SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

CSC3067

Video Analytics and Machine Learning

Group Assignment

Pedestrian Detection

Jesús Martínez del Rincón

j.martinez-del-rincon@qub.ac.uk

Baharak Ahmaderaghi

B.Ahmaderaghi@qub.ac.uk

QUEEN'S UNIVERSITY BELFAST

Learning outcomes

1. Explain when and how machine learning and computer vision is useful in industry, public institutions and research.

2. Know and apply a range of basic computer vision and machine learning techniques.

3. Demonstrate the ability to understand and describe the underlying mathematical framework behind these operations

4. Design and develop machine learning pipelines applied to computer vision applications

Learning outcomes

5. Formulate and evaluate hypothesis

6. Evaluate the performance of proposed machine learning solutions through rigorous experimentation

7. Analyse quantitative results and use them to refine initial solutions

8. Communicate finding effectively and in a convincing manner based on data, and compare proposed systems against existing solutions

Assessment - Coursework

- Practical assignment (40%)
  - More details on Week 6
  - In groups of 3
  - Learning outcomes 4 to 8

10%	Initial baseline system	Week 9
30%	Final report and code	Week 11

- Must attain at least 40 % (24 marks) of total marks available in the coursework

Critical Analysis

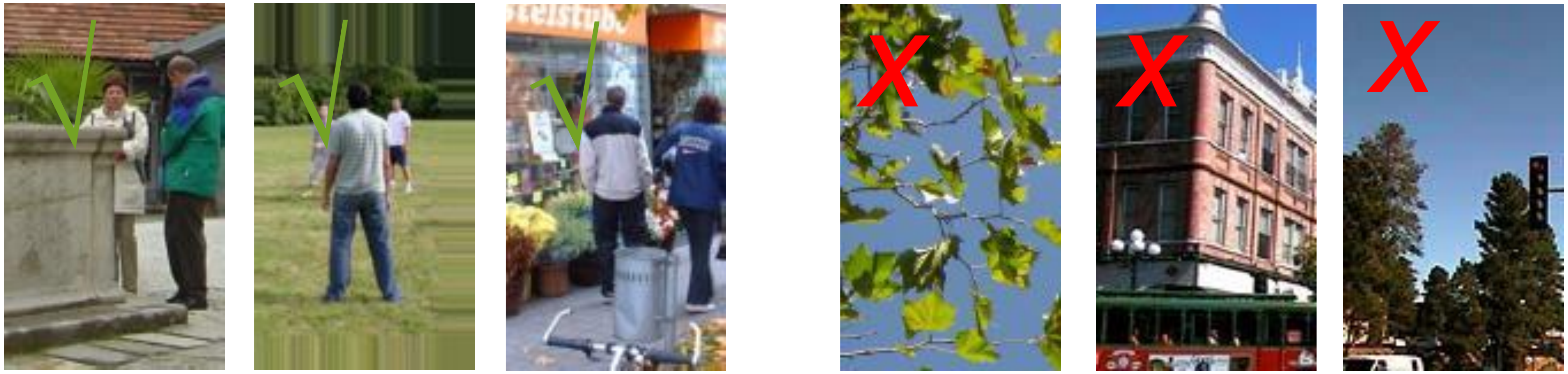
- Justify every single decision based on objective data and reasoning
- For example:
  - *"We applied SVM."* 🚫
  - *"We applied SVM because it is a good classifier."* **Mehh**
  - *"We applied SVM because its requires few training images, which fits our problem where few images were available. "* **Good**
  - *"We applied SVM because its requires few training images, which fits our problem where few images were available. This was supported empirically by its comparison with NN, where a 10% was achieved, as shown in table 1"* **Excellent!!!**

