A CNN-Based Facial Expression Recognition System

Ziyu Hu

School of Automation, Central South University, Changsha, China

huziyu@st.btbu.edu.cn

Abstract. Facial expression recognition has long attracted researchers from a variety of domains, such as human-computer interaction, emotion analysis, intelligent medical care, and so on. Various human-designed features have been used in FER to extract image appearance features, but human-designed methods are difficult to be used to recognize new face images, which poses challenges for FER in uncontrolled environments. Recently, CNNs and FNNs have been used for facial expression recognition. The new CNN-based recognition approach yields excellent FER results. The numerous convolution and pooling layers that CNN possesses allow it to extract higher and multi-level features of the entire face or local regions. Additionally, CNN has excellent classification performance when it comes to the feature extraction of facial expression pictures. However, the mechanism through which FNN remains largely controversial. In this work, the experiments are carried out on the comparison of three facial expression recognition algorithms: K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Convolutional Neural Network (CNN), and the experimental results show that, using CNN can get more stable recommendation results of better quality.

Keywords: Deep Learning; Convolutional Neural Network; Computer Vision; Facial Expression Recognition.

1. Introduction

Facial expression recognition (FER) is an important direction of human-computer interaction. Artificial intelligence has recently become increasingly popular, and FER has become one of the research hotspots. Facial expressions contain so much information that even slight ones can reflect changes in psychology. It is conceivable how exciting it would be if machines could keenly recognize the emotions expressed in human faces. FER has been a topic that researchers from many different fields, such as human-computer interaction, emotion analysis, intelligent medical care, etc., have been interested in for a long time [1].

FER is an acronym for facial expression recognition, which is the technique of assessing the emotional and psychological states of persons based on facial expressions extracted from still pictures or video sequences [2]. At present, there are many research results on FER. In the 1970s, American psychologists Ekman and Friesen defined six basic human expressions through a large number of experiments: happiness, anger, surprise, fear, disgust, and sadness [3]. Various human-designed features have been used in FER to extract image appearance features, including Gabor, LBP, LGBP, HOG, and SIFT. Then in the early years, Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) algorithms also began to be applied in FER [4]. Convolutional neural networks (CNN) and feedforward neural networks (FNN) have also been used to recover FER data in recent years [5].

However, most of the above algorithms, including FNN, still have many problems. Human-designed features are too constrained by the designed algorithm and cannot be integrated into an end-to-end model [6]. FER samples are mostly large and unbalanced, while SVM and KNN are not suitable for larger datasets, due to their large amount of calculation and low efficiency on noisy data sets with overlapping classes. As for FNN, a simple neural network, it ignores two-dimensional information of images, and the features extracted by the shallow convolutional network have poor robustness [7,8]. Due to the shortcomings of basic algorithms, this study tries to solve the problem of FER with deep convolutional networks.

2. Methods

2.1 Data Fetch & HOG Feature Extraction

2.1.1 Data Fetch

This study makes use of The Japanese Female Facial Expression (JAFFE) Database, which has 213 photos. In an experimental setting, ten Japanese women were instructed to make seven facial emotions, which were sad, happy, angry, disgust, surprise, fear, and neutral. The camera was then used to collect photographs of these facial expressions. Each person has about 3 to 4 patterns of each expression [9]. This paper divides the dataset into training and prediction sections for building models, the training section has been classified in advance, and images in the prediction section are unordered, after the model is trained, it can be used as a prediction.

2.1.2 Pretreatment

With great success, HOG feature paired with the SVM classifier has been utilized extensively in image recognition, particularly for pedestrian identification [10]. First, convert the photos to grayscale, then use Gamma correction to standardize the color space of the input image. The contrast of the image should be adjusted, the influence of local shadows and lighting fluctuations should be mitigated, and noise should be removed from the picture.

2.1.3 Calculate the Image Gradient

Before calculating a HOG descriptor, this study must determine the horizontal and vertical gradients. Ultimately, this effort will compute the gradient histogram [10]. It is simple to do by simply filtering the image with the given kernels.

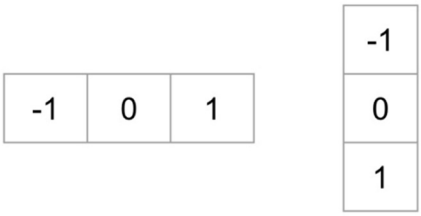


Fig 1. The kernels

The size and direction of the gradient using the following formulas [15]:

$$g = \sqrt{g_x^2 + g_y^2}$$
$$\theta = \arctan\frac{g_y}{g_x} \quad (1)$$

2.1.4 Calculated at 8 × 8-cell Gradient Histogram

8 × 8 Image patch has 192-pixel values (8x8x3=192). This gradient patch has two values per pixel (size and direction), resulting in an 8x8x2 value of 128. At the end of this section, a 9-Bin histogram represents 128 numbers (an array that can be stored as a length of 9). In patch histograms, not only is the representation more compact, but the algorithm is also more resilient. Individual color gradients may contain noise, but an 8x8 picture patch indicates that the gradient histogram is not noise-sensitive.

