**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

---------------o0o---------------



Capstone project 2

Verification on ID students



Instructor: Prof. Dr. Thuong – Le Tien

Student:

* Nguyen Truong Giang – 1551034
* Ta Quoc Vinh - 1551174

Table of Contents

**I. Introduction2**

1.Overview2

2.Purpose of Project2

**II. Theory3**

1.General Idea3

2.Face detection and alignment3

3 ID Student Detect5

**III. Design and implementation7**

1.Face detection7

2.Face verification8

**IV. Conclusion and future work10**

**References11**

1. **INTRODUCTION**
2. **Overview**

In this report we will make the comparison between the real information and the database specific the real face, ID with the picture in database by using pre-trained model in caffe framework to verify face and MSER method to detect ID student in ID card.

1. **Purpose of the project**

* To detect the ID in ID student card and display.
* To using model and detect face in ID card also in picture that take in real time.
* Verifying the face that taken to display the decision whether that face is the same person or different.
* Matching between student and picture on the ID card.

1. **THEORY**
2. **General Idea**

In this report, we will build a face recognition system. Many general ideal was taken from FaceNet [1] and DeepFace [2]

Face recognition problem commonly fall into two categories:

* Face Verification: “Is this the claimed person?”
* Face Recognition: “Who is this person?”

FaceNet learn a neural network that encodes a face image into a vector. By comparing such two vectors, you can determine if two pictures are of the same person.

Encoding face image into a 1024-dimensional vector

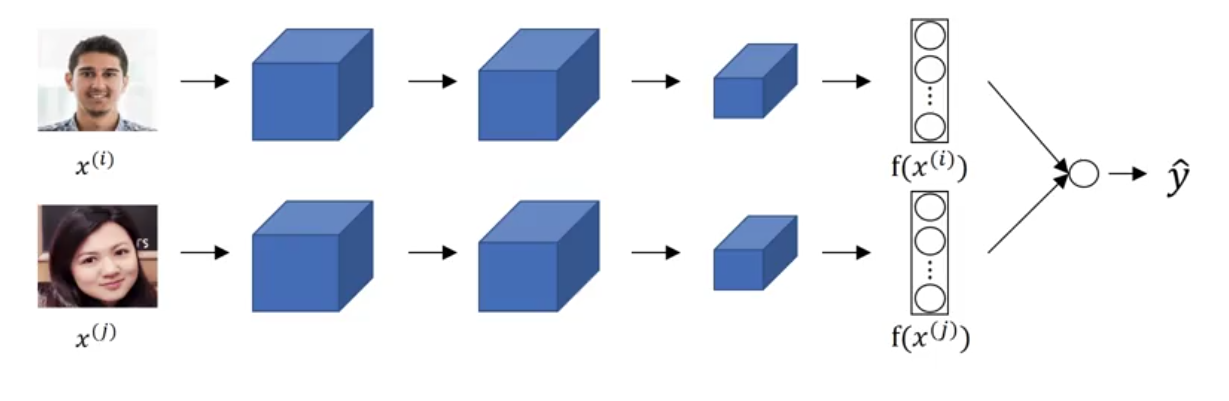


Figure 1 Neural network encode into vector and comparing

Using the FaceNet model, which was taken lots of data and take a long time to train. So in this project, in other to save time we will not train it from scratch. Instead, we load the pre-trained model written in caffe.

By computing a distance between two encoding and thresholding, we can determine if the two pictures represent the same person.

1. **Face detection and alignment**

Using Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks by Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li [3]

Given an image, we initially resize it to different scales to build and image pyramid, which is the input of the following three-stage cascaded framework:

**Stage 1:** We exploit a fully convolutional network, called Proposal Network (P-Net), to obtain the candidate facial windows and their bounding box regression vectors. Then candidates are calibrated based on the estimated bounding box regression vectors. After that, we employ non-maximum suppression (NMS) to merge highly overlapped candidates.

**Stage 2:** All candidates are fed to another CNN, called Refine Network (R-Net), which further rejects a large number of false candidates, performs calibration with bounding box regression, and conducts NMS.

