



Open Source Promotion Plan(OSPP)' 2022

Final Report

1. Project Description

- 1.1 Title
- 1.2 Brief Description of the Problem
- 1.3 Brief Description of the Solution

2. Project Timeline:

- 2.1 Project plan and Deliverables Schedule
- 2.2 Areas of project that were most/least challenging

3. Project Output

- 3.1 The Accomplished Work
- 3.2 Problem and Solution
- 3.3 Subsequent Work Arrangement

4. Notebook Guide, PR & Documentation

- 4.1 Step by step guide - Jupyter Notebook
- 4.2 Github *README.md* Guide
- 4.3 Pull Request on *'milvus/bootcamp'*

5. Community and Contribution

- 5.1 Communication and Feedback
- 5.2 Source Code

6. References

Project:

Add a face recognition Bootcamp project to Milvus.

[227410324]

Organization:

Milvus

1 Project Description

1.1 Title

Add a face recognition Bootcamp project to Milvus.

1.2 Brief Description of the Problem

Milvus is a world-class distributed vector database. We would like to add a great face recognition demo to demonstrate the ability of Milvus and to teach users step-by-step how to use Milvus in real-life AI applications.

1.3 Brief Description of the Solution

- Developed a Face Recognition Pipeline using Milvus Vector Similarity Search Engine, Computer Vision and Deep learning.
- The procedure commences with Face detection using MTCNN, then storing the extracted face embeddings from FaceNet on the Milvus Database, followed by Face classification.

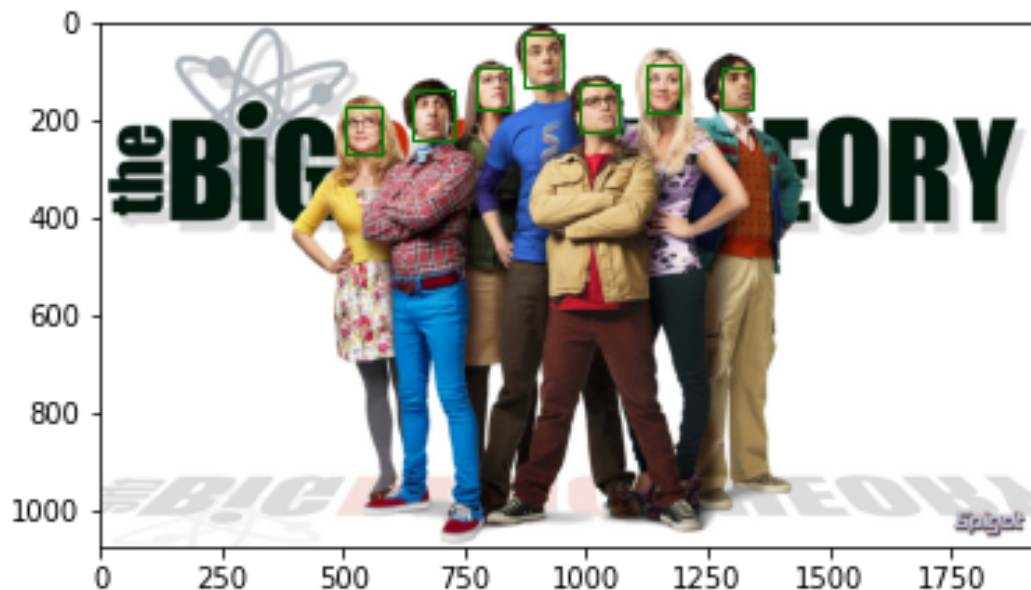
2 Project Timeline

2.1 Project plan and Deliverables Schedule

The objective is to develop a Face Recognition Pipeline on some training set of images and evaluate its performance on the basis of Accuracy and Inferencing speed, using Milvus Database, Image Processing and Deep learning.

- *Pre Coding Period*
After spending days and days searching for a jekyll theme for my OSPP logging, I finally found an amazing theme. Explored Milvus Database.
- *Week 1*
Completed the procedure of Face Detection using MTCNN (Multi-Task Cascaded Convolutional Neural Networks). Tested the detection algorithm on various images & faces.

