.4183&rep=rep1&type=pdf

- What Makes these papers important?

All of these papers show us how to get the data, pre-process our data and train it on a CNN or a NN, which is what we are trying to aim for in our project. It is important, because these papers show us that this is a feasible problem worth solving and that it can be done.

- What are there results and how did they achieve these results?

Ishak, et. al's results were very positive. They first looked for the color of the dataset's speed limit signs (red), then they looked for the shape of the speed limit signs (circle), then finally they parsed the actual speed limit digits. They were able to achieve 95% accuracy in detection and 92% accuracy in recognition. Finally, their solution has an average computation time of 3.6 seconds.

Peeman, et al's results were also very promising. After the model was trained with an initial dataset, the solution only misclasssifed 0.19% of test set images. This follows that 99.81% of the test set images were classified correctly by the solution.

- What's different about these approaches?

One of the main differences between Ishak, et. al's approach and ours is that they were modeling their solution off of Malaysian Speed Limit signs, which contain red color. Their model looks for red color from the luminense, chrominance red, and chrominance blue properiotes (Y, Cr, Cb respectively.) Our solution will not be looking for colors specifically. In addition to color, their model also looks for a circular shape as the second mode of detection, behind looking for red color. Our model will not be looking for specific shapes as all of our images will be US based

speed limits (rectangular shapes.)

A difference between Peeman, et. al's (Pennsylvania State University) approach and ours is that they feed the parallelize their neural networkalgorithm via a GPU. Our approach will be using standard CPU memory and we are not exclusively parallelizing our solution.

Open Source

- What open-source code is available that is relevant to your topic?
    - https://towardsdatascience.com/recognizing-traffic-signs-with-over-98-accuracy-using-deep-learning-86737aedc2ab
- What data is available for testing and/or training algorithms?
    - The data in the above project is both European speed limit signs, and the model was designed for street signs in general. Our goal is mainly to get a number readout. However, there does exist a dataset here that has exactly what we are looking for.
- Is labeled data available? How much? How is the data licensed? Is it under copyright protection?
    - There is labeled data available for what we are looking for as mentioned above. There are signs between 25 and 60 mph in intervals of five. There are 1100 images available, but there is a deficit in the higher speed limits. The data is available on Kaggle and carries an Open Database license. That being said, we have full access to the data.

Industry Solutions

- What companies are solving similar problems to yours?
    - Google, Apple, Waze
- It can be tough to tell exactly how proprietary solutions work, but what can you find on the internet?
    - Their solutions are currently semi-automatic using public government databases, external mapping companies, or sourcing information from

    - phones that are using GPS softwares. Google has access to street view images which means that they could already be implementing something similar to our project, but there is no way to tell if they are or not.
- If you have access to the product, what can you learn from using it?
    - From personal experience using Google Maps, most major road provide a speed limit while backroads or smaller roads (typically less than 45mph) don’t have any speed limits. Additionally, they don’t seem incredibly confident in displaying the speed limit as it shows up as a small box only when using the navigation. Waze on the other hand does show the speed limit at all times on most roads, and also shows the speed you are currently going. Apple Maps is similar to Google Maps in their displaying and how much data they have.

Data Preprocessing:

- Load raw images from data folder
- Resize so that all images are 128x128 pixels
- Normalize images (divide by 255)
- Convert to grayscale
- Apply a gaussian filter to reduce noise
- Use Sobel edge-detection to extract edges of the image