2.1.5 16x16 Step Normalization

The picture gradient-based histogram was produced by the stages that came before this one in this procedure. Gradients are very sensitive to the brightness level of a picture as a whole. If you do this operation on an image and it reduces the brightness by reducing all of the pixel values by two, the gradient will shift by one-half, and so will the value of the histogram. The histograms are going to be normalized (normalized) as part of this study in the hopes that this will prevent them from being affected by differences in light [10].

After completing the above steps, the HOG eigenvectors of the images can be calculated, because all the images in the Jaffe dataset are grayscale 256×256 images with the same pixel scale, resizing is not required.

2.2 KNN

2.2.1 Prerequisites for the KNN Algorithm

i) All characteristics of a sample are quantified comparatively:

The picture gradient-based histogram was produced by the stages that came before this one in this procedure. Gradients are very sensitive to the brightness level of a picture as a whole. If you do this operation on an image and it reduces the brightness by reducing all of the pixel values by two, the gradient will shift by one-half, and so will the value of the histogram. The histograms are going to be normalized (normalized) as part of this study in the hopes that this will prevent them from being affected by differences in light [11,13].

ii) Sample characteristics should be normalized:

Each of the sample's parameters has its domain and acceptable value range. They have distinct consequences on the estimation of distance; for instance, the bigger value will overshadow the lower parameter value. Therefore, the sample must contain scale processing parameters; the simplest method is to normalize all numerical characteristics.

iii) A distance function is required to calculate the distance between two samples:

Typical distance measurements include the Euclidean distance, the cosine distance, the Hamming distance, and the Manhattan distance, amongst others. As a rule, the Euclidean distance is chosen to represent the level of separation between two points [13].

2.2.2 Main Steps to Implement the KNN Algorithm

i) Distance calculation

Euclidean distance is generally used. Given a sample to be classified, the distance between it and each sample in the classified sample is calculated.

$$\mathbf{dist}(\boldsymbol{X}, \boldsymbol{Y}) = \sqrt{\sum_{i=1}^{n} (\boldsymbol{x_i} - \boldsymbol{y_i})^2} \tag{2}$$

ii) Determination of K value

The lower the value of K, the more straightforward the overall model will be to fit, despite its increased complexity. Cross-validation is frequently utilized as a tool for determining the best k-value to use.

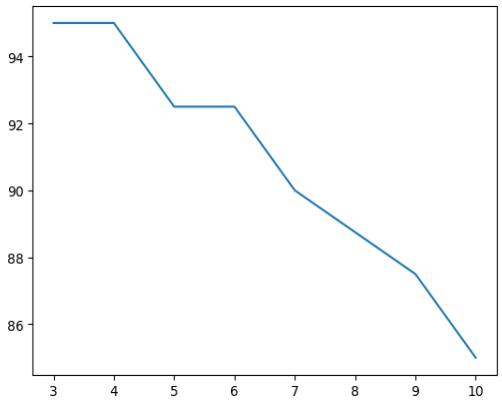


Fig 2. Recognition rate of different K values

By constantly trying to select K values, 3 and 4 will eventually have the highest recognition rate. Generally, K is an odd number, so this study takes k=3.

iii) Classification decision

Majority voting is generally used, most classes of K nearby training points determine the classes of input instances. It can be demonstrated that a majority voting rule is equivalent to minimizing empirical risk.

