Historical Face Comparison Assisted by Fuzzy Clustering

Runze Tang

Research School of Computer Science, Australian National University
u7102270@anu.edu.au

Abstract. Photographs are important to symbolize an individual and repose emotion. However, individuals in many historic photographs cannot be identified by human due to historic reason [1]. Hence, a face recognition model is needed to compare the unknown person's image with another one in which we know who the person is. Modern face recognition has high accuracy but training it needs many face images for one person [2] while historic photographs are rare. Hence, a model aiming at face recognition with few images are needed. This paper uses C-Means [3], a fuzzy clustering method to do clustering on the training set to get a set of centroids and calculate each faces' degree of membership as guidance. This paper proposes a model to compress the coordinates of key points in a face image together with the degree of membership to each cluster to an embedding vector on a high dimension unit hypersphere and use Euclidean distance to compute face similarity. Network Pruning based on angle of activation vector [7] is also implemented to reduce the complexity of the model. Since not all faces are in the same pose and position, a unification based on affine transformation is implemented on the dataset first. The result is that the accuracy of the proposed model can reach more than 90%, which is better than the accuracy in [1], which is using the distance of key points as input to avoid the effect of different face pose and position.

Keywords: Historical Face Recognition · C-Means · Network Pruning.

1 Introduction

[1] proposed to sample 14 key points in human's face like canthus, which is edge of an eye, nasion, which is root of nose or alare, which is wing of nose to identify whether two faces belongs to the same person. The first and third images on the right of Fig. 1 show the key points marked on one's face as yellow dots. Key points like those on ears are not taken into consideration because occlusion occurs in some photographs. [1] computed the distance of each two of these key points, 91 distances are selected in total, to avoid the effect of different pose and position of the face. The difference is that in this paper, the raw coordinate of key points after unification are used as input to a neural network. It has been proved theoretically [4] and practically [1] proved that key points in a face can be a signature to identify face. The reason why use key points in face to do comparison rather than the whole image is that there is no sufficient face images to train a model to distinguish the face and complex background pattern. For example, the only two images of a man could be different on beard, which may interfere the prediction. Another reason is that since the dataset contains only 36 faces, it is hard to train without overfitting a complex model with numerous parameters. Although there are many data augmentation method to enlarge the dataset [5], after the affine transformation to align each face, only few methods can be used like flipping, noise injection or color transformation.

The task of this paper is to generate an embedding vector by the input to represent a person's face. This is basically a compression model, which makes face recognition efficient. If using a whole model to compare a new photograph with images in a database, for each pair of key points set the model is run to compare. But if what stored in a database is the precalculated embedding vectors, the compression model only needs to run once to extract embedding vector, and then calculate the distance between the new face and those in the database, which is more efficient. Using the distance between two faces' embedding vectors to represent how much they are similar is also like how human thinks. In addition, the embedding technique has been using not only in face recognition [6], it has also been using in many other areas especially in multi-modality tasks like visual question answering to align visual and textual input [8, 9]. To solve the problem that accuracy fluctuate caused by scarcity of images for one person, a fuzzy clustering method C-Means are used to generate the degree of membership of each cluster as a guidance to stabilize the performance, the detail of which will be clarified in section 2.3. The result shows that this method can handle face recognition in historic photographs much better than the model in [1].

2 Method

In this section, five main parts of the method and the result are discussed in detail. First, coordinates of key points in the dataset are unified to avoid the following step being affected by the pose and position of faces in

images. Second, pros and cons of two dataset splitting method are discussed. After that, C-Means are used to calculate degree of membership to each cluster as a guidance, which will be a part of the input data together with the coordinates of key points. Then, a network to extract the embedding are applied and Euclidean distance are calculated to describe how similar two faces are. At last, pruning technique mentioned in [7] are applied to analyze its effect of reducing the complexity of the model.

