International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.1325-1330

Epitomic Analysis For Face Recognition In Machine Learning

Y. Angel Blessy[1],
Dr. G. Heren Chellam[2]
[1]Research Scholar, Register Number: 17221172162022,
Rani Anna Goverrnment College for Women, Affiliated to Manonmaniam Sundaranar University,
Abishekapatti, Tirunelveli - 627 012, Tamilnadu, India.
[2]Assistant Professor in Computer Science,
Department of Computer Science, Rani Anna Government College for Women,
Affiliated to Manonmaniam Sundaranar University, Abishekapatti,
Tirunelveli - 627 012, Tamilnadu, India.

Abstract

Comparision identical twins using their face images is a challenge in biometrics. Face detection and comparison is one of the most common place programs in Artificial intelligence. Human beings carry out face recognition automatically each day and practically without a effort. Eigenface Recognizer set of rules considers the truth that not all components of a face are equally essential and equally useful. When we take a look at someone we recognize the character his distinct functions like eyes, nose, cheeks, brow and how they range with recognize to each other. The idea is to surely consciousness on the areas of maximum exchange of the face. For example, from eyes to nostril there is a sizable exchange and the identical is the case from nostril to mouth. This work compare more than one faces you compare them with the aid of looking at these components of the faces because these parts are the most useful and important components of a face. Important because they seize the maximum exchange amongst faces, exchange this facilitates you differentiate one face from the other.

Keywords*: Machine Learning, Artificial Intelligence, Eigenface Recognizer, face recognition systems ,Matching Algorithms, Twins Identification.*

1. INTRODUCTION

Machine learning is an area of Artificial Intelligence (AI) that permits computer systems to 'analyze'. Traditionally, we constantly got computers to do things by means of providing it a strict set of instructions called laptop programs. Machine Learning makes use of a completely different approach. Instead of giving the computer a set of commands on the way to do something, we supply it commands on a way to discover ways to do something. We do that by giving it statistics and programming it to apply numerous mathematical and statistical models that make experience of the data and learn to make decisions primarily based on that. Think of a gadget that can classify facial pixels of different human beings. Instead of manually locating unique visual traits and patterns from photos of these faces after which coding it up, you can program the laptop to take in pixels of these faces and find visual patterns[1][4] and variations between faces of different human beings all via itself. That can be done the use of a range of different algorithms. The concept right here is that the pc 'learns' by using itself rather than being specifically programmed to do a certain challenge (in this case, classifying facial photos of various humans). The system of teaching the laptop (Giving it data to analyze from) is known as training. EigenFace Recognizer appears at all the education photos of all of the men and women as a whole and attempts to extract the components which are essential and useful (the components that capture the most variance/change) and discards the relaxation of the components. These critical components it extracts are called principal additives. EigenFaces Face Recognizer[3] trains itself (with the aid of extracting principal additives). Another point to be considered is that it also maintains a report of which principal issue belongs to which person. Face Recognition using Python and OpenCV follows a well described

ISSN: 2233-7857 IJFGCN

pattern[6][2]. When you meet someone for the primary time to your life, you have a look at his/her face, eyes, nose, mouth, color, and overall capabilities. This is your mind studying or schooling for the face reputation of that person with the aid of amassing face facts. Although it sounds like a totally simple undertaking for us, it has established to be a complex challenge for a pc, as it has many variables that may impair the accuracy of the methods, for example: illumination variation, low resolution, and occlusion, amongst other. In laptop science, face reputation[5] is essentially the task of recognizing someone primarily based on its facial photo. It has end up very popular in the remaining two decades, mainly because of the new strategies developed and the high first-class of the cutting-edge videos/cameras.

2. MATERIALS AND METHODS

Data collection

The data's are carried out experiments by using python on images of WIDGER database and our own Face Database of 100 images. We carried out experiments by applying our image recognition face method on the images in the database.