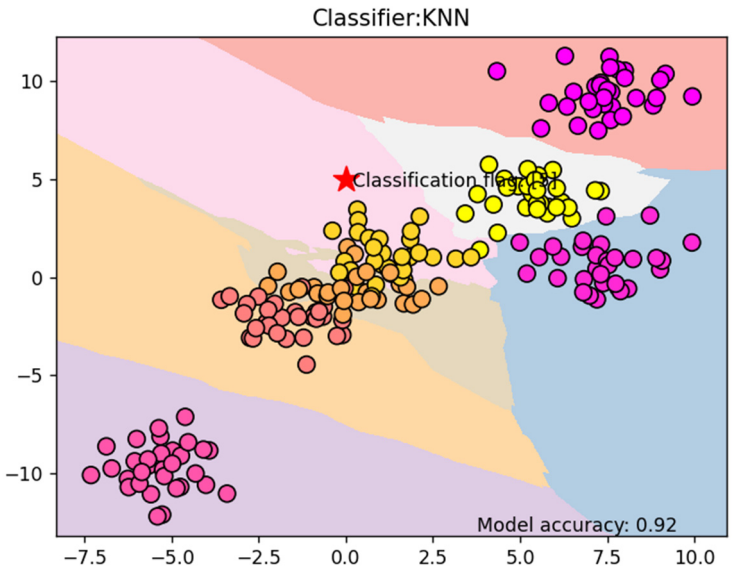


Fig 3. Classification results run in the training set

2.3 SVM

2.3.1 Initializing Data

Load iris data from datasets dataset, extract two columns in data as feature values, extract target as classification values, convert the feature values and classification values into pandas DataFrame data box, and merge them into data. Rename each feature x1, x2, and y.

Find the maximum and minimum values of x1 and x2 to generate a full coordinate system of points used to describe the hyperplane. This study uses seaborn's scatterplot to plot x1 and x2 as shown below.

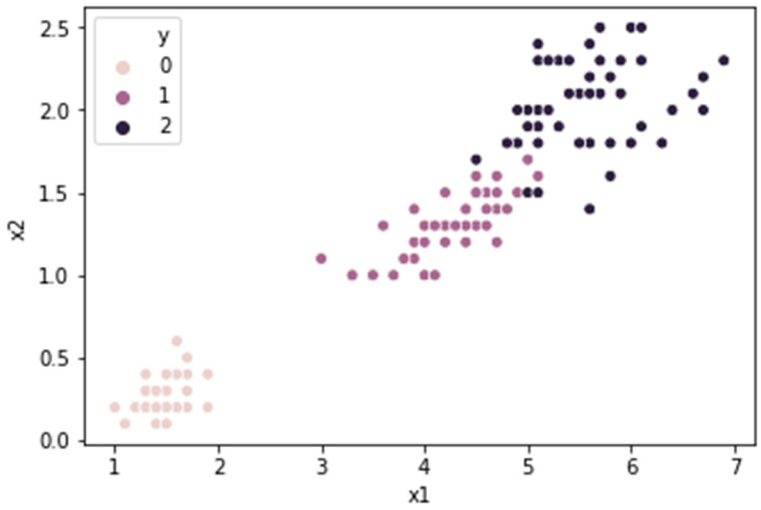


Fig 4. Scatter plot

In this figure, there are three categories, and then this work uses SVM to build a model for these training data.

2.3.2 Linear SVM Model

A linear SVM model with C=0.1 is built, using 80% of the data set as the training set, and all x1 and x2 of the data are used to predict y, y, and the accuracy of the original y calculation [12].

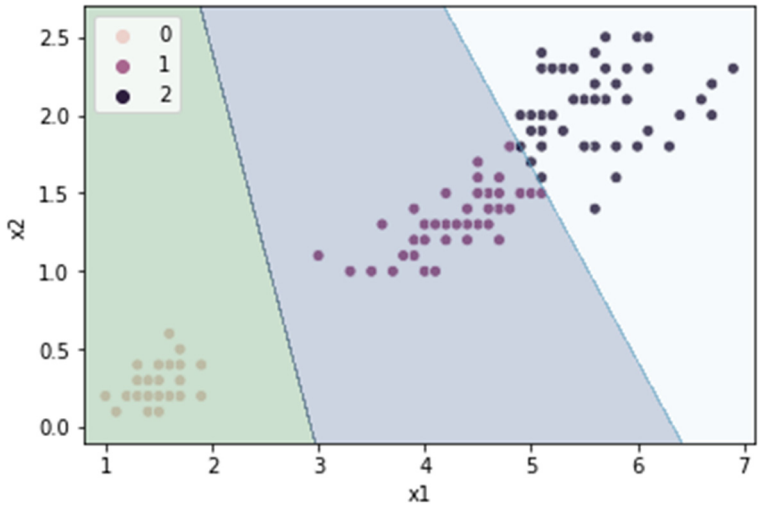


Fig 5. Scatter plot after linear SVM model classification

Results 0.9533333333337, which is about 95.33% accuracy, can be compared with the first figure, to find out which point is misclassified.

2.3.3 Find the Best Model

This study search through the grid to find the best model. GridSearchCV can configure a list of parameters (hyperparameters), in which the best model is automatically found. GridSearchCV has automatically divided the samples into 5 equal fractions with cv=5 for training and validation. The model parameters were automatically found, the accuracy rate was 96.67%, and the kernel function RBF fitted well.

