University of Bucharest
Faculty of Mathematics and Computer Science
Department of Computer Science

Alexandra Diaconu
Bogdan Alexe
MSC in Computer Science
June 2024

Computer Vision - Project 2

Visual surveillance of on-street parking spaces

Objective

The goal of this project is to develop an automatic system for video analysis that enables visual surveillance of on-street parking spaces for a particular scene (Figure 1). The system gets as input data the video stream from a static camera in a specific scene and should be able to: (i) classify on-street parking spaces as being occupied or not given a video frame; (ii) update configuration of parking spaces given the initial configuration and a video stream to process; (iii) track a specific vehicle in the scene containing on-street parking spaces in a given video; (iv) additionally, the system should enable to count the number of vehicles stopped at a traffic light in the scene containing on-street parking spaces in a given video.

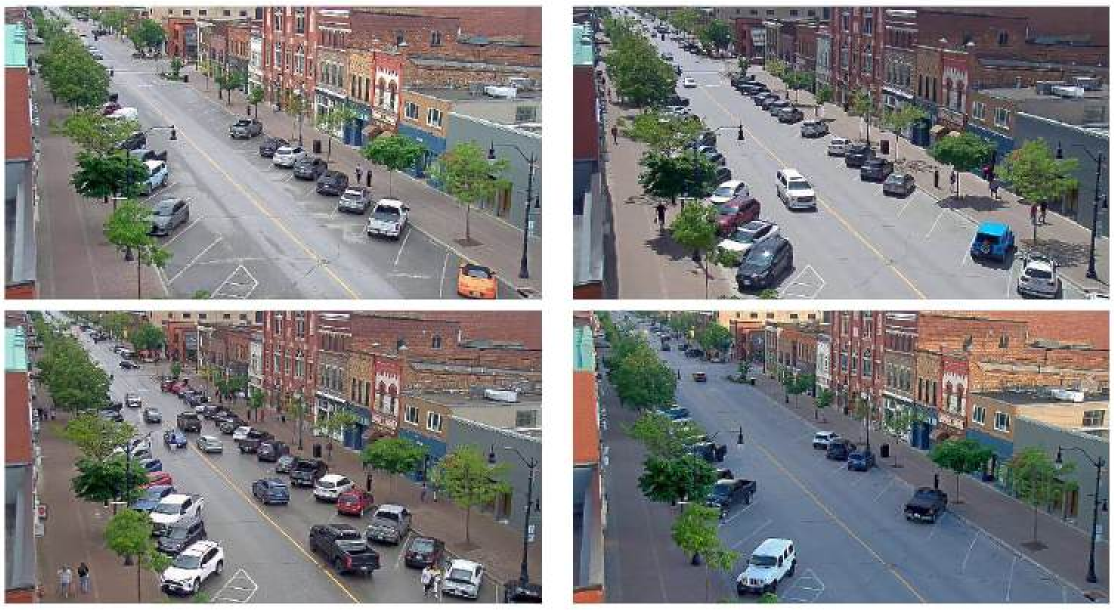


Figure 1: *The static camera acquires images of a scene containing on-street parking spaces at different times in a day/week.*

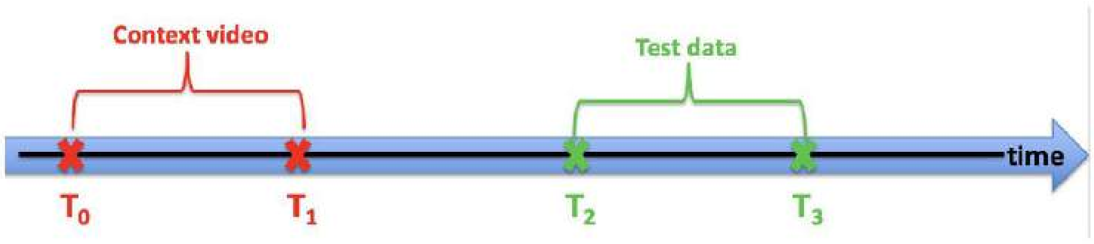


Figure 2: *The context video data is taken from time T_0 to time T_1, a few minutes before the test data is acquired. The test data is taken from time T_2 to time T_3.*

Introduction

Smart cities are being developed worldwide with the use of technology to improve the quality of life of citizens and enhance their safety. Video surveillance is a key component of smart city infrastructure, as it involves the installation of cameras at strategic locations throughout the city for monitoring public spaces and providing real-time surveillance footage to law enforcement and other city representatives.