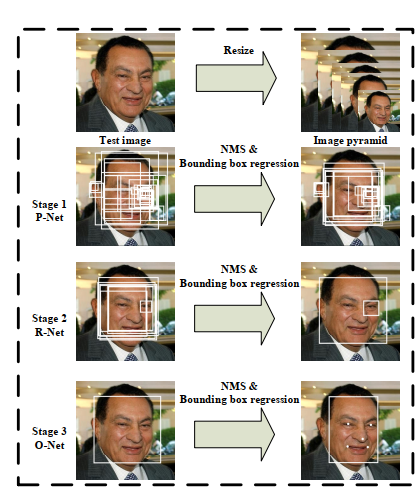


Figure 2 3-Stage model of using CNN to detect face

**Stage 3:** This stage is similar to the second stage, but in this stage we aim to identify face regions with more supervision. In particular, the network will output five facial landmarks’ positions.

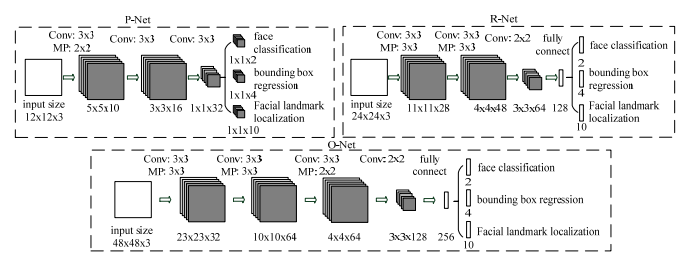


Figure 3 Detail Structure of P-Net, R-Net, O-Net for Face Detection

1. **ID student detect**

Using Automatically Detect and Recognize Text in Natural Images [4] [5] [6]

**Step 1: Detect candidate text regions using MSER** function to find all the regions within the image and plot these results. Notice that there are many non-text regions detected alongside the text.

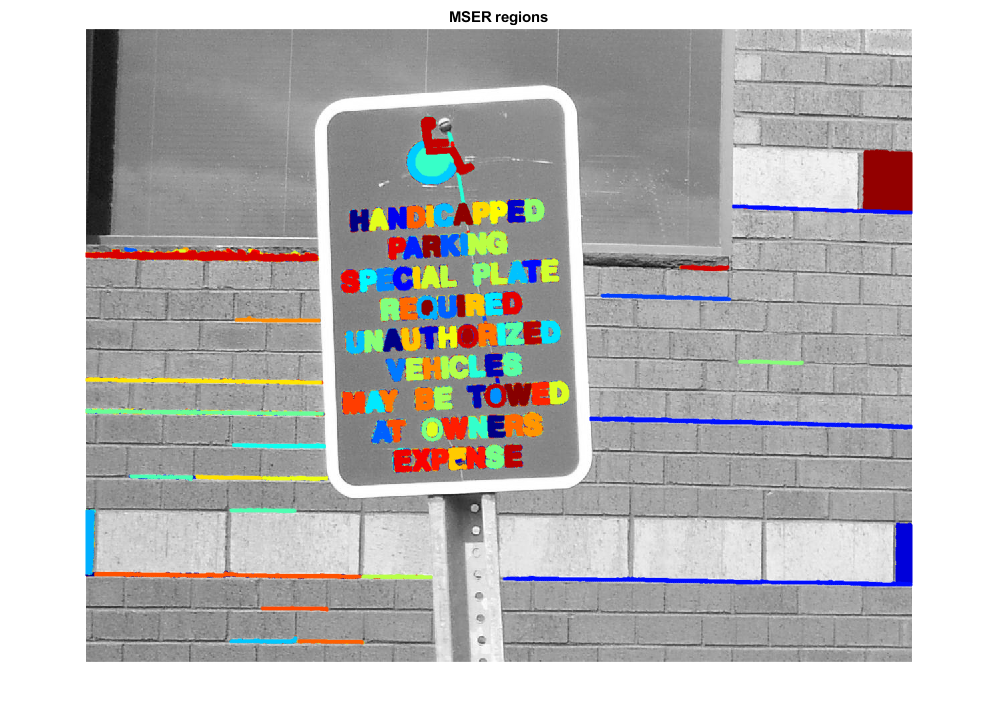


Figure 4 Detect candidate text regions

**Step 2: Remove non-text Regions based on basic geometric properties**

There are several geometric properties that are good for discriminating between text and non-text regions, including:

* Aspect ratio
* Eccentricity
* Euler number
* Extent
* Solidity

Use *regionprops* to measure a few of these properties and then remove regions based on their property values.