2.3.4 Optimal Model

This work directly uses the whole set to fit the SVM model with C=0.1 and kernel function RBF [14]. The results of the visualization as shown below.

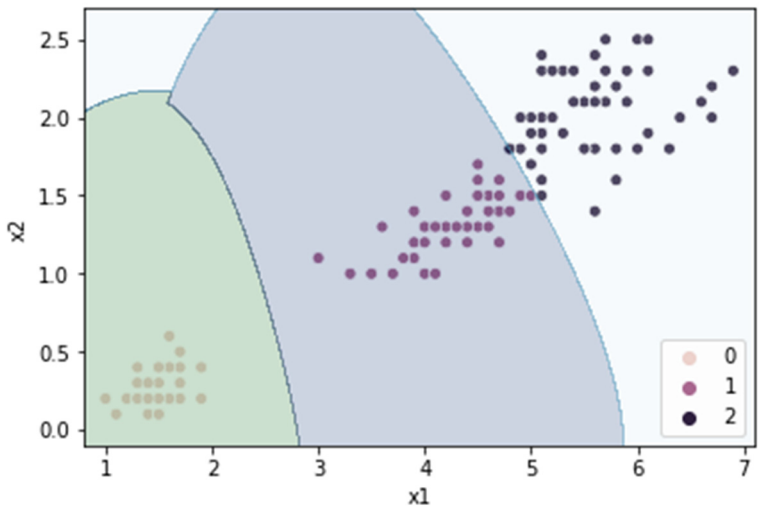


Fig 6. Scatter plot after classification of an optimal model

Running in the training set, results 0.96666666666666667, which is about 96.67% accuracy.

2.3.5 Learning Curve

The accuracy of the model appears to have reached a stable state following the introduction of fifty samples, as shown by the results of this study, which uses this model to analyze the learning curve of the sample:

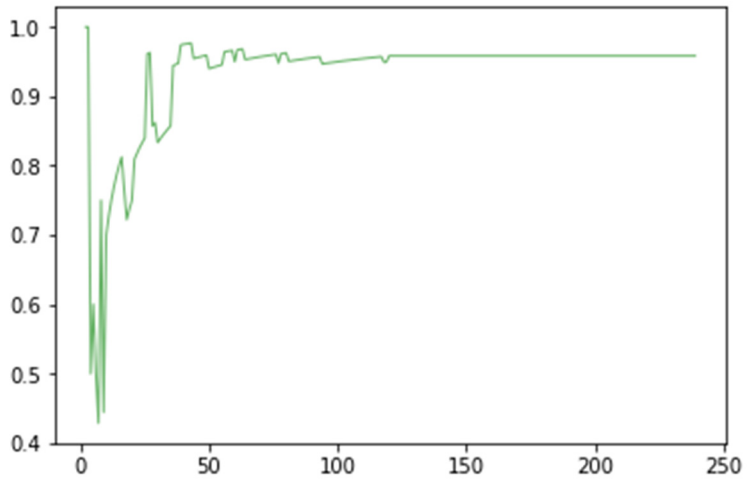


Fig 7. Learning curve

2.4 CNN

2.4.1 VGGNet

VGGNet is a convolutional neural network structure used for large-scale image processing and pattern detection. The VGGNet variation that this study created is seen below.

