Visual Surveillance of On-Street Parking Spaces

Bejenariu David-Cosmin, 407

I. Introduction

Visual Surveillance of On-Street Parking Spaces is the second task of the Computer Vision course. This project aims to develop an automatic system for visual surveillance of on-street parking spaces in a specific scene. The system processes video streams from a static camera to:

1. Classify on-street parking spaces as occupied or not given a video frame.
2. Update the configuration of parking spaces over time given an initial configuration and a video stream.
3. Track a specific vehicle in the scene.
4. Count the number of vehicles stopped at a traffic light in the scene.

II. Task 1: Classifying On-Street Parking Spaces

The objective is to correctly classify the on-street parking spaces listed in a query file as *occupied* (label 1) or *not* (label 0) given a video frame.

1. Data Acquisition and Preparation

The training set consists of 50 training images and 50 corresponding query files, along with their ground truth annotations. Each query file specifies the parking spaces to classify in the associated image, and each ground truth file specifies only the queried parking lots and their state.

To follow the approach explained below, a **manual annotation** of the training images was necessary to determine the complete configuration of the parking spaces in each image, where all 10 spaces were labeled as either occupied or free. This manual annotation ensured more consistent and accurate ground truth data for training the classifier.

```
10
1 0
2 0
3 0
4 0
5 0
6 0
7 1
8 1
9 1
10 1
```

Figure 1. Manual annotation of the training data

2. Preprocessing

As an initial preprocessing step, the coordinates of the four corners of each parking lot on the right side of the street were manually extracted from an example image by opening a frame in a separate window with the help of *open-cv* and drawing the four corners of each space to set the coordinates.

Figure 2. Parking lot coordinates extraction

This step is crucial for accurately defining the **regions of interest (ROIs)** corresponding to each parking space and can totally benefit from the fact that all footage has been recorded by a static camera, which means the region coordinates are not subject to change.

Each parking space region was then **extracted** from the images and **warped** to a **top-down view** for consistent feature extraction. The warping process standardizes the orientation and size of each parking space, making feature extraction more reliable.

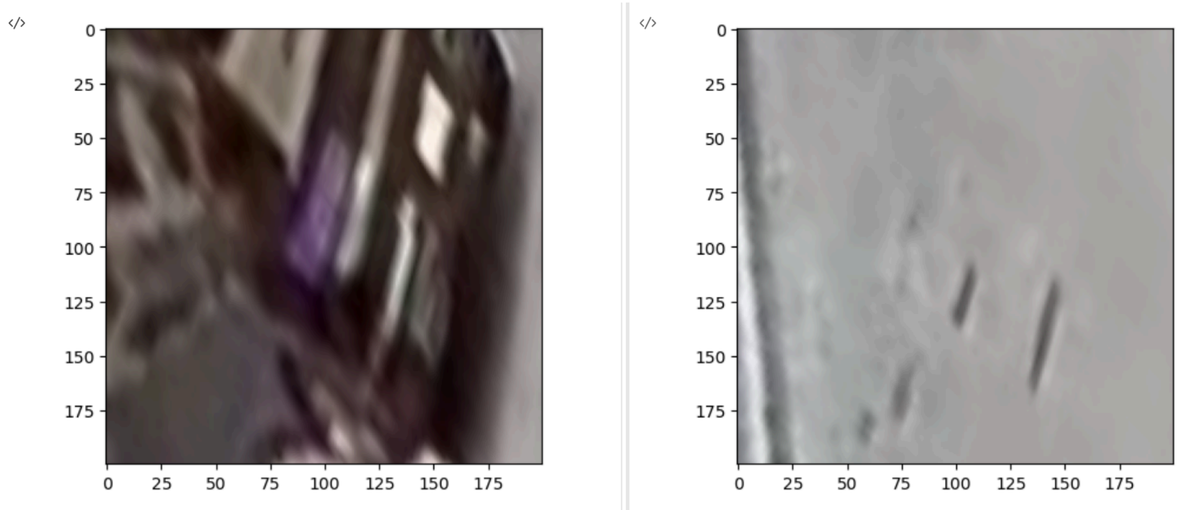

Figure 3. Occupied (left) versus non-occupied (right) parking space

Histogram of Oriented Gradients (HOG) was chosen for feature extraction due to its effectiveness in capturing edge and texture information, which are key indicators of occupancy status. The HOG features were computed for each warped ROI to generate a consistent and discriminative representation of the parking spaces.

3. Model Training

A **Support Vector Machine (SVM)** classifier with a linear kernel was selected for training due to its robustness in binary classification tasks and its effectiveness in high-dimensional feature spaces like those generated by HOG. The classifier was trained using the extracted HOG features from the annotated training images.

On a train-test split training approach, the model scored an accuracy of 100% in classifying whether a parking space is occupied or not.

4. **Prediction**

To classify parking spaces in a new image, the system:

- **Extracts and warps the ROIs** using the known coordinates.
- **Extracts HOG features** from the warped ROIs.
- Uses the trained SVM classifier to **predict the occupancy status** of each parking space.

III. Task 2: Updating Parking Space Configuration

The objective is to update the configuration of the 10 on-street parking places given an initial configuration and a video to process. The *initial configuration* characterizes the state of the on-street parking places at the beginning of the video, and the *final configuration* should characterize the state at the end of the video.

To correctly update the parking space configuration at the end of each video, the system would follow the steps below:

- **Extract final frame:** The final frame of the video is captured, as it represents the final state of the parking spaces. This frame is used for classification.

```python
def get_final_frame(video_path):
    cap = cv.VideoCapture(video_path)
    total_frames = int(cap.get(cv.CAP_PROP_FRAME_COUNT))
    cap.set(cv.CAP_PROP_POS_FRAMES, total_frames - 1)
    _, frame = cap.read()
    cap.release()

    return frame
```

Figure 4. Final frame extraction

- **Process final frame:** The final frame is processed to determine the state of each parking space, similar to the approach used in Task 1.
- **Extract and warp ROIs:** The same extraction and warping process as in Task 1 is applied to the final frame to standardize the parking space regions.
- **Feature extraction:** HOG features are extracted from the warped ROIs of the final frame to generate a consistent representation for classification.
- **Classification:** The already trained SVM classifier is used to predict the occupancy status of each parking space in the final frame.

- **Generate final configuration:** A binary vector representing the state of each parking space at the end of the video is generated. This vector is compared with the initial configuration to update the status of the parking spaces over time.

As a future improvement, it should be taken into consideration that, if the final frame is not clear due to occlusions, analyzing multiple frames or the entire video may be necessary to improve robustness. This can help handle transient events, such as cars temporarily blocking the view of parking spaces.

IV. Conclusion

This documentation outlines the methods used to solve the two primary tasks of classifying and updating the configuration of on-street parking spaces. The key steps involved manual annotation of training images, extracting and warping ROIs, and using HOG for feature extraction. A robust SVM classifier was trained to accurately classify parking spaces. The same approach was adapted for video processing to update the configuration of parking spaces over time.