2.1 Dataset error repairing and key points unification

In the dataset, not every photograph are taken as a passport photo. As sown in the first and third images on the left of Fig. 1, not everyone is facing at exactly where the camera is. Not every face are at the exact center of the image, instead, some are at lower left, some top right. [1] used distances between each pair of key points to avoid effect caused by what described above. In this paper, affine transformation are applied to make sure coordinates of key points of every face are at the same position and face the camera. But first, damaged data needs to be repaired.

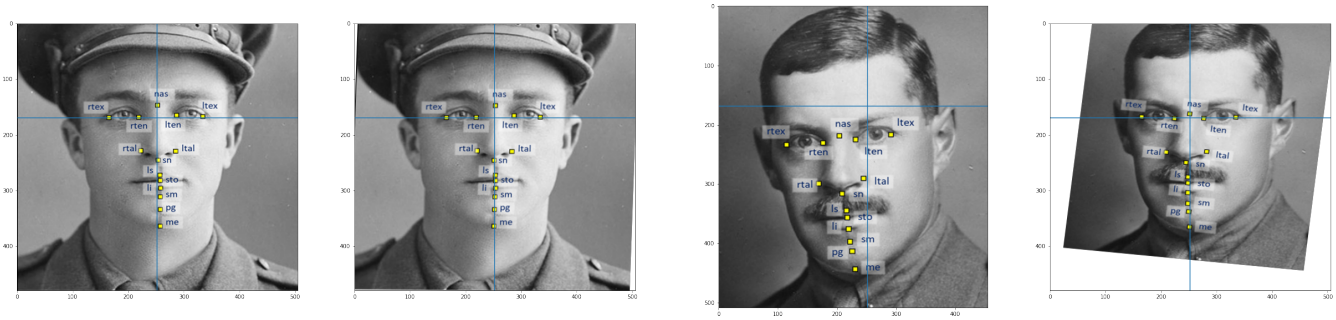

Fig. 1. The first and third images on the left are original images. The second and fourth on the left are images after affine transformation. The blue line in the images are at the same position.

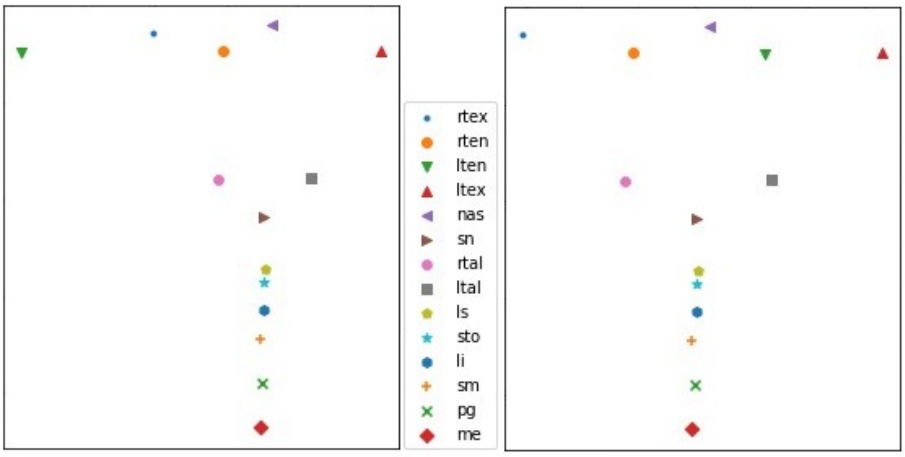

Fig. 2. Different marker marks different key points on the face like the yellow dots in Fig. 1. The right one is the correct data after repairing. The green inverted triangle, which represent the inner edge of left eye, in the top left of the left figure locates at the wrong position.

Since the key points on a face is recognized by another model, position of some key points are wrong. One example is shown as Fig. 2. Different markers represent different key points on a face like the yellow dots on the faces in Fig. 1. The marker to key points map is listed in the legend at the middle. The green inverted triangle, which represent the inner edge of left eye, at the top left of the left figure locates at the wrong position, which is fixed by manually pick point in the original image. The correct one after repairing is the right figure of Fig. 2, in

which the green inverted triangle locates at the correct position. There are 6 mistakes like the one described above in the dataset in total. Each of them is repaired by manually pick correct point in the original image to replace the coordinate of the wrong one. Since the dataset has 36 faces in total, 6 outliers will interfere the training. Hence, they must be corrected.