Assistance

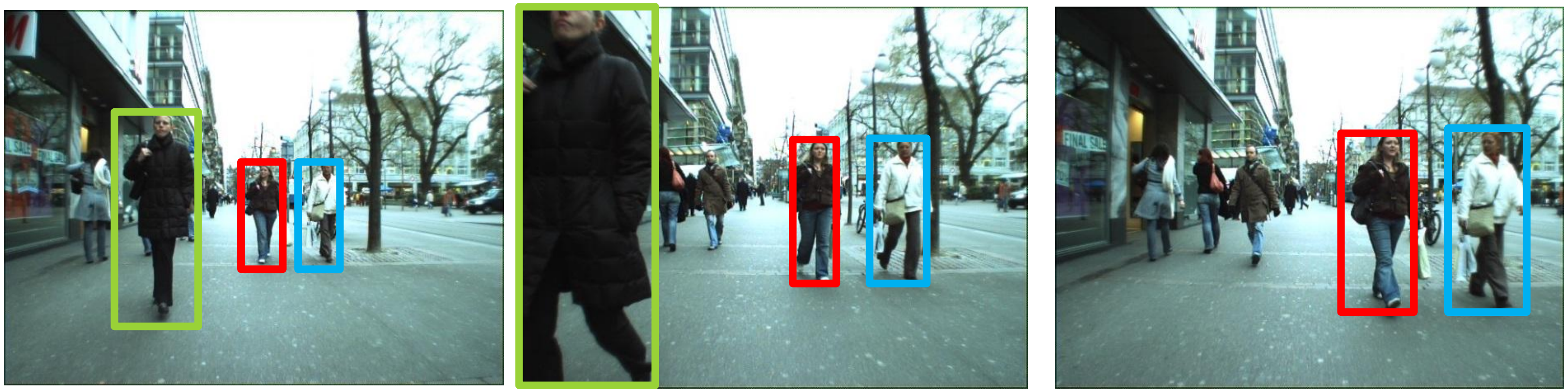
- We are here to help
- If you don’t understand then please ASK
  - Email: j.martinez-del-rincon@qub.ac.uk, B.Ahmaderaghi@qub.ac.uk - No response to queries for which the information is already provided
- Office Hours
  - Jesus: Wednesday 14:00 - 17:00 16 Malone Roal 01.008
  - Baharak: Tuesday 11:00 – 12:30
    Thursday 11:00 – 12:30 21 Stranmillis Road
  - MS Teams: Feel free to call, share your screen, etc
  - Drop us an email before!

