Binary Image Classification for Person Recognition Using SVM and Random Forests

Zac Howard

Introduction to Machine Learning ITCS-3156

December 5, 2025

Introduction

Object detection is a common problem when it comes to computer vision, with applications ranging from autonomous driving to security systems and even robotics. The main goal of object detection is to determine whether an image contains a specific object automatically. Over time many detection methods have been developed, ranging from simple pixel operations like thresholding and edge detection to advanced deep learning-based detectors like YOLO, Fast R-CNN, Faster R-CNN and SSD.

In this project the goal was a simple, meaningful task, binary classification of images based on whether or not they contain a person. Rather than performing full bounding-box localization, this project focuses on determining the presence or absence of a person anywhere within an image. This type of detection is valuable in many applications such as surveillance, autonomous systems, or robotics where merely knowing that a person is present can stress caution, alert a system to potentially important events, or help fluence safety-related behaviors. However, person detection can also be challenging due to variations in lighting, pose, scale, and background noise.

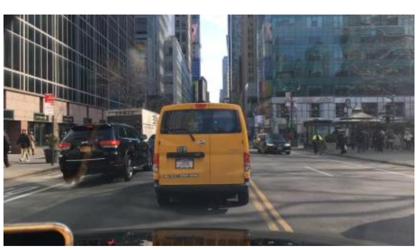
To address these challenges this project uses a classical computer vision approach, Histogram of Oriented Gradients (HOG) for feature extraction combined with machine learning classifiers. This approach makes sense for the task because HOG is proven reliable for capturing human silhouettes and edge structure. Furthermore, traditional models such as Linear SVMs and Random Forests perform well when trained with strong, structured features. The goal is to evaluate how these models perform using these curated features on a binary person detection problem.

Data

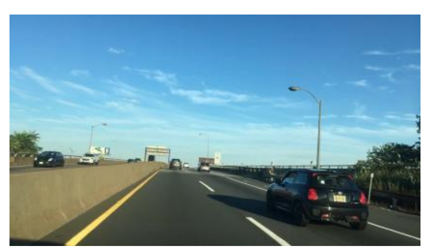
For this project, the dataset used was the BDD100k, a large-scale driving video dataset containing dashcam images of various traffic scenes. The images are originally 1280x720 pixels and include a wide range of objects such as cars, pedestrians, traffic lights, traffic signs and lane types as well. For the purposes of this project I focused on binary classification of images based on the presence or absence of a person anywhere within an image. To create a balanced dataset a total of 10,000 images were randomly sampled with 5,000 images containing at least one person and 5,000 images without a single person. This ensured that the model received equal representation of both classes during training. The sampled dataset was then split into 80% training and 20% testing sets, yielding 8,000 images and 2,000 images respectively.

Sample Images:

With Person(s):



Without Person(s):



Class Distribution:

```
(.venv) PS C:\Users\Zac\Documents\School Stuffs\Classes\Intro to ML\ML Final Project> python .\scripts\random_sample.py
{'selected_count': 10000, 'missing_images_for': []}
Preprocessed 10000 images. Positive samples: 5000, Negative samples: 5000
```

Data Preprocessing:

Prior to training the models the following steps were applied:

- Resizing: All images were downscaled to 256x144 pixels, maintaining the original aspect ratio to reduce computation time while also preserving important features.
- Grayscale Conversion: Images were converted to grayscale to simplify the feature extraction and reduce the input dimensionality.
- Feature Extraction: Histogram of Oriented Gradients (HOG) features were computed for each image.
- Train/Test Split: As described above, 80% of the images were used for training and 20% for testing.

The resulting HOG feature vectors along with their corresponding labels, person/no person, were stored together in a single file “preprocessed_data.npz” for easy loading during model training.

Methods

In this project I used two methods to determine whether or not an image contained a person or not, that being HOG + a linear SVM as well as the Random Forests Algorithm, an ensemble learning method used for both classification and regression tasks. HOG feature extraction captures the distribution of gradients and edges, which is useful for detecting objects like people without relying on color information.

Results

Model training didn’t finish in time..

Conclusions

References

https://www.youtube.com/watch?v=sDByl84n5mY

Source Code: https://github.com/ZacHowardUNCC/ML-Final-Project