Three key points at two outer corners of two eyes and the one on the chin are used to generate an affine transformation to transform faces to the same position. After the affine transformation, the three key points will locate at $(20, 20)$, $(120, 20)$ and $(70, 140)$. This process can align each face to the same position to reduce the effect of different pose of the person and the different position of faces in the image when taken the photographs.

One thing have to mention is that transform the coordinate into range 0-1 is also tested but was abandoned. More specifically, after this 0-1 range affine transformation, three key points mentioned above will locate at $(\frac{20}{140}, \frac{20}{140})$, $(\frac{120}{140}, \frac{20}{140})$ and $(\frac{70}{140}, \frac{140}{140})$. Since these three key points are the outer points of all key points, other key points' coordinates will be in range 0-1. After many tests, it is shown that model trained on data after this 0-1 range affine transformation has a similar performance as the finally used unification method describe above, but the accuracy dropped a lot after redo the C-Means on the split dataset, detail of which process will be interpreted in section 2.3.

2.2 Dataset splitting

In the dataset of [1], all 36 faces are formed into 12 groups. Each group has three faces, relation among them is listed in Table. 1. Each face only occurs in one group. There are two kinds of dataset splitting method. Both of them are analyzed. But finally split by group is used.

Table 1. Each row contains one sample of input, which contains two faces (the second and third column), and truth (the last column).

No.	key points coordinates group 1	key points coordinates group 2	Label	distance
1	face 1 of person A	face 2 of person A	1	0
2	face 2 of person A	face 3 of person B	0	1
3	face 3 of person B	face 1 of person A	0	1

Random by Sample This method split the dataset into training set and test set by randomly select each of the sample without considering groups, which means the probability of each sample is selected is independent and identically distributed. This method has a high probability resulting in data leakage because the data in a sample can be found in two other samples in the same group. For example, if the first two samples of a group are split to training set and the third sample is split to test set, before the test, machine has already met faces in the third sample by learning the first two samples. As shown in Fig. 3, if the network forced to remember the No.1 and No.2 or No.1 and No.3 samples in one group, it can directly get the result of the rest sample. Only one-third probability that machine cannot infer based on the other samples. What machine should do is draw dots from the beginning to infer the distance rather from pre-drawn dots.

Random by Group This method split the dataset by group. That is, samples in the same group are either all split to training set or all split to test set. The probability of each group is selected is independent and identically distributed. This method can prevent data leakage but due to the insufficiency of samples in the dataset, fewer samples in the training set has higher probability leads to overfitting in addition with that one face in a group occurs twice. But experiments show that this effect is not serious. For this method, 4 groups are split to test set and 8 groups are split to training set.

2.3 Fuzzy Clustering

Fuzzy C-Means [3] clustering is a clustering algorithm that each data point can belong to more than one cluster. The degree of membership to each cluster is used to describe one data point's belonging state. In the dataset of historical faces, there is no obvious cluster. But C-Means can use degree of membership to describe the degree a data point belongs to a cluster. Hence, it can be used to describe the degree of a face belongs to a certain face groups, in which all faces have similar features.

