Predictive Queue Times

Corey McLaughlin
Department of Computer Science
Ohio Northern University
Ada, Ohio 45810
Email: c-mclaughlin.3@onu.edu

Jacob Hartman
Department of Computer Science
Ohio Northern University
Ada, Ohio 45810
Email: j-hartman.3@onu.edu

Introduction

Queues are a large part of how people organize events and services. As a result, queues are also places where people can end up spending a lot more time than they would like. Our goal is to utilize and apply the technology available to improve the experience.

Our project's aim is to build a monitoring application for human queues. Our goal is to detect and identify queues that lack well defined structure, regularity, or organization (e.g. queues that exhibit clusters of people and movement), or queues whose shape is not completely predictable (e.g. queues that are formed spontaneously at different locations and cannot be monitored using fixed motion sensors). Once the queue is recognized by the system, the data collected will be able to be used in two ways. Firstly, it will be able to almost instantaneously predict the average wait time for a person in the queue. Secondly, businesses that utilize the system will be able to make intelligent decisions about staffing and logistics to provide better service to their customers. All of this will be possible due to the utilization of an artificial neural network

An artificial neural network, or ANN, is a computing structure that allows software to detect patterns in information and draw inferences based on those patterns[1]. Specifically, the type of ANN used in this project will be a convolutional neural network. In this type of network, information is looped through multiple times, allowing changes over time to be detected[2]. The proposed software will detect and track queues in order to predict an estimated wait time. In this instance, a queue is defined as a group of three or more individuals awaiting service. Queues tend to follow a particular direction, with the line drawn through all individuals being nearly straight. However, there are many instances (such as airports, bank tellers, etc.) where a queue may be unconstrained and could theoretically form in a variety of shapes. The use of the ANN will allow the proposed software to adapt to these irregular shapes. The tracking of such unconstrained queues is the primary goal of the project.

Methodology

Information will be input to the system via a single camera providing live feed of an area in which a queue may form. Use of the TensorFlow application, an open source implementation of an ANN, will allow the software to detect people in the camera's field of view. For the training of the neural network, we decided to use the COCO dataset. The COCO dataset is one of the largest object datasets available and is used to teach neural networks how to recognize certain objects in context. The dataset uses what is called "non-iconic images" to provide a more robust contextual library for the network to draw from. This means that if the network sees something it should recognize from an odd angle, or even just a partial image, the chances of recognition by the network are significantly higher.

The COCO dataset will be used to train the network to recognize the shape of a human, allowing detection to be possible. The detected objects' positions will then be analyzed to determine whether a queue has formed. Once detected, this queue will be tracked using the loop aspect of a convolutional neural network. The time it takes for an individual to receive service will be identified and stored. These service times will then be used to keep a running average of service time. This service time will be multiplied by the number of individuals in the queue to predict the wait time of the queue. These average wait time will be sent to a front end application for the public. This application will be used by the people who use the services of our clients to see average waiting times from wherever they are. So a busy traveler or a family of six will be able to see how long they should expect to wait in airport security and plan accordingly. This makes our system valuable to our clients as well as directly benefiting theirs.

A custom built computer system will be utilized to run the proposed software. This system will run a linux-based operating system with a python development environment set up to allow native interaction with the TensorFlow software. A GPU will be utilized to train the ANN, allowing for faster and more effective training[3]. Once the neural network is trained, it will be able to run efficiently on additional systems, allowing for rapid reproduction.

Constraints

Any system designed to identify humans may potentially save facial recognition information. There is a concern in regards to the privacy of individuals being tracked. To avoid this issue, all such data will be stored on a volatile basis, being removed from the system once it is no longer needed. The system as a whole needs to be robust enough to handle a variety of deployment locations. This includes but is not limited to, various temperature ranges and weather proofing in the case of an outdoor implementation. Additionally, the proposed system will possess a comprehensive user interface which gives easy access to the required information without requiring extensive training.

Solution Analysis

Several options were analyzed in order to solve the problem of queue tracking. The first and lightest solution was the careful placement of motion sensors around queue space. This solution fails when queues are unconstrained, since there may not be an appropriate location for the sensors. It also fails to provide an actual wait time, since the sensors cannot accurately track how many people are currently in the queue. The second solution was the use of a human to track queues and determine wait times by hand. This solution would require an additional employee, making the cost higher and less consistent over time due to wages per hour, possible employee benefits(if applicable), overtime, etc. There is also a high risk of error, since many factors can influence a human's focus. With these in mind, the best option would be to determine a method of using a computer system to consistently track queues. Due to the multifaceted and varied nature of queues, an artificial neural network is the best solution. It can easily identify people and queues with a high level consistency, and the setup and overhead are minimal compared to brute force methods. The identification of patterns in this manner is exceedingly difficult to code correctly, and is commonly accepted to be best solved by an artificial neural network.

Components and Cost

A lot of the cost for this project comes from the initial setup and training as it requires a custom built system with a higher end GPU to train the neural network efficiently. Once the initial training is complete the implemented system that will be used by any clients will be variable in cost since we can customize the different components for each client. For example, an airport may need more cameras and a more powerful system to handle all the data that they need to take in. Alternatively, a small outdoor event may only need one camera and laptop computer system for their intake. The great thing about neural networks is that once they are trained, the intense requirements are done. A final list of components needed for this projects would be: a camera, a computer containing an average powered CPU, and all the required cables to ensure proper connection between the two. The software being used is open source and has no cost to us and, therefore, no cost to the client.

Future Goals

As this is an active project at the time of this writing, we have come across complications that have set us back from our original plans and have not achieved a working prototype yet. Progress has been made since the writing of the previous paper however, and we are sure to have more by the 22nd of March. At that point, the focus will turn to collecting data and using that data for the predictive wait times mentioned before. We hope to have the client application running and showing estimated times alongside the working neural network by the date shown above.

Conclusion

While the final project has not been completed as of the writing of the paper, our standing conclusion is that this system will remarkably enhance any business or service that utilizes unstructured queues. Places such as airports or train stations will gain useful data to help them make better business decisions, while the front end application for the customer will allow them to see the average wait time before they even leave their house, reducing the chance of a missed flight or train. Other businesses or events such as festivals and amusement parks will also benefit from the system for the same reasons, which is all possible due to the artificial neural network.

1. Gurney, Kevin (1997). An introduction to neural networks. UCL Press. ISBN 978-1857286731. OCLC 37875698.
2. "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.
3. Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," In 20th International Conference Artificial Neural Networks (ICANN), pp. 92–101, 2010. doi:10.1007/978-3-642-15825-4_10.