CSC420 Project Report

Yubo Wang and Yun Lu

Panorama Face Recognition

Introduction and Related work

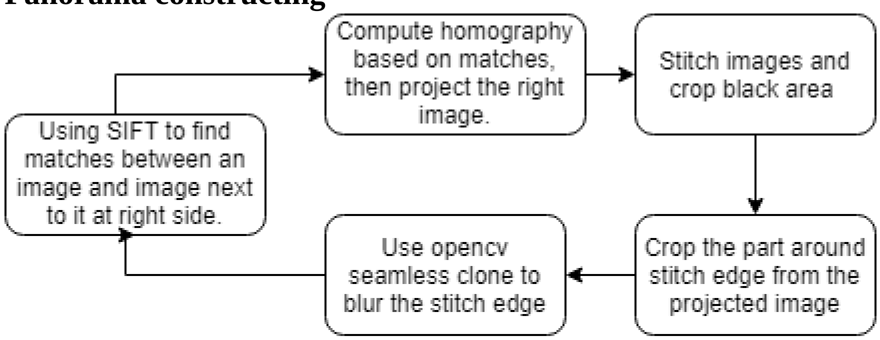
This project aims at recognizing faces in an environment with several people. A panorama picture will be constructed. Gender, age and distance to camera of all recognized faces will be labeled in the image. From my perspective, this is a quality project to show what we have learned through this semester. This project can be used as tools to produce panorama or wide angle selfie with face tag. And with additional hardware help, it can be an assistance software for blind that can tell them who are surrounding them and how far they are.

To accomplish this goal, we need to use SIFT, Homograph computing, image cutting, Geometry computing and neural network.

Methods

The main process is divided into several steps.

1. **Panorama constructing**



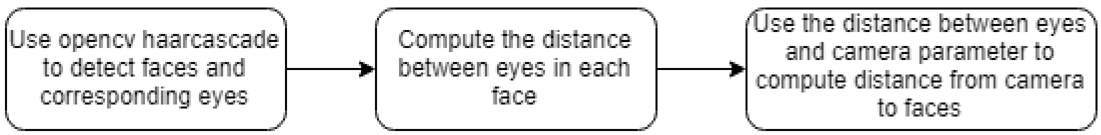
As the above flow chart, the algorithm begins with stitching the middle image with the image right to it.

a. Using SIFT to detect matches
Scale-Invariant Feature Transform is a feature detection algorithm that we use to detect and describe local features in images to find matches between several images. Keypoints recognized by SIFT of an object are extracted from reference images, then they are used to recognize the object in new images by computing the Euclidean distance of their feature vectors.
b. Compute the Homography based on matches
Matches gathered from previous step are filtered by applying a threshold to get quality matches that we can use to compute homogaphy matrix of the projected image. Computing Homography is necessary since in the image shooting process, the camera is rotating instead of moving in an axis, which means the adjacent image need to be projected to the same plane as previous image in order to stitch them together.

c. Crop the black area to have better effect and save time for future computing
d. Blend the stitching result using the crop part of image around the stitch edge.

Repeating the above 4 process for all right side images to get right side stitched result. Then flip the stitched result and all left side images since out algorithm stitch images from left to right. By applying 4 above steps to them, we can get a flipped full panorama. Finally flip the image to get the final result.

2. **Distance computing**



a. Locate faces and corresponding eyes using opencv haarcascade xml file.
b. Compute distance between eyes in each faces.
c. Use the distance between eyes and camera parameter to compute distance from camera to faces

$$\text{Distance to object}(mm) = \frac{f(mm) \times \text{real height}(mm) \times \text{image height}(pixels)}{\text{object height}(pixels) \times \text{sensor height}(mm)}$$

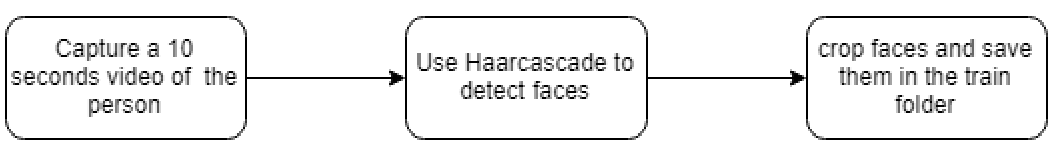
The camera and lens we use to capture images are Canon EOS 80d with 16mm lens. Thereby we can get focal length f = 16 mm, original image size 4000*6000 pixel and sensor size 22.5 x 15 mm. Image size is reduced to 667*1000 to increase compute speed.
Average of distances between eyes is used as real height in the formula.

