Final Project Presentation

JAEHO BANG & SIDDHARTH BISWAL

CREATING THE NEXT®

Motivation

- Video content is increasing exponentially
- Analysts want to explore large amount of data quickly
- State of the art methods are either limited in capability (3fps on $8000 GPU)

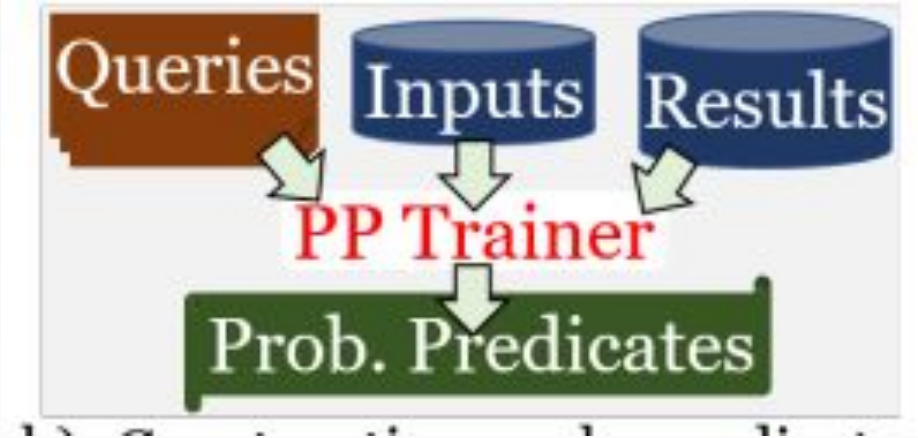
Goals

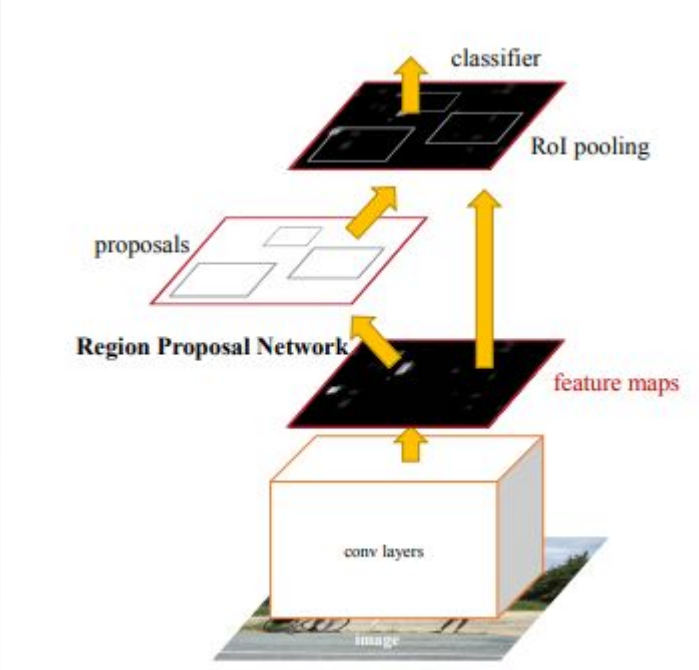
Build a Video Analytics Framework by

1. Replicate Probabilistic Predicates Paper by Yao et al. (80% goal)
2. Apply optimization techniques from BlazeIt by Kang et al. to further improve performance (100% goal)

Execution Overview

Training Overview





Dataset Details

- Dataset used: UA-DETRAC
- Details:
  - Traffic cameras and annotation label of objects in the frames
  - 10 hours of videos captured with Canon EOS 550D
  - 24 different locations
  - 25 frames per second with 960x540 pixels
  - >140,000 frames, 8250 vehicles
  - 1.21 million labeled bounding boxes
  - Vehicle categories:
    - Car, bus, van, others
  - Weather categories:
    - Cloudy, night, sunny, rainy

Architecture In Depth

Loaders

1. Image
2. Vehicle Type - Specified in XML
3. Speed - Specified in XML
4. Color - Color Detection Scheme
5. Intersection - Manual Labeling of Intersection

