Applications of Computer Vision in Order to Accurately Determine Footfall in Industrial and Commercial Environments

Ibrahim Anjum- 001011658

November 7, 2023

A Project Proposal for the Degree of Computer Science BSc

1 Introduction

In large scale industrial and commercial environments, the most recent wide-scale use of monitoring attendance and footfall has been lightgates- which are notoriously inefficient in both time and monetary values. Another solution would be to use IoT in the form on bluetooth connectivity in order to detect if a person has a bluetooth enabled device, this then adds one to the value of population Pu et al. (2021) The proposed solution put forth in this paper is a system of low power machines that are mounted on entrances and exits. The system splits the screen into two sections: the upper and lower areas- for the sake of clarity, the upper section will be called A and the lower section will be called B. If a person moves from section A to Section B, the internal counter of the system will count one down, this will subtract one from the total amount of people that have been in the room. Likewise, if a person moves from section B to A, the internal counter will move one in a positive direction, this adds one to the total amount of people that are in the room.

The goals to be achieved for this project are the following:

- Implement an algorithm which allows for an accurate way of object tracking
- Have a high overall accuracy: Chaabane et al. (2020) and Zhong et al. (2019) show that an overall accuracy of 88-95 percent are thought of as an acceptable accuracy range related to image classification.

In order to achieve these goals, the programming language python will be used. This is as there are pre-programmed modules and libraries such as pandas, numpy, keras, tensorflow, imutils, and various others. These libraries allow for seamless integration of industry wide used functions, these libraries also have various pieces of documentation written to allow them to be used easily. As

well as documentation, there are a multitude of forums and support websites available if errors occur and the user is unable to debug the error.

2 Literature Review

The topic of research is the implementation of a machine learning model that uses information gathered by a developed algorithm. The developed algorithm uses euclidean mathematics in order to form coordinates in a three dimensional space based upon other elements in the space.

Wu & Yang (2015) states that large scale classification tasks have fast speeds, namely in the category image classification, additive kernels have also shown state of the art accuracy. Even though fast algorithms have sped up this process drastically, large scale tasks are still open problems. This problem is tackled by using a linear regression based model for an SVM (Support Vector Machine) to approximate gradient computations in the learning process. a PmSVM (Power Mean Support Vector Machine) algorithm for all additive kernels using non symmetric explanatory variable functions was proposed. The non-symmetric kernel application does not require closed form Fourier functions and does not require extra training for the approximation. Wu & Yang (2015) was relevant to the current topic as a large amount of data was processed with a less efficient method, and an algorithm was developed in order to make this process more efficient.

Euclid et al. (2017) is a series of 13 books most commonly referred to as Euclid's Elements. It is a collection of definitions, postulates. propositions, and mathematical proofs of said propositions. The books cover plane geometry and Euclidean Geometry, elementary number theory, and incommensurable lines. This was instrumental in developing the algorithms required to make the project work.

According to **Rafique & Velasco (2018)**, complex systems can be represented and understood more easily by integrating machine learning- by showing use cases varying from cloud operations, metro transport system, and mobile connectivity all the way to streaming applications. Various machine algorithm and their use cases are looked at in detail also, the algorithms explored include q-learning, self organising maps, Support vector machines, unsupervised learning, and k nearest neighbours. This is relevant to the topic at hand as machine learning libraries are heavily used in order to form data- by using the methodologies and design patterns mentioned by Rafique & Velasco (2018), an insight was gained into how to best use these practices in the current use case.

Zhong et al. (2019) shows how a new approach to hyperspectral image classification was developed and how the altered algorithm works. Zhong et al. (2019) states that the algorithm works upon the principle of weighted classes,

each with a probability. These probabilities are then used to calculate the significance of each class- there are three main uses for these probabilities: to derive a specific training sample size for each class to be used in classification, another is to use the probabilities as weights to improve overall accuracy, and the last is to derive hyperspectral image classification. This paper has been insightful as it has shown a brief overview of developing an algorithm and how to implement it- This was used in order to develop and implement the Euclidean algorithm of forming three dimensional coordinates.

Zheng et al. (2020) state that the majority of deep learning models for facial recognition has progressed at an incredible rate- but that unconstrained video based facial recognition is still challenging due to various contributing factors. This problem is combated by using multiple SSD's (Single Shot Detectors) to localise regions of each frame where faces are. This use case was looked at when developing an approach to the problem of attendance and helped to develop early strategies and fundamentals of the Euclidean algorithm.

