Eagle Eyes: ML Powered Parking Enforcement System

Andrew Henley, Arthanios Ghaly, Chasely Goodman, Kirill Kozlov, Shritesh Bhattarai, Wesley Byrne

Introduction

Parking enforcement can be a major hassle for the management of any public or private location. It requires a lot of manpower to enforce designated parking spots. Parking enforcers have to repeatedly circle the parking lots. In many occasions, a video surveillance system is used. This too requires constant human involvement if actions need to be taken right away.

Machine Learning and Computer Vision

With the recent advancement in Graphics Processing Unit (GPU) hardware, previously intractable problems such as computer vision have become possible. Similarly, there is a proliferation in machine learning achievements where the computers learn on their own to solve a problem from a given dataset.

NAVSEA approached us with the idea of solving the parking enforcement problem by using machine learning and computer vision techniques.

We have built an end-to-end solution that solve some of the problems of parking enforcement and reduce human dependency.

Eagle Eyes

We decided to build a deployable solution that can be used at Tennessee Tech and national parks initially. It was decided that it use security cameras and the solution would be accessible via a web browser.

Elevator Statement

- **For** building managers and security personnel
- **Who** want to secure and monitor their parking lot,
- **Eagle Eyes** is an intelligent parking security system
- **That** monitors the parking lot and alerts you when suspicious activities occur
- **Unlike** traditional security cameras and human parking enforcers,
- **Our solution** uses state-of-the-art computer vision and machine learning technology so that you always get notified of suspicious activity without having to constantly watch the video footage or circle the parking lot.

Video Clips and Anomaly Score

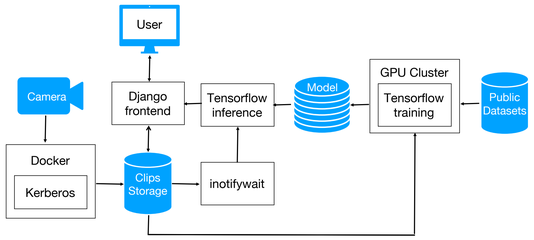
Our solution involves assigning an anomaly score between 0 and 1 to video clips of parking lots. A score towards 0 signifies that there is no anomaly, and a score towards 1 signifies that the video is anomalous.

The anomaly score is generated by a neural network that has been trained from public datasets and has been adapted to work on the deploying location using transfer learning.

TensorFlow was chosen as the deep learning framework of choice, because of the features provided by it and the wide support and adaptation from both industry and academia.

Training and Inference Pipeline

We separated training and inference right from the start. The training pipeline would run in a datacenter or powerful GPU cluster while the inference system would run in a on-premise personal computer attached to a camera.



End to End Machine Learning Pipeline

The training pipeline requires considerably more processing power and time to create the model. This process involves tweaking the hyperparameters of the neural network by data analysts.

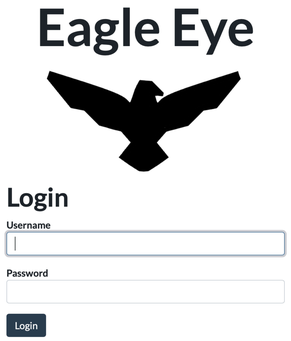
We used Jupyter Notebooks and TensorFlow using publicly available datasets to train our model and transfer them our deployment pipeline. The dataset we collected totaled more than 48 gigabytes of labelled data.

On the inference pipeline, we used the open source Kerberos.io surveillance system to record the clips via a USB camera under a docker environment. Those clips were stored on a Hard Disk Drive. The Linux utility inotifywait is used to execute the inference script and create a new entry on the Django frontend.

User Interface

We used Django as our web framework and sqlite as the database to store anomaly score.

A user first logs into the Eagle Eye website and enters their credentials. This prevents unauthorized individuals from accessing the video clips.

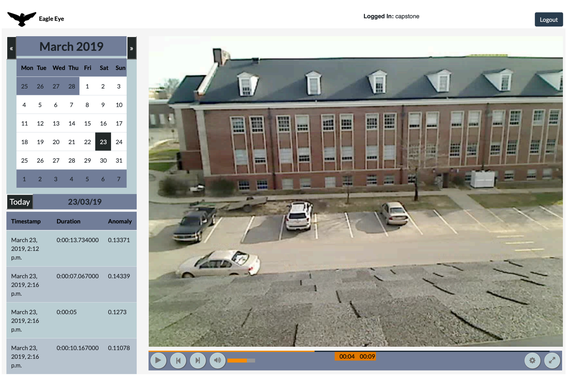


Eagle Eyes Web UI: Login

Authenticated users get access to the dashboard where they get can view footage from a specific day in chronological order in a table. The table shows the timestamp, duration and anomaly score. Entries from the table can be clicked and the specific video clip plays on the video player on the right.

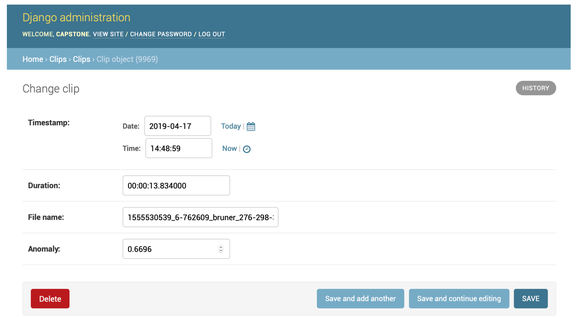
There is also a calendar interface for easy navigation to a specific day.

Once the user is done with the dashboard, they can log out by clicking the logout button on the top right next to their name.



Eagle Eyes Web UI: Dashboard

Users that have elevated permissions, also known as superusers, can access the Django backend itself, so that they can manage the users and also access all the records stored in the database.



Eagle Eyes Web UI: Admin

Results

We built the end-to-end pipeline and deployed it with a web-camera pointed behind Bruner Hall at Tennessee Tech university. We have collected over 10,239 video clips and run it through the inference engine that is accessible through the web interface. The number of video clips is increasing on average of 600 each day.

Our initial machine learning model has achieved an accuracy of 69% on public datasets.

Future Work

- **Manually label the captured dataset**: The dataset can be released publicly so that it can help other researchers.
- **Increase the accuracy of the model**: We used a simple model to validate the idea and build a minimum viable product. Data analysts and domain experts can help us get the accuracy even higher.
- **Deployment:** The solution has been designed right from the beginning to be deployed in multiple establishments. We will be working with NAVSEA and Tennessee Tech to deploy in the planned locations.

Special thanks to Tennessee Tech University, Computer Science Department and the Naval Sea Systems Command, Dahlgren