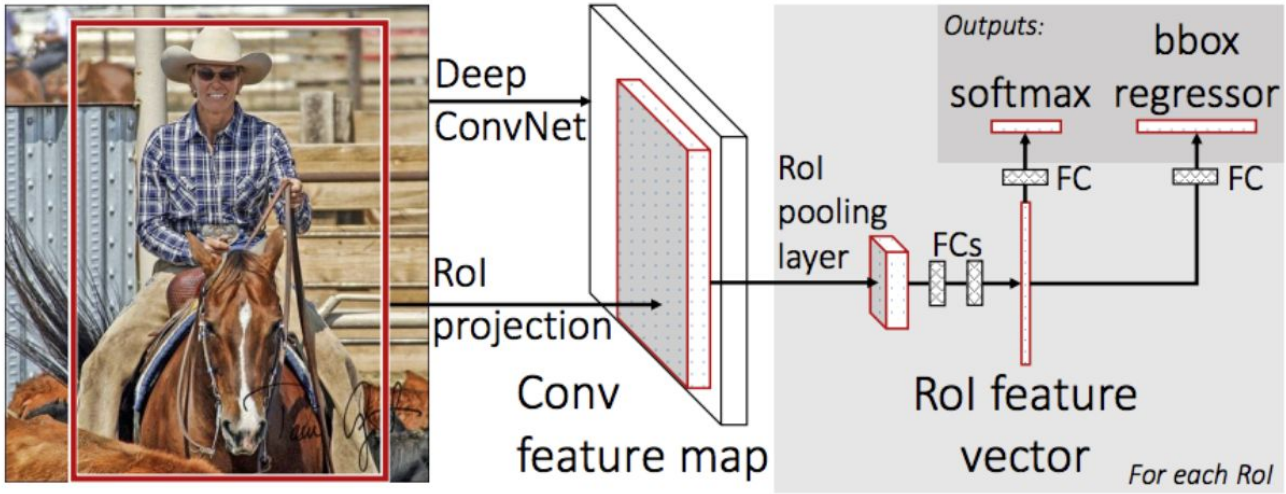
Query Optimizer

1. Wrangler step - generates transformed queries that are all sufficient conditions of the original query
2. Compute Expressions step - Probabilistic Predicates matching, calculating the final reduction rate, and generate the final plan

Probabilistic Predicates

1. Data Preprocessing (scikit-learn)
   a. Principal Component Analysis
2. Data Classification Models (scikit-learn)
   a. Kernel Density Estimation
   b. Multi-Layer Perceptron
   c. Random Forest
   d. Support Vector Machines
3. Generates the binary labels
4. Generate accuracy, cost (time it takes to build the filter), and reduction rate for Query Optimizer to use
5. Focused on extensibility - only need to implement three functions to use custom models

UDF - Faster RCNN



Evaluation of System

- Test for correctness
- Test for accuracy
- Test for speed

Evaluation - Correctness

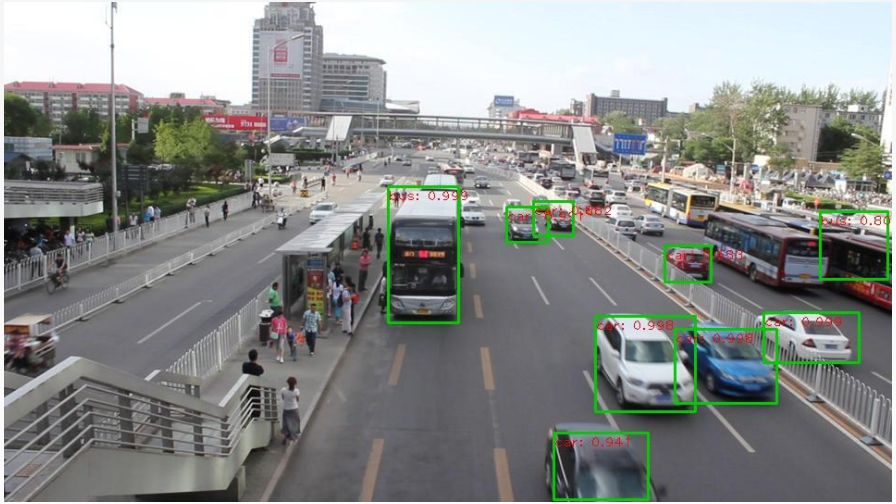
1. Loader
   a. Randomly sampled frames for manual examination of labels (vehicle type, color, intersection, and speed)
2. Probabilistic Predicates
   a. Examined filtered result statistics
      i. Multiple models - we can develop general ideas of cost, accuracy, reduction rates
   b. Manual examination of the frames
3. Query Optimizer
   a. We input synthetic data and TRAF-20 query samples used in Yao's paper
   b. Manually calculated possible queries and its corresponding reduction rate to make sure it works
4. UDF
   a. Extract detected images with detection boxes laid over