★ Face Detection Results:



★ Colab Notebook Link:

<https://colab.research.google.com/drive/1WOPVxNc33eonhDD4gX2VGQ8zofbKm3fF?usp=sharing>

- *Week 2*

Tried & tested Face Detection using RetinaFace and Explored Face Embeddings Extraction Methods.

★ References:

- [Research Paper- Feature Extraction](#)
- [A study of Face Embedding in Face Recognition](#)

- *Week 3*

Completed the implementation of Facial Embeddings Extraction through Facenet & OpenCV.

★ GitHub Repo [Draft]:

https://github.com/Spnetic-5/ospp_face_rec_draft

- *Week 4*

Loaded the FaceNet model to obtain embeddings from the images of cropped faces Embeddings vectors and they were then successfully converted back to numpy arrays.

★ Google Colab Notebook: [FaceNet Model](#)

- *Week 5*

Connected to the Milvus server & created a collection in the Database. Inserted vectors & Stored these embeddings on Milvus Database. Tried to resolve errors & warnings in the code due to Ubuntu 21.01

- *Week 6*

Created a collection on the Milvus Database if not already present. Imported all the face embeddings into the created collection. Performed Indexing on the collection. Searched for the nearest neighbor of the given image using vector similarity search with Milvus.

Demo Results:

```
sp-lit@sp-lit:~/Desktop/OSPP/face_rec_draft/face_rec_bootcamp$  
python celeb_finder.py test.jpg  
Running on device: cpu 🖥️  
Creating a collection on Milvus Database... 📊  
Collection created. ✅  
Loading in encoded vectors...  
(51598, 512) float32 [ 1 1 1 ... 3040 3040 3040]  
(51598, 512) float32 [3040 3040 3040 ... 534 534 534]  
(51598, 512) float32 [ 534 534 534 ... 7592 7593 7593]  
(51597, 512) float32 [7593 7593 7593 ... 9999 9999 9999]  
Vectors loaded in.  
Indexing the data... 📊  
Indexed! ✅  
Searching for the image ... 😬  
Wohoo, Similar Images/Faces found! 🎉  
[8328, 5876]
```

- *Week 7-10*
Getting the work reviewed from the Mentor and resolving minor issues. Performing suggested improvements and maintaining well structured code and following good coding practices.
- *Week 11*
Created a step by step guide using Jupyter Notebook which is required for running the project and made it easier for developers to replicate my work. Composed a well documented and instructive GitHub [README.md](#) for proper understanding of the project working.
- *Week 12*
Created the Pull Request on Milvus Bootcamp repository, requested for the review from Mentor. Made a Final Report of the project for submission on OSPP portal.

