CLOUD-BASED FACIAL RECOGNITION SYSTEM

STUDENTS: DAVID YU, ERIK WHEELER,
HAOBO ZHANG, SIMON WANG

more than just money®
BECU

Problem Statement

Over the past decade there have been numerous advances within the banking industry in the use of technology, specifically facial recognition. This revolution brings a new level of security within business transactions with authorized access. BECU, working together with the University of Washington ECE Department, would like to bring facial recognition technology to 50+ Neighborhood Finical Center (NFC) locations and 1,000+ ATMs. This new facial recognition software would be deployed with hopes of preventing malicious actors from impersonating BECU customers, thus creating a more secure banking experience.

Requirements

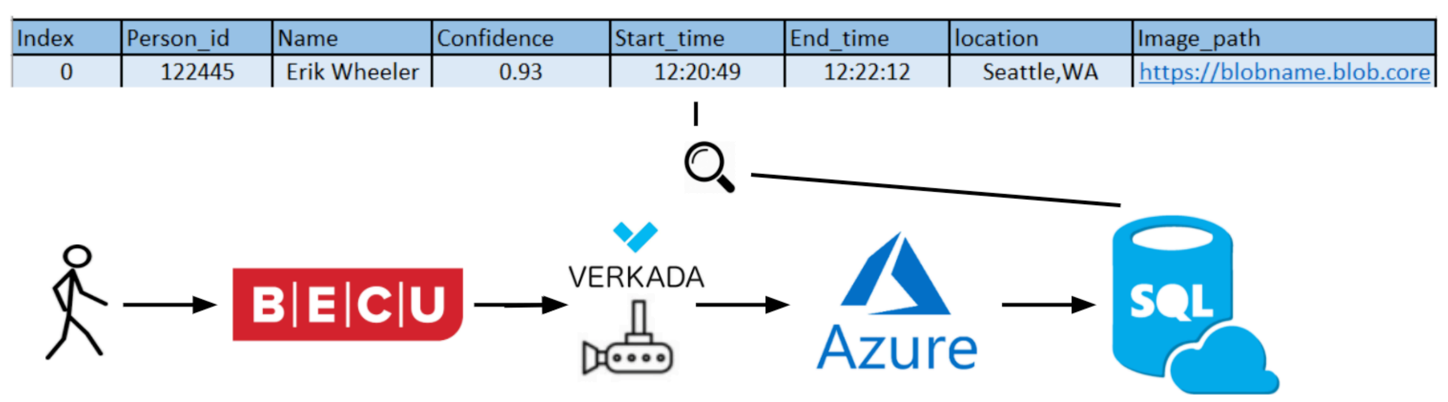


Figure 1: High Level Design Overview of the System

- The system needs to perform computation and facial recognition non-locally in Microsoft Azure.
- The system should utilize IP-based security video camera technology provided by Verkada.
- The system should be able to scale to handle 1,000+ ATMs and 50+ NFCs with over a million customers.
- The system needs to achieve accurate facial recognition in real-time.
- The system should be able to record all facial recognition results as transactions in a SQL server.
- The system will train Azure Cognitive Services (ACS) model on one headshot picture for each customer i.e., the customer's driver's license picture.

Different Use Cases		
	ATM	NFC
Camera Distance	1-3 feet	10-15 feet
Camera Angle	Side view	Overhead
Customers to be recognized per Frame	Assume one customer	Assume multiple customers
Time Customer Spends in Camera View	1-3 minutes	5-15 seconds

Figure 2: How the Use Cases differ between ATM and NFC

Overall System Implementation

- Our system is completely deployed on Microsoft Azure.
- Our system takes input from the video stream of the IP camera mounted either on BECU ATMs or Financial Centers and logging the visit information of BECU customers to the the SQL database.
- The subsystems are containerized as a docker image that is deployed on Azure VM.
- Preprocessing subsystem, recognition subsystem, Azure Storage Queue and post-processing subsystem are scaled up to be deployed on every ATM and NFC.
- Depending on whether the system is deployed on an ATM or an NFC, the preprocessing subsystem and postprocessing subsystem are designed differently.