Filters (Probabilistic Predicates)

Figure 1) Frame with Car

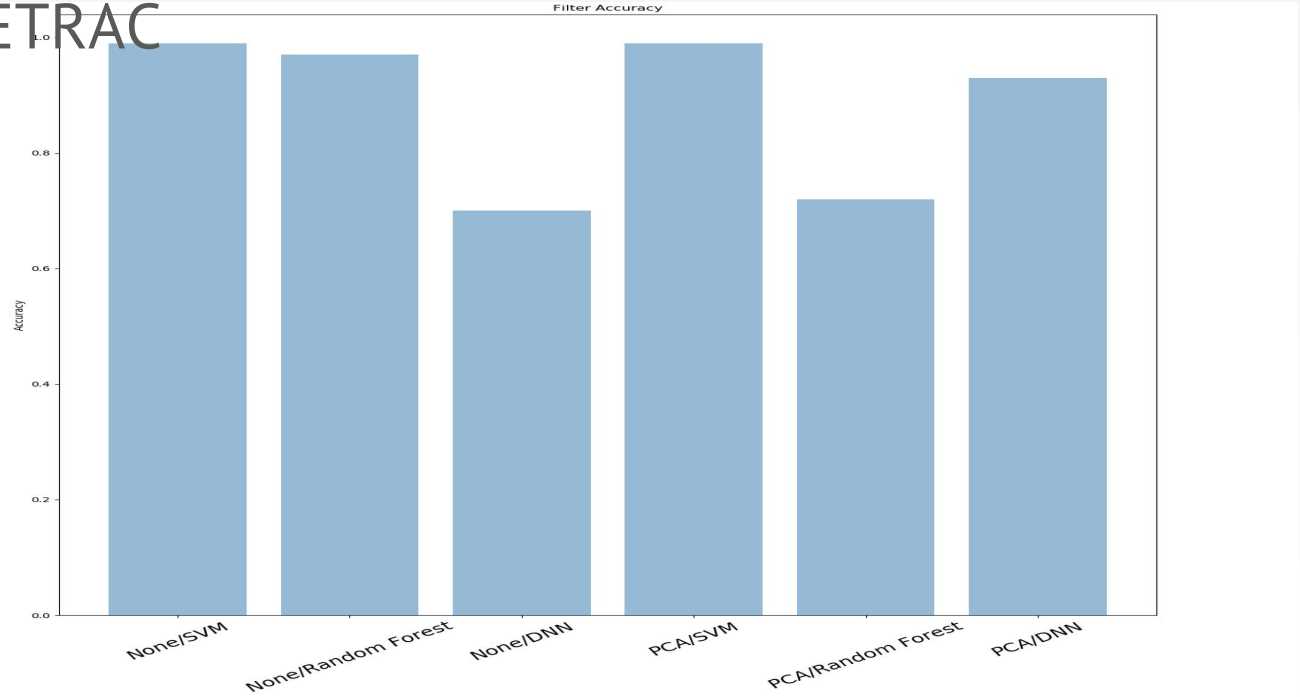
Figure 2) Frame Without Car

UDFs



Evaluation - Accuracy

Predicate: t=van, Dataset: 664 images from UA-DETRAC



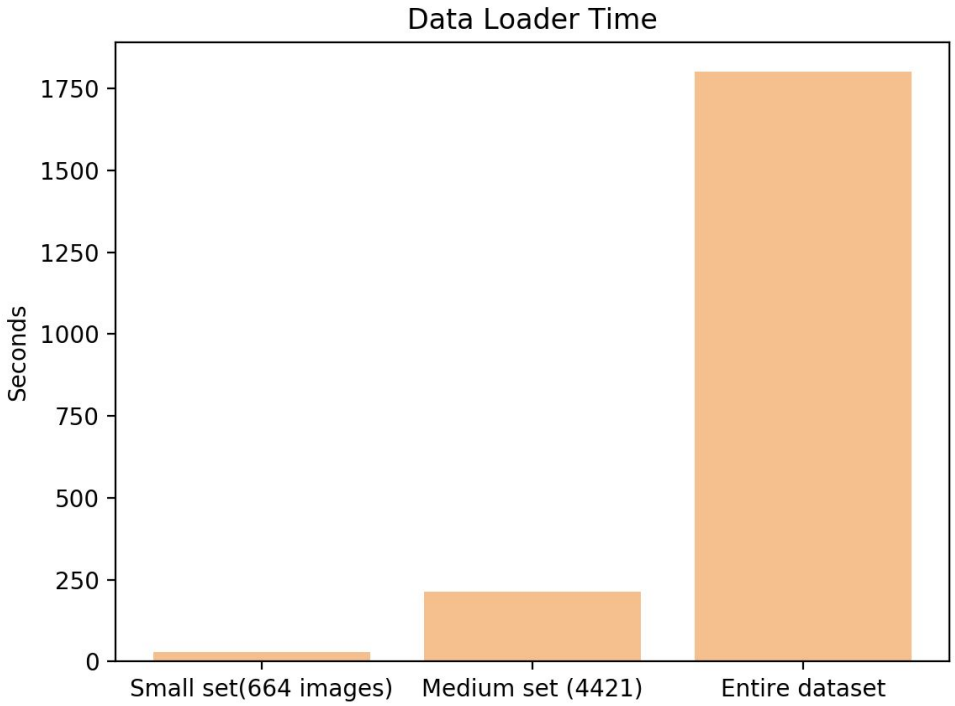
Evaluation - Accuracy