In this project you have to analyze video data obtained from a static camera with the desired goal of visual surveillance of on-street parking spaces. The videos are collected at different times during day/week so it is easy to observe that there are some changes in illumination in the scene specific to each video (Figure 1). For each of the four tasks that you have to tackle in this project you are given a "context" video that is taken with a few minutes before the test data (images or video) that you have to analyze was acquired using the same static camera (Figure 2).

Data description and grading scheme

The release data directory (available at `https://tinyurl.com/CV-2024-Project2`) contains three directories: *train*, *test* and *evaluation*. The directories *train* and *test* have similar structure, although the *test* data will be made available after the deadline. The *train* directory contains the data organized in four subdirectories corresponding to the four tasks that you need to solve and the context videos shared by these tasks. The subdirectories are:

- *context_videos_all_tasks* - this directory contains 15 training videos which represents the context video data for each of the four tasks. Notice that each video is taken at a given moment of a day from T_0 to T_1.
- *Task1* - this directory contains 50 training images (3 images corresponding to the first 10 training (context) videos and 4 images corresponding to the remaining 5 training videos - so in total there are $3 \times 10 + 4 \times 5 = 50$ training images) and 50 query txt files, each query file corresponding to a training image. Notice that all images are taken from T_2 to T_3.

  The Task 1 consists in correctly classifying the on-street parking spaces listed in the query file as being occupied (label 1) or not (label 0). The on-street parking spaces are