2.2 Areas of project that were most/least challenging

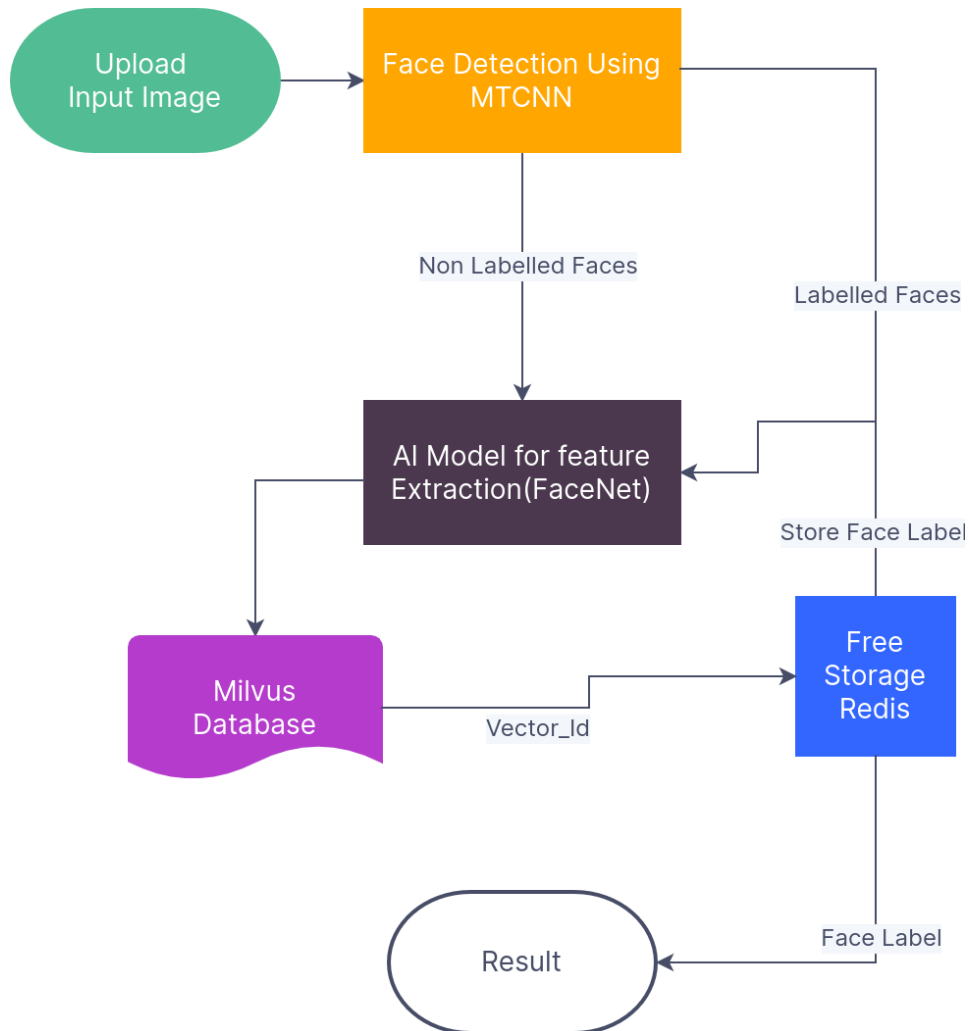
- ★ Performing similarity search on images was a bit challenging as I was not very familiar with this Milvus Database but due to excellent documentation I was able to do it.

3. Project Output

3.1 The Accomplished Work

- Developed a modern **Face Recognition Pipeline** which consists of 5 common stages: detect, align, normalize, represent and verify.
- **Face Detection** using **MTCNN**:
Trained an efficient, accurate and faster Face detection model for the detection of faces in the image or video.
- **Face Embedding** using **FaceNet**:
Extracted the important features and face embeddings from each detected face in the image.
- **Face Classification** using **Image Similarity Search**:
Classified each face based on its extracted features, performed face similarity search on Milvus vector database.
- **Created the Documentation & Jupyter notebook** with step by step guide for making custom real-life AI applications using the Milvus, which would allow anyone to duplicate my work on developing a Artificial Intelligence problem statement.
- **Github Pull Request & README.md** ReadMe file for the project is created with detailed information required for using and contributing to the Face Recognition Bootcamp.