Training time for faster RCNN

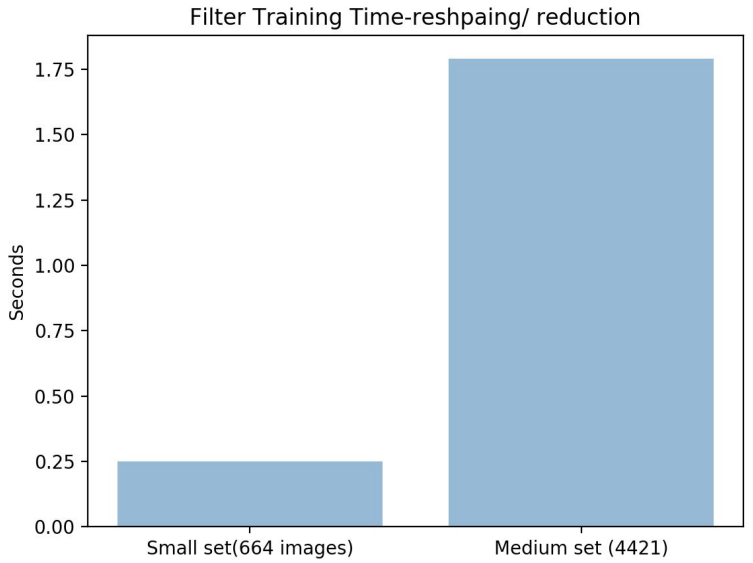
It's important to note that UA-DETRAC dataset contains 60 different scenes with around 1000 images in each scene. So, in order to train on the entire dataset it will take around **7.5 hours.**