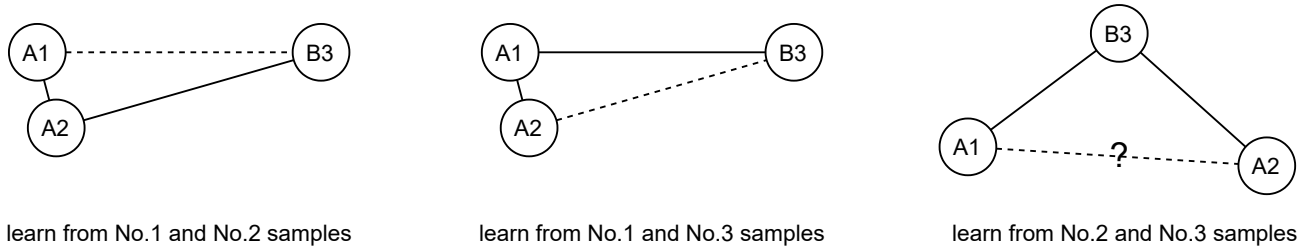

Fig. 3. Faces A1 and A2 are the same person and B3 is another person. Solid line represent the knowledge in training set and the length of dash line is what we want machine to calculate. For example in the left most image, we can directly know that A1 is far from B3 if we know A1 is close to A2 and A2 is far from B3. But we cannot infer whether A1 is close to A2 if we only know that A1 and A2 are both far from B3.

It is simple to use C-Means. First, treat the x and y coordinates of 14 key points on each face as one 28 dimension data point to calculate centroids using only training set to prevent data leakage. Then, use the obtained centroids to calculate degree of membership for each face in both training set and test set. At last, append each face's degree of membership after the coordinates of key points of the face.

To decide the number of clusters, fuzzy partition coefficient (FPC) is used. The higher FPC is, the clustering is better. Since the data we are clustering have no obvious partition, the FPC will not be high. But it is unexpected that FPC rises when the number of clusters is 5, which means there are potentially 5 clusters. Hence, number of clusters is set to be 5. Other parameters when implementing C-Means are: $m = 2$, $stop\_criterion = 0.005$, $max\_oteration = 1000$.

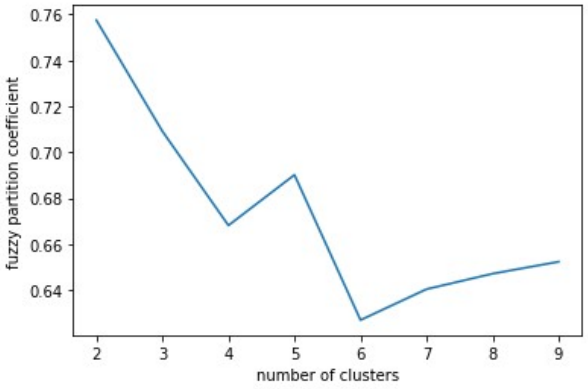

Fig. 4. Tests of number of clusters from 2 to 9. Fuzzy partition coefficient (FPC) is used to describe how well the data being clustered. It is in range from 0 to 1, with 1 being the best. The FPC is averaged value of 10 runs.

2.4 Model structure

FaceNet [6] uses a point on a high-dimension unit hypersphere to represent a face. This paper will take the same representation as FaceNet. To train FaceNet, triplet loss are used. In a triplet there are three faces, x^a is anchor face, x^p is another face of the same person as anchor face and x^n is another face belongs to different person. The loss function of triplet loss is,

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ , \tag{1}$$

where $[t]_+$ means $max(t, 0)$, α is margin between different person's face, and function $f(x)$ calculates a high-dimension point as feature embedding to represent face x. Triplet loss is aiming at shorten the distance between same person's faces to 0 and extend the distance between different person's face to α.

Based on the format of the dataset and inspired by FaceNet, a model is constructed. The model takes two images' key points coordinates together with the degree of membership to the 5 centroids as input, which is 28+5 in total, and out put the probability that the two images are not the same person represented by distance. Hence, the ground truth for same person's two face images are 0 and 1 for different persons' faces. As shown in Fig. 5, the model first use the same three-layer network to extract embeddings from the two input images' key point coordinates and degree of membership one after another, then project them onto a high-dimension unit hypersphere, i.e. $\|v\| = 1$, finally calculate their distance. Since in the dataset each sample data contains two faces and whether they are the same person, mean square error is used as loss function. Because the output is distance, use MSE to train this model is to shorten the distance between same person's faces to 0 and extend the distance between different person's face to 1. The compression network in this model has 33 input, 15 nodes in hidden layer and 10 output nodes and all the activation functions are sigmoid. Learning rate is set to 0.00001 and Adam is used as optimizer.

