Engineering | Lancaster University

# DESIGNING A VISUAL SENSOR SYSTEM FOR ESTIMATING BUILDING OCCUPANCY

GATE REVIEW

THOMAS LUMBORG

# Contents

# 1. Project Introduction

## 1.1 Project Objective

The objective of this project is to design and test a system to estimate the occupancy of a building using a visual sensor. The aim is to give the owner/manager of the building information on the amount of people at a given time, which can be used to adjust lighting, heating or to just show how busy the building gets.

The system will realistically comprise of a camera and some sort of small computer such as a raspberry pi. It could also use a network connected camera and a computer in a different location.

The system will use object detection and tracking to determine when a person enters or exits the building and therefore estimate occupancy.

Object detection is the classification of type of object (person, dog etc.) along with the location of said object. This differs from object tracking where a pre-detected object is simply tracked, and the location updated. It is less accurate but requires much less computational power.

There are potential ethical concerns with this project. Mounting a video surveillance system in a public space and having object detection and tracking systems running can be an invasion of privacy. It might be prudent to take measures to ensure privacy of the people in the frame such as blurring of faces.

## 1.2 Previous Work

Counting the number of people in a location can be very useful; "counting the number of viewers in a shopping mall may provide valuable information for optimizing trading hours, as well as evaluating the attractiveness of some shopping areas or shopping items" [10]. Knowing how many people are doing something can help determine what a business focuses on and if its customers approve of what the business is offering.

People counting can be thought of in two categories, contact and non-contact detectors [11]. Contact detectors can include turnstiles and mat-type foot switches which require contact with the person being counted and can disrupt the flow of people. Non-contact methods meanwhile are cameras or IR sensors, and don't require contact with the person being counted. Contact methods can be more reliable and accurate and something like a turnstile also offers security as it can be a physical barrier. A non-contact method can however be more subtle, doesn't disrupt the flow of people and can be cheaper. Non-contact methods mainly consist of computer vision systems using object detection.

Cai et al [12] suggest that as detecting an object is computationally intense a tracking system as well as a detection system should be used. One proposed idea for person detection is background modelling where the background of the video is learnt and as such anything in the foreground could be a person. As it is an older paper people/object detection has moved on considerably. For tracking they suggest Kalman filters and Meanshift, both built into OpenCV.

A more recent attempt at a people counter uses the Caffe framework containing the MobileNet Single Shot Detector as the detection system and correlation filters as a centroid tracking system. Together these form the basis of the people counter. Again, the use of both detection and tracking is justified as 'The benefit of this hybrid approach is that we can apply highly accurate object detection methods without as much of the computational burden'. [13]

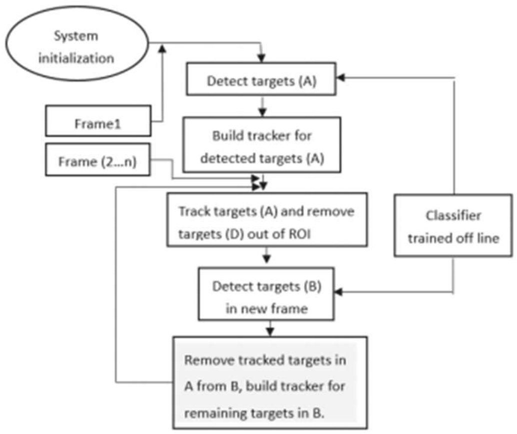

Figure 1 - Proposed processing flow [12]

## 2. Available Technologies

Object detection is nothing new and there are plenty of freely available, open source programs and algorithms to use. For my project I will use a pre-made and pre-tested person-detection model, rather than train my own. This will make the project much more achievable. Python seems to be the programming language of choice and will be what I will use for my project. All algorithms discussed below have integration with python. This review is to show what technologies are out there.

### 2.1 OpenCV

OpenCV is the industry standard library for computer vision. Used by everyone including Google and Microsoft. It provides a group of algorithms, libraries and more; all useful for object detection and object tracking. As the name implies it is an open source library developed by Intel around the turn of the century. It supports the deep learning frameworks TensorFlow and Caffe. OpenCV is very useful for its tracking capabilities using Meanshift.

### 2.2 R-CNN

Two successive papers have built on the original R-CNN (Region-based Convolutional Network) to create Fast R-CNN and Faster R-CNN; each as the names suggest, getting progressively quicker.

R-CNN is based on using Selective Search to find bounding boxes of Regions of Interest (RoI) which may contain objects. It then uses CNN, a deep object detection network, to classify objects in the regions. A regression model is used to correct the bounding box size and shape and non-maximum suppression helps stop repeated detections of the same object.

Fast R-CNN speeds up R-CNN by aggregating all region proposals into one CNN forward pass.

Faster R-CNN builds on Fast R-CNN by solving its bottleneck; having the region proposal algorithm (Selective Search) independent to the CNN model. Faster R-CNN uses the CNN that has already found features to then identify regions of interest and bounding boxes.

[1], [2], [3], [4]

### 2.3 Modern detection systems

Each paper in the so-called trilogy of R-CNN tries to develop a faster network for object detection but ultimately fail in creating a real-time object detector. [5] Forson says that 'Training the data is unwieldy and too long, training happens in multiple phases and the network is too slow at inference time'. More recent advances in object detection have resulted in new modern architectures for real time object detection, that are both faster to train and detect

#### 2.1.1 You Only Look Once (YOLO)

YOLO claims to be a fast and accurate alternative to other object detectors. It uses a single neural network to predict bounding boxes (location) and classes (object type) in a full image in one evaluation.

[6]

#### 2.1.2 Single Shot Detector (SSD)

Developed by google employees as a response to slow, outdated algorithms. Similarly, to YOLO, the single shot detector only needs a single look at the image to detect objects. SSD utilises MultiBox developed by C. Szegedy; a bounding box regression technique. SSD differs from YOLO in that it uses multiple sizes of activation maps to predict classes and bounding boxes on objects of differing sizes better.

[7]

### 2.4 Machine learning frameworks

With the wide variety of object detection methods, it is sometimes easier to utilise a machine learning platform to create your program. The frameworks offer pre-trained models for lots of different algorithms and ways to train you own. Their true capabilities will be much beyond the scope of this project.

#### 2.4.1 Tensorflow

Tensorflow is a framework by the Google Brain team to bring together various forms of machine learning on a single platform. It has a large group of premade models.

[8]

#### 2.4.2 Caffe

Caffe is the Convolutional Architecture for Fast Feature Embedding. Built to be used by scientists for machine learning projects, it offers a range of deep learning algorithms and models.

[9]

## 4. Project Plan

### 4.1 Project breakdown

There will be 3 major parts of my project

1. Detection system
2. Tracking system
3. Counting system

The detection and tracking system will work together to form the basis of the program. Detection will happen once every few frames and the tracking system will track the object in the other frames. Detection will be used to check for new objects/people and to confirm the tracking system. Each object/person will be given an ID to allow location data to be stored between frames to determine direction. Direction and location then allow me to determine whether the person has left the building or just entered and so whether to add one to the count or deduct one.

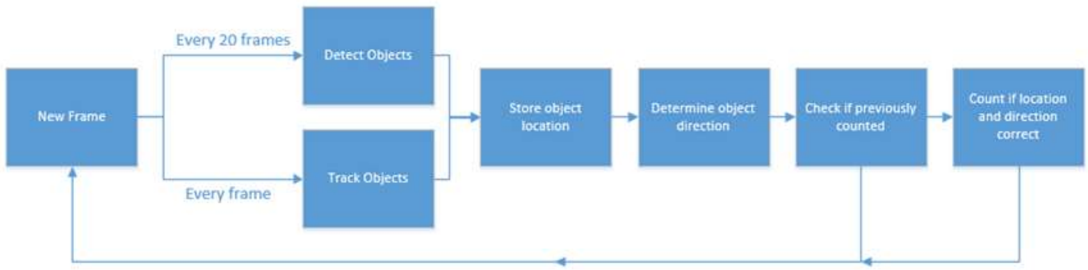


*Figure 2 - Pseudo-code diagram*

The project will be programmed in Python, utilising OpenCV. The detection system will be decided at a later date, from the group of systems previously stated. I will also be using one of the machine learning frameworks (Tensorflow and Caffe) for their pre-trained models.

## 4.2 Milestones and Gantt chart

There will be a few major milestones in my project.