IPD values (mm) from 2012 Army Survey

Gender	Sample size	Mean	Standard deviation	Minimum	Maximum	Percentile				
						1st	5th	50th	95th	99th
Female	1986	61.7	3.6	51.0	74.5	53.5	55.5	62.0	67.5	70.5
Male	4082	64.0	3.4	53.0	77.0	56.0	58.5	64.0	70.0	72.5

Then we have all values we need to compute the distance between faces and camera.

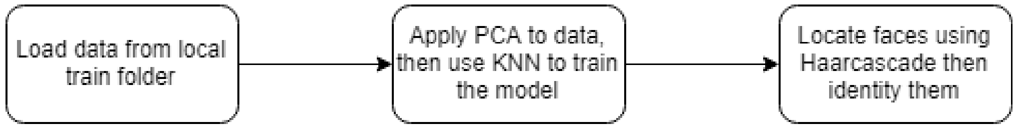
3. **Training set gathering**



a. Use opencv videocapture to get frames from the video
b. Locate faces using haarcascade
c. Crop faces and save them as train images in train folder

Training images should include multiple angles and multiple conditions.

4. Name identifying

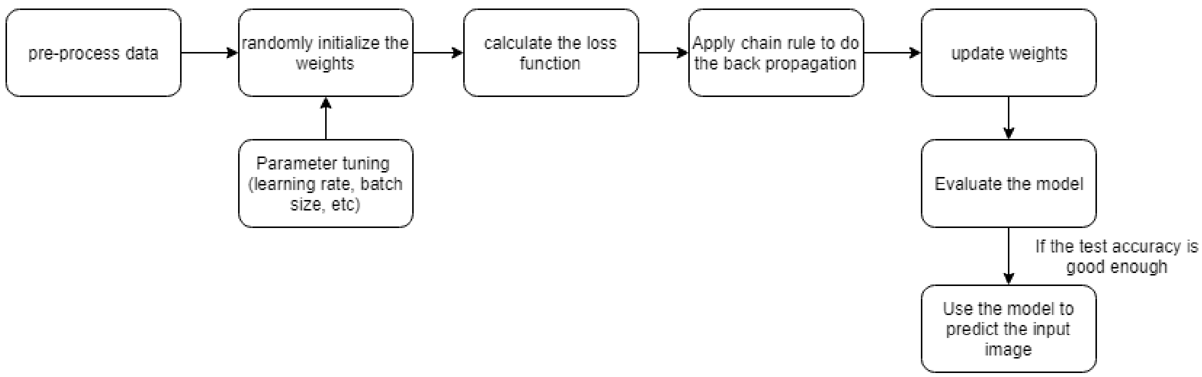


a. Load training images captures from previous step
b. Using PCA to speed up the algorithm, then use KNN to train the model.
   In K-Nearest Neighbors, nearest k training instances to the target instance is recorded. SVM can also be used to replace KNN.
c. Locate faces using Haarcascade, then use the trained model to predict the class. The name of the person will be printed in the picture.

All above functions are implemented by Yubo Wang. Code file panorama.py, compute distance.py, capture dataset.py, detect people.py.

Functions below this are implemented by Yun Lu.

5. Gender and age recognizing



Specific procedure refer to Yun Lu's Report.