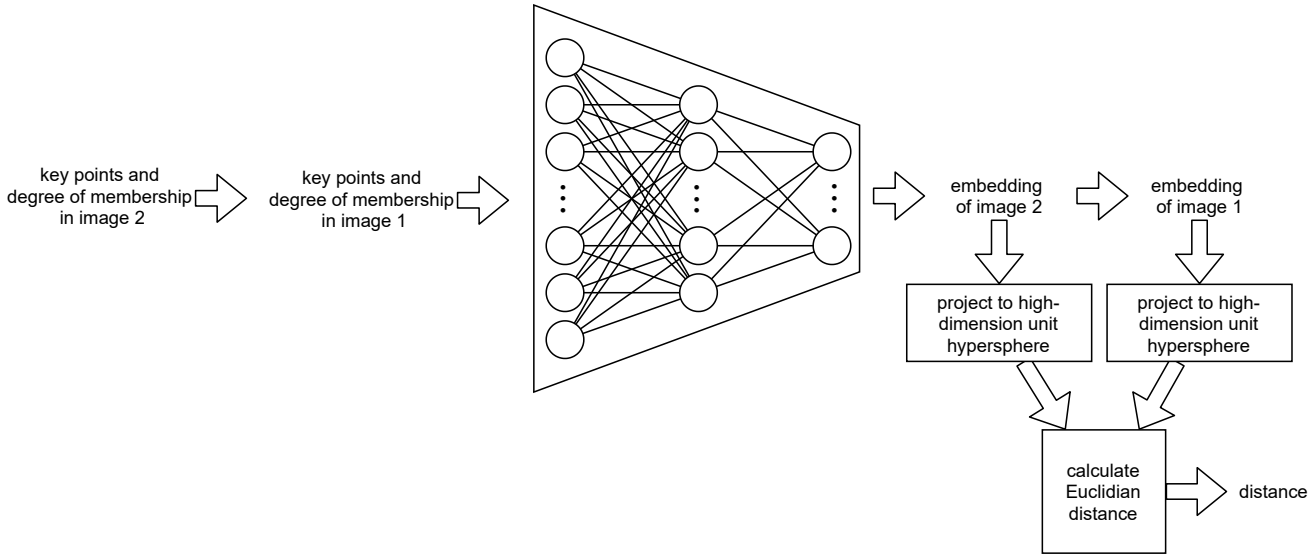

Fig. 5. The structure of the model.

When implementing this model, an auto-save mechanism is used. This auto-save mechanism can automatically save the whole model if the training accuracy and test accuracy are both higher or equal to a preset value, which are both 0.8 in default. And it will automatically update the saved model during training if the two accuracy is both raised or one is raised and the other stays the same.

2.5 Pruning

Pruning technique [7] is implemented to analyze the performance of simplify this model.

In [7], three-layer compress and decompress network are used on an image. With pruning more nodes in hidden layer, the image is compressed progressively. However, in this paper, pruning technique is used in hidden layer, which is not exactly the result of compression. Since the dimension of embedding vectors and the input are fixed, the network in this model can be regarded as a simple encoder-decoder to extract embeddings. In this case, the output of hidden layer can be regarded as a compressed vector. Hence, using pruning technique on it can enhance the compression degree.

Pruning is done after training the model. Accuracy in training set and test set are used to analyze the pruning performance. The pruning steps are clarified as follows. First, calculate each neuron's activation vector. Each component in activation vector is the output of this neuron when the corresponding training example is input to the network. In other words, component v_i in the activation vector v of a neuron is the output of this neuron when the i^{th} training example is fed to the network. Second, calculate the angle range of 0-180 among these activation vectors centered to 0.5, that is, minus 0.5 before calculate the angle. At last, handle the neurons based on the angle

of activation vector. If the angle is less than 15°, the two vector is considered as similar, one of them should be removed and the weight of the removed one is added to another one, which will not be removed. If the angle is over 165°, the two vector is considered as complementary, both of the neurons should be removed.

