Computer Vision Full Project Proposal

EN.601.661.01.FA24

Problem:

Walking and cycling in the United States can be very unsafe compared to other developed countries. Sidewalks and bike lanes are narrow and rough, and cars drive fast, creating a high chance of getting into an accident. Some streets are safer for walking and cycling though, namely the ones that have lower vehicle traffic, narrower lanes for cars, and wider sidewalks. However, typically only locals of the area will know the safest paths to walk or cycle through to get to their destination because Google Maps' directions are optimized for time and not safety, and some paths that are unnamed or offroad will not be considered. If one is cycling, it can also be difficult to know in advance where bike racks around the destination are located. Additionally, the problem of unsafe roads can be applied to hiking trails. While there exists data on trails about length, incline, and expected difficulty, paths can change daily based on inclement weather, and is much more difficult to predict. In the best case, this is just a mismatch between the data available and the actual trail, and in the worst case, can lead to accidents for hikers, especially those who are less experienced.

Context:

There are several existing solutions regarding map directions regarding pedestrian and cycling safety. Washington Trails Association provides trip reports for trails in the Pacific Northwest. Hikers can report the latest hiking conditions such as road condition and weather. The city of Baltimore provides a map of safe cycling routes in the city and bikemaps.org is a website that allows cyclists to report cycling hazards in the area such as collisions, roadblocks, and bike thefts. Rackfinder is an app that shows where bike racks are located in the user's area.

Data:

For the biking data, we will use Google StreetView for the conditions of the streets. There already exists an image scraper, so we can use this to obtain our dataset, which could be billions of photos large. To simplify this dataset, we will focus on biking trails in Baltimore first and try other cities depending on how well our algorithm works. To craft a more robust solution, we can also take obstruction data for biking lanes to add in as a heuristic for whether or not a path should be taken. For training data for the hiking context, since these trails are not usually available on Google StreetView, we can use the data available on AllTrails including lots of images users have taken over time. This data encompasses most, if not all hiking trails in the United States, but we will first focus on hiking trails in Baltimore. There are currently hundreds or thousands of reviews for each trail, but only a select few have photos, narrowing down our dataset. There are also metrics about each trail, including weather, difficulty, length, and incline. For the bike rack add-on, we can scrape the location data from Rackfinder, which should include all bike racks in the United States, as an additional data point.

Proposed Solution:

Our goal is to improve route recommendations for bikers by providing safer and more efficient paths. We will use **computer vision algorithms** to analyze Google StreetView images for biking routes and AllTrails images for hiking paths. The primary algorithm will be **Canny edge detection** to detect the boundaries of bike lanes and sidewalks, and **Hough Transform** variations to identify obstacles such as parked cars, curbs, or other potential hazards. We can also train an object detection model with YOLO to provide further fine tuning. A **voting algorithm*** will compile the safety score for each route based on multiple metrics, including weather, obstacle count, bike lane presence, incline, and overall road width.

We will start with the biking context in Charles Village/Baltimore, and then apply our algorithm to the hiking context in Baltimore, and then extrapolate to other cities.

To evaluate the performance of these algorithms, we will compare the safety scores generated by our system with known safe routes, such as those listed in the Bike Baltimore map, and through real-world feedback from cyclists using our app. The key evaluation metrics will be **route safety score accuracy**, **user satisfaction (feedback surveys)**, and **consistency across multiple datasets**.

An example workflow could look as follows:

How: open source google path data/api -> generate more than 3 paths from point a to point b -> map the images from Google Streetview for each path through a feature detection algorithm to associate regions of a street to a path-> calculate the score for each image based on the voting algorithm (below) and additional metrics -> compile scores for each path -> pick the safest path -> confirm with the existing Bike Baltimore map scores and possibly our own assigned scored

* Voting algorithm (accumulated array) - each path has a score which might come from a couple different metrics (weather, difficulty, length, and incline) and add up all these

scores to produce a final safety score which the algorithm will choose the highest to determine the safest route.

Potential Challenges:

We anticipate challenges such as **dataset diversity** (images across different environments may vary in quality), **scalability** (applying the solution to larger datasets), and **weather condition detection** (accurately interpreting street images affected by varying weather). Moreover, fine-tuning the algorithms to detect obstacles and assign accurate scores will require several iterations of testing and refinement. Additionally, coming up with a sound safety score interpretation might become troublesome due to the majority of the metrics we will be using and the differences between streets and terrain. Thus, this might entail creating 2 different scales of our safety evaluation to suit both environments.