3. METHODOLOGY

INPUT

Every Machine Learning set of rules takes a dataset as input and learns from these records. The algorithm is going through the information and identifies patterns inside the facts. For instance, assume we desire to become aware of whose face is present in a given photograph.

OUTPUT

Similar faces have similar dimensions. The difficult element is to convert a specific face into numbers – Machine learning algorithms most effective apprehend numbers. This numerical illustration of a "face" (or an element inside the schooling set) is termed as a function vector. A function vector contains of diverse numbers in a selected order.

PROCESSING

Face_recognition Library

face_recognition library in Python can perform a huge wide variety of tasks:

•Locate faces in the image
•Find and manipulate facial functions in an picture
•Use face_encodings to go back 128 D facial Vector of face image
•Return face_distance
•Prints the message twins or not

Compare_faces

face_recognition.api.compare_faces(known_face_encodings, face_encoding_to_check, toler
ance=0.5)

Compare a list of face encodings against a candidate encoding to see if they match. Returns A list of True/False values indicating which known_face_encodings match the face encoding to check.

Face_Distance

face_recognition.api.face_distance(face_encodings, face_to_compare)

Given a list of face encodings, compare them to a known face encoding and get a euclidean distance for each comparison face. The distance tells you how similar the faces are.

Face_Encodings

face_recognition.api.face_encodings(face_image, known_face_locations=None,
num_jitters=1, model='small')

Given an image, return the 128-dimension face encoding for each face in the image.

ISSN: 2233-7857 IJFGCN

4. RESULTS AND DISCUSSION

However, a facial reputation library called face_recognition is much simpler to put in and use. We use this
library to get face encodings of each faces and compare the encodings of each faces to inform us if they're twins.

These are the steps:

1.Load the pictures of persons
2.Find the face encodings of two persons
3.Compare the face encodings
4.Print if they're twins or not.

As a easy example, we will map a “face” into a feature vector which can comprise numerous capabilities
like:

•Height of face (cm)
•Width of face (cm)
•Average color of face (R, G, B)
•Width of lips (cm)
•Height of nose (cm)

Table 1
Map out various features and convert it into a characteristic vector

Height of face (cm)	Width of face (cm)	Average color of face (RGB)	Width of lips (cm)	Height of nose (cm)
23.1	15.8	(255,224,189)	5.2	4.4
22.9	16.8	(255,232,175)	4.9	4.2
19.5	15.9	(254,219,168)	4.2	4.0
24.2	17.1	(239,240,180)	5.1	3.9
25.3	14.5	(253,239,152)	3.8	4.8
18.7	19.3	(250,249,148)	4.7	4.5

So, our photo is now a vector that would be represented as (23.1, 15.8, 255, 224, 189, 5.2, 4.4). Of direction there could be limitless other capabilities that would be derived from the photograph (for instance, hair color, facial hair, spectacles, etc). However, for the example, allow us to don't forget just these five simple functions. Now, once we have encoded every picture into a feature vector, the problem turns into lots vectors derived will be pretty similar. Put it the opposite way, the “distance” between the 2 function vectors will be quite small. Machine Learning can help us here with 2 things:

1.Deriving the characteristic vector: it is tough to manually listing down all of the features because there are just so many. A Machine Learning algorithm can intelligently label out lots of such functions. For instance, complex capabilities ought to be: ratio of peak of nose and width of forehead. Now it's going to be quite difficult for a human to list down all such “2nd order” capabilities.
2.Matching algorithms: Once the feature vectors had been obtained, a Machine Learning set of rules wishes to suit a new image with the set of characteristic vectors present in the corpus.
Now that we've got a basic knowledge of the way Face Recognition works, allow us to construct our very own Face Recognition algorithm using some of the well-known Python libraries.