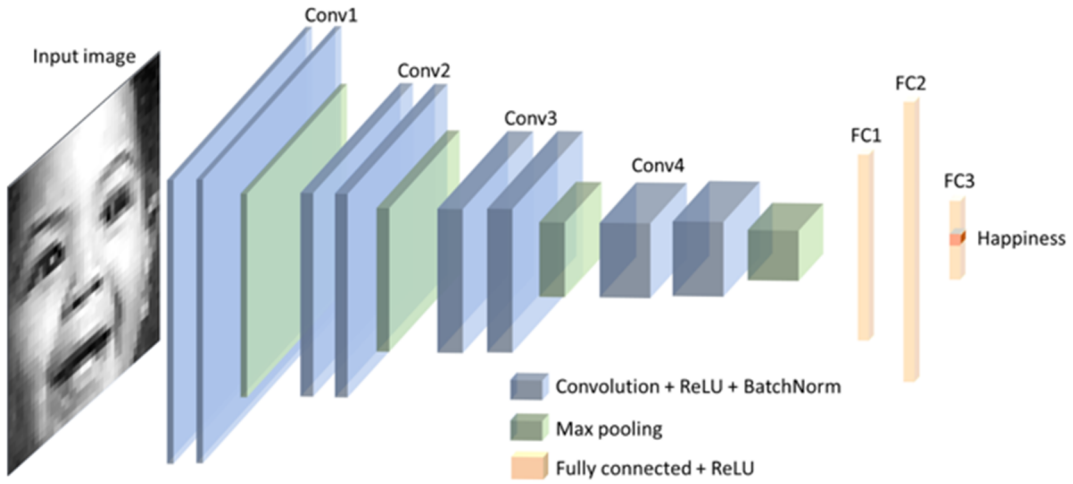


Fig 8. VGGNet

The network is comprised of four Convolutional Stages as well as three Fully Connected Layers. Each level of convolution is made up of a maximum of one Max Pooling and two Convolutional Blocks. Convolution blocks are comprised of a Convolution layer, a ReLU activation function, and a Batch Normalization layer. Batch standardization speeds up neural network learning, decreases internal covariance offsets, and prevents gradients from vanishing or expanding. Following the initial two complete connection layers is the ReLU activation function. The SoftMax activation function is

applied to the final classification in the third completely linked layer. The functions of the convolution level include feature extraction, dimension reduction, and nonlinearity. Following training in the fully connected layer, input pictures may be categorized based on the retrieved characteristics [16,17].

The following table lists the parameters of each layer:

Table 1. The parameters of each layer of VGGNet

Layer		Input size	Convolution Kernel Size	Pooling interval	Output interval
Conv Block 1	Conv	48×48	64,3×3		64@48×48
	BN	64@48×48			64@48×48
	ReLU	64@48×48			64@48×48
Conv Block 2	Conv	64@48×48	64,3×3		64@48×48
	BN	64@48×48			64@48×48
	ReLU	64@48×48			64@48×48
Max Pooling 1		64@48×48		2×2	64@24×24
Conv Block 3	Conv	64@24×24	128,3×3		128@24×24
	BN	128@24×24			128@24×24
	ReLU	128@24×24			128@24×24
Conv Block 4	Conv	128@24×24	128,3×3		128@24×24
	BN	128@24×24			128@24×24
	ReLU	128@24×24			128@24×24
Max Pooling 2		128@24×24		2×2	128@12×12
Conv Block 5	Conv	128@12×12	256,3×3		256@12×12
	BN	256@12×12			256@12×12
	ReLU	256@12×12			256@12×12
Conv Block 6	Conv	256@12×12	256,3×3		256@12×12
	BN	256@12×12			256@12×12
	ReLU	256@12×12			256@12×12
Max Pooling 3		256@12×12		2×2	256@6×6
Conv Block 7	Conv	256@6×6	512,3×3		512@6×6
	BN	512@6×6			512@6×6
	ReLU	512@6×6			512@6×6
Conv Block 8	Conv	512@6×6	512,3×3		512@6×6
	BN	512@6×6			512@6×6
	ReLU	512@6×6			512@6×6
Max Pooling 4		512@6×6		2×2	512@3×3
Flatten		512@3×3			1×4608
Fully Connected 1		1×4608			1×4096
Fully Connected 2		1×4096			1×4096
Fully Connected 3		1×4096			1×7

2.4.2 Train the Neural Network

This study uses two strategies to monitor the network:

1. Over-fit monitoring, if there is no smaller verification loss, the network stops training
2. Learning rate monitoring, reduces learning rate if there is no better validation accuracy

According to the output, there are three learning rate adjustments during the training process, and the training is terminated earlier at the 49th iteration. Observing the curves below shows a slight over-fitting phenomenon in the later stage of the neural network.

