Artificial Intelligence in Distributed Edge Networks: Optimizing for Security and Latency in Surveillance Systems

Ameena Golding
Katie Li
Rosanna Neuhausler

Abstract

Machine learning frameworks in conjunction with hardware advancements has allowed IoT applications to support more data-intensive tasks. The proximity of resources not only garners benefits in terms of latency and bandwidth, but also allows developers to address security concerns of sensitive data. We propose a method of decreasing the amount of data sent to the cloud in the context of surveillance systems in order to preserve the security of video data, as well as benefit from decreased latency by processing data on the edge. By utilizing caching techniques, in conjunction with CNN abstractions for minimizing stored data per individual, we present a proof-of-concept built on currently available commercial hardware to experiment and evaluate our design. Our results show that facial recognition done on the edge is a viable solution to the security concerns of current surveillance networks.

1. Motivation

1. Increased risks in leaking sensitive information associated with Cloud computing.

The increase in the number of paths over which data is transported (i.e. movement from edge sensors to the cloud) and the concentration of millions of users' data into a small number of databases (i.e. clouds themselves) has provided hotspots for malicious attacks (bandwidth monitoring and risky administrative access).

2. Developments in the hardware of edge devices and machine learning methods.

The optimization of machine learning algorithms and the increasing capabilities of running them on the edge has created a space for edge computing capabilities.

2. Objective

Our main objective is to reduce risks associated with sending and processing confidential information in a cloud-dependent surveillance system.

A secondary goal is to expand on ideas on how artificial intelligence fits into edge operating systems.

3. Materials and Methods

We eliminate the sending of confidential data to the cloud through computing the task of intruder detection at the edge.

Confidential data is defined as real-time information of individuals that the security system is meant to protect. (i.e. video streams of residents and workers in homes and office settings, respectively).

Cloud independence is possible by making the edge "smart" through running a face recognition machine learning framework, dlib and the face_recognition library, on the cameras generating the surveillance data. This data-heavy task is made possible through reducing storage and memory concerns using the following two techniques:

1. Encoding images

3 dimensional data object
size = f(resolution)

128 numbers
size ⊥ resolution

2. Caching encodings

We use the Amazon DeepLens hardware and its AWS cloud connection to simulate cloud-independence and dependence. We measure and record resulting latencies and scaling capabilities based on the set-ups shown in Figure **1.1** and Figure **1.2.**

n number of people
m number of encodings per person
* cached
main channel of communication
backup channel
memory
storage

AWS DeepLens

- 4-megapixel camera with MJPEG
- 8 GB RAM
- 16 GB of storage capacity
- Optional 32 GB SD card for additional memory
- Intel Atom Processor (2 cores at 1.30GHz)

