Intelligent Queue Management System at Airports using Image Processing and Machine Learning

Diksha Wuthoo[1], Ketan Bedarkar[2]

[1]B.E. Information Technology, MIT College of Engineering, Pune, India
[2]B.E. Computer Engineering, Maharashtra Institute of Technology, Pune, India

Abstract - *In a world where population is growing exponentially, crowds have become a quotidian menace to such extent that even the sight of a queue can test the virtues of the most patient person. Such aggravation is somewhat alleviated if the time to be spent waiting in a queue is known beforehand. Our proposed system provides such an estimate for the various waiting queues at the airport by using image processing and machine learning. The image processing module uses human detection, facial recognition and tracking algorithms to detect the number of passengers in a queue, the employee at that particular counter and the time required by an individual to traverse the queue. A multiple linear regression model calculates the said estimate which is displayed as output.*

Key Words: Image Processing, Machine Learning, Haar cascade, multiple linear regression, OpenCV.

1. INTRODUCTION

Due to the ever-expanding economy and an increase in the availability of air travel, crowds have become a massive problem in areas like airports and controlling these crowds has become a vital task for their smooth functioning. In an attempt to restore some order to such chaos, the simple solution of queues is implemented. However, due to the uncertainty about the time they will be required to wait in the queue, many times people become impatient and this elevation in their anxiety makes it even more difficult to control the traffic in queues like check-in, security, immigration, etc.

The proposed system provides an elegant and easy-to-implement solution that uses image processing and machine learning to provide the passengers with an estimate of the time they will require to traverse a particular queue that doesn't interfere with the airport security in any way, shape or form. This estimate depends on factors like the length of the queue i.e. how many people are standing in that queue, what time of day it is, and the person at the counter. The passengers will be able to view this estimated time and then decide which queue to stand in to reach the counter as soon as possible which facilitates even distribution of the traffic and avoids congestion in a particular area.

The system mainly comprises of two modules: the image processing module and the machine learning module. The image processing module analyses the video feed obtained from the CCTV cameras at the airport. It detects the people standing in the queue as distinct objects and tracks the time taken by each passenger to move from the back of the queue all the way to the front. A facial recognition algorithm identifies the airport employee handling that particular counter and passes this information along with the timestamp to the machine learning module as training data. A multiple linear regression model is trained on this data and then, it is used to predict the waiting time in the queues.

We have implemented the image processing module of the proposed system using OpenCV library of Python. Pre-processing techniques like grayscale conversion and background subtraction are implemented using the functions of OpenCV. For human face detection, Haar cascade algorithm is used which is a machine learning based approach that provides optimal accuracy and performance. The LBPH method is used for facial recognition as the environment of the airport is a controlled environment and the subject is unlikely to move at a very fast speed. The prediction algorithm is implemented using multiple linear regression as it is a scalable and dynamic approach that provides a numerical output.

2. RELATED WORK

The paper is based on two major fields of study:

1. Image Processing for human face detection.
2. Machine Learning algorithm for the estimation of time taken by a passenger to reach the front of the queue.

Feng Su, Gu Fang, and Ju Jia Zou proposed a different method for human detection without pre-training the model. Their method involves four stages - scaled gradient mapping, detection of body parts, estimating body region by gap detection between the body parts, and lastly figure extraction and segmentation into head, upper body and lower body. Though their model had good accuracy, it is limited to detecting a single person only [1]. In [2], Kruti Goyal, Kartikey Agarwal, and Rishi Kumar have talked about the various techniques for face detection and tracking like AdaBoost, Haar cascades, etc. They have compared those techniques based on the factors such as time complexity and performance, and concluded that Haar

Cascade is the most efficient way of face detection. Also, they have stated the prps and cons of the libraries used for the face detection like OpenCV and MATLAB. Li Cuimei, Qi Zhiliang, Jia Nan, and Wu Jianhua proposed a human face detection algorithm which uses Haar cascade classifier with three additional weak classifiers. The main Haar cascade classifier filters out most of the non-human images and then the filtered result is passed to the three weak classifiers which detects the face skin, eyes and mouth [3]. Wenxiang Yu, Jiapeng Xiu, Chen Liu, and Zhengqiu Yang in [4] proposed a face detection algorithm, for obtaining a strong classifier from a weak one, which is a combination of AdaBoost algorithm and Haar classifier using OpenCV. They have also compared the success rate and detection time for the combined algorithm with the AdaBoost algorithm and results show that the recognition rate is greatly improved with the combined algorithm. Nehal N Ghosalkar and Sudhir N Dhage in [5] proposed a method for real estate value prediction using linear regression. They have considered multiple factors for the training of the model like physical conditions, locations, financial plans of the customer, etc. They have used the linear regression algorithm as it gives the exact numerical target value rather than just classifying the output.

