MAGPANTAY, NINO JANDEL C.

BSCS-4B

CCST-106 Perception and Computer Vision

MR. BERNARDINO

Exercises No. 4

Module 2.0: Feature Extraction and Object Detection

Performance Comparison: HOG-SVM vs. YOLO vs. SSD

When comparing HOG-SVM, YOLO, and SSD object detection techniques, each method offers unique strengths and weaknesses in terms of accuracy and speed. Below is a performance analysis and discussion of their advantages and disadvantages.

Comparison

• **Accuracy Comparison**: HOG-SVM relies on the extraction of gradient-based features (HOG) and uses a traditional Support Vector Machine (SVM) classifier for object detection. While it can provide reliable results for simpler objects in controlled environments, HOG-SVM's accuracy is often lower when compared to modern deep learning approaches like YOLO and SSD, especially when handling more complex, real-world images. YOLO, with its single-pass detection framework, offers real-time object detection with impressive accuracy, often outperforming traditional methods in terms of object localization. Similarly, SSD, which utilizes anchor boxes and a more fine-grained detection approach, tends to perform better than HOG-SVM in accuracy, though it is generally more resource-intensive.

• **Speed Comparison**: Speed is a critical factor in real-time applications, and here the differences are clear. HOG-SVM, while relatively lightweight, can be slower compared to YOLO and SSD due to its reliance on feature extraction and a separate classifier. YOLO, designed for real-time applications, is one of the fastest deep learning models available. It can perform detections in a single forward pass of the network, making it ideal for applications like autonomous driving. SSD, though slightly slower than YOLO, balances speed and accuracy by detecting objects at multiple scales, which allows for more precise detections but at the cost of computational overhead.

• **Advantages and Disadvantages**: The main advantage of HOG-SVM is its simplicity and lower resource requirements, making it suitable for applications where computational power is limited. However, its lower accuracy and inability to handle complex objects make it less desirable for modern applications. YOLO's primary advantage is its speed, allowing it to perform in real-time while maintaining high accuracy, although it may struggle with smaller objects in dense environments. SSD, on the other hand, excels at detecting objects of various sizes but tends to require more processing power, making it less suitable for real-time applications on resource-constrained systems.

• **Real-World Application**: In practical terms, YOLO and SSD are often preferred in scenarios that require both speed and accuracy, such as surveillance or autonomous systems. HOG-SVM, despite its limitations, can still be used for simple detection tasks or when hardware constraints necessitate lower computational demands. In scenarios requiring a balance between object detection speed and accuracy, SSD may provide the best results, particularly when deployed on powerful hardware.

• **Algorithm Complexity**: YOLO and SSD are more complex than HOG-SVM due to their deep learning architecture, requiring large datasets and GPU resources for training. HOG-SVM, however, can be trained with a smaller dataset and does not require GPU acceleration for inference, which is a significant advantage in environments where deep learning infrastructure is not available.

•**Computational Cost**: HOG-SVM is relatively cheap in terms of computational resources, while YOLO and SSD require more memory and computational power. YOLO is optimized for real-time detection but still requires a significant amount of processing power compared to HOG-SVM. SSD, with its multiple layers of detection, may have higher computational demands, making it more suitable for systems with ample processing resources.

Table of Comparison

Feature	HOG-SVM	YOLO (You Only Look Once)	SSD (Single Shot Detector)
Accuracy	Moderate (for simple tasks)	High (especially for larger objects)	High (effective for various object sizes)
Speed	Moderate (slower due to separate SVM)	Fast (real-time detection)	Moderate (slower than YOLO)
Real-Time Capability	Limited	Excellent	Good
Object Scale Handling	Poor (struggles with small objects)	Moderate	Excellent (detects objects at multiple scales)
Computational Cost	Low (CPU-friendly)	High (requires GPU for optimal use)	High (requires GPU)
Use Case	Simple object detection	Real-time applications	Balanced detection across object sizes
Disadvantages	Lower accuracy, slow classification	Struggles with smaller objects	Higher resource consumption