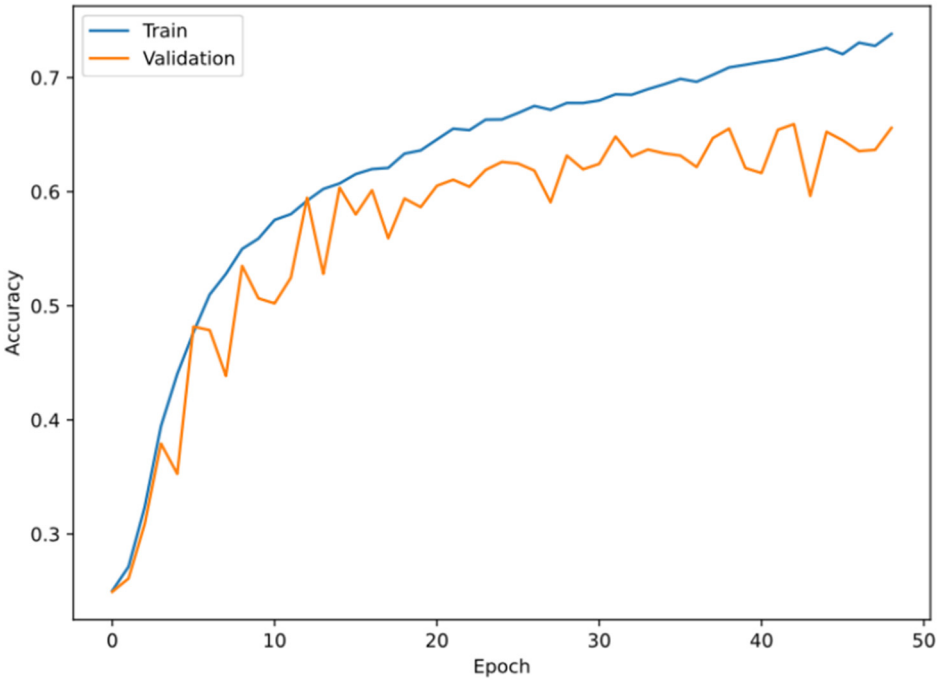


Fig 9. Model accuracy

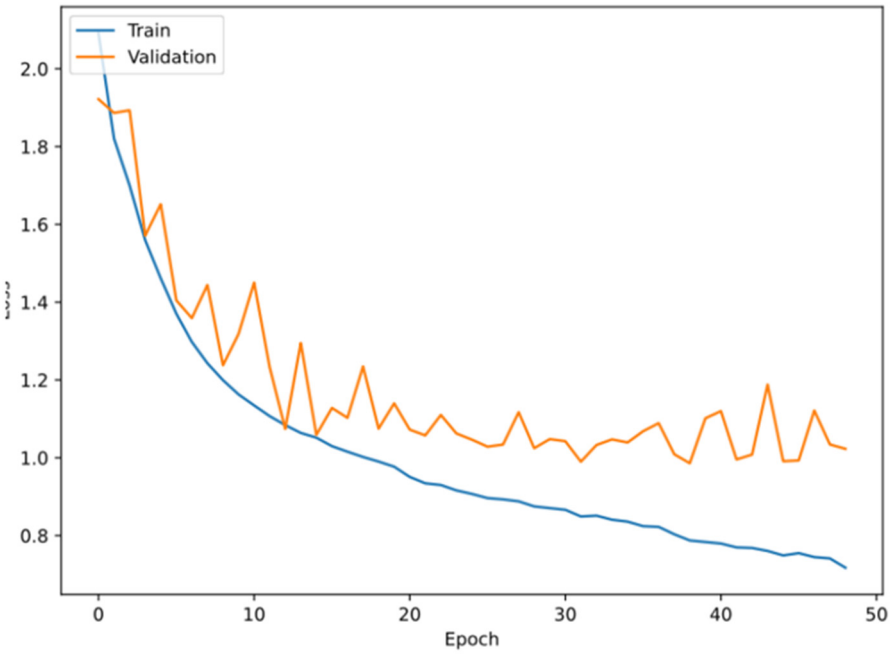


Fig 10. Model loss

3. Results

3.1 Comparison of KNN, SVM, and CNN

Running in the testing set yields the following confusion matrix

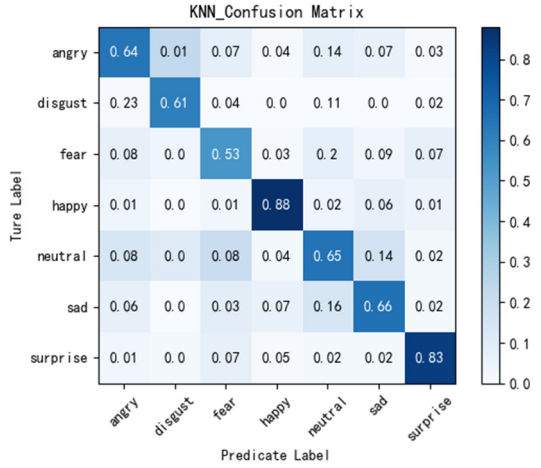


Fig 11. The confusion matrix of KNN

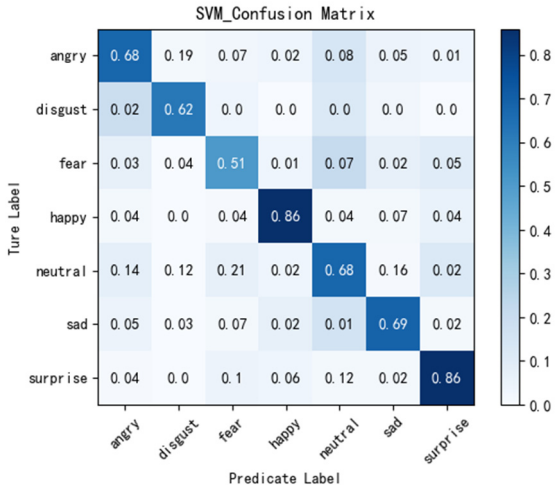


Fig 12. The confusion matrix of SVM

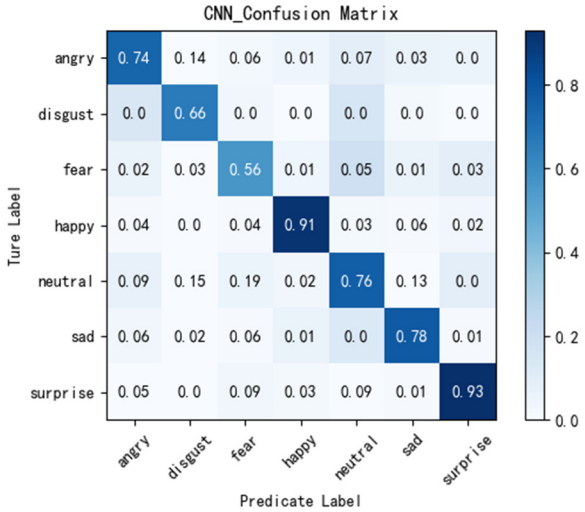


Fig 13. The confusion matrix of CNN

The accuracy, precision, and recall of each model can be calculated:

Table 2. Experimental results in the training set

Model	Expression	Accuracy	Precision
KNN	Angry	68.52%	64.00%
	Disgust		60.40%
	Fear		53.00%
	Happy		88.89%
	Neutral		64.36%
	Sad		66.00%
	Surprise		83.00%
SVM	Angry		68.00%
	Disgust		62.00%
	Fear		51.00%
	Happy	70.00%	86.87%
	Neutral		68.00%
	Sad		69.00%
	Surprise		86.00%
CNN	Angry		74.00%
	Disgust		66.00%
	Fear		56.00%
	Happy	76.28%	91.92%
	Neutral		76.00%
	Sad		76.47%
	Surprise		93.94%

All three models are capable of producing accurate classification results on datasets. Furthermore, CNN is superior to the other two models. And the precision for happiness and surprise is substantially higher than that of other emotions, but the precision for fear is poor.

3.2 Problem Analysis

CNN takes into account all classifications, whereas the SVM and KNN methods only consider two classifications at a time. Additionally, the SVM and KNN methods can only learn whether it is this sample or not, and this sample isn't necessarily the correct sample, which makes classification even more challenging.

The accuracy rates for happiness and surprise are greater than those of other emotions, but the accuracy rate for fear is quite low. One explanation for this issue is that the quantity of varied expressions in the dataset is unbalanced, leading to classification mistakes; another reason is that the four categories of emotions, anger, disgust, fear, and sorrow, have significant commonalities. People find it difficult to differentiate these four sorts of emotions in real life, especially when they do not know each other. Moreover, errors in judgment are inevitable in certain classes that are difficult to discern and easily confused.