3. PROPOSED STUDY

Figure 1 shows the proposed system architecture. The footage captured by the CCTV cameras at the airport shall act as input for the system. The resultant output will be the estimated time required by a passenger to reach the front of a particular queue, based on which he/she will be able to determine the shortest possible route through the airport. The overall functioning of the system can be described in 4 phases:

1. Firstly, the video captured using CCTV cameras at the airport is given to the first primary module i.e., the image processing module which processes it frame by frame. Each frame is then pre-processed by performing operations like grayscale conversion, background subtraction etc. A trained Haar-cascade is then used to detect the outlines of people and determine how many people are present in that particular queue by analyzing the frames.

2. Then, the passenger who is detected at the rear end of the queue, is tracked using OpenCV's built-in tracker to find when this passenger reached the front of the queue. Thus, the tracking module calculates the time required for traversal of the queue by a particular passenger, and how many people were in the queue before him/her when this passenger joined the queue.

3. Subsequently, this data is given to a machine learning module that uses this data as training to develop a regression analysis model that gives an estimate for how much time shall be required to traverse a particular queue by a particular passenger based on how many people are detected to be in front of him/her.

4. Finally, the predictions for each queue at the airport are given to the reporting module that displays the estimated time on a screen near the queue. Thus, the time is shown for each queue in an easy-to-understand user interface.

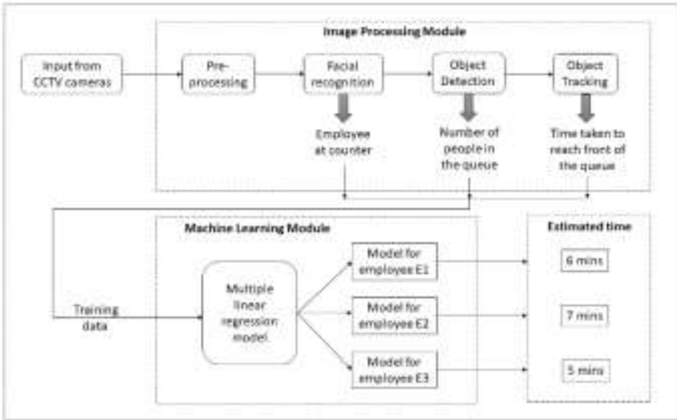

Fig -1: Architecture Diagram for proposed system

The algorithms used are explained in detail below:-

3.1. Image Processing

3.1.1 Background Subtraction:

This algorithm is based on the concept of running average. The running average is an entity that is used to differentiate between foreground and background. The video is analyzed frame-by-frame and running average over the current and previous frames is computed using the formula:

dst(x,y) = (1- alpha).dst(x,y) + alpha.src(x,y), where src is the source image, dst is the destination image and alpha is the weight of the image.

This gives us the background model and any new object introduced during the sequencing of the video becomes part of the foreground. Then, the newly introduced object is distinguished from the background in the current frame. Thus, the calculation of the absolute difference between the background model (which is a function of time) and the current frame (which is newly introduced object) is done.

3.1.2 Object Detection:

In the proposed system, the object to be detected is the face of a human. The algorithm used is the Haar Cascade algorithm, a machine learning based approach, where a cascade function is trained from a lot of positive and negative images (positive images are those in which, the object that is to be detected, is present and negative images are the ones where it is not). The size of the set of negative images is kept greater than twice that of the positive images' set for optimal accuracy. The pre-trained OpenCV cascade classifier is used in the system for detection of faces. As shown in Figure 2, the face and eyes are detected in an image.

Fig -2: Passengers' face detection in a queue at an airport

3.1.3 Facial Recognition:

The facial recognition algorithm is implemented using the LBPH (Local Binary Patterns Histograms) method. This algorithm uses four parameters:

1. The radius is used to build the circular local binary pattern and represents the radius around the central pixel.
2. The number of points required to build the pattern.
3. The number of cells in a row.
4. The number of cells in a column.

A greater number of cells increases the finesse of the grid and consequently, the dimensionality of the resulting feature vector. As the environment of the airport is controlled, this is an easy and optimal approach that is made available by the OpenCV library. The facial recognition is used in the system to identify which employee is at the counter, which is a required input for estimation of the queue traversal time.

3.1.4 Object Tracking:

To find the traversal time, a particular individual is tracked from the back of the queue, all the way to the front. The Centroid Tracking algorithm is used as it exhibited adequate speed and accuracy for the required purpose. The algorithm relies on Euclidean distance between existing object centroids (i.e., objects the centroid tracker has already seen before) and new object centroids between subsequent frames in a video. The individual is tracked so as to find the time required by him/her to traverse the queue. When any individual is detected as a new object on video, the detected object is assigned an ID. It is against this ID that the traversal time shall be recorded into the system.

The training data for the machine learning model is generated using this image processing module itself. The data is given in the form of an n*3 array, where n is the number of people the data is collected for. The two values given for each passenger detected to be in the queue are:

1. The number of people ahead of this passenger in the queue.
2. The time required for the passenger to reach to the front of the queue.
3. The time of the day at which the passenger first arrived into the queue.

