

Facety

A Directory Sub-structuring Using Face Recognition First Milestone Report

Anmol Chugh (ac4624), Bhavya Gupta (bg2682), Mansi Khemka (mk4282)

Project Aim:

The project aims at building a directory sub-structuring tool for images where image segregation among the various directories depend on the faces recognized in those images. In scenarios where an image can contain multiple faces, the image shall be present in the directory for each of those faces. The user is then equipped with accessing all images containing a particular face from the corresponding directory. The common use cases can be ease of access and privacy-centric sharing of images.

Existing Tools and Technologies Used:

Extensive open source libraries are present online which incorporate several deep learning algorithms. One such library is dlib by Davis E. King. This library includes a very clean code on Face Recognition based on a clustering mechanism called Chinese Whispers. The input to the program is a set of image paths. The code responsible for clustering the images essentially consists of two modules:

- The first module parses through each image and constructs an array of face matrices. Each such matrix is just a clipping of a face detected in that image.
- The second module is the core of the code. It treats each face matrix as a node and constructs a graph. The criteria for the existence of an edge between any two nodes in the graph is that the discrepancy between those nodes should be less than 0.6. Once the graph is constructed, it is passed to a function named 'chinese_whispers' which implements the famous Chinese whispers clustering algorithm. This function finds the clusters in the graphs and returns an array of labels for each node in the graph. The number of unique labels/clusters is determined by the algorithm itself. Each element in the label array is for the face matrix at the corresponding index (constructed in the first module).

One thing to note is that the code can only run on images where the faces are camera aligned or vertical. We will try our best to modify the code such that it can also run on rotated orientation of faces in images.

We are using C++17's Filesystem library for all file and directory operations including parsing directories, locating images, creating directories and copying images.

What is your plan?

Since we have access to a really awesome Face Recognition code, our focus is on the directory sub-structuring aspect of the application mentioned above. This is where the main design aspect comes into play. Our plan is divided into the following milestones:

1. Implement dlib locally and modify code pieces to interface well with the rest of the code. Test with sample images.
2. Write directory creation file for the clusters generated from the face recognition algorithm. Unit test the code with example images.
3. This is the main part of the project. Currently, whenever the code is run on a set of images, it creates some clusters with numeric IDs. For each new set of images, a local instance of the graph is created and the cluster indexing is reinitialized to 0. We want to be able to support the 'update' feature wherein we can save some sort of a global representation of the graph or cluster. Thus, when the user inputs a new set of images, it can append the new information/images to the previous one and prevent overwriting the previous clusters. Thus, either new directories will be generated if new faces are detected or old directories will be updated if a familiar face is seen. This is a little ambitious target but we will try our best to achieve it.
4. The final milestone is to build a GUI which supports all the functionalities in a user friendly manner.

What is the current state of your plan?

Currently, we have covered two milestones. We have successfully replicated dlib locally. Moreover, we have modularized the example code provided on Face Recognition, suited to our needs and from the perspective of optimized designing. We have built a directory sub-structuring routine that builds directories corresponding to each cluster. All code segments have been unit tested. The interface between these two code snippets have also been built and integration testing has been performed until this stage.

How far have you gotten?

We have currently completed the basic pipeline of our vision of the project. There are three major modules of the code :

1. Input : Name of the folder to be organized
Output : List of all image files along with their paths that need to be analyzed.
2. Input : Vector of images (in the form of image paths)
Output : Map of cluster labels and the images associated with that label.
The images are first fed to the face detector to create a vector of faces which are then fed to the chinese_whispers' clustering algorithm as explained above. The labels are then organized in the form of a map that relates each cluster/face/label to the images that the face is found in.
3. Input : Map of cluster labels and the images associated with that label.
Output : An organized directory substructure

What is your next step?

Our next step is to tackle the third milestone. We will work on the update feature and on maintaining a global clustering structure.

Directory Sub-structure created after the pipeline was executed:

