ENGN4528/6528 Term Project Guideline

Abstract

The goal of the term project is to offer students the opportunity of design and implementation of a working computer vision system, providing hands-on experiences and related research methods. The term project will involve some algorithm implementation and experimentation and a submission of a written project technical report at the end of the semester. There are several project topics to choose from, which will be described below.

- The technical report should be 8 pages long in an A4-PDF format and be submitted to Wattle on the submission due date (TBC).
- In solving the term-project, students are asked to present solutions by combining existing methods or by proposing their own solutions.
- A half-page proposal with the topic chosen, group members, and a brief plan how to address the problem should be submitted to Wattle by the due date (Friday 14 April).
- It is important that you decide on the project group and topic that and then focus on the project throughout the semester. Later change of mind is discouraged, and each of the later changes will result in a loss of 20% marks of the total mark of your term project.
- You may use Matlab (preferred), Python, Java or C/C++. You may call OpenCV functions if you want, though this is optional. Other programming languages are also acceptable.

Operating System

Windows is preferred. Tutor may compile and run your source code on a Windows-based machine.

Group Policy

ENGN4528: The term project is a "group project", and each group can have up to 4 undergraduate students. Teamwork is important and essential for doing term project, so the minimum size of each group is 2 students per group. Each individual's contribution will be assessed through a peer-assessment reported by your team-members confidentially to tutors, and the final marks for students from the same group can be different.

ENGN6528: The project can be done either individually or as a group with maximum 2 students. For a group type work, each individual's contribution should be mentioned in the report (e.g. Student-A 50% and Student-B 50%).

Assessment

ENGN4528: The assessment of the project will be based on

1. Final project demonstration (**30 marks**) – During your demo session, tutors will ask technical questions, and your answers will directly affect your final project mark.
2. A 8-page project group report (**20 marks**) – It is a group report but should be submitted individually, containing at least the following contents:
    - Introduction: Problem statement –what, why, with real-world applications
    - Related work: What has been done, according to literature survey, with a few references
    - Your approach: Conceptual design, method, algorithms, equations and figures
    - Your implementation and experiment results: What you do, what are your results, with images, tables, and numbers
    - Conclusions: Itemized conclusions, observations and discussions, future work
    - Learning outcomes: Provide at least 3 learning outcomes and experiences about what you have learned from doing the project.
    - Commented source codes as a separated zip-file with a "readme.txt" file: The readme file should explain how to run the codes so that tutors can test the codes

ENGN6528: The assessment of the project will be based on

1. Final project demonstration (2**0 marks**) – During your demo session, tutors will ask technical questions, and your answers will directly affect your final project mark.
2. Literature survey and review (**10 marks**) – It is a report on literature review should be submitted individually (during week 12 and the due date: TBA).
    - The report should have 1500 – 2000 words (roughly 3–4 pages) in the area of modern computer vision research. The starting point can be the term project and students should search for the open research problems and gaps.
    - The report is not a simple summary of the existing work. Instead the existing work should be summarised, evaluated and criticised. Successful literature often provides the ground for the new research.
3. A 8-page project group report (**20 marks**) – It is a group/individual report but should be submitted individually, containing at least the following contents:
    - Introduction: Problem statement –what, why, with real-world applications
    - Related work: What has been done, according to literature survey, with a few references
    - Your approach: Conceptual design, method, algorithms, equations and figures
    - Your implementation and experiment results: What you do, what are your results, with images, tables, and numbers

- Conclusions: Itemized conclusions, observations and discussions, future work
- Learning outcomes: Provide at least 3 learning outcomes and experiences about what you have learned from doing the project.
- Commented source codes as a separated zip-file with a "readme.txt" file: The readme file should explain how to run the codes so that tutors can test the codes

In principle, your term project will be marked based on all the following components.

- Successful demonstration of your project
- The efforts and thoughtfulness that you have put into the project
- Any innovation & creativity idea that you have proposed and/or have implemented
- Teamwork and collaboration (based on the confidential individual contribution ratio report)

Project Topics

- **ENGN4528**: Please choose one topic out of four projects listed below. In some case, students can propose their own project idea but it should be approved beforehand.
- **ENGN6528**: Please choose one topic out of four projects listed below, or students can decide their own project proposal.

	Proposed Project Title
1	Fruit Counting: how many orange/pears/bananas are there?
2	Angry birds: Spot the bird, pigs, etc.
3	Face Detection and Recognition
4	Panoramic Image Stitching

Project Option 1: Fruit Counting: how many orange/pears/bananas are there?

Abstract: You are given 6 video clips which contain five types of fruits in the videos: apple, orange, banana, pear, and tomato. You program should be able to detect, recognize and count at least 3 type of fruits types. You decide which three are the targets to detect, recognize and count. The program should be able to

(1) Read in the video clip

(2) Display the video frames

(3) Detect all the fruits in each frame, segment the corresponding pixel region, count the number of pixels (i.e., the size of each fruit), and draw a bounding box on each fruit

(4) Track the detected fruits until they disappear

(5) Count the number of fruits of each type, and update the counting result in real-time

(6) At the end of each clip, report the success rate (100%) of your program. You can manually count the objects in the video, and obtain the ground-truth data.

