RUTGERS
WINLAB | Wireless Information Network Laboratory

Smart Intersection

Nicholas Meegan, Kevin Zhang, Bryan Zhu

Advisors: Dr. Zoran Kostic, Ivan Seskar, Jennifer Shane

Project Objectives

Using machine learning paradigms, detect and track vehicles as they move through a traffic intersection, then analyze those detections to quantify traffic flow during different times of day over different days of the week.

- Perform inference on traffic intersection footage using an object detection machine learning model to track vehicles travelling through the intersection
- Implement a system that allows us to perform inference on one machine and view the original footage with superimposed inference outputs on a different machine
- Create a method for estimating vehicle count and traffic flow statistics using the results from this system

Network Data Flow

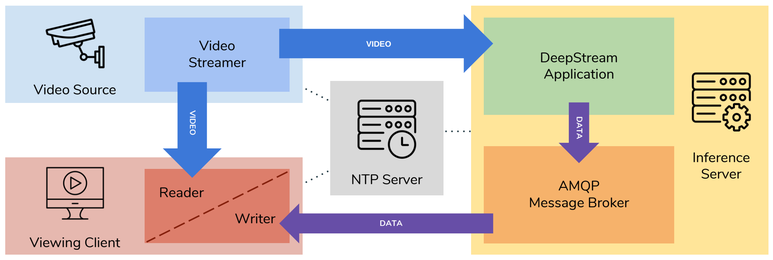

Fig. 1: Visualization of Data Flow Across Network for Implemented System

Employed Technologies

NVIDIA DeepStream
Video analytics SDK used to build application that optimizes performance of inference engines

YOLOv3
Algorithm used for real-time object detection

AMQP 0-9-1
Messaging protocol used for data transmission

OpenCV
Library used to draw bounding box data on video

NTP
Network protocol used to synchronize system clocks across all machines in system

System Implementation

Video Source

- Video stream made available to both inference server and viewing client
- System times of each machine synchronized with WINLAB central server reference clock using NTP
- Same video frames arrive at each machine within microseconds of each other, barring network issues

DeepStream YOLOv3 Application

- Incoming video stream decoded using URI source bin
- Decoded video stream pushed through a multiplexer to adjust video resolution for inference
- System timestamps recorded for each frame for viewer client use during syncing
- Real-time inference performed on video frames using YOLOv3 to detect objects
- Detected objects tracked and assigned unique identifiers
- Analytics performed on tracked objects to determine direction of travel and whether any user-defined boundaries were crossed
- Object locations and analytics information compiled together and converted into message payload
- Messages published to AMQP message broker exchange for routing

Fig. 2: DeepStream YOLOv3 Application Pipeline

AMQP Message Broker

- Message payloads contain bounding box, tracker, and analytics metadata
- Messages routed to AMQP topic queue for consummation by subscribed viewing client

Viewing Client

- Client operates using multi-threaded paradigm consisting of a reader thread and a writer thread
- Shared queue data structure utilized to pass information between both threads
- Queue mutex locked as necessary to prevent reader and writer threads from accessing queue concurrently
- Writer thread saves output data obtained from the message broker to shared queue
- Reader thread synchronizes and draws queue data over incoming video stream using OpenCV
- Boundary crossings counted and tallied to determine when and from where vehicles are entering and exiting traffic intersection

Results

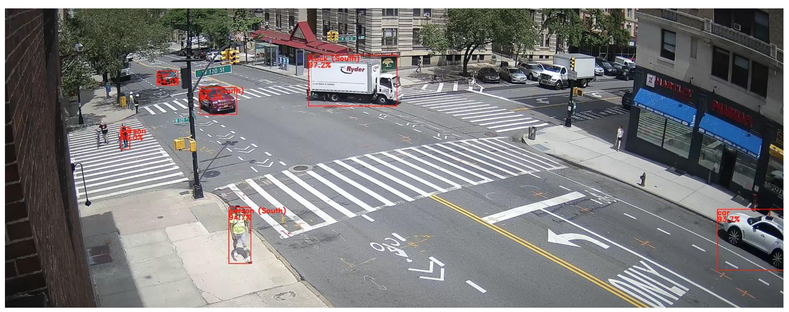

Fig. 3: Output Drawn by Implemented System

- Viewing client able to successfully draw bounding boxes obtained from DeepStream application on top of video streamed from different network location
- Timestamp synchronization allowed for accurate combination of data and video on client with little to no information loss

Future Work

- Train YOLO model specifically for vehicle detection
- Upgrade object detector from YOLOv3 to YOLOv4
- Adapt project to perform inference on 3D point cloud and depth map data

Project Resources

Project Website
https://bzz3ru.wixsite.com/smartintersection

WINLAB Wiki Page
https://www.orbit-lab.org/wiki/Other/Summer/2020/SmartIntersection

WINLAB GitLab Repositories
https://gitlab.orbit-lab.org/si2020-smartintersection

WINLAB