Theme Park Wait Time Estimation Optimization

TeamMembers

Overfitted Misfits

Problem Statement

Current wait time approximation methods for theme park lines are unreliable, leading to theme park guest frustrations. There is no existing method that provides an accurate, real time estimate of theme park wait times. Theme parks such as Disneyland use physical landmark references and portable reference points to estimate and verify wait times. For physical landmarks, the theme park workers estimate wait times based on where the line ends in relation to a physical reference point. An example of portable reference objects would be the lanyard that Disneyland uses. The workers at each ride will periodically give one of the incoming guests a lanyard, instructing them to carry it through the line and give it to the last worker they see as they board the ride. This method is unreliable due to its physical nature. It relies on the efforts of humans to identify the time required for an incoming person to get on the ride, creating space for human error. The human error related to the physical landmark references arises because of the variation in distance between people in line. If the distance is larger, the line will reach the physical landmark with less people in line. Therefore, the actual wait time will be shorter than expected. The opposite is true with less distance between people. As for the portable reference points, there is human error when the guests forget to hand it to the worker or if they lose it somewhere along the way. In addition, the wait time updates are slow since no actual time updates can be identified until the next lanyard successfully goes through the line.

Potential Methodology

This project will utilize facial recognition to maintain a real-time wait time for theme park ride lines/queues. The process by which this is accomplished is by deploying a facial identification ML model to the microcontrollers which will determine the faceprints of the people in line. These faceprints will be reported to a central device for storage and comparison.

There will be at least two of these devices deployed for each ride; one at the beginning and end of a line. This can be extended to utilize many more devices to produce more time deltas rather than one large time delta. This would provide more insight into the movement speeds (flow rate) of each portion of the line.

The system works by the device monitoring the beginning of the line (watching people enter the line) and performing facial recognition on each person to produce their faceprints. The faceprints (the output of the edge's device's ML model) are then sent to the central device which will add them to a data structure and record a timestamp with that faceprint. If the faceprint is already in the data structure, then the timestamp will be updated (this is conditional on which edge device reported the detection).

Then, at the exit of the line (for a 2-device system), each faceprint produced by the exit device for people exiting the line will be sent to the central device. The central device will check the data structure to see if that faceprint has been seen before, and, if yes, will calculate the time delta from the previously recorded timestamp stored in the data structure for that faceprint with the current time. This time delta represents how long it took that person to move through the entire line. A rolling average will be calculated to average out all the data points, detect outliers, and represent the current wait time for the ride.

The edge devices will be performing facial recognition and producing the faceprint data and will consist of a MCU such as an Arduino, ESP32, or Pi Pico, and a camera.
The central device is, for our purposes, a Raspberry Pi or other SBC which is used to store the faceprints in a data structure, search the data structure each time a faceprint is produced to see if it has been previously reported, update/retrieve the timestamp records for the faceprints, and calculate and report the rolling average which represents the current ride wait time (and possibly display said value at the beginning of the ride line).

The central device is used due to potentially large storage requirements of faceprints, and to aggregate the data from the edge devices due to this being a networked arrangement. Privacy is also preserved due to all processing being local and no internet connectivity is required.

Dataset

To achieve our goal, we will use existing human detection or face recognition datasets. There are many datasets available, one of which is the CrowdHuman dataset. It is a large-scale benchmark dataset aimed at detecting people in crowd scenes. It contains a total of 20,370 high-resolution images, 15,000 for training, 4,370 for validation, and 5,000 for testing. A total of 470,000 human instances are included in the training and validation subsets, with an average of 23 persons per image and various occlusions. Potential datasets for face recognition would be 1) Labeled Faces in the Wild (LFW) dataset - contains more than 13,000 facial images collected from the web. These images contain significant variations in pose, lighting, and expression; 2) CelebA dataset - contains over 200,000 celebrity faces annotated with 40 attributes such as gender, age, and hair color; 3) VGGFace2 dataset - contains over 3.3 million face images from over 9,000 identities. The images are collected from the web and vary significantly in pose, lighting, and expression.

References

Dataset:
- ● HumanDetection
- ○ CrowdHuman:ABenchmarkforDetectingHumaninaCrowd
- ○ PKUHumanIDDataset
- ● FaceRecognition
  - ○ LabeledFacesintheWild(LFW)dataset:http://vis-www.cs.umass.edu/lfw/
- ○ CelebAdataset:http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
- ○ VGGFace2dataset:https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/