Chaabane et al. (2020) states how multispectral SITS (Satellite Image Time Series) enables the production of a certain type of map. Efficient models that are used to form these maps already exist, such as statistical learning methods- but the model can be improved by iomplementing deep learning. Chaabane et al. (2020) proposes that a direct comparison of an SOTAG (Spatial Object Temporal Adjacency graph) SVM (support vector machine) based classification, and an RNN (recurrent neural network) LSTM (long short term memory) model trained by historical SITS. By comparing both models and reviewing overall accuracy, an acceptable rate for a model that doesn't overfit and is concise is around the region of 88-95 percent. This will be used as a factor to determine the overall acceptance level for the trained model.

Pu et al. (2021) state that Bluetooth and internet enabled devices can be used in order to determine the origin, route, and flow routes of passengers on metro transit systems. This information could then be used in order to optimise routes for passengers to ease congestion, and identify areas of interest on transit systems. This study contributed most closely to the overall concept of this paper, the machine learning model, and the implementation of it due to the use case being so similar.

Frohlich et al. (2021) presented the problem of attempting to derive pose estimation in a three dimensional space based upon a physical two dimensional space, without the need for any special calibration equipment. This was achieved by using the obtained two dimensional in nonlinear equations and functions in order to extrapolate data. This inspired the approach of forming coordinates in three dimensional space.

3 Problem Domain

The problem domain for this proposal would be in environments where population monitoring would be used; an example-based upon experience in a placement year- would be in a museum: if a museum were to introduce a new exhibit or interactive, the current implementation of population monitoring is the use of lightgates- these are highly inaccurate and costly. Lightgates are used as they are the only widely available solution for commercial business/environments, The reason lightgates are highly undesirable are the following:

- Highly expensive to implement- commercial businesses have finite resources
- False positives are recorded- the nature of a lightgate using IR radiation means that it is sensitive to any moving objects, therefore it picks up any moving object and counts them as a person
- Requires specialized support when errors occur- As lightgates are a specialized piece of equipment, specialist contractors/trained staff are required in order to fix them. As lightgates aren't very commonplace, the time taken to diagnose and fix an issue would take a great deal of time.

4 Methodology

4.1 Translating coordinates

For full transparency, Rosebrock (2018) was used in order to gain a boilerplate to develop this project off. Images were also used from Rosebrock (2018)

In order to determine what the best approach to tackle this problem would be, an overall analysis of all methods that could be used were gathered and gone through recursively until the most efficient solution came forth. (Rafique & Velasco 2018) performs a brief overview on machine learning for network automation- this includes an overview on the most commonly used algorithms under the section labelled "algorithms".

The second step would be finding out how to gather the three dimensional co-ordinates on a 2 dimensional plane. As a regular webcam/camera can only translate movements in three axes- three axes as a human is able to move in ijk vectors- into movements in two axes- a screen will only show i and j movements- a way of forming meaningful three dimensional coordinates is needed. Based upon Frohlich et al. (2021) and prior knowledge, Euclidean Geometry will be used in order to do this.

4.2 Euclidean geometry

Euclidean Geometry works upon the principles of Euclid's Axioms. In this case, all theorems or "true statements" are defined by a small number of base

axioms. The theorems stated in Euclid et al. (2017) states to let the following be postulates (axioms) for any plane geometry be true:

- To draw a straight line from any point to any point.
- To produce (extend) a finite straight line continuously in a straight line.
- To describe a circle with any centre and distance (radius).
- That all right angles are equal to one another.
- The parallel postulate: That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than two right angles.

These are the rules to which the application must be coherent with in order to calculate each objects Euclidean co-ordinates

4.3 Languages

Python will be used as there are pre-programmed open source libraries that assist with a multitude of functions. For this specific use case- OpenCV, imutils, time, dlib, argparse, and numpy will be used. OpenCV will be used as it provides a multitude of libraries that can be used for computer vision applications; imutils is a library that contains various utilities to make image manipulation easier- one that will be used is the videostream component, this allows the user to input a video direct from a source without coding their own library, time is used in order to keep track of how long the program has been running for, dlib is originally a C++ library that allows a seamless implementation of various machine learning algorithms such as SVM's, K-Means clustering, Bayesian networks, and various others- it also allows utilities such as networking and threading, argparse is an argument parsing library which allows custom arguments to be set so the application can be launched from a command line, lastly, numpy is used as it allows easy manipulation of numeric data.

4.4 The algorithm:

The algorithm to be used consists of two main phases, the detection phase and the tracking phase. The algorithm works by running each phase sequentially, as the detection phase is more computationally expensive- it is used one in every 5 frames in order to reduce the overall cost of the algorithm. The detection phase is more expensive as it computes the euclidean coordinates of every object in frame- these coordinates are then used in the next cycle of the applications and so on.