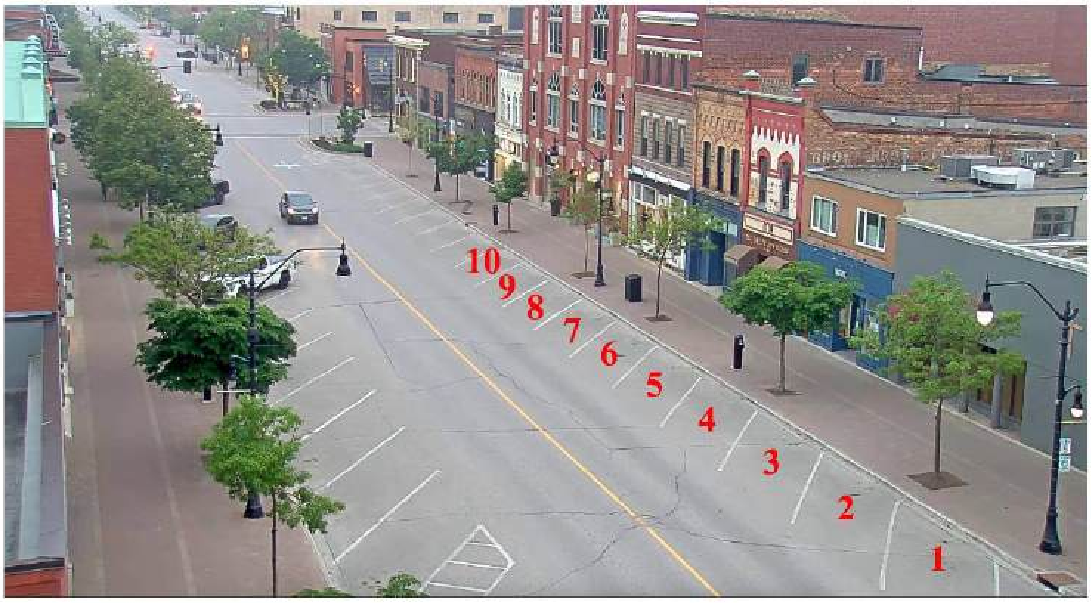


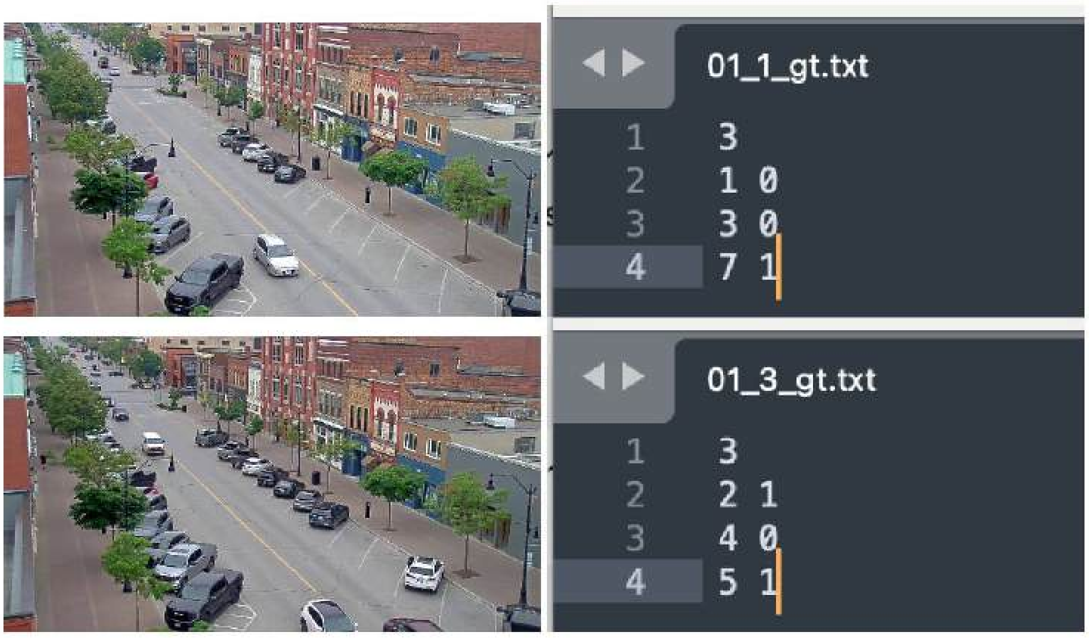
Figure 3: *On-street parking places are numbered in the right part of the street, from 1 to 10 from the lower part of the image/scene to the upper part of the image/scene. We do not consider parking places positioned in the left part of the street.*

numbered as in Figure 3, in the right part of the street, from 1 to 10 from the lower part of the image/scene to the upper part of the image/scene. We do not consider parking places positioned in the left part of the street.

The format of the query files is the following: on the first row it is specified the number P of parking places for which you have to do the binary classification and then each of the following P rows lists a parking place number. Figure 4 shows two training examples and the corresponding query files/ground-truth files. Please also refer to the examples presented in Lecture 13.

Grading scheme for Task1. In the test scenario we will release 15 testing videos that will offer the context data. For Task 1 we will release 50 testing images, following the same distribution as in training ($3 \times 10 + 4 \times 5 = 50$ testing images). By correctly solving the Task1 on all images you will get 1.5 points. You get 0.03 points/image if your algorithm solves correctly the binary classification for all parking places whose position is specified in the query file.

- *Task2* - this directory contains 15 training videos taken in the interval from T_2 to T_3. The task is to list the configuration of the 10 on-street parking places given the initial configuration and a video to process. The initial configuration is represented by a binary vector of length 10, with component i being 0 (free parking space) or 1 (occupied parking space) corresponding to the i-th parking space, as numbered in Figure 3. The initial configuration characterizes the state of the on-street parking places at the beginning of the video. The final configuration should characterize the state of the on-street parking places at the end of the video.



```
3
1 0
3 0
7 1
```