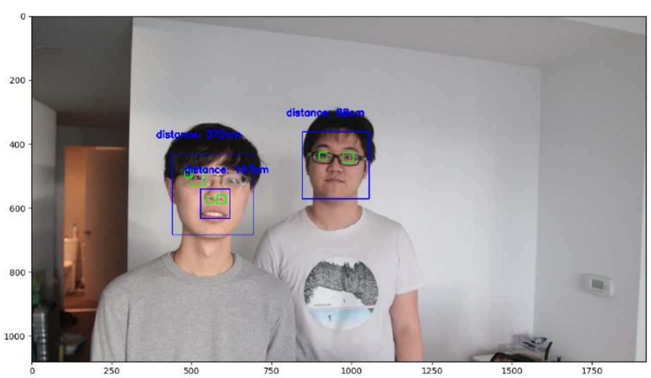
Results and Discussions

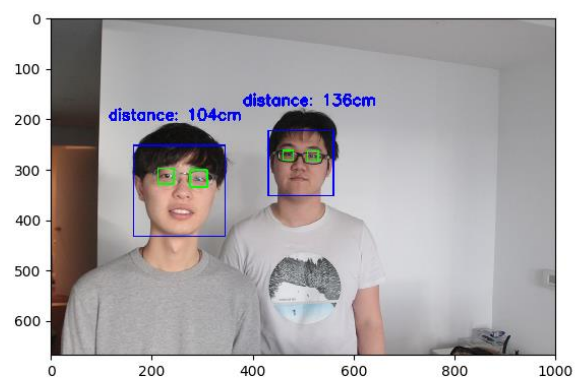
Results for panorama stitching:

The algorithm works as expected. There is a small mismatch at bottom part (monitor) since I use a wide angle lens to capture the picture which cause the distortion of original images. To eliminate this error, we can use a 50mm lens instead of 16mm to capture original images with accurate shape.

In addition, only center part of the panorama picture we created can be used in later steps to recognize face since sides part of the image are projected image. Using other panorama stitching method like spherical or cylindrical instead of perspective may enhance the result.

Results for distance computing:



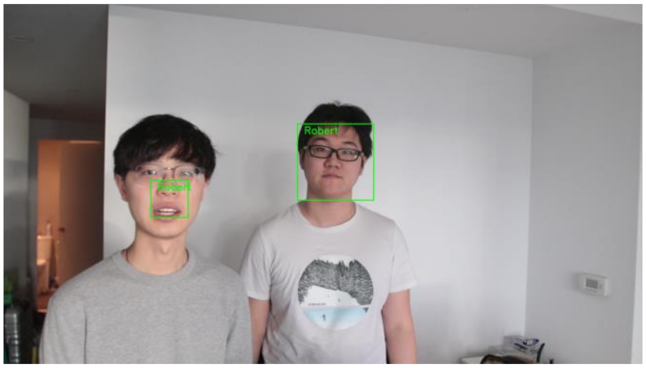


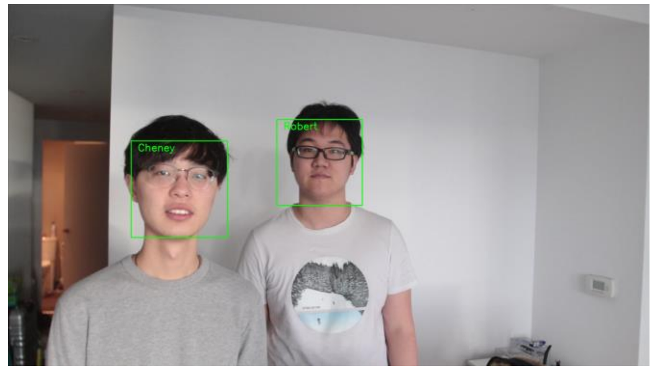
Result before tuning parameter Final Result

At first, the algorithm cannot locate faces and eyes correct. For instance, nostrils, mouths and places near hairs are always detected as eyes. By setting appropriate min size when classify faces, we can recognize faces correct. Then filter all eyes detected in bottom half faces to eliminate wrong detections at nostrils and mouths. Finally pick two detections with most similar size and height as eyes location to compute the distance.

This algorithm will now work if eyes cannot be detected in the face. Sometimes this happens on subjects wearing glass, or they only show side faces in the frame.