3 Result and Discussion

Using the unification method in range 20-140 and degree of membership to the 5 centroids in training set, the accuracy of this model in both training and testing set can reach more than 90%. The degree of membership calculated by C-Means stabilize the accuracy from fluctuating from 40% to 90% if not using it to fluctuating from 70% to 90%. Loss and accuracy during training of one of the best models is in Fig. 6, the blue vertical lines in which are the time when auto-save mechanism works to save the whole model. This model can still keep an over 90% accuracy even if re-split the dataset and recalculate the C-Means centroids of the training set to calculate a new degree of membership for each face in training and testing set.

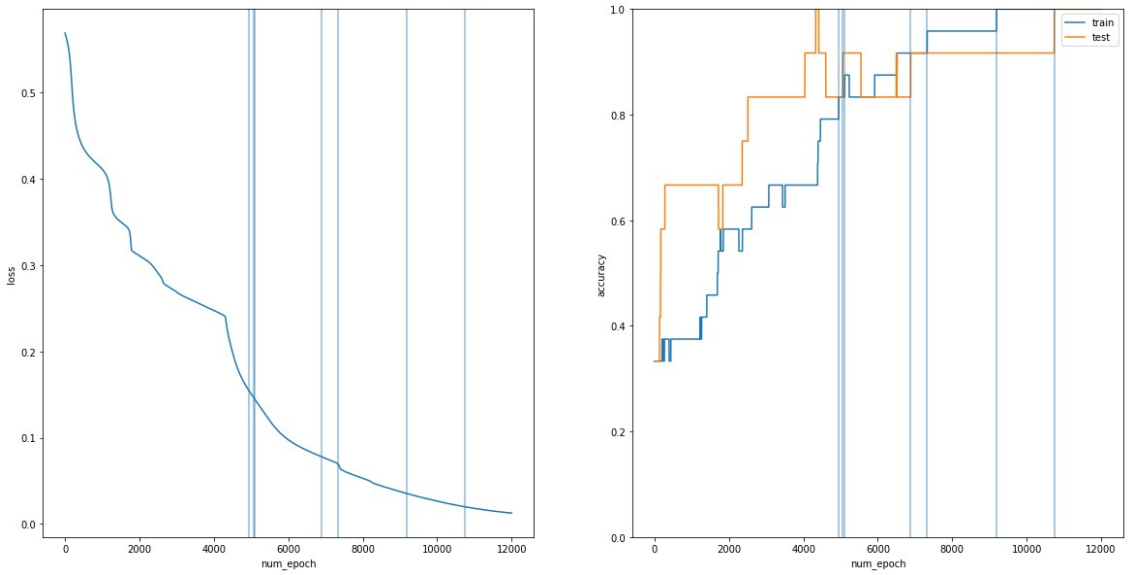

Fig. 6. The left plot is loss over training epochs. The right plot is accuracy over training epochs. The orange line in the right plot is test accuracy and blue line is training set accuracy. The blue vertical lines are the time when auto-save mechanism works to save the whole model and for each saving process it will cover the previous save. Hence, after training, the saved model will be the last saved one which is the rightmost blue vertical line.

It is found that when calculating angles between activation vectors range in 0-90, some angle is NaN, that is, the norm of some neurons' activation vector is 0. Experiments show that remove these neurons whose activation vector's norm is 0 will not affect the result at all. It can also be proved by mathematical deduce easily. Remove neurons whose activation vector's norm is 0 can be considered as another basic pruning technique to reduce model complexity without losing performance.

