Machine Learning Mobility Data Through Security Camera Feeds

Students: Yifan Bai, Zhe Han, Austin Miller

Sponsor: Sound Transit | Industry Mentors: Tom Wilson, Brian Brooke, Michael Berman | Faculty Mentor: Prof. Jenq-Neng Hwang

Sound Transit

UW Electrical & Computer Engineering

PROJECT STATEMENT

> Deploy multiple cameras to generate passenger mobility data at UW Link station.
> Design machine learning software to track objects and produce:
  > Traffic counts
  > Duration of stay
  > Path of track
> Without using visual attributes that can be associated to individuals through matches to other data sources (e.g. facial recognition, license plates).

Data/Technology Used

> Security camera video data from UW Link Light Rail Station

python™ YOLO PyTorch

PERFORMANCE CRITERIA & SYSTEM REQUIREMENTS

> For accurate movement tracking, frame processing needs to be a minimum of 10 FPS.

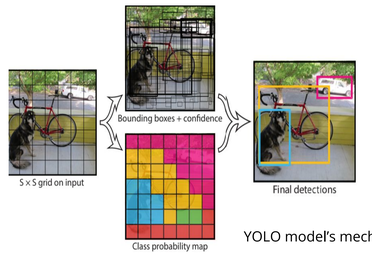
> The system needs to be low power.

> The system uses existing security cameras in UW Link Light Rail Station.

> The system needs GPU machines for improved performance of object detection

OBJECT DETECTION

YOLO V3 (You Only Look Once) Model



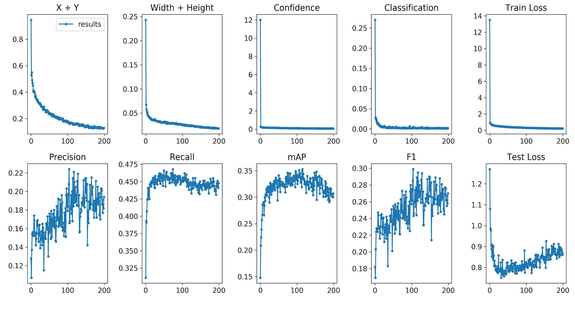
YOLO model's mechanism

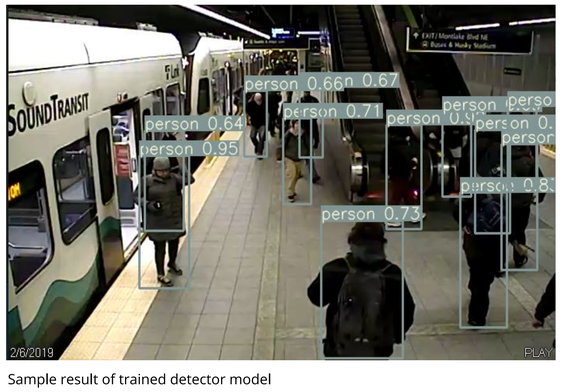
> Has 53 convolutional layers
> Divides image into regions
> Predicts class probabilities for each region
> Outputs final detection and confidence
> Faster than most other classifiers.

Detector Model Training

> We trained a 2-Class detector for person and bicycle.
> We trained model on more than 8000 annotated images.
> Overall mAP of model is 0.35.
> Model having 20+ avg FPS performance on a GTX 1060 GPU.

Metrics in 200 Training Epochs

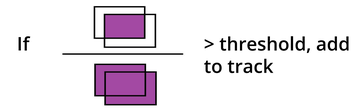




Sample result of trained detector model

OBJECT TRACKING

In order to obtain higher frame rate for tracking, we use Intersection-Over-Union method to check overlap of detections in new frame over existing tracks without extracting features from tracked objects or other image information used by more sophisticated tracking algorithms.



IOU Tracker's mechanism

> Original design assumed that the detector produces a detection per frame for every object tracked and high frame rate necessary for high overlap IOU.

> Modification required for catching missed detections between frames.

> Let an inactive track wait for 10 more frames before adding it to finished tracks, picking back up when the same object is detected a few frames later.

Catching the missed detection in modified tracker

PERFORMANCE METRICS

Commonly used metrics may not accurately measure the performance of our tracker as we are applying the tracker for tasks of generating traffic counts.
To measure the performance of tracker, we analyzed where the errors come from and developed some quantitative metrics.

Error Sources:

> Failure to catch missed detection leads to over-count
> Failure of detection leads to under-count
> A different object picks up the track.

Quantitative Metrics:

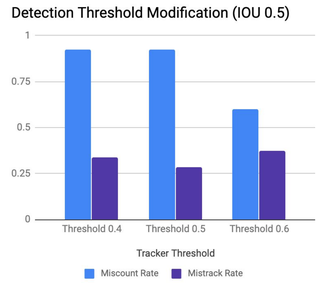
> Miscount Rate:

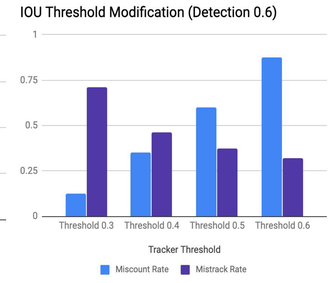
$$\frac{|\#\ of\ tracks - \#objects\ appeared|}{\#objects\ appeared}$$

> Mistrack Rate:

$$\frac{\#tracks\ assigned\ to\ different\ object\ between\ frames}{\#\ of\ tracks}$$

RESULTS

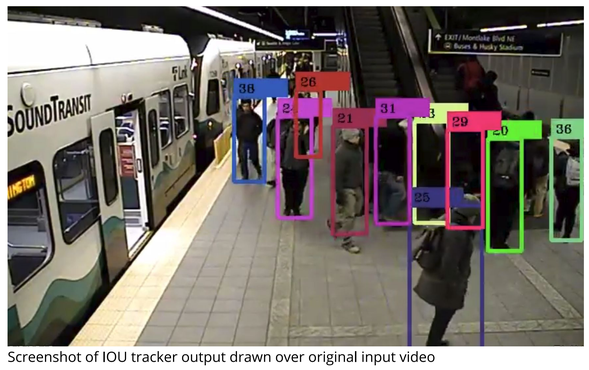




Metrics with different detection and tracking thresholds

We adjusted several parameters and ran the tracker on some short video clips. We found out that the performance is related to our detection threshold and IOU threshold.

> Overall, the tracker works well. Individuals who pass through the video with high detection accuracy are followed by the tracker and assigned just one ID. Mistakes occur when detection boundaries for two individuals occur in one spot.



Screenshot of IOU tracker output drawn over original input video

FUTURE IMPROVEMENTS

> Introduce features for more accurate object tracking.
> Deploy onto real-time video stream.
> Adapt for functionality onto moving spaces such as elevators and transit vehicles.
> Track an object through multiple cameras with overlapping fields of view and different lighting conditions.
> Develop a database system to store the tracked information.

REFERENCES

> Joseph Redmon, Ali Farhadi: YOLO V3: An Incremental Improvement
> Erik Bochinski, Volker Eiselein, Thomas Sikora: High-Speed Tracking-by-Detection Without Using Image Information