Different Design Parameters		
	ATM	NFC
Motion Trigger Threshold	High	Low
Number of Frames Processed per Second from Stream	Low	High
Number of Intermediate Logs Processed per Second	Low	High
Minimum Transaction / Visit Time	High	Low

Figure 3: ATM and NFC Use Case Design Choices

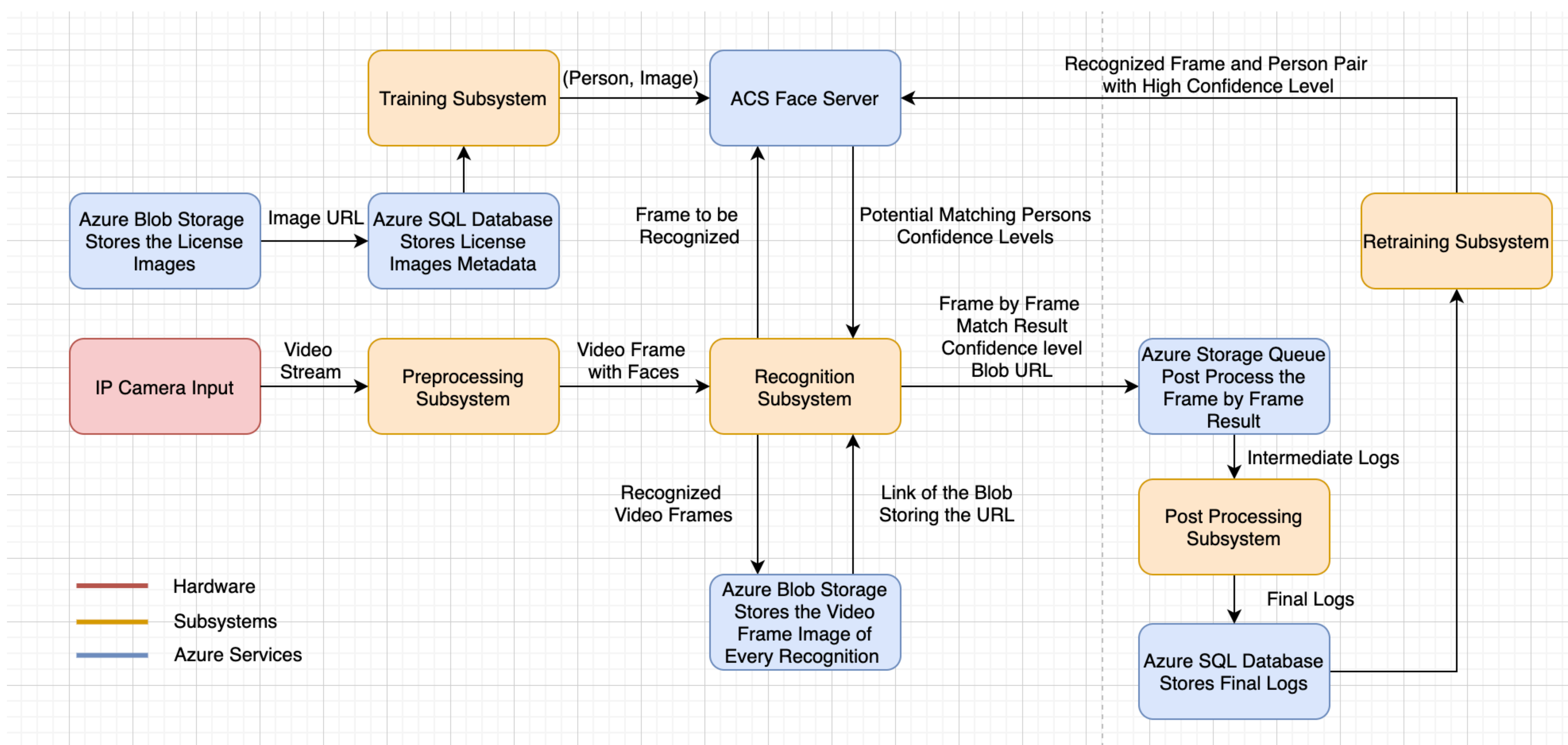


Figure 4: Full System Diagram

- **Preprocessing subsystem** reads from camera video stream, performs motion detection and extracts a frame only when there is motion detected.
- **Recognition subsystem** sends the frame image to the ACS Face Server and gathers recognition results. It also uploads the frame image to Azure Blob Storage and saves the URL of the blob.
- **Postprocessing subsystem** processes the intermediate logs stored on the Azure Storage Queue and generates the final logs.
- **Retraining subsystem** reads from the final log SQL Database, extracts the recognition logs with high confidence level and uses them as training data to the facial recognition model.
- **Training subsystem** trains the the facial recognition model with the BECU customer names and their corresponding license pictures.