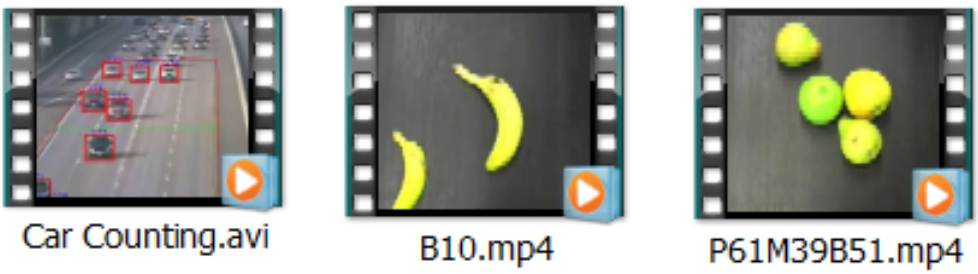

Figure 1 Car counting example and fruit videos

Requirement: Your program must perform in near real-time (e.g., process speed is at 5+ frames per second.) Hint: this system is in fact a toy-example of traffic video monitoring system: car detection, car tracking and car counting.

In the zip file, you will find an AVI video, which gives such an example. Your program's GUI (user interface) should look like this one.

Marking Criteria

- Detect and track each fruit, and report its size and name. Succeed on the first 3 video clips.
- Succeed on all the 6 video clips.
- Detect and play a short alarm sound whenever a non-fruit object is detected on the belt.
- Project report and codes

Project Option 2: Angry birds: Spot the bird, pigs, etc.

(Project idea credit: Dr Jochen Renz, Dr Steve Gould: Angry Bird Aus-AI Contest;**)**

Abstract: In this project, you are given a video clip of the "angry birds" (in game trailer). You are asked to code a computer vision program that can:

(1) Read in the video clip, and display the video frame by frame

(2) Detect all the pigs in each frame, draw a bounding box on each pig

(3) Detect the slingshot

(4) Detect the bird which is either waiting on the slingshot, or is flying. Draw a bounding box around the bird in each frame

(5) Compute the (x, y) coordinates of the detected bird, relative to the slingshot. Set the slingshot's base point as the origin

(6) Fit the trajectory of the bird by a parabolic curve. Display the (parametric) equation of the parabolic curve. Plot the fitted parabolic curve on screen, overlay with the video.

Figure 2 Angry bird detection in a slingshot

Requirements:

- Your pig or bird detector must be able to handle rotation and scale change.
- Your program must perform in near real-time (e.g., process 5+ frames per second.)
- Programming language: Matlab, or Python, or C/C++, or Java.

Marking criteria:

- Detect and track all pigs, and draw bounding boxes.
- Detect and track the bird, and draw a bounding box around it.
- Convert all the coordinates of the bird relative to the slingshot's position.
- Draw correctly the parabolic curve of the bird's trajectory; display its parameters of the curve
- Project report and codes

Project Option 3: Face Detection and Recognition

Abstract: We wish to develop a computer vision technique that can automatically tell who are present & absent in a group photo. For this purpose, a computer program should be able to the following tasks.

(1) Given two photos as input:

- A class photo taken in lecture-theatre during the class time, and
- A frontal face photo of a student

(2) Automatically performs the following two tasks

- Count how many students are in the class photo;
- Verify whether or not the given student's face appears in the class photo.

In doing this project, you may choose to use any face detection, and face verification or face recognition techniques, and are not limited by deep-learning.

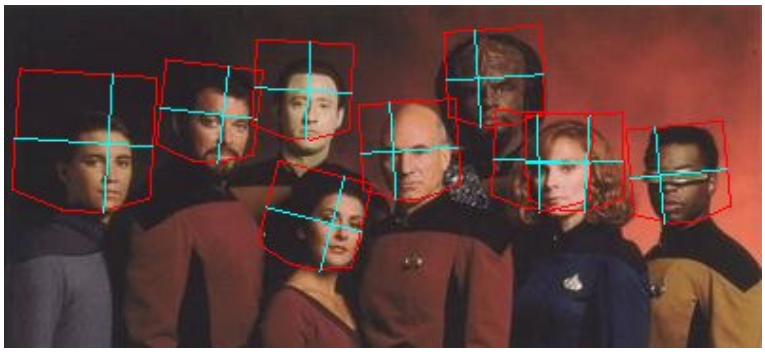

Figure 3 Face detection example (From https://www.cs.nyu.edu/~yann/research/cface/)

Real-time requirement: In the testing time, your program must complete both tasks within 20 second's time on a regular laptop/PC.

Data set and Privacy Note: If you choose to work on this project, please contact Lecturer/Tutors to get the class photo. Privacy Note: Please do not use the photos for any other purpose. Please never share or re-distribute the photos anywhere in any circumstance.

Project Option 4: Panoramic Image Stitching

Abstract: In this project, you are asked to implement a program to automatically stitch selected key frames from a video clip into a single 360-degree panorama. The recommended workflows are

(1) First, you need to take a continuous 360-degree video of an outdoor scene, using your mobile phone or a camcorder while rotating 360 degrees.

(2) Implement a program in Matlab or other language, which reads in the video, and then automatically select some key-frames.

(3) Detect SIFT points on the key-frames. Matching these SIFT points between neighbouring key-frames. For matching, the nearest neighbour using L2 (Euclidean) distance and sorted. A match is defined as one whose distance is less than 0.6 times the distance of the second best nearest neighbour.

(4) Estimate a homography between the key-frames. (Hint: Before doing this, you may use RANSAC to prune the matches and remove those outliers. The basic idea is to randomly select four points, compute the homography from these four points and checks how many other points agree (measured in terms of SSD (Hx1 - x2)). Repeat many times and choose the homography with the most agreement. The final homography is computed based on all inlier points. Alternatively, you can also manually remove outliers.)

(5) Warp the neighbouring frames using the estimated homography, and blend the pixel colour by a suitable method. You need to design a suitable blending method to reduce the colour distortion.

(6) Repeat the above steps until a full 360-degree panorama is generated. Crop the resulted image to a rectangular one.

(7) Save the panorama in a jpeg file.

(8) Display the original video, and also display the panorama on screen.

Figure 4 Panoramic stitching example (From: http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/f07/proj4/www/lisachan/)