Fig. 7 shows the representative result of pruning the same saved model as Fig. 6. After calculating the angle matrix of activation vectors range in 0-180, two rows contains $< 15°$ angles and $> 165°$ angles are chosen and each pair in them are processed before recalculate the accuracy to show the difference of pruning ordinary neurons and similar or complementary neurons. In Fig. 7, the upper chart show the different accuracy handling similar neurons pairs and ordinary pairs. The lower chart show the different accuracy removing complementary neurons pairs and ordinary pairs. It can be found that accuracy stays unchanged after handling similar neuron pairs, detail of which is removing one of the neuron in the pairs and add the weight of the removed neuron to the other one. Accuracy drops after removing complementary neuron pairs, removing ordinary neuron pairs and removing one of the neuron in the ordinary pairs and add the weight of the removed neuron to the other one. This controlled trial shows that only handling neuron pairs with similar activation vector angle can reduce the complexity of the model without accuracy dropping.

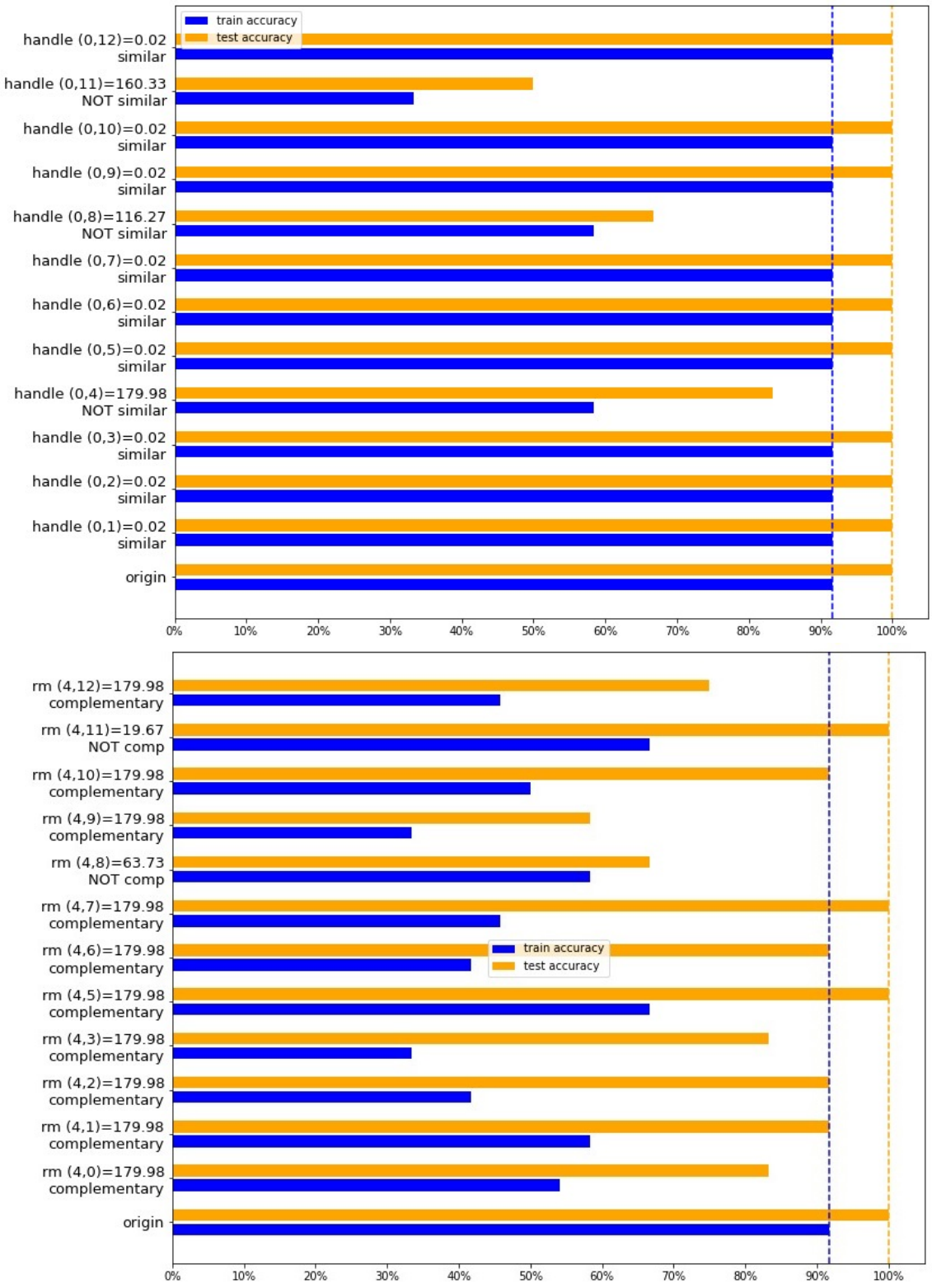

Fig. 7. The upper chart show the different accuracy after handling similar neurons pairs and normal pairs. The lower chart show the different accuracy after removing complementary neurons pairs and normal pairs. The orange bar represent test set accuracy and blue bar represent training set accuracy. The two bars at the bottom of each chart is the accuracy before pruning. The orange and blue dashed lines is the accuracy of training set and test set before pruning. The number of bars is different with the number of neurons of the middle layer is because neurons whose activation vector's norm is 0 have been removed. In the upper chart, only similar pairs has the same accuracy as the original one, others are all dropped. In the lower chart, accuracy of every pair drops.

There are some limitations in this study. First, the dataset splitting method has drawbacks. Random by group may lead to overfitting since it is emphasized by contains each face twice. Another is that since the dataset contains only 36 samples, the training result is unstable.

4 Conclusion and Future Work

The proposed face embedding compression network generate a point at high-dimension hypersphere to represent one's face and using distance to show the similarity of two faces. This method is feasible and can reach up to 90% accuracy, which is higher than the performance in [1]. Pruning technique based on similarity of angles of activation vector is proved to be useful and a new pruning technique that remove neurons whose activation vector's norm is 0 without affect the performance is proposed.

