Guide install and demo face recognition

1 Installation

1.1 Requirements

1.1.1 Operating system

- MacOS
- Linux

1.1.2 Language programing and packages

- Python 3.3+ or Python 2.7
- packages
  - PIL.Image (install from pypi using pip3 or pip2)
    - `sudo pip install pil`
  - dlib (library contain machine learning algorithm)
    - `sudo pip install dlib`
  - numpy (library use for matrix, useful linear algebra, …)
    - `sudo pip install numpy`
  - face_recognition (contains api and model for face recognition)
    - `sudo pip install face_recognition`

1.2 Advance

- GPU acceleration (via nvidia's CUDA library) is required for good performance with this model when run with dlib.
  - Note: use GPU have compute capability >= 5.0

2 Application

2.1 Face detection

2.1.1 Find faces in a photograph and video

- Find faces in photograph
- Find faces in video
- Find faces in camera real time

2.1.2 Facial recognition

- Find and recognition unknown faces in a photograph based on photographs of known people
  - In terminal use command:
    - `face_recognition --cpus 4 -tolerance 0.4 -show-distance true ./pictures_of_people_i_know ./unknow_pictures`
    - parameters:
      - --cpu 4: use 4 core cpu (if --cpu 1: use all core cpu)
      - --tolerance 0.4: value tolerance distance highest between two images to match, higher is not match. Tolerance default is 0.6
      - --show-distance true: show distance between two images to match.
      - Pictures_of_people_i_know: folder contain images people you known name (note name pictures is name people in image)
      - Pictures_of_people_i_unknow: folder contain images people you unknow name
  - Result:
    - If show name people is unknow, mean don't people in folder image I know match with that people
    - If show two name people for a face, mean have faces people match that people (distance similarity of two face less than tolerance and first people have tolerance lower than second people)

- If one picture have more two people match two people you know, it will match with all that people
- We can adjust tolerance to fit match faces have accuracy best. (experiment: 0.4-0.45)

■ Recognition faces in live video using your webcam

- Use add code auto load images in folder auto assign name ever people
-

3 How face recognition works

3.1 Face recognition is really a series of several related problems:

■ First, look at a picture and find all the faces in it

■ Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.

■ Third, be able to pick out unique features of the face that you can use to tell it apart from other people

■ Finally, compare the unique features of that face to all the people you already know to determine the person's name

3.2 Face recognition – step by step

3.2.1 Step1: Finding all the faces

■ Use a method invented in 2005 called Histogram of Oriented Gradients (HOG)

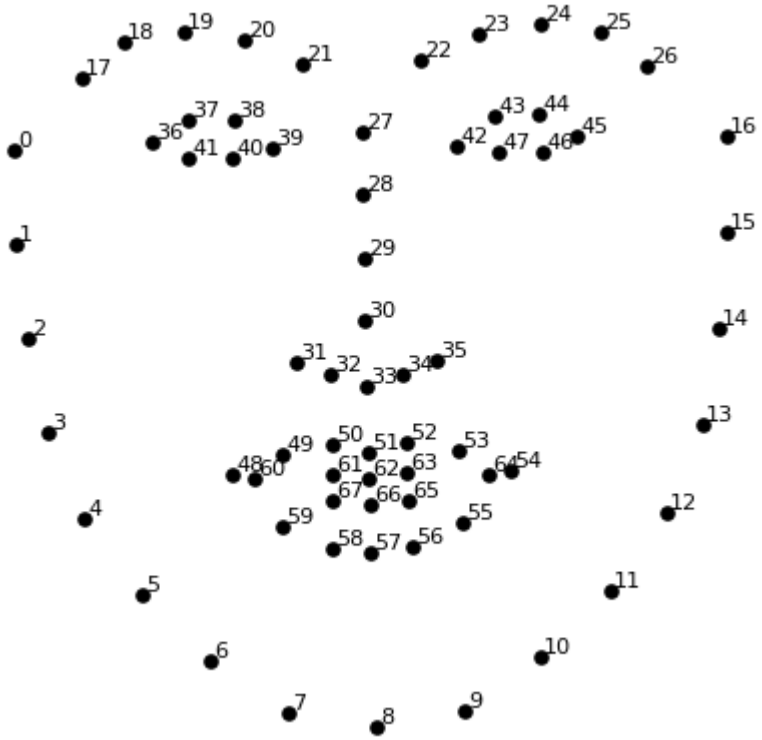
■ To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces

3.2.2 Step2: Posing and projecting Faces

■ We are going to use an algorithm called face landmark estimation

■ The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face—the top of the chin, the outside edge of each eye, the inner edge of each

eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:



- Now that we know were the eyes and mouth are, we'll simply rotate, scale to center the eyes and mouth are in roughly the same position in the image. This will make next step more accuracy.

3.2.3 Step 3: Encoding faces

- Use Triplet neural network
- Model

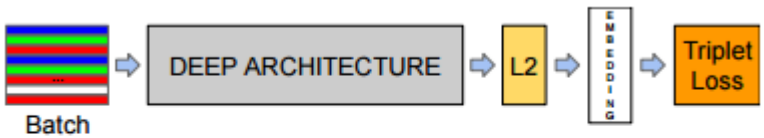
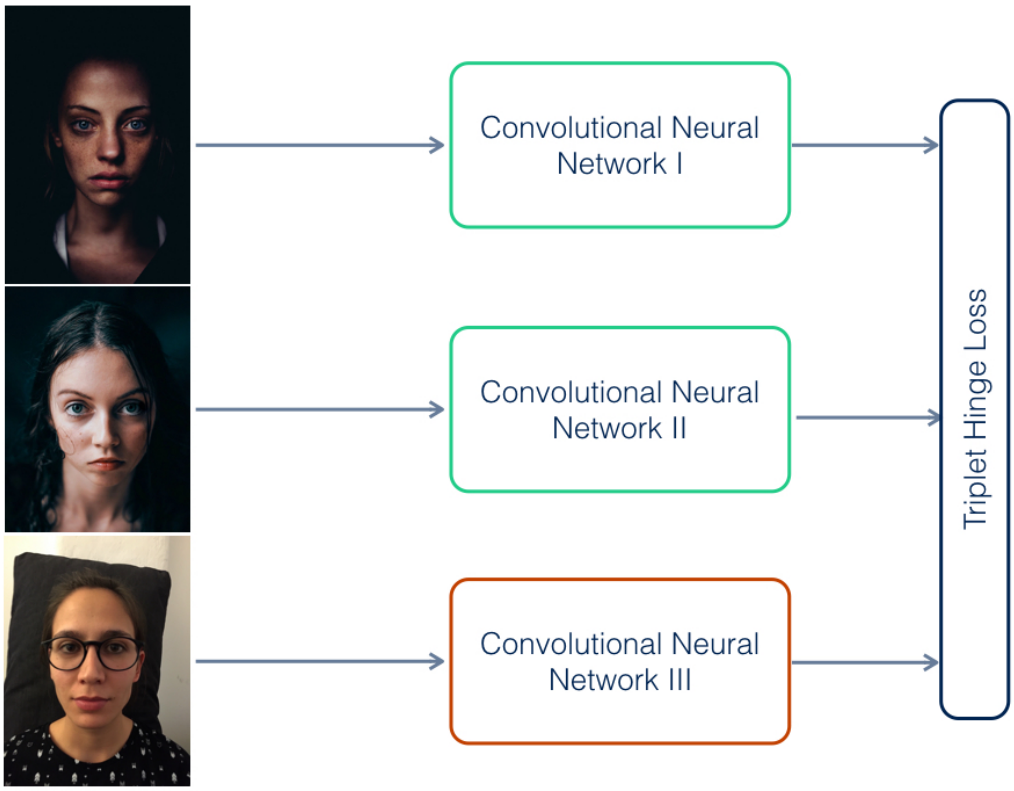


Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

- Triplet loss



3.2.4 Step 4: Finding the person's name from the encoding

- This last step is find the person in our database of known people who has the closest measurements to our test image.
- Can use KNN, SVM classifier.. embeddeding features (128 measurements for each face).
-