1. Detector able to detect people
2. Tracking system able to track accurately
3. Assigning and storing IDs and location
4. Recalling previous locations linked with IDs
5. Counting system able to count people who cross a predetermined line
6. Finishing the project with a fully working building occupancy sensor system

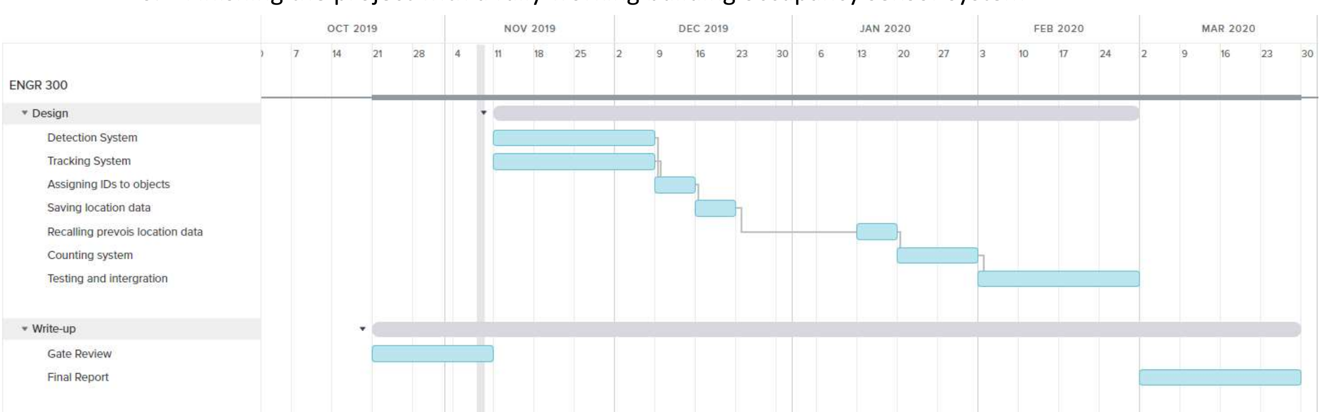


*Figure 3 - Gantt Chart*

## 4.3 Use of departmental resources

The only real resource I require for my project is a camera, which has already been provided. My own personal laptop and university computers will be used for the programming and writing of reports

# Appendix A - Risk Register

| Prob. (P) | Risk Score (P x I) | | |
|---|---|---|---|
| 3 | 3 | 6 | 9 |
| 2 | 2 | 4 | 6 |
| 1 | 1 | 2 | 3 |
| | 1 | 2 | 3 |
| | Impact (I) | | |

| | |
|---|---|
| | high level |
| | medium level |
| | low level |

*Figure 5 - Risk Register key*

| Category | ID | Risk | Probability | Impact | Severity | Mitigation | Contingency | Residual Risk |
|---|---|---|---|---|---|---|---|---|
| Personal | 1.1 | Illness stopping work | 2 | 2 | 4 | Depending on severity of illness, work can still be done | Have spare time left over at the end just in case | 1 |
| Personal | 1.2 | Personal Issues stopping work | 1 | 3 | 3 | Maintain healthy work/life balance | Have spare time left over at the end just in case | 1 |
| Planning | 2.1 | Insuffucuent time to finish project | 1 | 3 | 3 | Ensure proper time management and planning | Have spare time left over at the end just in case | 2 |
| Planning | 2.2 | Insuffucuent time to write reports | 1 | 3 | 3 | Ensure proper time management and planning | Have spare time left over at the end just in case | 2 |
| Equipment | 3.1 | Data loss | 1 | 3 | 3 | Secure backups of all work should be collected | Have spare time left over at the end just in case | 1 |

*Figure 4 - Risk Register*

# Appendix B – Risk Assessment

ENGINEERING DEPARTMENT

**SAFETY**

3rd/4th YEAR PROJECTS

| | |
|---|---|
| SUPERVISOR DENES CSALA | DATE 07/11/2019 |
| STUDENT/S THOMAS LUMBORG | |
| PROJECT TITLE DESIGNING A VISUAL SENSOR SYSTEM FOR ESTIMATING BUILDING OCCUPANCY | |
| RISK CATEGORY FOR SUPERVISION A B C D | |