Discussion

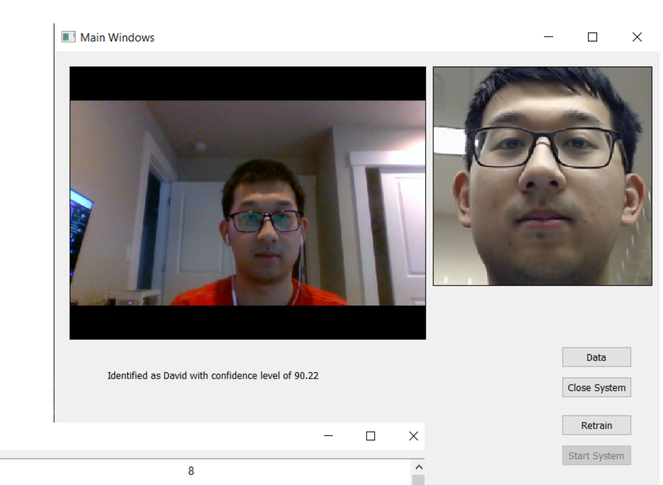
We made several implementation decisions to utilize one choice over the other.

- Azure Cognitive Service vs Siamese Model
  - Siamese Model we implemented is free to use.
  - ACS offers better accuracy and better synergy with other Azure Services than the Siamese Model.
- Post-Processing vs Non-Post-Processing
  - Intermediate logs generated per second (~ 2000 logs per second) are too many to directly log into a SQL database.
- Azure Storage Queue vs SQL Database (for intermediate logs)
  - Intermediate logs are generated on each instance of deployment (every ATM or NFC). It is too expensive to maintain a SQL server for all logs (About $900 per month).
  - It is cheaper to create and maintain multiple Storage Queues (Only about $20 per month).
- Multiple ACS Face Servers vs One ACS Face Server with Producer-Consumer Model
  - The facial recognition should be performed in real-time. Adding a queue introduces unwanted delays.
  - 1 ACS Face server with a standard subscription can perform recognitions up to 10 faces per second, which satisfies our peak load.

Testing & Results

We tested our system by feeding the system pre-recorded videos of team members walking towards a camera, stimulating video inputs of customers approaching ATMs and NFCs.

Our system was able to identify group members(small sample of 4 people) in mock tests with 100% accuracy and confidence level of 62% the first time and up to 91% on average after retraining.



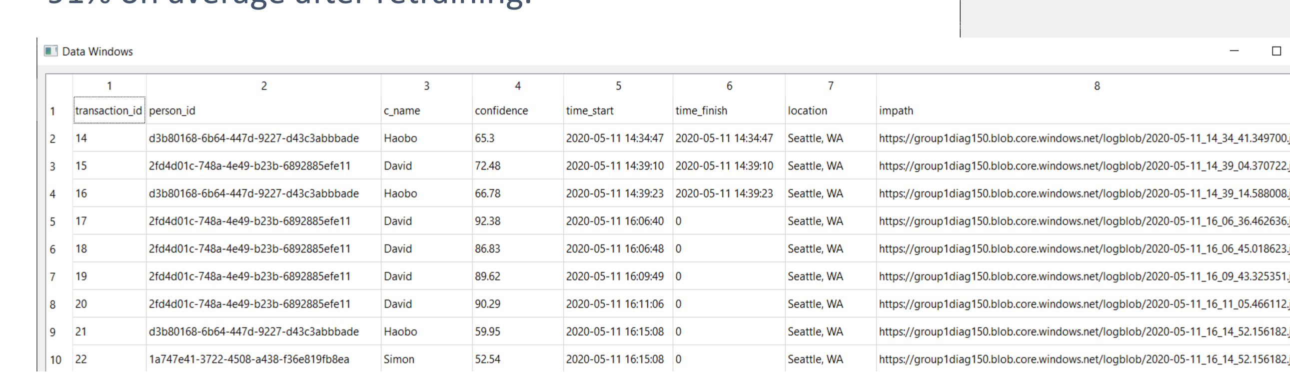


Figure 5: User Interface (UP)

Figure 6: Transaction Log (LEFT)

Future Work & Acknowledgments

- Handle more edge-cases in post-processing such as a person whose facial data is unavailable.
- Develop a more sophisticated Siamese Model to replace the current use of ACS.
- Fully utilize the Verkada Security Camera (its security settings are incompatible with our current system).

We would like to thank Jim Jenkins and his team as well as Professor Rania Hussein for their guidance on this project.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISORS: JIM JENKINS, RANIA HUSSEIN

SPONSORS: BOEING EMPLOYEES CREDIT UNION