A Digital Image Library – Making it possible with Facial Recognition

Lokesh Peta & Max Prud'homme

TCDL — 2023-05-17
The project

Objectives:

→ Building a robust and dynamic search engine to search for images related to a particular person using facial recognition

→ Delivering a trainable model for any image dataset to be retrieved using an interactable UI with DB

Challenges:

→ Computing power, resources, scalability, sustainability

Wrapping up:
There are two types of users:

- **Admin Support**
- **Patron**

**Home Screen**
- **Image Search by Face Recognition**
- **Search by uploading an Image**
- **Search by Name**
Technical Specifications

To work on the project, you will need the followings:

Software:
› Python 3.9
› Anaconda distribution

Hardware:
› 16GB RAM
› NVIDIA Quadro P400
Flow and Architecture

User Interface
- Search by Image or Name
  - (React Native)
  - Trigger application

Get embeddings of face to be searched
- Feed those embeddings to already trained model
  - Model Trained and saved
  - Returns the group of images related to the search
  - Labels and embeddings are used to train a classification model

Large Files of images
- Face Detection using MTCNN
  - Face Embeddings using Facenet

Along with face embeddings store shared space file path, drive file path, face file path, vectors (face embeddings) will be stored in SQL table
- Currently stored in shared space, for dev will use a hard drive to make processing a bit quick and avoid network issues
- Currently using csv files, will use SQL DB, later can be upgraded to Hadoop

Label for each face is created and stored in the same table (new column)

Note: planning to use only one model going forward
Progression

Original images

Detection of faces

Face Embeddings

```
array([[-0.01682856, -0.00642363, 0.03465647, ..., 0.0078016,
        0.00345464, -0.01197146],
       [0.00851609, 0.01758024, 0.01723476, ..., -0.12584084,
        -0.1199290, -0.00433318],
       [0.02066858, 0.0024223, -0.00118933, ..., -0.03829551,
        -0.00993131, -0.06472497],
       ...,
       [-0.0219016, -0.01447556, -0.02132493, ..., 0.03964908,
        -0.05926018, -0.04692425],
       [-0.01163085, 0.05863634, 0.04538924, ..., 0.02364511,
        0.05153345, 0.00697211],
       [-0.01444273, 0.03160005, 0.04416583, ..., 0.02467633,
        0.06337241, 0.0571871]], dtype=float32)
```

```
def train(embeddings, labels):
    softmax = LogisticRegression(solver='lbfgs', multi_class='multinomial', C=1.0, max_iter=10000)
    # If args.grid_search:
    #     clf = GridSearchCV(
    #         estimator=softmax,
    #         param_grid=[{'C': [0.001, 0.01, 0.1, 1.0, 10.0, 100.0]},
    #                      'C': 3}])
    # else:
    #     clf = softmax
    clf.fit(embeddings, labels)
    return clf
```

Prediction of new face

Train classification

Clusters
There are two types of users:

- Admin Support
- Patron
User can search a name

Name provided by user

Search based on Name

All images related to the Name provided
User can upload an Image

Image Uploaded by user

Face Detected

Search based on detected face

All images related to the face provided
Challenges

Computing:
- Batch processing phase
- Clustering phase

Scalability:
- Starting with SQL DB, planning to adapt Hadoop (Big data) Hive tables
- Applying technical stack to different scenarios of image search using facial recognition

Sustainability:
- Universal approach— a product to be utilized for different datasets
Wrapping up

» Current demo is for only small set of data

» While dealing with the entire set of data, accuracy starts dropping at each of the phases (detection, clustering, prediction)

» Planned to achieve best accuracy by continuous preprocessing of images based on the metadata collected

» Once finalising the model part, will start building an UI

THANK YOU!!!
References


