Adrien Loizeau
Wednesday 15th March, 2023

CentraleSupélec

Assignment 2 Report
Vision par ordinateur

This project entails a comprehensive approach that integrates HOG features, machine learning models, sliding windows, and heatmaps to accomplish the objective. This report will provide a detailed account of the various stages involved in developing our final model. For visual aid, figures that effectively illustrate the outcomes of our study are presented on the final page.

1 Description of the pipeline

1.1 ML Model

The first step of the project was to explore different methods for predicting the position of a bounding box.

HOG Features

To achieve this, the Histogram of Oriented Gradients (HOG) feature extraction technique was utilized. HOG features are effective for object recognition tasks, especially for pedestrian detection. This method captures the gradient information of an image and encodes it into a histogram, which can then be used as a feature for training a machine learning model. HOG features are invariant to changes in lighting and contrast, making them robust to different environments, and can capture the shape and edges of objects, making them effective for detecting pedestrians in images. An example of HOG feature extraction is presented in Figure 1. In the repository, the HOG features have been explored in the "notebooks/exploring-hog.ipynb".

Dataset

After deciding to use HOG features to extract information from the data, a labeled classification dataset consisting of "cars" and "no cars" was searched for. Since no suitable dataset was found, the "Car connection picture dataset" [3], containing over 60,000 images of cars, was utilized. Initially, it was planned to train a model with both the Car connection picture dataset and the images from the training dataset of the course (VIC dataset) [1] that had no cars. However, this approach did not work out as the images from the VIC dataset without any vehicle were all taken at night, as shown in Figure 2. Finally, the Vehicle Detection Image Set dataset [2] was utilized, which contains 17,000 labeled images with cars and without cars. After exploring and shuffling the dataset, it was used to train the machine learning model.

Model Training

To train the models, the dataset was split into 70% training data and 30% testing data. For this classification task, the Adaboost Classifier, Random Forest classifier, and SVM classifier were tested. Eventually, the random forest classifier was selected as it achieved the best results for the classification task. A comparison of the different models is presented in Figure 4. The final model had an accuracy of 97%.

1.2 Vehicle Detection

After obtaining the model, work began on a sliding window.

Sliding Window
The sliding window has four parameters: the size of the window, the step size, and the starting and ending y position. These parameters provide flexibility to the algorithm and enable focusing only on the interesting parts of the image. To define the y position at which to start looking for a vehicle, a data exploration was performed on the common position of the bounding boxes. This was presented in the form of a heat map in Figure **??**.

Prediction and Heatmap
For each window, a car was searched for. If it was predicted that a car was present, the position of the window was saved, and the prediction probability was added to a heat map. After exploring all the windows of an image, a complete heat map was generated, providing the most relevant position of a car. An example of this heat map is presented in Figure 6.

Using an adaptive threshold, the most relevant position of the bounding boxes was estimated from this heat map by creating a binary heat map based on its values.

Bounding Boxes Generation The binary heatmap (Figure 7) provides the spatial information of the final bounding boxes. In cases where multiple bounding boxes overlap, they need to be merged to create a bigger box. The connect_components function is responsible for merging these boxes. The output of this function is a list of positions of the boxes that can then be plotted on the image, as shown in Figure 8. The final output is a visual representation of the bounding boxes around the detected vehicles in the original image.

2 Results

Results
The results achieved by the pipeline are not very reliable compared to the state of the art deep learning methods. It took more than 40 minutes to predict the bounding boxes of the whole training dataset and is therefore very much not usable in a real-word case scenario. Never-the-less, if we compare the finale solution to the baseline one proposed, our pipeline achieves better performances and is more reliable. In summary, the model predicts very well when we have many cars in the image. If not, the model starts to found cars pretty much anywhere. This aspect should be more explored.

Improvements
To have better performances, we could try adding more information to our classification model. One way of doing so would be to not extract our HOG features from the RGB spectrum but from the YCrCb one. A code to do this was started in the "notebooks/exploring-hog.ipynb" file.

References

[1]

[2] Baris Dincer. Vehicle detection image set, May 2021.

[3] Paul. 60,000+ images of cars, Feb 2020.

Thank you for reading.

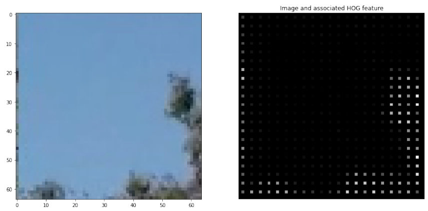

Figure 1: Example of HOG feature extraction on the first image of the training set

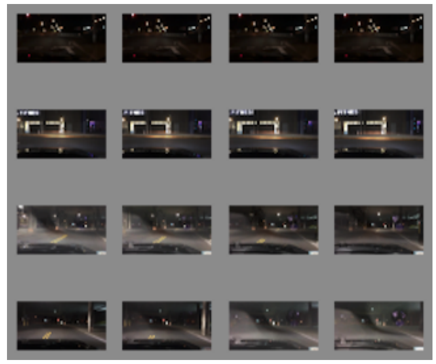

Figure 2: Sample from the VIC dataset where there are no cars

Figure 3: Sample from the Vehicle Detection Image set

	Model	Accuracy	F1score	Auc
	Adaboost raw classifier	0.94	0.94	0.94
	Random Forest raw classifier	0.97	0.97	0.97
	SVM raw classifier	0.97	0.97	0.97
	Extremely Randomized Trees raw classifier	0.97	0.97	0.97

Figure 4: Comparison between different machine learning models for the classification task

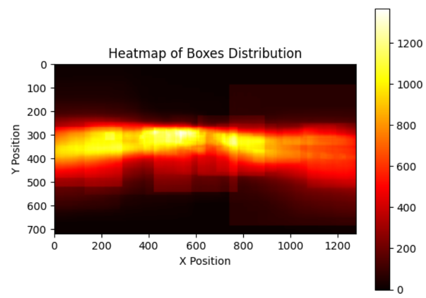

Figure 5: Heatmap the most common position for a bounding box

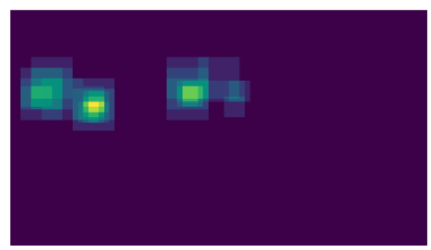

Figure 6: Heatmap of an image

Figure 7: Binary heatmap of an image

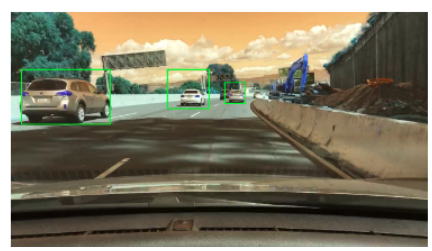

Figure 8: Final boxes of an image

Figure 9: Images