Topic

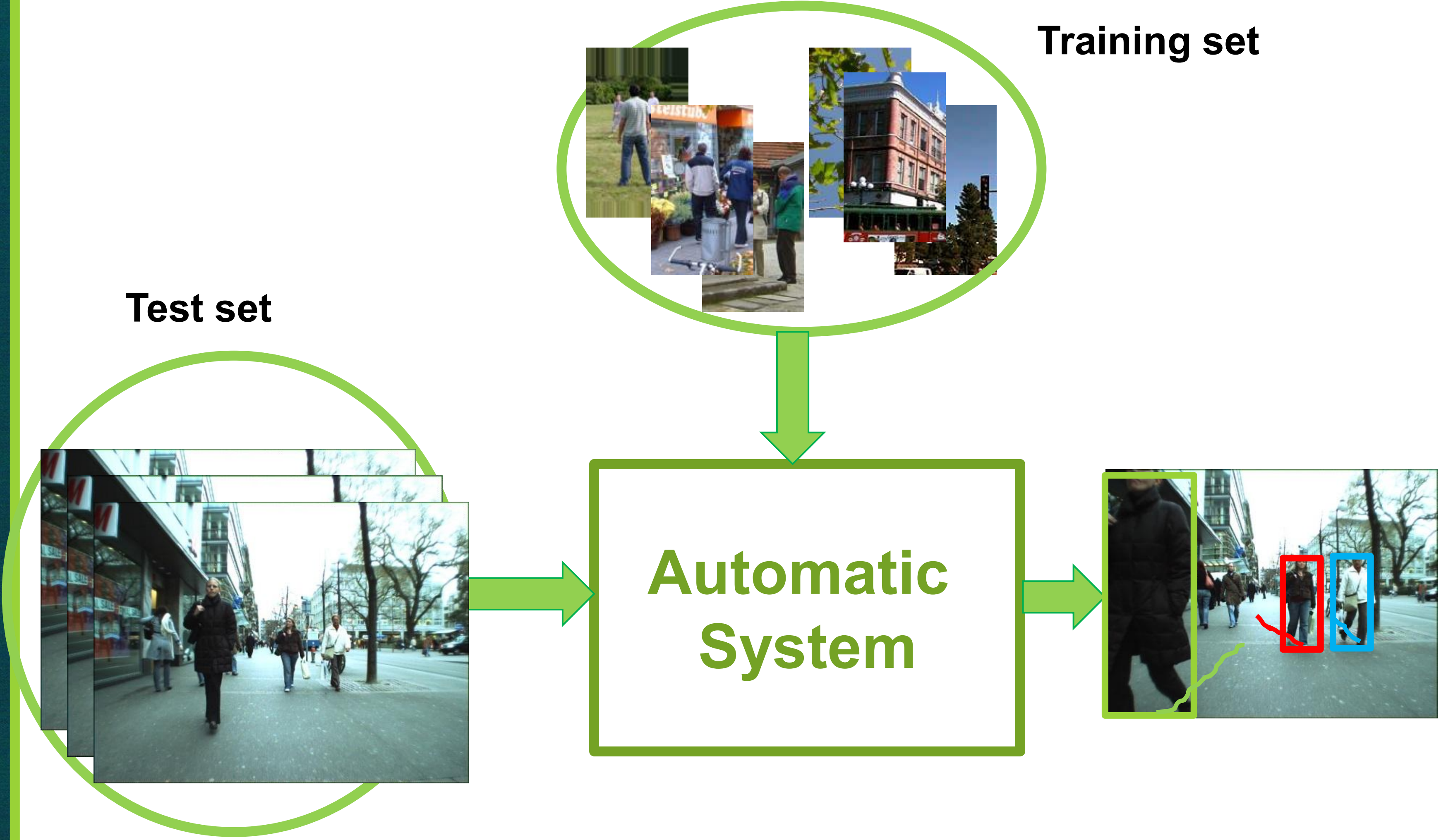
- Pedestrian Detection
  - Machine Learning



  - Detection



Pedestrian Detector



Training set

Test set

Automatic System

Training the automatic system (32 marks)

- 2 Main components
  - **A feature descriptor**.
    - Describe an image region with a high-dimensional descriptor.
    - Options:
      - ***Histogram of gradients (HOG)*** features. (Prac 5)
      - Full image (Prac 4 & 5)
      - Dimensionality reduction techniques (Prac 6)
  - **A learning method**. (Prac 4)
    - Learn to classify an image region (described using one of the features above) as a pedestrian or not.
    - Options:
      - **SVM**
      - Nearest Neighbour,
      - K-NN
- Folders *images\neg* and *images\pos* contains crops to be used for training
- Justify your choice and **the parameter values**. Analysis

Testing the classification system (21 marks)

- In order to justify previous choices, we need to divide dataset (both positive and negative examples) in 2 subsets: training and testing
  - Training samples and their labels for learning
  - choose the best techniques, strategies and parameters for each block using the testing
  - Options:
    - Half/half
    - Cross validation

- Evaluate the performance of your final choice
  - Options:
    - Recognition rate
    - TP, FP, TN, FN (Prac 6)
    - Precision, recall, specificity, sensitivity, etc… (Prac 6)

Detection implementation (37 marks)

- Apply your classification/verification system to implement a pedestrian detector
- 2 important components
  - **A sliding window detector**. (Based on Prac 4)
    - Crop the image at every location and use the classifier to tell if that image region contains a person. By scanning every location on the full image, it will detect all instances of pedestrians in that image.
    - In order to detect pedestrians at multiple sizes, our sliding window detector should run at multiple scales (will require resizing image)
  - **Non-maxima suppression**. (Prac 5)
    - Overlapping detections are a common problem. NMS removes overlapping detection to improve performance. It keeps best detections in each region by selecting the strongest responses.
- Using *test.dataset* file, we can read the real location of people in the image sequence
  - For each image, the (x, y, width and height) parameters of the bounding box of each person in that frame, so it can be compared against our output
  - Calculate the performance of our detector quantitatively
  - Generate output video
  - Reflect and explain the results that you have obtained
    - Why and when it fails?