| | YES | NO |
|---|---|---|
| 1. Will the project require <u>only</u> computer, drawing or library facilities?...................... | ✓ | |
| If YES then there is no need to proceed with the following questions. | | |
| 2. Has the student carried out similar work in previous lab classes?......................... | | |
| 3. Are the manufacturing processes similar to those usually carried out in the dept?.... | | |
| 4. Will the student be using standard lab equipment?............................................. | | |
| If YES the Department Safety Handbook could be used. | | |
| Please continue:- | | |
| 5. Will the student require any special training or instructions?................................ | | |
| 6. Will any materials require a COSHH Assessment?................................................. | | |
| 7. Will the student be working away from the department?....................................... | | |
| 8. Will the student be lifting heavy and/or awkward loads?...................................... | | |
| 9. Will the student be working at heights?........................................................... | | |
| 10. Will the project require guards to protect against entanglement, heat, noise etc?.... | | |
| 11. Will the project require a special area, (eg is it mobile and hazardous)?................. | | |
| 12. Will the project incorporate hydraulic or pneumatic equipment?........................... | | |
| 13. Are the voltages/currents higher than those normally used in the labs?................. | | |
| 14. Are there any other hazards associated with the project (microwaves,etc)? .......... | | |

<u>If YES to ANY of the above (5 to 14) then a specific risk assessment MUST be carried out. Please state the date you will complete each risk assessment by in the box below. All assessments MUST be complete BEFORE any practical work is done and before the end of the week 10.</u>

| Risk Assessments to be completed | Date to be completed by |
|---|---|
| | |

Signed: Supervisor Student/s

## Appendix C – Table of figures



## Appendix D - Works Cited

[1] D. Parthasarathy, "A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN," 22 April 2017. [Online]. Available: https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4. [Accessed 6 November 2019].

[2] R. Girshick, J. Donahue, T. Darrell and M. Jitendra, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. (pp. 580-587), 2014.

[3] R. Girshick, "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision,* pp. (pp. 1440-1448), 2015.

[4] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks.," *Advances in neural information processing systems,* pp. (pp. 91-99), 2015.

[5] E. Forson, "Understanding SSD MultiBox — Real-Time Object Detection In Deep Learning," 18 November 2017. [Online]. Available: https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab. [Accessed 2 11 2019].

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection.," *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. (pp. 779-788), 2016.

[7] W. Lui, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. Berg, "Ssd: Single shot multibox detector," *European conference on computer vision,* pp. (pp. 21-37), 2016.

[8] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean and M. Devin, "Tensorflow: A system for large-scale machine learning," *12th Symposium on Operating Systems Design and Implementation,* pp. (pp. 265-283), 2016.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadrrama and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *Proceedings of the 22nd ACM international conference on Multimedia,* pp. (pp. 675-678), 2014.

[10] D.-Y. Chen and K.-Y. Lin, "A novel viewer counter for digital billboards," *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing,* pp. (pp. 653-656), 2009.

[11] M. Kamel, M. Fkry, A. Mashat, N. Biqami, H. Barhamtoshy and I. Beedewy, "Monitoring, Surveillance and control of the crowds in the holy sites using SCADA System," *The Seventeenth National Computers Conference,* 2004.

[12] Z. Cai, Z. Yu, H. Liu and K. Zhang, "Counting people in crowded scenes by video analyzing," *9th IEEE Conference on Industrial Electronics and Applications,* pp. (pp. 1841-1845), 2014.

[13] A. Rosebrock, "OpenCV People Counter," 13 August 2018. [Online]. Available: https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/. [Accessed 7 11 2019].

[14] H. Cetinkaya and M. Akcay, "People counting at campuses," *Procedia-Social and Behavioral Sciences 182,* pp. (pp. 732-736), 2015.

[15] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," *12th IEEE International Conference on Automatic Face & Gesture Recognition,* pp. (pp. 650-657), 2017.

[16] P. Viola and M. Jones, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96, 3(14), 2,* 2003.

[17] V. Prabakaran, A. Arthanariee and M. Sivakumar, "Crowd Safety: A Real Time System For Counting People.," *International Journal Of Innovative Technology & Creative Engineering,* pp. (pp. 6-11), 2011.

[18] H. Ziang and E. Learned-Miller, "Face detection with the faster R-CNN," *IEEE International Conference on Automatic Face & Gesture Recognition,* pp. (pp. 650-657), 2017.