Number of Images	Training Time/epoch
936	450 second

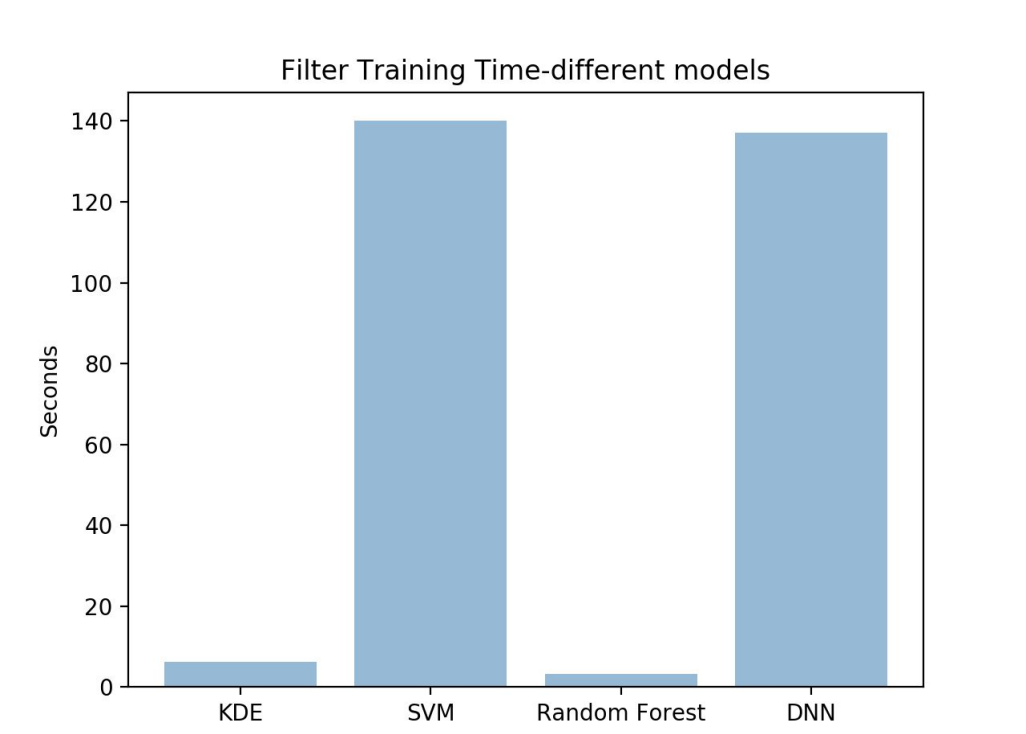
Evaluation - Speed (Loader)



Evaluation - Speed (Filter)



Evaluation - Speed (Filter)



Evaluation - Speed (All)

Dataset - 664 images from UA-DETRAC

Data Loader	29.113 seconds
Filters - train	339.52 seconds
Filters - execution (one query i.e t=car)	0.03 seconds
Filters - execution (TRAF-20)	0.55 seconds
Query Optimizer - execution (one query i.e. t=car)	0.0003 seconds
Query Optimizer - execution (TRAF_20)	0.023 seconds
UDF - train	x
UDF - execution (one object detection)	25 seconds
UDF - execution (entire TRAF-20 with object detection tasks)	x

Aggregate Optimization

Sampling based aggregation
- adapt the sampling procedure described in BlazeIt paper which means that we select K/epsilon samples

Specialized Neural Networks
- Train another multiclass classifier for detecting objects in the video

Query: SELECT COUNT(*) where class="car" error within 0.1
E.g find number of cars in scene

Results
- without Aggregation Optimization
  100 images- 45 mins
- with Aggregation Optimization: Sampling
  - 100 images- 21 mins
- with Aggregation Optimization: Sp NN
  - 100 images- 14 mins

Discussion - Filters

Predicate: t=van, Dataset: 664 images from UA-DETRAC

Preprocessing	Classification	Accuracy	Cost (Training time)
none	SVM	0.99	22.9 (s)
none	Random Forest	0.97	0.16 (s)
none	Multi-Layer Perceptron	0.7	47 (s)
PCA	SVM	0.99	3.9 (s)
PCA	Random Forest	0.72	0.03 (s)
PCA	Multi-Layer Perceptron	0.93	4.2 (s)

Future Work

1. Dataset loading time takes too long - we should implement a small database where we can read the data from.
2. Saving a downsampled version of the data - currently all images are stored as 960x540, for the UDFs and filters we don’t need such a big image.
3. If we could add in a timing constraint for the system, (ex: perform training task in 6 minutes) the system could be more configurable to its users.
4. Labels other than vehicle type does not have UDF - Add in more UDFs for textual labels
5. Faster RCNN tests - ROI, mAP numbers
6. Smarter Way of Filter Selection / UDF Selection for training
7. Smarter Way to perform Video Data Saving and Analysis - Utilizing P, I frames in video compression