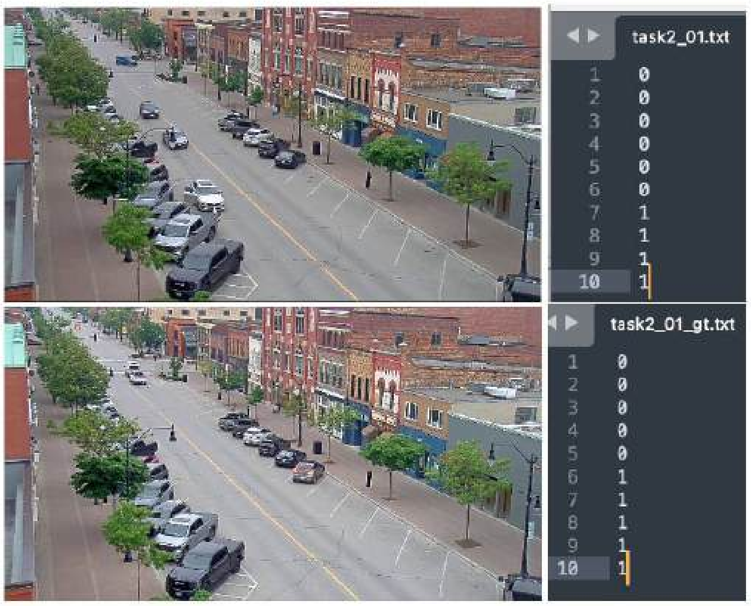
```
3
2 1
4 0
5 1
```

Figure 4: *Examples for Task1. Left column: query images. Right column: ground-truth annotation files, containing on the first row the number P of parking places for which you have to do the binary classification and then each of the following P rows lists a parking place number. Each ground-truth file includes the correct answer for each query.*

Grading scheme for Task2. In the test scenario we will release 15 testing videos similar with the training videos. By correctly solving the Task2 on all videos you will get 0.75 points. You get 0.05 points/video if your algorithm correctly outputs the final configuration of the on-street parking places. Partial points are not awarded if you solve only partially the task (for example correctly classifying only two of the total three parking places).

- *Task3* - this directory contains 15 training videos taken in the interval from T_2 to T_3. The task is to track a specific vehicle from the initial frame to the final frame of the video (see Figure 6). The initial bounding box of the vehicle to be tracked is provided for the first frame (the annotation follows the format [xmin ymin xmax ymax], where (xmin,ymin) is the top left corner and (xmax,ymax) is the bottom right corner of the initial bounding-box).

  In each video we will consider that your algorithm *correctly tracks the vehicle* if in more (greater or equal) than 80% of the video frames your algorithm *correctly localizes the vehicle to be tracked*. We consider that your algorithm *correctly localizes the vehicle to be tracked* in a specific frame if the value of the IOU (intersection over union) between the window provided by your algorithm and the ground-truth window is more than 30%. The format that you need to follow is the one used in the ground-truth files (located in the sub-directory *ground-truth*) with the first line containing the number of frames N of the video, and each line having the format [frame_index xmin ymin xmax ymax]. The first frame for which we provide the bounding box initialization has frame_index 0, the last frame of a video with N frames has frame_index $N-1$.



task2_01.txt

```
0
0
0
0
0
0
1
1
1
1
```

task2_01_gt.txt

```
0
0
0
0
0
1
1
1
1
1
```

Figure 5: *Examples for Task2. First row: the first frame of the query video and the corresponding initial configuration. Last row: the last frame of the query video and the corresponding final configuration.*

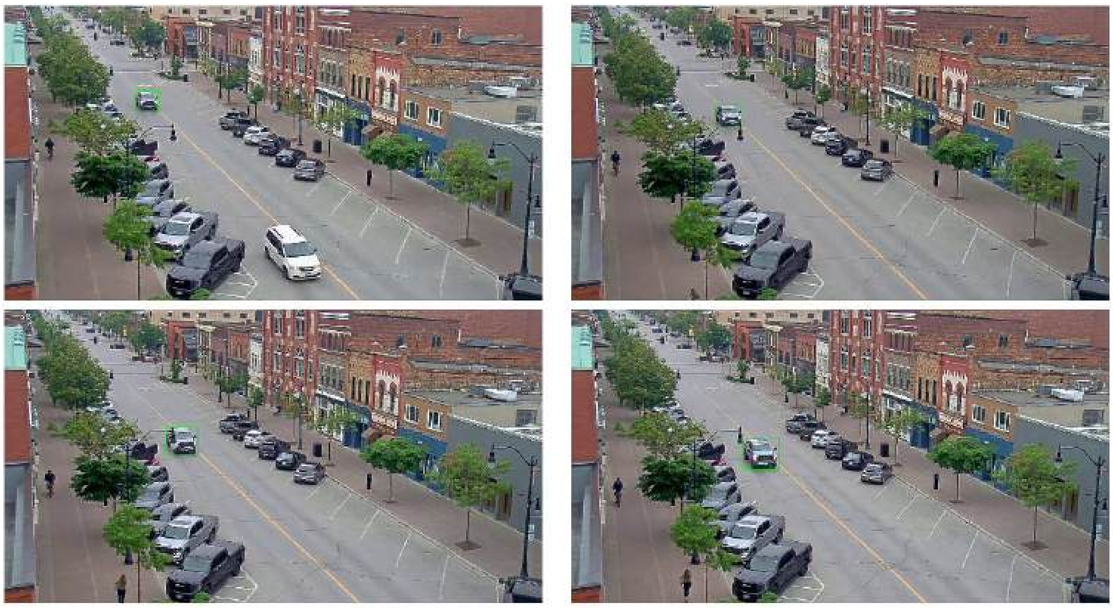


Figure 6: *Examples of train videos for Task3: the figure displays four frames of a training video containing annotation (green bounding-box) of a specific vehicle.*