Tests show that unify key points coordinate to range 0-1 caused accuracy drop after redo the dataset splitting and C-Means process but unify to range 20-140 won't. Hence, in the future unify key points coordinate to different range should be test to figure out the most suitable range. If only use key points on the face to predict rather than the whole image, it is feasible to directly use modern human's face to enlarge the dataset in the future. The proposed embedding compression network is a simple three-layer network. As future work, more complex structure should be considered and the most suitable number of neurons for each layer could be proposed. If using CNN in the future, the dataset can be enlarged by filling it with antiqued modern photographs. This paper use task that compare whether two faces are the same person by distance of embedding are used to train the compression network. In the future, more tasks could be proposed to train the compression network. With a larger dataset, future work can analyze more details about the condition to pruning and try to find a way that the pruning can be done automatically with the lowest performance lose.

References

1. Caldwell, S. (2021) "Human interpretability of AI-mediated comparisons of sparse natural person photos" CSTR-2021-1, School of Computing Technical Report, Australian National University.
2. Masi, I., Wu, Y., Hassner, T., & Natarajan, P. (2018, October). Deep face recognition: A survey. In 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI) (pp. 471-478). IEEE.
3. Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & geosciences, 10(2-3), 191-203.
4. Caple, J., & Stephan, C. N. (2016). A standardized nomenclature for craniofacial and facial anthropometry. International journal of legal medicine, 130(3), 863-879.
5. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48.
6. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).
7. Gedeon, T. D., & Harris, D. (1992, June). Progressive image compression. In [Proceedings 1992] IJCNN International Joint Conference on Neural Networks (Vol. 4, pp. 403-407). IEEE.
8. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., ... & Gao, J. (2020, August). Oscar: Object-semantics aligned pre-training for vision-language tasks. In European Conference on Computer Vision (pp. 121-137). Springer, Cham.
9. Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., & van den Hengel, A. (2017). Visual question answering: A survey of methods and datasets. Computer Vision and Image Understanding, 163, 21-40.