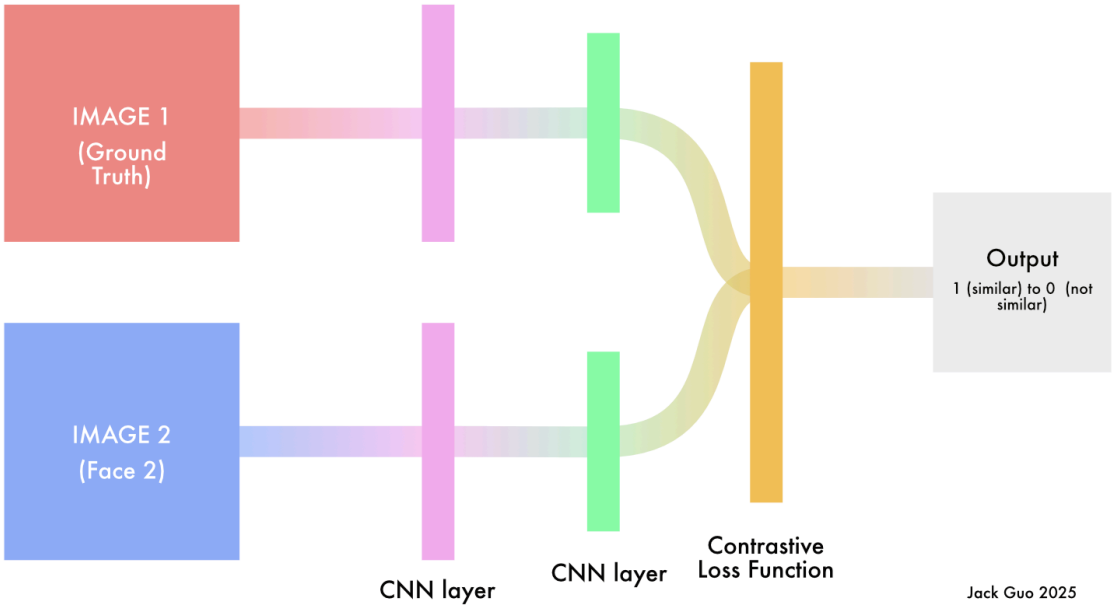
Project name: Smart Doorbell

Team members: Marilyn De Leon, Jack Guo, Aidan Shepston

Progress & Plan:

I. *Setup completed:*
   a. Successfully configured the Raspberry Pi
   b. Installed required dependencies: Python, TensorFlow Lite runtime, and OpenCV
   c. Downloaded datasets locally for testing purposes

II. *Machine Learning implementation:*
   a. In order to pick out faces from video, we want to use an existing tensorflow lite model,https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB/tree/master, which will provide the actual face detection and cropping so the system knows what to perform the facial recognition on. This model will box faces and orient them in a common way, which makes input into the next stage much cleaner.
   b. We want to use a Siamese Network to detect similar faces. The network is set up in such a way to detect similarities in 2 different images. This is accomplished by having 2 convolution neural networks with the same weights and bias that we input two images, one is the anchor image and another is either positive and negative. *[1]* The loss function is contrastive, meaning that it pairs data points closer together in feature space with a higher score. Our output in the trained model will be a float with 1 being the most similar, and 0 being the least similar.
   c. We will be training using the Labelled Faces in the Wild (LFW) Dataset from kaggle. I am preprocessing data using google collab and tensorflow.image to rescale all images into 100 by 100 pixel size. We will also divide and label the images positive, negative and anchor.
   d. https://www.kaggle.com/datasets/atulanandjha/lfwpeople



III. Results generation plan:

a. Collect and store sample images of people for model testing (2-3 people, 15-20 images each)
b. Measure model inference time, accuracy, and false detection rate of the doorbell system on Raspberry Pi in a number of different environments
c. Record and analyze results to demonstrate success of recognition and performance metrics in different environments and with different people

IV. Demonstration plan:

a. Integrate real-time inference pipeline:

   when a person appears -> capture image -> detect and crop faces -> perform pre-processing on images -> run Siamese Network to compare similarity -> display/log classification result

b. Prepare a short demo video showing system functionality and accuracy running in real time on a raspberry pi
c. Demonstration could also be done in person in real time in a class setting, but this may come with additional issues with regards to environment and lighting so will determine later if this is feasible
d. If time permits we may implement a notification system when an unknown person is detected

V. *Writing plan*

a. Update GitHub repository to include setup instructions, code structure, and hardware configuration details
b. Update GitHub repository to include final reflection and performance summary after testing phase
c. We also plan to extensively document our design choices, procedure and overall pipeline in a final report. This will go more into depth than the github, explaining the underlying systems we chose as well as how they function. Within this report, we also plan to discuss our results as well as what areas could be improved in the future.

Progress attributed to each team member:

I. Marilyn De Leon

a. Contribution progress:
    i. Researched TensorFlow Lite workflows for facial recognitions focusing on preprocessing pipelines and model conversion steps that could be used for running inference on edge devices.
    ii. Reviewed multiple open source implementations and face verification datasets to better understand how embedding distances can be computed for recognition.
b. Contribution plan:
    i. Assist with pre-processing the dataset by rescaling images, and organizing the data to support the SIamese model training.
    ii. Work with the team to help test the TensorFlow Lite model once generated, and identify any pre-processing adjustments needed before integration with the hardware.

II. Jack Guo

a. Contribution progress:
    i. Researched the use of Siamese Network for face recognition through articles and videos. Designed an architecture of the Siamese Network that can be used for our project.
    ii. Researched and gathered Datasets for the training of the network. Started preprocessing data by making all resolutions the same.
b. Contribution plan:
    i. Finish preprocessing the data using google collab and tensorflow.image to rescale all images into 100 by 100 pixel size. Then implement the

Siamese Network in google collab. Train the model using the Labelled Faces in the Wild to produce a model for Tensorflow lite to run on edge devices.

III. Aidan Shepston

a. Contribution progress:
    i. Set up Raspberry Pi 4 model B with Raspberry Pi OS Lite and connected it to a computer over ethernet.
    ii. Picked out a tensorflow lite model for face detection (not recognition) and loaded it on a computer.
    iii. Downloaded relevant datasets (LFW) locally for testing purposes.
b. Contribution plan:
    i. Connect and interface with Pi camera once it arrives
    ii. Integrate face detection model with future face recognition model including pre-processing images to allow for a higher accuracy.

VI. References

[1]

C. Song and S. Ji, "Face Recognition Method Based on Siamese Networks Under Non-Restricted Conditions," in IEEE Access, vol. 10, pp. 40432-40444, 2022, doi: 10.1109/ACCESS.2022.3167143.

keywords: {Face recognition;Feature extraction;Neural networks;Image recognition;Target recognition;Artificial intelligence;Artificial intelligence;face recognition;Siamese neural network;LBP algorithm;frequency feature perception},