RTCHub : Home Monitoring with Machine Learning

1) Abstract

Home video monitoring systems have been expensive and complex setups with a high barrier to entry. Existing easy to use solutions are single box setups like Nest, Piper, Withings or Canary which are too expensive, has a subscription fee and lacking intelligent features. On the other-side, the saturated CCTV market is not open for innovation, hard to setup, relies on static ip, requires expertise and not portable.

To fill this gap I have been working on a low cost, low maintenance, low barrier to entry device with an intelligent software stack which also bridges the two sectors. It has an ergonomic feature set not found in any of the devices currently in the market.

2) Features

- “Faces of the day” : Faces seen through the day is maintained as a daily face log.
- “lastseen” : Shows when the family members were last seen in the house
    - Overtime learns the behavior of the family members and predicts the presence of a certain member and can also trigger warnings when a certain member is not seen at home for a long time or when a new face is found in the house.
- “Timelapse” : Each day is available as a small clip which can be used to review the whole day in minutes.
- “Timelapse to DVR Mapping” : The camera feed is recorded to local storage. It is then made easy to access only the interesting parts by using the “Timelapse” feature and “Faces of the day” Feature
- The IP cameras are wireless and paired to communicate with the HUB, which in-turn will expose the streams securely to the app. This network can consist 2-4 wireless cameras spread around the house
- Hub will have a speaker which is used to have two way audio communication between the hub and the speaker.

3) Implementation

I have built a proof of concept by reverse engineering a Chinese IP camera and using an embedded Linux system as the central HUB. I have successfully implemented the “Faces of the day”, “Timelapse” , “Timelapse to DVR Mapping” and “lastseen” feature set on the linux embedded system.

Some implementations are entirely rewritten to run on the low resource embedded systems and none of the facial recognition and ML algorithms are run on the cloud, which means the user privacy is respected.

The mobile app is written using a cross platform infrastructure to be able to release both iOS and android versions together with minimal code rewrite.

4) Feasibility

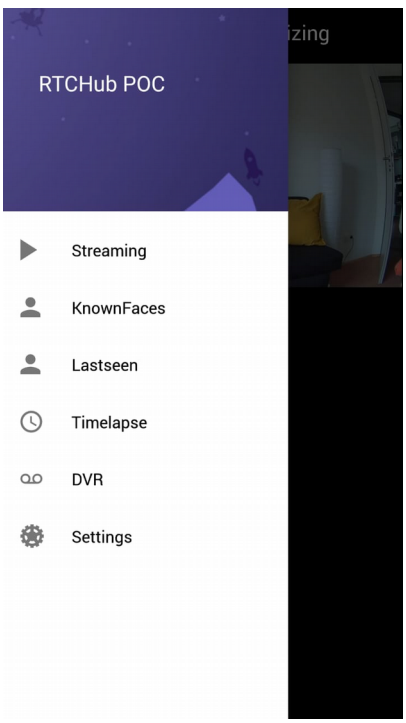
Hardware: The choice of hardware used to implement the HUB is both easy to source in mass and low cost. IP camera is also easy to source and the choices are many.

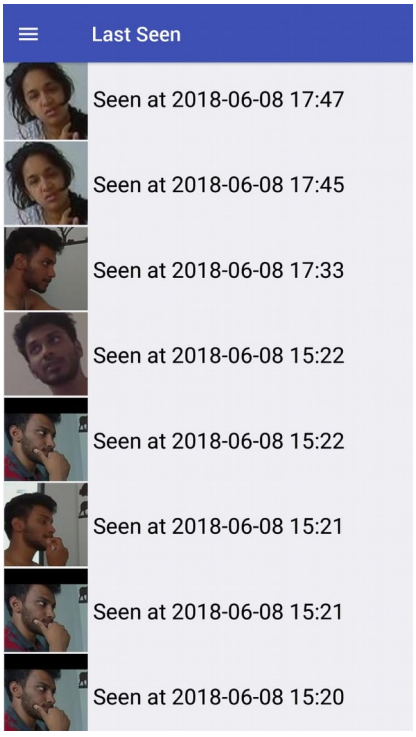
Software - Embedded System: The core architecture is based on making use of the embedded system for all computationally intense tasks and use less resources on the cloud infrastructure. This makes long term maintenance cost effective and scalable.

Software – Mobile: Apps are built using cross platform infrastructure which decreases the development time and improves efficiency for a small trade off in performance.

IP Camera and Embedded System

Live Streaming with axis control





Features implemented on app