Synthetic-to-Real Object Detection Using HOG+SVM

Justin Wenzel

March 26, 2025

1. Introduction and Motivation

Object detection is a growing computer vision task across multiple applications ranging from surveillance to self-driving cars and robots. Deep learning-based methods such as YOLO (You Only Look Once) [1] have begun to dominate object detection applications because of their high speed and accuracy. These deep learning based models are often more complex and prone to requiring a large amount of annotated data, high-speed hardware, and significant training time. In contrast, there are well-established classical approaches, such as Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM) using a sliding window approach to perform object detection originally proposed on humans [2], which can make HOG+SVM a potential option for low-resource and constrained environments for other object detection uses.

Motivated by the potential simplistic benefits of HOG+SVM, this project applies to a different context by exploring the training approach potential of the model. Others, such as Kaplan and Saykol, have explored a HOG+SVM object detector more recently trained on real-world vehicles, which could be a more complex object compared to humans, and resulted in a detection accuracy of 57.8% [3]. This helps understand some potential limitations of HOG+SVM on certain complex objects. This project will explore how a HOG+SVM detector model trained on synthetic data created using the Microsoft Airsim simulator [4] can perform on real-world data. Sometimes, during development, collecting and annotating real data can be expensive, but synthetic data can offer a potential low-cost alternative for data gathering. By training on synthetic Airsim data and testing on real data, this project aims to explore the feasibility and limitations of using a classical detector model in a synthetic-to-real data transfer setting.

2. Project Goals and Methodology

The goal of this project is to deploy and evaluate a HOG+SVM-based object detection model trained only using synthetic data and see how well the model generalizes to real-world data. This allows for testing the feasibility and limitations of a traditional object detection method combined with simulation-based training.

The project begins by creating a training dataset through the help of Microsoft AirSim, where the simulated wolf moves through an outdoor setting. Frames are then extracted from the simulated scene using different camera angles. Finally, a bounding box is applied around the wolf to create the synthetic dataset.

Then, the features of HOG will be learned from wolf and non-wolf-containing images. The implementation for the detection pipeline will be related to the face detection pipeline outlined and demonstrated in the book *Python Data Science Handbook* by VanderPlas [5][1]. The pipeline begins with demonstrating how HOG features paired with SVM can form an object detection system. The book provides a step-by-step guide for implementing the system and covers HOG feature extraction, linear SVM training, and applying a sliding window for final detection. This pipeline will be adapted to this projects domain instead of faces.

In addition to the baseline linear SVM demonstrated by VanderPlas, this project will also explore an additive kernel using the intersection kernel, as demonstrated by Maji and Malik [6]. Their results showed that introducing HOG features with an additive kernel resulted in a high accuracy and low training cost for number classification, making it a more relevant technique than the initial HOG+SVM proposal. This method will be another test to see how performance might change for an SVM to generalize on synthetic to real-world data.

Lastly, the trained detector will be tested on real-world images in the final step of the project. The detection performance will be evaluated through standard metrics such as precision for how many predictions are actually correct out of all detections, and intersection over union (IoU) to verify how well the predicted and true bounding box overlap. This helps determine how well the model generalizes to real-world data after being trained solely on synthetic images.

3. Roadmap/Timeline

Date Range	Tasks
Mar 1 – Apr 7	Simulate and record wolf movement scenes in Microsoft AirSim; begin dataset creation.
Apr 8 – Apr 11	Extract HOG features, train SVM classifier, and implement detection pipeline using sliding windows.
Apr 12 – Apr 16	Test detector on real-world data; evaluate precision and IoU metrics.
Apr 17 – Apr 21	Analyze results, document key findings, and begin the final report.
Apr 22 – Apr 26	Finalize report and prepare a class presentation.

[1] Code available at: `https://github.com/jakevdp/PythonDataScienceHandbook/tree/master/notebooks`

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[3] Ö. Kaplan and E. Saykol, "Comparison of support vector machines and deep learning for vehicle detection." in *RTA-CSIT*, 2018, pp. 64–69.

[4] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[5] J. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, 2016. [Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/

[6] S. Maji, A. C. Berg, and J. Malik, "Efficient classification for additive kernel svms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 66–77, Jan. 2013.