4. Discussion

KNN is a broader category of technology, which uses specific training examples to predict without maintaining the model based on data. And KNN algorithm is suitable for the case of the irregular boundary of sample classification like FER. The KNN is superior to other methods for splitting a sample set with overlapping or overlapping class fields. This is due to the fact that KNN focuses mostly on the limited adjacent samples to pick the category, rather than the method of identifying the class field itself [11].

However, KNN has low computing efficiency and consumes large computing resources. KNN requires that the distance between each test point and each training data point be calculated, and the calculation of this distance must include all characteristics. The data set used in this article is not large, but KNN still takes a lot of time to run. Another point is that the KNN model is unstable and has weak repeatability. KNN can generate decision boundaries of arbitrary shape. Such decision boundaries have high variability because they depend on the combination of training samples [11,13].

SVM is an algorithm for linear and nonlinearly separable data (using kernel techniques). The only thing to do is to find the regularization term, and SVM works well in both low and high-dimensional data spaces. It can effectively work on high-dimensional datasets because the complexity of training datasets in SVM is usually represented by the number of support vectors rather than dimensions. Even if all other training examples and repeat the training are deleted, the optimal separating hyperplane is the same [12].

Although SVM can work on smaller training data sets because they do not depend on the entire data. Although SVM can work on smaller training datasets, they do not depend on the entire data. However, it is also not suitable for large datasets, because the training time of using SVM on large datasets may be high, the computational load is larger, and they are inefficient on noisy datasets with overlapping classes [12,13].