Please note that the first line of the annotation file has the format [N -1 -1 -1 -1] as it is easy to load an entire matrix $M \times 5$ (first line is [N -1 -1 -1 -1], then the following $M - 1$ lines are of the form [frame_index xmin ymin xmax ymax], so $M \leq N + 1$) to assess the correctness of your algorithm. Notice that if the vehicle disappears from the video your algorithm should not output any detection in the corresponding frames (in this case $M < N + 1$).

Grading scheme for Task3. In the test scenario we will release 15 testing videos similar with the training videos. By correctly solving the Task3 on all videos you will get 1.5 points. You get 0.1 points/video if your algorithm correctly tracks the vehicle in a video.

- *Task4* - this directory contains 15 training videos. The goal here is to count the number of vehicles which queue to the traffic light located in the upper part of the scene (see Figure 7).

  Grading scheme for Task4. In the test scenario we will release 15 testing videos for Task4. By correctly solving the Task4 on all videos you will get 0.75 points. You get 0.05 points/video if your algorithm correctly predicts the number of vehicles queued at the traffic light in a given video.

- 0.5 points - ex officio. Please note that we will award the 0.5 points only to those students who submit their results in the REQUIRED format.

The oral presentation of this project (face-to-face or online) will be scheduled after the deadline, in the week $26 - 28$ of June. It will take around 10-15 minutes in which Alexandra or Bogdan will ask questions regarding implementation. The oral presentation will count for 0.5 points and is mandatory for each student submitting his solution for this project.

Evaluation

The directory *evaluation* shows how the evaluation will take place on the test data after the deadline. It contains the following subdirectories:

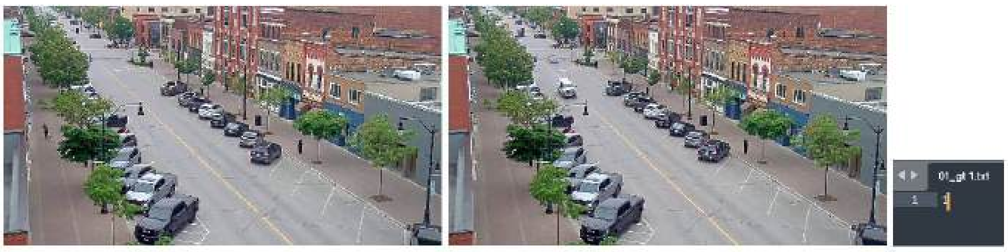


Figure 7: *Example for Task4. There is just one vehicle queueing at the traffic light in the upper part of the scene.*

- *fake_test* - this directory exemplifies how the test data will be released in the *test* directory (similar with Project 1), keeping the structure of the previously described *train* directory;
- *submission_files* - this directory exemplifies the format of the results data that we expect from you to submit in the second stage. You will have to send your results in this format, uploading a zip archive of a folder similar with the one called *Alexe_Bogdan_407*. Please note that if you don't submit your results in this format you will not get the 0.5 points from ex officio;
- *code_evaluation* - this directory contains code that we will use to evaluate your results using the ground-truth data. Make sure that this code will run on your submitted files. The ground-truth data will be released after you send us your results.

Deadlines: Submit a zip archive containing your code, all auxiliary data that you are using (templates, models, etc.) and a pdf file describing your approach until Tuesday, 25th of June using the following link `https://tinyurl.com/CV-2024-PROJECT2-SUBMISSIONS`. Please do not include in your archive training images or videos. Notice that this is a hard deadline, no projects will be accepted after the deadline. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run the solution and where to look for the output file. Students who do not describe their approach (using a pdf file) will incur a penalty of 0.5 points.

On Wednesday 26th of June we will make available the test data. You will have to run your solution on the test images provided by us and upload your results in the same day as a zip archive using the following link `https://tinyurl.com/CV-2024-PROJECT2-RESULTS`.