IP Camera

- 1.3-megapixel camera
- 32GB max local storage
- Supports TCP/IP, UDP, HTTP

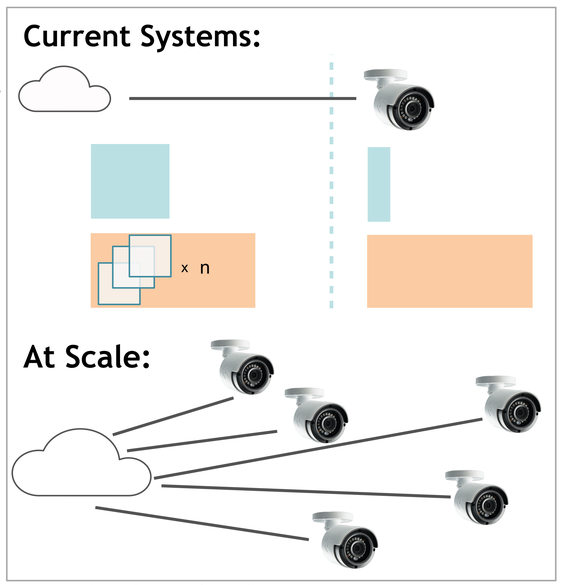

Figure 1.1: General set-up of surveillance systems

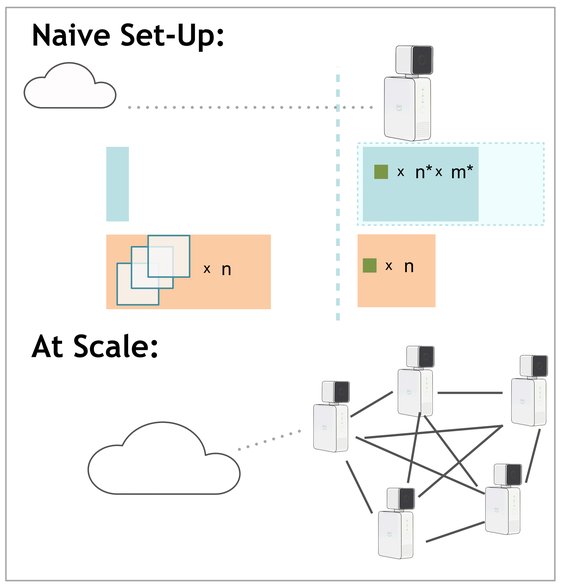

Figure 1.2: Cloud independent set-up using encodings and caching

4. Results

Cache Set-Up	Max Entries in Cache	Encodings per Person
1	10	3
2	20	10

Figure 1.3: Cache configurations on the edge device

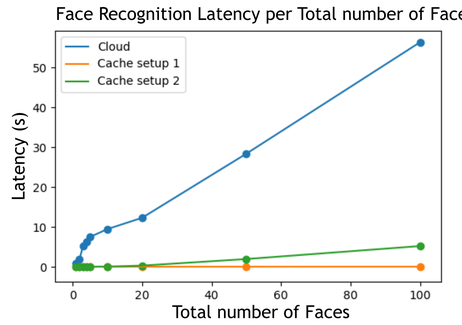

Figure 1.4: Scaling behavior of face recognition on edge and cloud

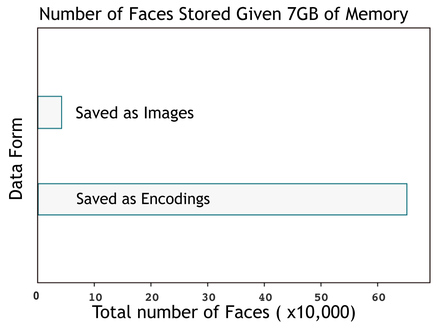

Figure 1.5: Number of faces that can be stored using images or encodings

5. Discussion

5.1 Results

As Figure **1.4** shows, the scaling behavior of a pure cloud model is not only unsustainable for real-life application, but also heavily dependent on network bandwidth. The experiment was conducted on UC Berkeley's campus network, but relative performance between cloud and edge devices should behave similarly under different network conditions. Figure **1.3** shows two different cache set-ups utilized in order to determine the optimal configuration. Main concerns with having too many encodings per person involved the delay incurred by the linear comparisons done for each face. The on-device storage is used as a backup for the database of faces in case of power or other failures. Figure **1.5** highlights that by only storing the encodings, we are able to store around 6 million unique encodings more as opposed to storing images of faces at a given resolution of 480p at 300x300 pixels.

5.2 Increased Security and Privacy

By reducing the amount of traffic sent to the cloud for processing, multiple security concerns are addressed. Consider the case where a constant stream of video data is sent to the cloud for facial recognition processing. From network traffic, the current location of individuals appearing in the data can be easily accessed. Although this might be desirable for CCTV video surveillance networks, our targeted use case of intruder detection assumes that the set of people with granted access is known, and that the whereabouts of these individuals should not be trivially revealed regardless of intruders in system. As we also choose to store the image encodings of individuals instead of pictures, the identity of each permitted person can also be obfuscated as faces cannot be regenerated from encodings.

5.3 Other Benefits

Our approach also addresses computing concerns, namely latency and failure tolerance, of the system. By storing the list of encodings for permitted individuals in persistent storage, intruder detection is able to continue in the event of network failure or powering off of a device. Our results have shown that although it is slower to go to persistent storage first before the cloud, it is useful to keep an on-device list of encodings to improve failure tolerance.

6. Conclusion

By reducing dependence on the cloud, we address the main concern of security and privacy through reducing sensitive data sent in the network. We are also able to decrease latency through our design of caching and improve storage efficiency by saving encodings instead of images.

7. Acknowledgements

We'd like to thank Joey Gonzalez and Marcel Neuhausler for introducing us to the issues of privacy in regards security camera streaming and to the concepts around AI at the edge. We'd also like to thank John Kubiatowicz for teaching us advanced topics in computer systems this semester.

8. References

https://www.nortekcontrol.com/pdf/literature/InterProCameras_Lit.pdf
https://aws.amazon.com/deeplens/

riselab
UC Berkeley