Step 1 involves generating bounding box coordinates and then generate a centroid for each bounding box. The bounding boxes are provided by an object detector class (Such as a Histogram of Oriented Gradients and a Linear SVM

Wu & Yang (2015), a Faster R-CNN (Lee et al. 2020), or a Single Shot Detector). To implement the counter, OpenCV and dlib were used. OpenCV was used due to it's built in standard image processing/computer vision processes.
Dlib was used due to its abundance of correlation filters, OpenCV could be used for this, but dlib's objet tracking implementation throws less errors and was easier to implement seamlessly into this project. In *figure 1* we have two bounding boxes and two centroids.

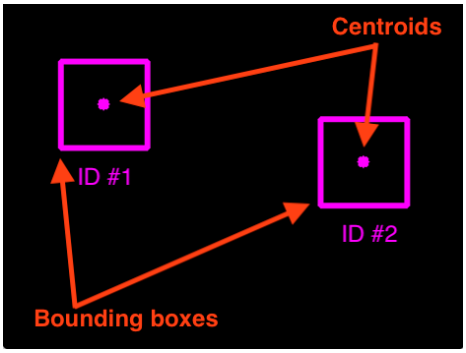

Figure 1: The first step is to accept bounding box coordinates and use them to compute centroids

Step 2 involves computing the distance between any *new* centroids (shown in yellow) and any *existing* centroids (shown in purple), see *figure 2*. A drawback is that the centroid tracking algorithm makes the assumption that pairs of centroids with minimum Euclidean distance *must be the same object ID*. Once the Euclidean distances have been calculated, the algorithm attempts to calculate the object ID's in **Step 3**.

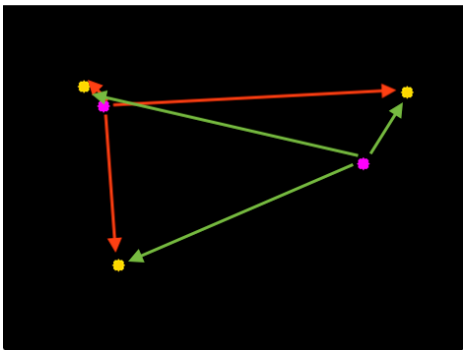

Figure 2: Three objects are present, the program needs to compute the Euclidean distance between each pair of original centroids (*red*) and new centroid

Step 3 involves the calculation of the object ID's based upon Euclidean coordinates that were calculated in the previous step. In *figure 3* the algorithm

has chosen to associate centroids that minimize their respective Euclidean distances. The point on the bottom left isn't associated with anything as **Step 4** will identify it as a new object.

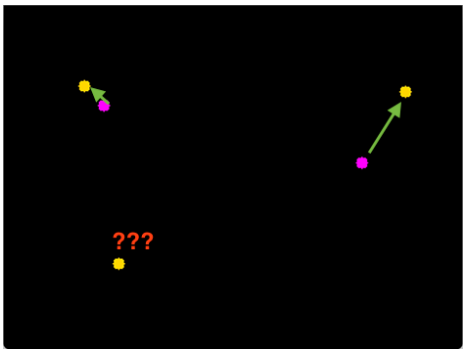

Figure 3: The object centroid tracking algorithm has recognised and assigned associated objects with minimized object distances. The object on the bottom left however isn't recognised so is treated as a new object

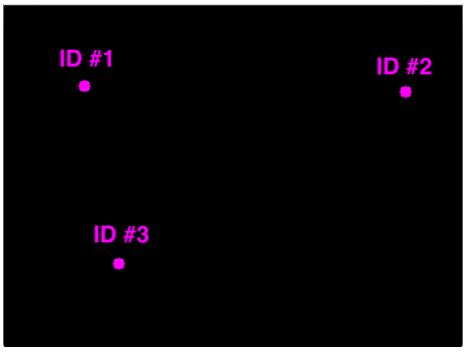

Figure 4: In the object tracking example, there is a new object that wasn't matched with an existing object, so it is given a new ID

Step 4 consists of two main phases: registering a new object and assigning it an ID, see *figure 4*, and storing the centroid of the bounding box coordinates for the new object. In the event that an object is lost or leaves the field of view- an object is "Lost" when it is no longer in the window for 40 consecutive frames- the algorithm de-registers the object in order to optimise resources

In order to create a "trackable object", one that can be tracked and counted on a video stream, a way to store metadata about the object must be created. The metadata includes:

- The object's ID
- The objects previous centroids

- whether or not the object has already been counted

5 Evaluation

The goal of this application is to determine whether it will be a better solution than the current system that is in place, lightgates- the metrics of performance will be judged by using the known population of people/peoples in a certain area, running the application over a certain time period, and comparing the results gathered compared to the actual result. The data gathered will be first hand recorded data. The program will be deemed "successful" if an overall accuracy of 88-95 percent is reached in detecting the number of people entering and leaving a room.

