FACE RECOGNITION

Using Raspberry Pi, Opencv and Deep learning

Group 1

OVERVIEW

Face recognition is one of the primary tasks especially in computer vision field, becoming more and more important as advances in technologies such as Internet, digital cameras require increased security key features. Face recognition can operate either on still images or videos(image sequences). Also, it can manage either or both tasks: face verification and authentication. Face recognition is extreme challenging task as the performance significantly changes with change in lighting, pose, scale or occlusion.

GOALS

To develop a face recognition system which could help in some practical applications like Security, Immigration, Banking, Mobile unlocking and Verification.

System Specifications: Azure Cloud Platform - NVIDIA Tesla K80 with 1 GPU and 12 GPU memory GiB

Raspberry Pi: Model B, Pi camera

DATA

Data Preparation is performed manually:

1. Taking realtime videos and converting them into image frames.
2. Using the Bing API from Azure Cognitive services and obtaining images of people through API calls.

In addition, LFW: a database of face photographs designed for studying the problem of unconstrained face recognition is also used.

METHODOLOGY

To develop and implement different approaches for face recognition using keras, opencv and Raspberry Pi.

❏ Opencv is used for data preprocessing which includes image handling.

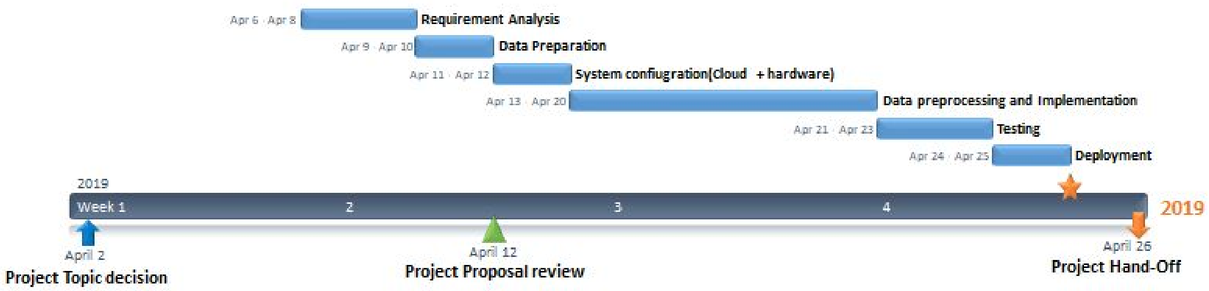
❏ Facial detection is performed using deep convolutional neural network.

- Most efficient method is chosen for face recognition post comparison of all methods.

LITERATURE REVIEW

TITLE	DATA	METHOD
FaceNet: A Unified Embedding for Face Recognition and Clustering	Labelled Faces in Wild and Youtube Faces	It is based on learning a Euclidean embedding per image using a deep CNN with Triplet Loss
Deep Face Recognition	Labelled Faces in Wild and Youtube Faces	Convolution neural networks using Triplet Loss
OpenFace: A general purpose face recognition library with mobile applications	Labelled Faces in Wild	Deep CNN which uses FaceNet's triplet loss so that network provides and embedding on euclidean distance similarity

Project Milestones

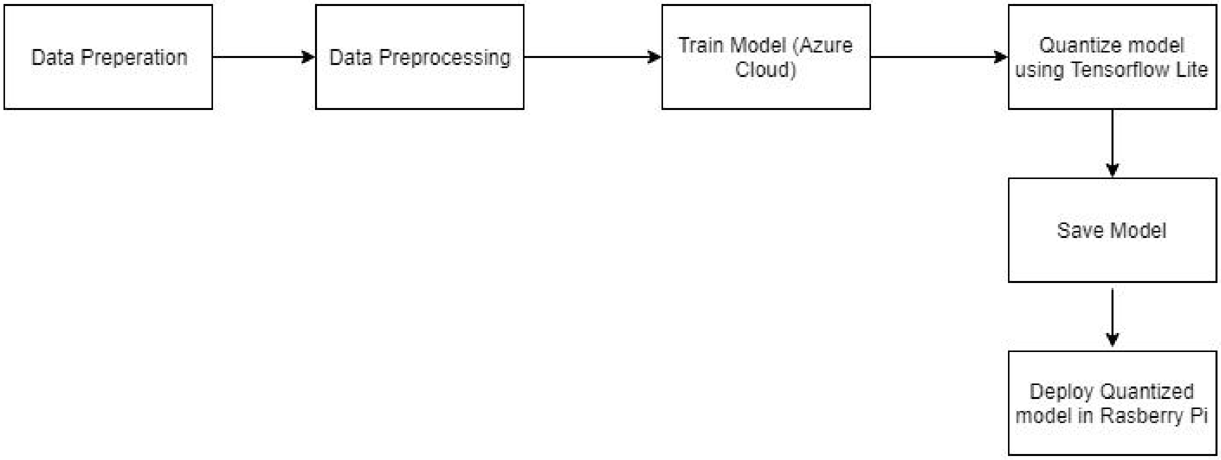


PROCESS PIPELINE

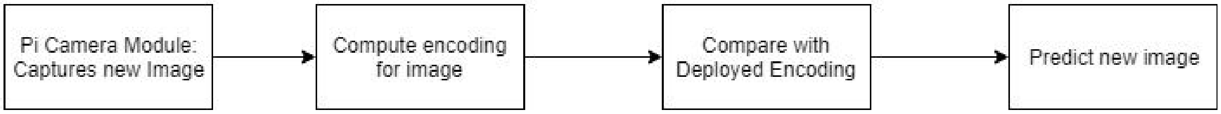
1. Data Preparation
2. Literature review about different approaches used for facial recognition
3. System system on Cloud: Azure VM
4. Data Preprocessing
5. Exploratory Data Analysis
6. Hardware setup: Raspberry Pi
7. Trial of different approaches for handling the problem and choosing the best one.
8. Design of a pipeline to implement the best approach obtained.
9. Post-training quantization for the best model built using tensorflow lite.
    a. **Why Quantization?**
        i. Raspberry Pi doesn't have much memory to process large models due to which quantization can help optimizing weights.
        ii. The linear spread makes multiplications straightforward.
        iii. A symmetric range for weights enables downstream hardware optimizations.
10. Deploy the model in Raspberry Pi and test with real time video/image.

SEQUENCE DIAGRAM

Part 1: Training and Deploying the Model


Data Preperation
Data Preprocessing
Train Model (Azure Cloud)
Quantize model using Tensorflow Lite
Save Model
Deploy Quantized model in Rasberry Pi

Part 2: Testing the model


Pi Camera Module: Captures new Image
Compute encoding for image
Compare with Deployed Encoding
Predict new image

DEPLOYMENT DETAILS:

Language: Python

Deep learning library used: Keras with tensorflow backend

Computer Vision library used: OpenCV

Container: Docker

Cloud Tools/Platform used: Azure VM with Linux OS, Bing Cognitive services API

Raspberry Pi OS: Raspbian which uses Debian

REFERENCES

http://elijah.cs.cmu.edu/DOCS/CMU-CS-16-118.pdf https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf

https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html

https://docs.microsoft.com/en-us/azure/cognitive-services/bing-web-search/tutorial-bing-web-search-single-page-app