**Step 3: Merge text regions for final detection result**

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters {'X','E','T','I'}, where the meaning of the word is lost without the correct ordering.

One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with regionprops. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.



Figure 5 Merge text regions for final detection result.

**Step 4: Recognize detection text using OCR**

After detecting the text regions, use the ocr function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the ocr function would be considerably more noisy.

1. **DESIGN AND IMPLEMENTATION**

* Software requirement:

1. **Face detection**

%% Detect face of people in single picture

% Input: path to image

% Output: 112x96 face align image in same directory

function I = facedetect2(img)

%% mtcnn settings

minSize = 20;

factor = 0.85;

threshold = [0.6 0.7 0.9];

%% caffe settings

modelPath = '../../MTCNNv1/model';

PNet = caffe.Net(fullfile(modelPath, 'det1.prototxt'), ...

fullfile(modelPath, 'det1.caffemodel'), 'test');

RNet = caffe.Net(fullfile(modelPath, 'det2.prototxt'), ...

fullfile(modelPath, 'det2.caffemodel'), 'test');

ONet = caffe.Net(fullfile(modelPath, 'det3.prototxt'), ...

fullfile(modelPath, 'det3.caffemodel'), 'test');

%% Detection process

if size(img, 3)==1

img = repmat(img, [1,1,3]);

end

% detection

[bboxes, landmarks] = detect\_face(img, minSize, PNet, RNet, ONet, threshold, false, factor);

if size(bboxes, 1)>1

% pick the face closed to the center

center = size(img) / 2;

distance = sum(bsxfun(@minus, [mean(bboxes(:, [2, 4]), 2), ...

mean(bboxes(:, [1, 3]), 2)], center(1:2)).^2, 2);

[~, Ix] = min(distance);

facial5point = reshape(landmarks(:, Ix), [5, 2]);

elseif size(bboxes, 1)==1

facial5point = reshape(landmarks, [5, 2]);

else

facial5point = [];

end

%% alignment settings

imgSize = [112, 96];

coord5point = [30.2946, 51.6963;

65.5318, 51.5014;

48.0252, 71.7366;

33.5493, 92.3655;

62.7299, 92.2041];

facial5point = double(facial5point);

% load and crop image

transf = cp2tform(facial5point, coord5point, 'similarity');

cropImg = imtransform(img, transf, 'XData', [1 imgSize(2)],...

'YData', [1 imgSize(1)], 'Size', imgSize);

I = cropImg;

