Detecting whether two images are matched using Fuzzy Logic representation's and Neural representation's Data Fusion

Haoyi Chen
Research School of Computer Science
Australian National University
Canberra ACT 2601 Australia
u6790253@anu.edu.au

Abstract.
Photos are used to keep history information and people's memory. This paper is to classify whether two images are the same person using the Facial Feature Marks distance dataset. This paper extends my previous paper [2]. This paper compares the performance of the data fusion network(combine fuzzy logic representation and Neural representation) with the performance of the Two-Layer Neural Network which I implemented in my previous paper [2]. We find that the Fusion network performs better in accuracy and F1 score. It has 90% accuracy and 0.85 F1 scores. But for Two Layer Neural Network, only has 81% accuracy and 0.7 F1 scores. I suspect that the reason is the parameters in Fusion Network are more than Two-Layer Neural Network.

Key Words: Neural Network, Fuzzy Logic, Data Fusion, Facial Feature.

1 Introduction

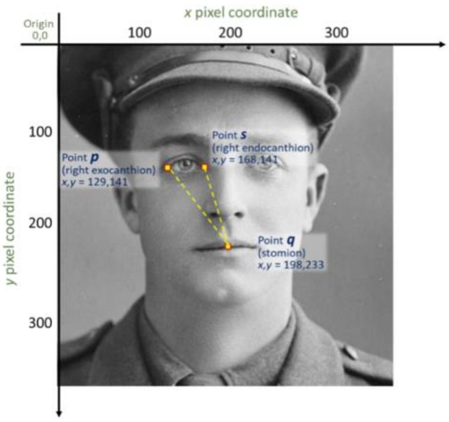
1.1 Background

Photographs are commonly used to keep history information. Photos can refresh people's memory, can record social relationships, and so on. However, for many historical photos, it's hard to recognize people in those photos. The problem that this paper solves is to classify whether two photos are the same people or not (match or not). It's meaningful to solve this problem. [3] shows that although AI can solve facial recognition successfully in the modern sense of term using digital photography and video capture, in cases where there are only very few images of individuals, it is of limited help in identifying individuals. Therefore, distinguish whether two photos are a match or not is worth solving. Dr. Sabrina Caldwell[3] showed average accuracy of NN is 75% in his paper. And my previous work[2] has implemented a Neural network and Casper network to solve this problem. [3] This paper will show data fusion of fuzzy logic representation and neural representation to extend its work on this problem.

1.2 Data inspection and choice

This paper provides two data sets related to history photos. The first one uses Facial Feature Markup(FFM) to show people's characteristics in the images. The second one uses the distance of each FFM. For the FFM dataset, we have 14 FFMs for each image. Because each FFM points have x and y coordinates, we have 28 features for each image and we have 56 feature for comparison (two images). For the Distance dataset, because we have 14 FFMs, we combine every two points in the data. So we have 91 distance features for each image and we have 182 for comparison (two images). And the last column of our dataset means matching or not. We have a total of 12 photo groups and each group has three photos and we combine every two photos from one group, which means for each group, the first two photos are matching and the next two pairs are not matching.

Dr. Sabrina Caldwell [3] introduces three ways to extend the dataset using FFMs. First, as the dataset is given: distance. Secondly, the proportions of two distances of the facial features of an individual in one image(better understanding in Fig 1). Thirdly, the angles of the derived right triangle relative to the FFMs(Fig2).



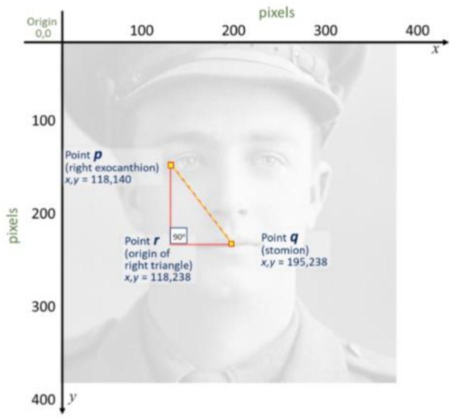


Fig 1. the proportions of two distances **Fig 2.** The angles of the derived right triangle

I have implemented three different datasets for my NN and Casper network at my previous work[2]. The performance of the proportion and angle dataset is performed very badly. Then reason may be Euler angles. The proportions' value and angle's value are not that big as the distance is, which means it will be affected more when people take photos at totally different Euler angles. So in this paper, I will choose distance as my dataset.