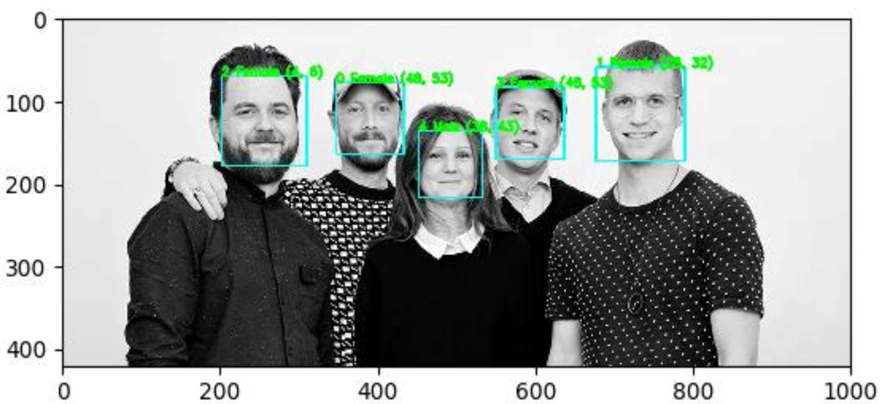
Results for name identify:



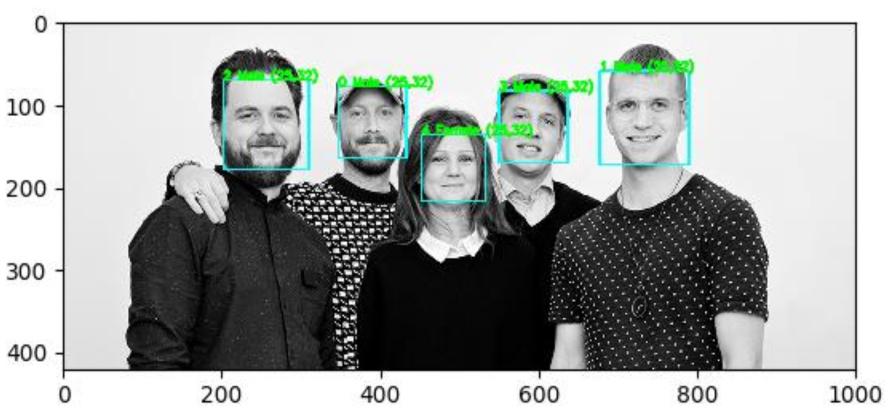


Same as previous step, nose and mouth may be detected as faces without setting min size for face detection. After testing, k=5 for KNN will produce the most accurate result, though it will still mismatch sometimes since two subjects are similar (both are male, wearing glasses, similar hairstyle). The error rate can be reduced by creating better training set. (training sets includes more angles, more light conditions and more facial expressions)

Results for gender and age prediction:



For this step, the accuracy of gender and age prediction improve after tuning training parameter like epoch and learning rate.



Main challenges

The main challenges of this project are parameter tuning and parameter picking in distance computing step. In order to get accurate location of faces and eyes, parameter such as scaleFactor and MinSize in CascadeClassifier need to be tuned. For different pictures, those parameters need to be adjusted every time since ratios of face size compared with image size are different.

Pupillary distance (distance between eyes) is used to compute the distance between faces and cameras. We have tried other distance that can be identified on the face such as single eye size and face length, but both of them did not perform well. For eye size, the result is incorrect since eye size of different people have large standard deviation, thereby mean of eye size cannot be used as real height in the formula to compute distance. For face length, the result is incorrect since face lengths detected in the image are usually not precise. The face detector in haarcascade only consider faces under hairline as faces, thereby it cannot be used as object height in the formula to compute distance.

Conclusion and Future work

Summary:

In this project, I implement panorama stitching, distance computing and name identifying process.

The algorithm can stitch a panorama images from several images taken at same place, then compute distance between faces in the image and camera. It can identify name of faces in the image if we have the training data of that face, and it can predict gender and age of people in the image.

Future work:

This overall quality of this project can be enhanced if panorama and training set capturing process are improved. For panorama stitching, using spherical or cylindrical transform can produce a panorama picture with less distortion at sides. For training set capturing, more angles of faces need to be captured (even side faces) to improve the accuracy of name identify process.

References

SIFT detector

https://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html

Homography computing

https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Harrcascade

https://github.com/Itseez/opencv/tree/master/data/haarcascades

Dataset for Pupillary distance

Gordon, C. C., Blackwell, C. L., Bradtmiller, B., Parham, J. L., Barrientos, P., Paquette, S. P., Corner, B. D., Carosn, J. M., Venezia, J. C., Rockwell, B. M., Murcher, M., & Kristensen, S. (2014). 2012 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics. Technical Report NATICK/15-007. Natick MA: U.S. Army Natick Soldier Research, Development and Engineering Center.