end

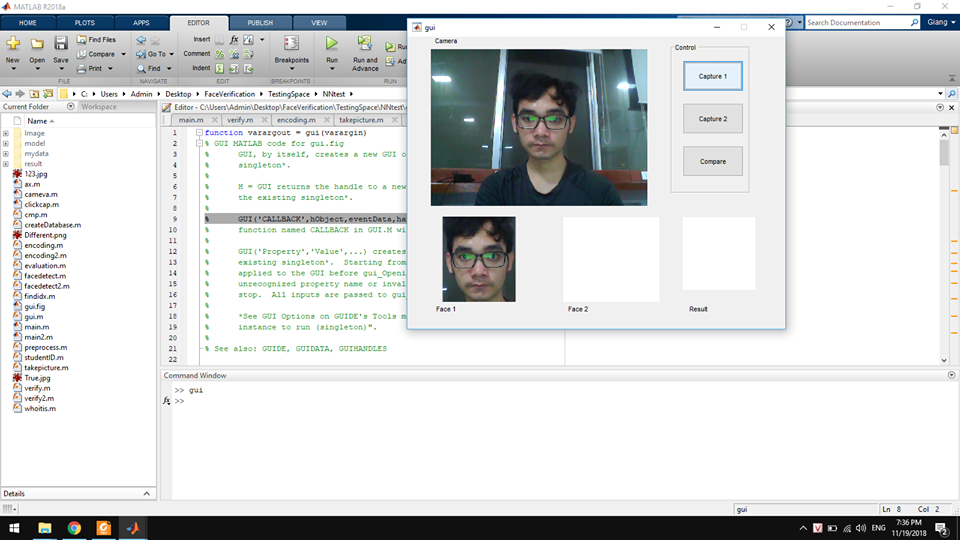


Figure 6 Face detection demonstration

1. **Face Verification**

%% Check if 2 112x96 image is the same person

% Input: 2 image and 2 id

% net pre-trained model

% Output: Correct or Incorrect

function cmp = verify2(path1,id1,path2,id2,net)

%% Forward Propergationn 2 image in 2 path to get 2 feature matrices

feature1 = encoding(path1,net);

feature2 = encoding(path2,net);

%% Compute distance between 2 feature

dist = norm(feature1 - feature2)/100;

%% Return True if having same id and distance between 2 image < 0.4

if id1==id2

if dist<0.4

cmp = 1;

disp('SAME');

else

cmp = 0;

disp('Diffferent');

end

else

cmp = -1;

disp('Person not in database');

end

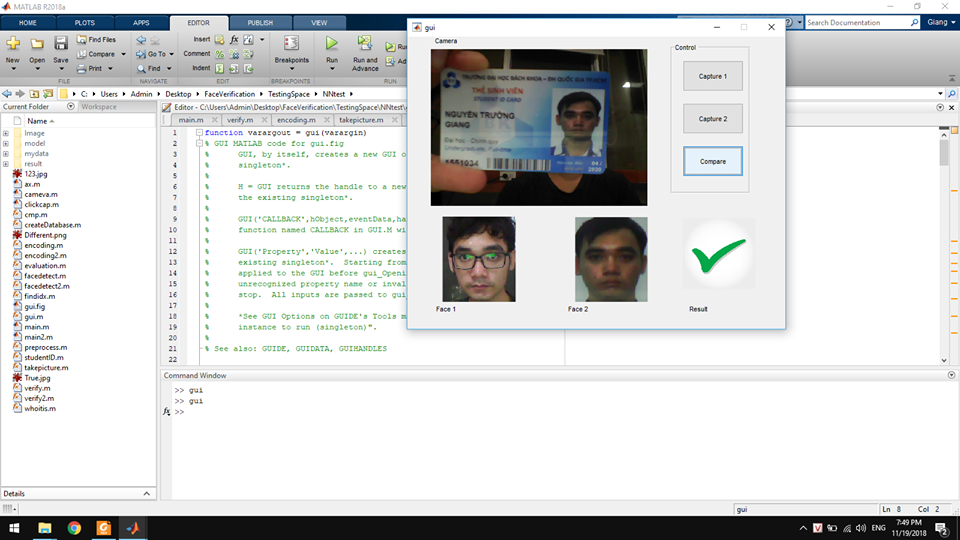


Figure 7 Face verification of Positive (Left) and Negative (Right) Test

1. **CONCLUSION AND FUTURE WORK**

* By using pre-trained model, we are able to detect face and alignment for in the pre processing step. Besides, during experience, we briefly understand about caffe model and our future work. By using Graphic User Interface, we designed a system that matches two faces: on ID card and the student himself.
* In future work, we consider that changing our system into Face Verification for Organizations by design a database of staffs and wireless connection through Microcomputer.

# References

|  |  |
| --- | --- |
| [1] | Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Yu Qiao, [Online]. Available: https://arxiv.org/abs/1604.02878. |
| [2] | Yu Qiao, Kaipeng Zhang, [Online]. Available: https://github.com/kpzhang93/MTCNN\_face\_detection\_alignment. |
| [3] | K. Zhang, "arxiv.org," [Online]. Available: https://arxiv.org/abs/1604.02878 . |
| [4] | G. Alvaro, "Text location in complex images," in *2012 21st International Conference on. IEEE*, 2012. |
| [5] | "mathworks.com," [Online]. Available: https://www.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html?s\_tid=srchtitle. |
| [6] | C. Huizhong, "Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions," in *2011 18th IEEE International Conference on. IEEE*. |