1.3 FDNN introduction

To extend the Neural network in previous work[2]. Yue Deng, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghai Dai[1] raise a fused fuzzy deep neural network(FDNN). It can extract information from both fuzzy and neural representations. The fuzzy system will automatically learn fuzzy membership functions and get fuzzy rules from some specific tasks. Then the fuzzy logic values are linearly combined in a defuzzifier to form the final decision of some specific tasks. Compared with traditional deterministic representations, the fuzzy logic representation can construct fuzzy rules flexibly.

2 Methods

2.1 Data Preparation

My pre-processing steps are applied to have a better performance. I use distance as my dataset. Firstly, I drop the first column which is the index number of the dataset. This column does not include any information about photos. If we include this, it will affect our results. Then I use MinMaxScaler to keep all of my data in the range from 0 to 1 so that it is normalized. In addition, because our dataset is very small, which only has 36 sets for us to train and test. To avoid overfitting, I use ShuffleSplite to have 10-fold cross-validation. I also set we have 70% dataset as our training set and 30% dataset as our test set. The last column will be used as our target output. We use the rest features as our input of Neural and Fusion networks.

2.1 Fused Fuzzy Deep Neural Network (FDNN) Implementation

This Neural Network is made up of four learning parts. The whole process is shown in Fig 1. The purple one is input data and it goes through two paths to respectively get fuzzy logic representation and the neural representation(DR). The FL representation is black part and DR is blue part in Fig 3. For the green part, which is the fusion part of FL representation and DR. We combined the FL representation result and DR result as our input of the fusion layer. The fourth part is our task-driven layer, which can classify data into different categories.

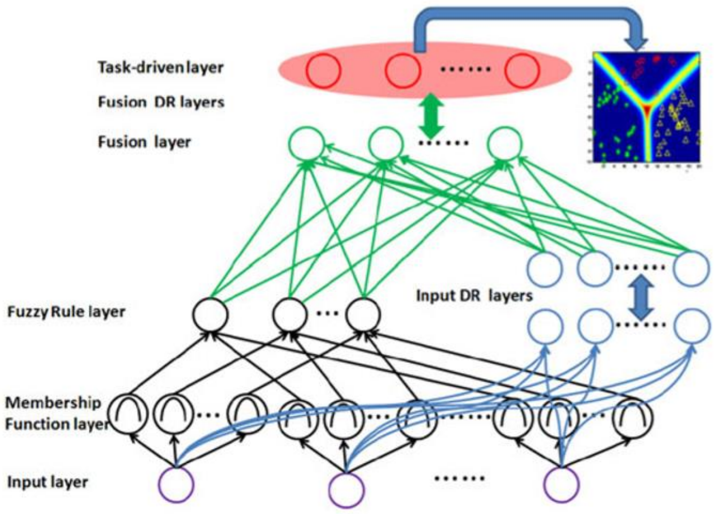


Fig 3. The whole process of FDNN.

2.1.1 Fuzzy Logic Representation Part:

Each node in the input layer, it's connected with many different membership functions, which will assign linguist labels to every node's input variable. This function will calculate the fuzzy degree, in which the input variable belongs to a fuzzy set. Fig 4 shows that the formula of getting membership function layer, where ak(l) represents the input of the number i node. oi(l) is the corresponding output. Follow [1]'s suggestion, we use the gaussian membership function's mean μ and variance σ. Then we use the "AND" fuzzy logic to be our fuzzy rule layer's operation. So that we get the result of Fuzzy Logic Representation's result, which is Fuzzy Rule layer shown in Fig 3.

$$o_i^{(l)} = u_i(a_k^{(l)}) = e^{-(a_k^{(l)}-\mu_i)^2/\sigma_i^2} \quad \forall i.$$

Fig 4. The formula to get the Membership function layer.

2.1.2 Neural Representation Part:

This representation is generated by using a fully connected layer, just like I did in my previous work [2].

2.1.3 Fusion part:

Because of the above 2 parts, we have got FL representation and Neural representation. We combine these two representations as our input of the fusion part. And we go through a fully connected layer to get our fusion representation. For better understanding, we can consider the output of the fuzzy part and the Neural part as the features of the fusion part.