I found 1 face(s) in this photograph.
A face is located at pixel location Top: 46, Left: 87,Bottom: 108,Right: 149
Face Landmarks: [{'chin': [(92, 64), (93, 72), (94, 79), (96, 86), (98, 93), (103, 99), (108, 104), (114, 108), (121, 109), (128, 108), (133, 103), (138, 97), (142, 91), (145, 84), (146, 76), (146, 69), (145, 61)],

ISSN: 2233-7857 IJFGCN

'left_eyebrow': [(96, 60), (99, 57), (104, 57), (108, 58), (113, 59)], 'right_eyebrow': [(125, 58), (129, 56), (133, 54), (138, 54), (141, 56)], 'nose_bridge': [(119, 63), (119, 69), (120, 74), (120, 79)], 'nose_tip': [(114, 81), (117, 82), (120, 83), (123, 82), (126, 80)], 'left_eye': [(102, 64), (105, 62), (108, 62), (111, 65), (108, 65), (105, 65)], 'right_eye': [(126, 64), (129, 61), (133, 61), (136, 62), (133, 64), (130, 64)], 'top_lip': [(106, 86), (111, 86), (116, 86), (120, 87), (124, 86), (130, 86), (135, 86), (133, 87), (124, 88), (120, 88), (116, 88),
(107, 87)], 'bottom_lip': [(135, 86), (130, 93), (124, 96), (120, 96), (116, 96), (110, 93), (106, 86), (107, 87), (116, 93), (120, 93), (124, 93), (133, 87)]}]

Image Encoding: [-0.10918969 0.05871312 0.0408542 -0.06019697 -0.10294548 0.01360412 0.08500479 -0.09815396 0.15060733 -0.06221908 0.24178138 -0.07154021 -0.39367965 -0.0237739 0.00304269 0.20419566 -0.17272574 -0.17633975 -0.13117629 -0.04831306 0.0281482 -0.03804543 0.09490065 0.10478658 -0.10058157 -0.34221318 -0.02140308 -0.1206454 0.06677216 -0.03786279 0.00894486 0.048531 -0.24902867 -0.02675017 0.047709 0.06537563 -0.13443364 -0.1782544 0.26810068 - 0.02364166 -0.25852883 -0.14864774 0.08606881 0.1555845 0.18748291 -0.00956187 - 0.00118036 -0.06227811 0.16787404 -0.37732667 0.08700009 0.14873056 0.07409905 0.10313466 0.03445666 -0.19078815 0.06482672 0.17286412 -0.29035679 0.03791903 0.07335825 -0.08366778 - 0.03197778 -0.07987785 0.18667535 0.10712483 -0.16231443 -0.11667757 0.22332871 -0.17329119 0.03906258 0.15775968 -0.11305131 -0.24493803 -0.24496554 0.02567299 0.4531489 0.13643602 - 0.0458652 -0.04578734 -0.08630373 -0.05609186 -0.00838592 0.05721835 -0.04352496 -0.06519517 - 0.11014339 0.00696597 0.21505132 -0.04698859 -0.04316361 0.30890778 -0.02543232 -0.00635569 0.03206065 0.12059457 -0.09148124 -0.02700478 -0.16144527 0.00739614 -0.07139891 -0.12586801 - 0.03315173 0.02262232 -0.1410794 0.08979748 -0.01184114 -0.00229226 -0.03227814 -0.08939727 - 0.18242672 -0.05710357 0.12462614 -0.27632198 0.20639135 0.15978315 0.09406153 0.11583151 0.06681114 0.03583489 0.05557906 -0.13444494 -0.18923435 -0.0889639 0.13226181 -0.03040397 0.0062706 0.00938414]

