Video As a Sensor

Kyle Begovich, Daniel Huang, Lucas Gong, Jianfeng Xia, Yuxuan Ren,
Dongjung Seung, Yanbing Wang, Mayank Kathuria, Dingyang Chen
Graduate Leaders: Tanushree Nori, Deborshi Goswami, Fangyu Wu
Faculty Advisors: Professors Richard Sowers (Math, ISE) and Daniel Work (CEE)

Introduction

Goals:

Optimize our existing system to make real-time detections
Deploy our system on a hardware platform capable for our use case
Package our working software with additional open source technologies

Background

Past Work

Pre-Project:

Our software, You Only Look Once (YOLO) was developed by Joseph Redmon, a PhD candidate at the University of Washington. He wanted to develop a Computer Vision system that was real time and made use of Graphics Processing Units (GPUs) highly parallelizable computations and therefore great speed. He made the project into an Open Source Software (OSS) project on GitHub, and we've built on top of his original work and contributed back to Redmon when we had presentable code additions for their project.

Spring:

In the spring, we worked on getting all the software running, and discovering what types of systems, software and hardware, that we could leverage to develop on top of the given code. A lot of this time was discovery into Convolutional Neural Networks and C level programming.

Summer:

Over the Summer, we spent a lot of time retraining the network. We displayed that capability by getting YOLO to recognize bicycle helmets from people apart from hats and full-mask helmets. We also worked to improve context maintained between frames of video, instead of the previous version without such features. We also added contextual detections to the system, so that a frame containing both a "bicycle" detection and a "person" detection intersecting would be detected as a "bicyclist" instead.

The New System

Features

Hardware:

We used two systems for our time over this semester, a large desktop computer system that has a twoway SLI Titan X GPU setup and the other (here today) is an NVIDIA Jetson system. The Titan X machine provided a sandbox enviornment with more than sufficient capability for our software, but more interestingly, the Jetson is a more modest powered and priced system that would be practical in real applications.

FaceNet and FaceYOLO:

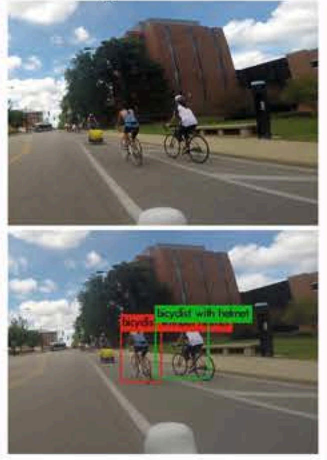
We started the semester working with an out-of-the-box software called FaceNet, which uses Google'd Tensorflow to detect people's faces. While it is very successful in detection accuracy, it did not meet the speed needs of our project, and we were forced to scrap it in favor of a pre-trained YOLO system from another group online. This is the software described on the right, and the one we use in our demo.

ALPR:

Using the ALPR system as described on the right, we are able to hash away licence plates to protect the privacy concerns of other drivers. Privacy has been a big concern since day one, and we spent a lot of time working to keep it safe, in addition to burring faces.

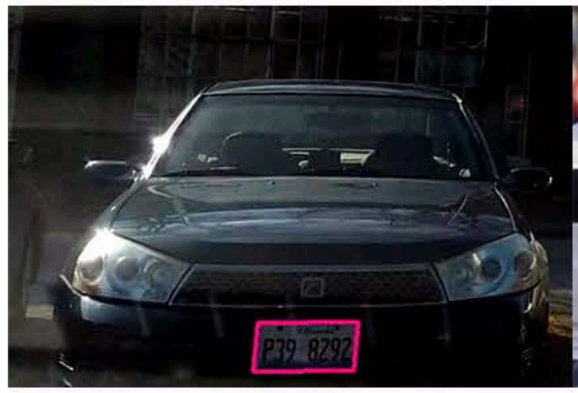
Parallelized YOLO:

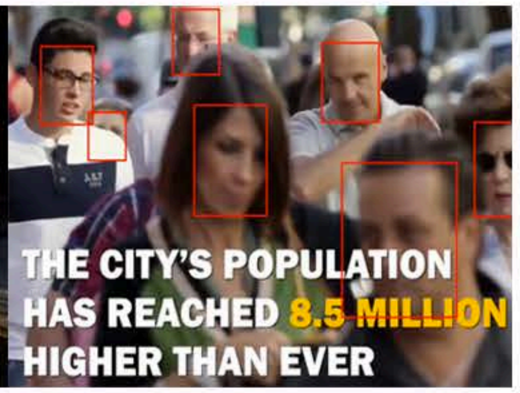
This semester, we spent a large portion of time working to parallelize multiple instances of YOLO (specifically, the helmet version from the Summer and the original set of classes) on the same GPU at the same time. We were very successful in this, and have been able to get the system running at the improved speeds we were looking for.



ALPR

Since the mid-term presentation, we have integrated ALPR package into YOLO as a parallel thread, and the detected license plates can now be highlighted with bounding boxes by calling a "draw_box" function inside of our program. Compared with the original YOLO, the ALPR embedded YOLO brings down the overall frame rate from 30fps to an average of 20fps, when running on our server. In terms of detection accuracy, the original ALPR package can correctly detect and locate 80% of the license plates in a 1.5-min test video. However, after integrated into YOLO, the combined package (YOLO+ALPR) only detects less than 10% of the license plates appear in the same video. We found out that the license plates which were beyond a 30-degree horizontal or vertical camera angle were not detected, and the accuracy degraded with a significant amount of false positive errors. In addition, detection stability has not been fully addressed. The bounding boxes drop as soon as a license plate is identified. Since the original ALPR algorithm is not able to have the same detection performance in every frame, the bounding boxes may lose track at a later time. We will further examine detection stability after improving detection accuracy.





Face Detection

We use YOLO Facial Recognition on Redmon's Darknet Framework to detect and recognize multiple faces. The algorithm learns facial features by training on different classes and identifies individual faces. Our current goal is to run YOLO facial detections in a parallel manner with high accuracy to maintain high speeds with the rest of the system. Once we achieve these goals, we would like to train our facial recognition system to detect for vulnerable groups like the elderly and the very young.

Support for this project was provided by the Illinois Geometry Lab and the Department of Mathematics at the University of Illinois at Urbana-Champaign.