2.1.4 Task-driven part:

This part is assigning the fusion representation into its corresponding category. I have used the softmax function to classify the data. Softmax can produce the probability of our result.

2.2 Hyper-parameter Tuning:

For this part, I set the same hyper-parameter for both FDNN and my baseline NN.

For the learning rate part. A too large learning rate may skip the optional results and a too small learning rate may train too slow. At the start, I set a very small learning rate, such as 0.00001. Then, the network is updated after each batch, and the learning rate is increased at the same time, and the loss calculated by each batch is counted so that we can find 0.0001 is a relatively good learning rate.

For epoch times, I will observe how many times my loss's change is very small. Too large epoch times will cause overfitting and too little epoch time will not train completely. I set 100 as my max epoch time.

For batch size. Commonly batch size is not lower than 16 but if we have a large batch size, our memory does not have enough capacity and it will lead to training loss can not decrease and generalization gap problem. So I choose 16 as my batch size.

2.3 Train and test methodology:

Firstly, I split input and target. Use the last column as our target and use the rest as our input. Because we need to use 10-fold cross-validation, I split our train set and test using the corresponding index. I set CrossEntropyLoss as my loss function and Adam as my optimizer. For these problems, I use loss, accuracy, and F1 as my testing indicator. They are all average value for 10-fold cross-validation

3 Result and Discussion

Table 1. Comparison of Two-layer Neural Network with Fuzzy Deep Neural Network

Method	Train loss	Test loss	Accuracy	F1
FDNN	0.1217	0.5638	0.9	0.8504
Two Layer Neural Network	0.3343	0.4889	0.8182	0.7013

The Result shows that FDNN performs better in Accuracy and higher in F1 score. Just as I thought before, FDNN performs better than a Two-layer Neural Network. The accuracy of FDNN varied from 72% to 100% and the accuracy of FDNN varied from 45% to 100%. It means that FDNN has a better generalization. The reason why FDNN performs better than Baseline is that fuzzy representation reduces the uncertainties and NR removes noises. FDNN integrates those two advantages so it will perform better when dealing with data with ambiguity and noise.

The limitation of this experiment is that the quantity of parameters of FDNN is more than the Two-Layer Neural Network. FDNN has more than 1000 parameters, which means it can store more information and learn deeper knowledge. It's the advantage that Two-Layer Neural Network does not have.

4 Conclusion and Future Work

4.1 Conclusion

This paper introduces an FDNN algorithm to classify whether two images are matching or not. FDNN fuses Fuzzy Logic and Neural Network representations to extend our Neural Network and increase the performance of the Neural Network. From my experiment, we can conclude that FDNN can reach 90% average accuracy and our baseline NN only can reach 80% average accuracy. The F1 score of FDNN is also higher than NN. The main reason might be that Fuzzy Logic can reduce ambiguity using different kinds of fuzzy rules and FDNN has more quantity parameters than NN.

4.2 Future Work

Firstly, because we have limitations for this experiment, the quantity of parameters, the quantity of FDNN's parameters is higher than NN. For this, we have to keep their quantity be the same for future work.

Secondly, I have observed that the distribution of our dataset is abnormal. The amount of match is 12 but we have 24 as not match amount, which will affect our result. In addition, the dataset is too small, we can have a larger dataset and the target distribution should be the same proportion.

Thirdly, we can try to compare FDNN with other data classification strategies, for example, self-constructing fuzzy neural network and self-constructing fuzzy neural network, so that we can verify the performance of FDNN is good enough or not.

5 References

[1] Yue Deng, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghai Dai. 2017, "A Hierarchical Fused Fuzzy Deep Neural Network for Data Classification", Automation Department, Tsinghua University

[2] Haoyi Chen, 2021, "Analyzing Classification Problem Using TwoLayer Neuron Network and "Casper" Method with Facial Feature dataset.", Research School of Computer Science, Australian National University

[3] Dr. Sabrina Caldwell. 'Human interpretability of AI-mediated comparisons of sparse natural person photos', Research School of Computer Science The Australian National University

[4] N.K. Treadgold and T.D. Gedeon. 'A Cascade Network Algorithm Employing Progressive RPROP'. School of Computer Science & Engineering, the University of New South Wales, Sydney N.S.W. 2052 AUSTRALIA.