I found 1 face(s) in this photograph.
A face is located at pixel location Top: 39, Left: 93, Bottom: 101, Right: 156
Face Landmarks: [{'chin': [(96, 59), (97, 66), (99, 73), (100, 80), (102, 86), (106, 92), (111, 96), (117, 100), (124, 102), (130, 101), (136, 97), (141, 93), (144, 87), (146, 81), (147, 74), (148, 67), (149, 60)], 'left_eyebrow': [(102, 55), (105, 53), (110, 52), (114, 53), (119, 55)], 'right_eyebrow': [(130, 54), (134, 52), (138, 51), (142, 51), (145, 54)], 'nose_bridge': [(124, 60), (125, 64), (125, 69), (125, 73)], 'nose_tip': [(119, 76), (121, 77), (124, 78), (127, 77), (130, 76)], 'left_eye': [(108, 60), (110, 58), (114, 58), (117, 61), (114, 62), (110, 62)], 'right_eye': [(130, 61), (134, 58), (137, 58), (140, 60), (138, 62), (134, 62)], 'top_lip': [(113, 83), (117, 82), (121, 82), (124, 83), (126, 83), (130, 84), (134, 85), (132, 85), (126, 85), (124, 85), (121, 85), (115, 83)], 'bottom_lip': [(134, 85), (130, 88), (126, 88), (123, 88), (121, 88), (117, 86), (113, 83), (115, 83),
(121, 84), (123, 85), (126, 85), (132, 85)]}]

Image Encoding: [-0.07398597 -0.02009123 0.03823597 0.00421214 -0.16739346 -0.00465336 0.05099295 -0.11636978 0.21984862 -0.11650382 0.18676609 -0.09976622 -0.31520846 0.02302535 - 0.02383655 0.18998617 -0.1915966 -0.17466778 -0.13230452 -0.0788852 0.05575123 -0.04191837

ISSN: 2233-7857 IJFGCN

0.12022785 0.09953832 -0.11492452 -0.35946226 -0.06240363 -0.14119582 0.08151479 -0.05876678 -0.00587528 0.09118646 -0.16144346 -0.04161035 0.07209611 0.02322891 -0.08829007 -0.19806099 0.2862688 -0.05998234 -0.25691274 -0.07940799 0.02804109 0.13820174 0.15445398 0.02681155 0.00325723 -0.09317183 0.18497521 -0.36054644 0.0610332 0.19438219 0.06665215 0.07553193 0.06063908 -0.2089352 0.05173872 0.15292831 -0.28374258 0.04443013 0.04660149 -0.04862349 0.00168553 -0.07528736 0.16528164 0.14989132 -0.14543183 -0.10086985 0.16047226 -0.16724789 0.07742374 0.15431719 -0.17771418 -0.2992855 -0.17489854 -0.02972459 0.5332737 0.15078923 -0.09255248 -0.06977516 -0.08132343 -0.01285464 0.00617093 0.08526745 -0.06057783 -0.06507812 -0.09143516 0.02621652 0.14758776 -0.07814189 -0.0943881 0.29901218 0.00668417 0.00245914 0.02540283 0.09898197 -0.08960936 -0.03259089 -0.126661 -0.02159739 -0.0214469 -0.04106335 -0.00936818 -0.01305922 -0.10830653 0.10681744 -0.05578717 -0.02082123 -0.0671987 -0.07864486 -0.1461345 -0.01337939 0.14929308 -0.28774139 0.16395988 0.17061138 0.05216811 0.19987752 -0.01572407 0.04891727 0.03100496 -0.09952989 -0.12566552 -0.07192289 0.10817552 0.01097944 0.01761024 -0.02141884]

Euclidean Distance: [0.42192085]

[True]