3.2 Proposed Solution:



3.3 Subsequent Work Arrangement:

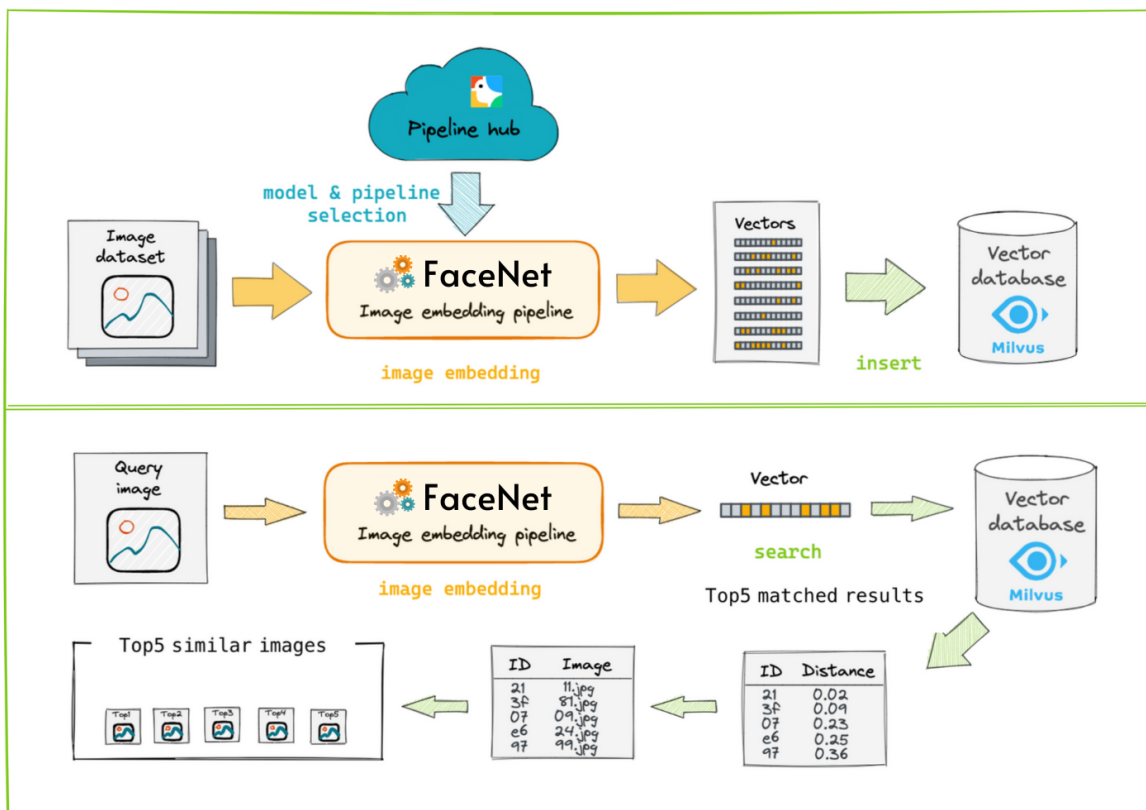
- ★ Migrating entire face recognition bootcamp from **Milvus v1.1.0** to **Milvus v2.0**, performing optimization and minor improvisation in the code .
- ★ Making Github README more informative and simpler.
- ★ Writing the *docker-compose.yaml* file for **Milvus v1.1.0** for quickly deploying project using docker.

4. Notebook Guide, PR & Documentation

4.1 Step by step guide - Jupyter Notebook:

★ **Link to Notebook:**

https://github.com/Spnetic-5/bootcamp/blob/dev/ospp_v1/solutions/face_recognition_system/face_recognition_bootcamp.ipynb



4.2 Github README.md Guide:

★ **Link to README.md:**

https://github.com/Spnetic-5/bootcamp/blob/dev/ospp_v1/solutions/face_recognition_system/quick_deploy/README.md

4.3 Pull Request on `milvus/bootcamp`

★ **Link to Pull Request:**

<https://github.com/milvus-io/bootcamp/pull/1111>

5. Community & Contribution:

5.1 Communication and Feedback

My mentor Jack Li was very supportive and helped me throughout the period. He was always available whenever I had doubts regarding the project. Along with him, excellent Milvus documentation was also very helpful & it played vital role in my project completion.

5.2 Source Code

★ **Link to Source Code:**

https://github.com/Spnetic-5/bootcamp/tree/master/solutions/face_recognition_system

6. References

1. <https://milvus.io/docs>
2. <https://github.com/milvus-io/bootcamp>
3. https://github.com/milvus-io/bootcamp/tree/master/solutions/reverse_image_search
4. <https://github.com/fhalmayer/milvus>
5. <https://cdmana.com/2021/06/20210620042502199c.html>