3.2. Machine Learning

3.2.1. Wait time estimation:

To predict the time, that an individual will have to wait in order to reach the front of the queue, the Multiple Linear Regression algorithm is employed. The training data received from the image processing module is used to engineer the regression model. The core design principle of the model is the simple dependency of the traversal time on factors like the number of people already present in the queue and time of day at which the passenger arrived. The model is trained to achieve minimum value of the cost function which is given below and then used to predict the time that will be required by a passenger if he/she joins a particular queue at that particular instance.

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

3.2.2. Considering effect of change in personnel:

The design understands that different employees at the airport will work at different rates or speed, i.e., the number of people processed per unit of time will vary from employee to employee. Hence, a different regression model is trained and maintained for each employee. While predicting the queue traversal time, using facial recognition, the employee at the counter is identified and the prediction model is loaded accordingly. Thus, a model with increased accuracy is obtained.

4. EXPERIMENTAL RESULTS

We tested the proposed system with the data of a normal day at an airport. After the image processing module identifies the number of passengers in a queue and the employee of an airline at the counter, the data is passed for the calculation of estimated time in all the queues. The estimated time differs from employee to employee depending upon their speed to cater to the people in the queue. Chart 1 shows the estimated time a passenger has to wait in a particular queue to reach the front of the queue at the check-in counter.

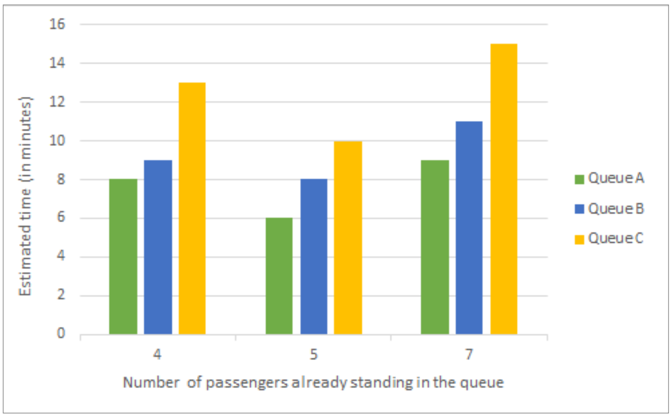

Chart -1: Graph showing estimated waiting time in three queues for different no of people standing in these queues.

5. CONCLUSION AND FUTURE WORK

The system takes input from the CCTV cameras at the airports and using image processing detects the passengers in waiting lines (like check-in counters, immigration counters) as well as the airport employees at each counter. This time taken by each passenger to reach the front for every queue is used to train the machine learning algorithm. Thus, when a passenger arrives near the queues at the airport, he/she would be able to see the estimated time to reach the counter for every queue and decide accordingly which queue to stand in. Our system uses the data captured by the CCTV cameras at counters of the airports for the implementation but without hampering the security of these airports.

In future, the system can be improved to identify overlapping faces of multiple people. It can also be further expanded to provide extra features for the airport security like unattended baggage tracking.

REFERENCES

[1] Feng Su, Gu Fang and Ju Jia Zou, “Human Detection using a Combination of Face, Head and Shoulder Detectors,” 2016 IEEE Region 10 Conference (TENCON) - Proceedings of the International Conference, pp.n 842-845.

[2] Kruti Goyal, Kartikey Agarwal, and Rishi Kumar, “Face Detection and Tracking Using OpenCV,” 2017 IEEE International Conference on Electronics, Communication and Aerospace Technology ICECA 2017.

[3] Li Cuimei, Qi Zhiliang, Jia Nan, and Wu Jianhua, “Human face detection algorithm via Haar cascade classifier combined with three additional classifiers,” 2017 IEEE 13th International Conference on Electronic Measurement & Instruments ICEMI 2017.

[4] Wenxiang Yu, Jiapeng Xiu, Chen Liu, and Zhengqiu Yang, “A depth cascade face detection algorithm based on Adaboost,” 2016 IEEE Proceedings of NIDC2016.

[5] Nehal N Ghosalkar and Sudhir N Dhage, “Real estate value prediction using linear regression” 2018 IEEE Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

[6] Li Hou, Wanggen Wan, Kang Han, Rizwan Muhammad, Mingyang Yang, “Human detection and tracking over camera networks: A review,” 2016 IEEE ICALIP conference, pp.n 574-580.

[7] Thombre D.V., Thombre D.V., and LekhaDas, “Human detection and tracking using image segmentation and kalman filter,” 2009 IEEE IAMA conference.

[8] Cha Zhang and Zhengyou Zhang, “A Survey of Recent Advances in Face Detection,” extracted from an early draft of the book “Boosting-Based Face Detection and Adaptation” published in 2010.