They are twins

Percentage of matching accuracy: 86.91%

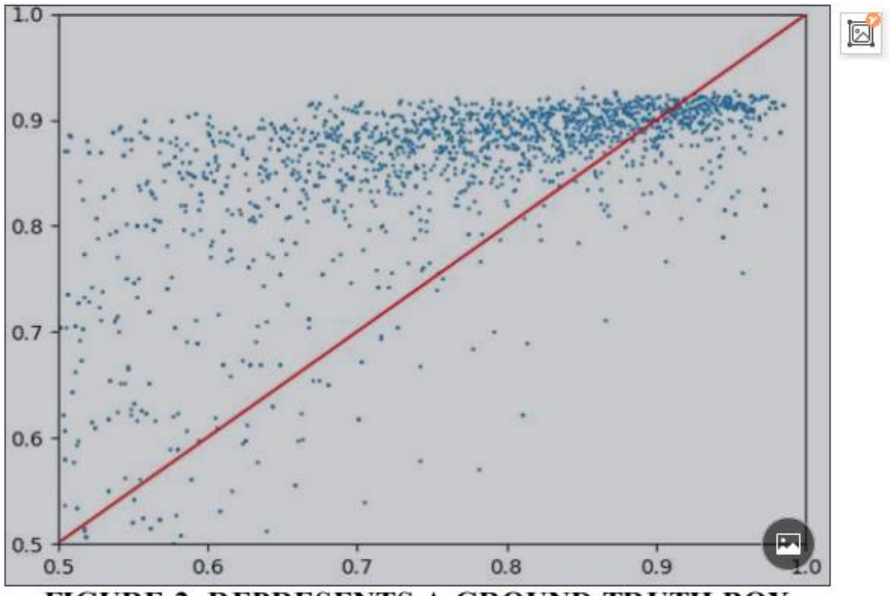

FIGURE 2: REPRESENTS A GROUND TRUTH BOX

Though it's miles still hard for a model to learn how to be expecting among a region and a floor truth, we do see that there is a more potent correlation between localization confidences. Here this correlation is visualized by the slope of the data points better equivalent to the crimson line (the unit line y=x) than in the earlier graph plotting classification self belief. Given this stronger correlation we assume the model to perform higher when using localization self belief as opposed to classification self assurance.

5. CONCLUSION

The built in library face recognition has been used to identify the given two human images are twins or not. We have seen how face recognition has followed the same transition as many other computer vision applications. Traditional methods based on hand-engineered features that provided state-of-the-art accuracy only a few years ago have been replaced by deep learning methods based on CNNs. Indeed, face recognition systems based on CNNs have become the standard due to the significant

ISSN: 2233-7857 IJFGCN

accuracy improvement achieved over other types of methods. Moreover, it is straightforward to scale-up these systems to achieve even higher accuracy by setting the value of face distance or Euclidean distance between two faces to 0.5. We normalize the value of Euclidean distance and calculate the percentage of accuracy between facial images.

REFERENCES

[1] Lee, C.C., Mower, E., Busso, C., Lee, S., Narayanan, S.: Emotion recognition using a hierarchical binary decision tree approach. Speech Commun. 53(9), 1162–1171 (2011)
[2] Lopes, A.T., de Aguiar, E., De Souza, A.F., Oliveira-Santos, T.: Facial expression recognition with Convolutional Neural Networks: coping with few data and the training sample order Pattern Recogn. 61, 610–628 (2017)
[3] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep Convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
[4] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 6, pp. 1113–1133, 2015.

[5] T. Zhang,"Facial expression recognition based on deep learning: A Survey" in International Conference on Intelligent Systems and Applications, Springer 2017, pp. 345-352 [6] Donato, G., Bartlett, M. S., Hager, J. C., Ekman, P., & Sejnowski, T. J. (1999). Classifying facial actions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(10), 974-989.

[7] Pantic, M., & Rothkrantz, L. J. (2000). Automatic analysis of facial expressions: The state of the art. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(12), 1424-1445.
[8] Samal, A., & Iyengar, P. A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern recognition,25(1), 65-77.
[9] Kim, B. K., Lee, H., Roh, J., & Lee, S. Y. (2015, November). Hierarchical committee of deep CNNs with exponentially-weighted decision fusion for static facial expression recognition. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (pp. 427-434). ACM.
[10] Pss, S.; Bhaskar, M. RFID and pose invariant face verification based automated classroom attendance system.In Proceedings of the 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, India, 23–25 January 2016; pp. 1–6.

ISSN: 2233-7857 IJFGCN