Available code

- No implementation, only usage
  - Prac 4 provides a SVM toolbox and matlab library alternative
  - Prac 5 provides a HOG feature extractor
  - Prac 6 provide LDA dimensionality reduction method
- You should already developed in the practicals (but you can improve them)
  - Preprocessing: Prac 1-2
  - NMS at Prac 5
  - K-NN classifier at Prac 4
  - Tracking filters for comparison at Prac 7
  - Concepts of sliding windows at Prac 4 and 5
  - Video recorder at Prac 3
  - Evaluation metrics at Prac 6
  - PCA at Prac6
- Main bits to implement:
  - Multi-scale sliding window

Basic System

- As many combination of methods (specially feature extraction and classifiers)
  - The final performance does not determine the mark
- Correct experimental setup
- Correctly evaluated (quantitative data)
  - Tables, graphs,…
- Good parameter setting
- Good analysis and explanation of results and decisions taken

Excellent system +Extra Credit (10 Marks)

- Use of Boosting/bagging, Random trees or NN/DNN for classification
- Cross validation (leave one out) as experimental setup
- ROC curves to evaluate
- Explore your own feature extractor
- Part-based detectors
- Tracking by detection

Initial Demo (10%)

- Date: 12th -14th November
  - With Baharak or me or a demonstrator
  - 10 min per group
  - Using Ms Teams
- Formative and summative
  - Opportunity to get feedback on your progress, alternative things to try, what to do until final submission, etc..
  - You will get a marking template and written feedback
  - 10% of the final mark
- Goal and expectations:
  - Show real progress at that stage. Results of some initial combinations of features+classifiers, initial attempt to the detector, parameter tuning,….
    - Prac4 and 5 completed give you a good baseline
    - The more, the better
  - You will need to submit the presented slide deck, tables, results etc..
    - The more professional the better

Deliverables

- Written report (4th December) } Through Canvas
- Code (4th December) } Gitlab Repo
- Peer Assessment (5th December) } Through Canvas

Report

- In order to obtain all the points in each section, the report must be clear and explain and justify **each** decision
  - Good structure
  - Logical order: training, testing, etc…
  - The more pictures and tables with results, the better

<u>Module guide</u>
Skills:
1. Communication
5. Working with others

Code

- I must be able to run it
  - And understand your code
    - Comments
    - Good structure
- Gitlab - Compulsory
  - I will provide you a git per group in the next few days
  - Code
  - Videos of your results
- I should obtain the same results that in the report
  - Not all the intermediate results, but the final ones

Peer Assessment

- Professional behaviour
  - Respect to the others
  - Me as a moderator. Raise issues as soon as possible.
- WARNING!! All the members are supposed to contribute in all the aspects
- Criteria: 4 marks (out of 10) in 4 categories
  - Design, Implementation, Testing, Report
  - Give also a general comment summary (1 paragraph)

Peer Assessment

- Peer assessment
  - Mark *M* given by the moderator
  - Individual mark $M_i = M/2 + a_i * M/2$
    - Where a_i is calculated by members marking anonymously.

Peer Assessment

- Hanna’s Formula $a_i = \frac{P_i}{\sum_{j=1}^{N} P_j}$

- Example: 3 member, M=15

Member 2 about Member 1	**Member 3 about Member 1**
Design =8	Design =7
Implementation=7	Implementation=7
Testing=7	Testing=7
Report=9	Report=8
Total =31	**Total=29**

⇨ P1=31+29=60

$$a_1 = \frac{P_1}{(P_1+P_2+P_3)/3} = \frac{60}{(60+58+66)/3} = \frac{60}{61.3} = 0.97$$

$$M_1 = \frac{15}{2} + 0.97 * \frac{15}{2} = 14.7$$

Issues and demo

- Raise group problems as soon as possible
- Moderating the mark
  - Analysing group member contribution
  - Use Gitlab to measure individual contributions
  - I keep the right to fully remark individuals and groups
- If needed a group interview will be called
  - All members of the group should be present
  - We will agree a date