In order to test the data, a stock video of various people walking down a public high street will be used- see *fig 5*- this was used as this video presents the most ideal conditions in order for the program to perform well.

Figure 5: A frame from a stock video of multiple people walking along a high street

5.1 Results

To define the success parameters for this specific video, the number of people travelling from the bottom of the frame to the top of the frame were counted, and the number of people travelling from the top to the bottom were also counted- A total of 31 people travelled upwards and 28 travelled down. The program detected 29 people travelling upwards and 26 travelling downwards. The program had a 93.5 percent accuracy in detecting upwards motion and 92.8 percent accuracy in downwards motion.

5.2 Improvements and Constraints

Whilst the program has achieved it's goals and can accurately detect a person moving upwards and downwards in a frame- there are a few instances where this program could be improved.

The first instance is wherein the implementation of this code is not as efficient as possible. In order to track multiple objects, multiple instances of the

correlation tracker object need to be created. The program then needs to compute the location of the object in the subsequent frames, and then loop over created objects and take the updated position. In order for this to happen, the program would need to compute this on the main execution thread- which would drastically reduce frame rate.

Given the opportunity to improve upon this, dlib's multi object tracking could be implemented in order to use multiprocessing and increase fps.

6 Conclusion

The goal of this program was to allow an easier, less expensive, and more accessible way to monitor population to be developed. Not only was this goal achieved, but the program is able to run regardless of hardware configuration due to the nature of the programming language used.

The goal was achieved successfully, the program was able to accurately track multiple people with an overall efficiency of 93.15 percent, which was within the original project guidelines set earlier on in this paper:

- Implement an algorithm which allows for an accurate way of object tracking
- Have a high overall accuracy: Chaabane et al. (2020) and Zhong et al. (2019) show that an overall accuracy of 88-95 percent are thought of as an acceptable accuracy range related to image classification.

All of these goals were achieved to great effect, an algorithm was developed using Euclidean geometry that allowed an accurate way of tracking objects; An accuracy of 93.15 percent was achieved, which falls within the 88-95 percent intended accuracy.

For future work, if this program were to be remade, a faster language such as C would be used along with dlib's multi object tracking modules. Caffe would still be used as it allows faster deployment of configuration files

References

Chaabane, F., Rejichi, S. & Tupin, F. (2020), Comparison between multitemporal graph based classical learning and lstm model classifications for sits analysis, Institute of Electrical and Electronics Engineers Inc., pp. 144–147.

Euclid, Heath, T. L. & Densmore, D. (2017), *Euclid's elements: All Thirteen books complete in one volume: The thomas L. heath translation*, Green Lion Press.

Frohlich, R., Tamas, L. & Kato, Z. (2021), 'Absolute pose estimation of central cameras using planar regions', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 377–391.

Lee, H., Eum, S. & Kwon, H. (2020), 'Me r-cnn: Multi-expert r-cnn for object detection', *IEEE Transactions on Image Processing* **29**, 1030–1044.

Pu, Z., Zhu, M., Li, W., Cui, Z., Guo, X. & Wang, Y. (2021), 'Monitoring public transit ridership flow by passively sensing wi-fi and bluetooth mobile devices', *IEEE Internet of Things Journal* **8**, 474–486.

Rafique, D. & Velasco, L. (2018), 'Machine learning for network automation: Overview, architecture, and applications [invited tutorial]', *Journal of Optical Communications and Networking* **10**, D126–D143.

Rosebrock, A. (2018), 'Opencv people cunter', `https://pyimagesearch.com/2018/08/13/opencv-people-counter/`.

Wu, J. & Yang, H. (2015), 'Linear regression-based efficient svm learning for large-scale classification', *IEEE Transactions on Neural Networks and Learning Systems* **26**, 2357–2369.

Zheng, J., Ranjan, R., Chen, C.-H., Chen, J.-C., Castillo, C. D. & Chellappa, R. (2020), 'An automatic system for unconstrained video-based face recognition', *IEEE Transactions on Biometrics, Behavior, and Identity Science* **2**, 194–209.

Zhong, S., Chang, C. I., Li, J., Shang, X., Chen, S., Song, M. & Zhang, Y. (2019), 'Class feature weighted hyperspectral image classification', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **12**, 4728–4745.