The only thing that CNN does is map inputs to outputs. It is able to discover an enormous number of mapping links between inputs and outputs without requiring explicit mathematical formulations to be provided [16,17]. As long as the convolution network is trained with known patterns, it will have the capacity to map input pairs to outputs.

The most significant advantage of CNN is, it shares the convolution kernel, so it can easily process high-dimensional data and does not need to manually select features. This work can get a good feature classification effect by training the weights well. But during the experiment, there are also some shortcomings of CNN. It needs to adjust parameters, needs a large number of samples, and GPU is better for training. The sample size this work used is relatively small, so the improvement of experimental results is not as obvious as this study expect [18,19].

5. Conclusion

In order to get an even higher level of precision in this investigation, it is possible to carry out by hand some alterations to the images, such as flipping, rotating, and cutting. The process is referred to as data augmentation. A third advantage of data operation is the development of the database's data capacity, which makes the training network more robust. This benefit comes as a direct result of the data operation. This study will continue its investigation by randomly clipping 4444 photographs during the training phase, mirroring those pictures, and then sending those pictures to the training phase. During the period of testing, an integrated strategy may be used to cut down on the number of outliers. The photos were cut and mirrored in the higher left, lower left, upper right, and lower right locations, respectively, as part of this research. This method increases the size of the database by a factor of ten, and then it supplies the model with ten different pictures. After that, the obtained probability is averaged, and the phrase that matches is considered to be the most accurate output classification. This method is quite effective in lowering the number of classification errors.

References

[1] Pietikainen, M. Hadid, A. Feng, X. Facial Expression Recognition with Local Binary Patterns and Linear Programming[J].Pattern recognition & image analysis: advances in mathematical theory & applications in the USSR.2005,15(2).

[2] Shan C. Gong S. Peter W. McOwan.Facial expression recognition based on Local Binary Patterns: A comprehensive study[J].Image & Vision Computing. 2009,27(6).

[3] Pitas I.,Kotsia I..Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines[J].IEEE Transactions on Image Processing.2007,16(1).

[4] McOwan P.W.,Anderson K..A real-time automated system for the recognition of human facial expressions[J].IEEE transactions on systems, man, and cybernetics, Part B. Cybernetics: A publication of the IEEE Systems, Man, and Cybernetics Society.2006,36(1).

[5] Pu Xiaorong, Zhou YI, Zhou Ruiyi.Face Recognition on Partial and Holistic LBP Features[J]. JEST.2012,(1).

[6] Zhang Chongsheng, Wang Pengyou, Chen Ke. Identity-aware convolutional neural networks for facial expression recognition[J].Systems Engineering and Electronics.2017,(4).

[7] Ruta A, Li Y, Liu X. Real-time traffic sign recognition from video by class-specific discriminative features[J]. Pattern Recognition, 2010, 43(1): 416-430.

[8] Deepak G, Joonwhoan L, Geometric. feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines.[J].Sensors (Basel, Switzerland). 2013, 13 (6). 7714-34.

[9] Xiang, C, Lin, H, Gu, W.Facial expression recognition using radial encoding of local Gabor features and classifier synthesis[J].Pattern Recognition: The Journal of the Pattern Recognition Society.2012,45(1).

[10] Yang Lei, Aghaabbasi M, Ali M, Jan A, Bouallegue B, Javed M, Salem N. Comparative Analysis of the Optimized KNN, SVM, and Ensemble DT Models Using Bayesian Optimization for Predicting Pedestrian Fatalities: An Advance towards Realizing the Sustainable Safety of Pedestrians[J]. Sustainability, 2022, 14 (17).

[11] Ghouchan N, Jalali M. A Graph-Based k-Nearest Neighbor (KNN) Approach for Predicting Phases in High-Entropy Alloys[J]. Applied Sciences,2022,12(16).

[12] Wang Zhoufeng, He Xiangqi, Shen Huiling, Fan Shangjie, Zeng Yilong. Multi-source information fusion to identify water supply pipe leakage based on SVM and VMD[J]. Information Processing and Management,2022,59(2).

[13] Khan M., Khanam S. Feature-based performance of SVM and KNN classifiers for diagnosis of rolling element bearing faults[J]. Vibroengineering PROCEDIA,2021,39.

[14] Nandyal S, Kattimani S. Umpire Gesture Detection and Recognition using HOG and Non-Linear Support Vector Machine (NL-SVM) Classification of Deep Features in Cricket Videos[J]. Journal of Physics: Conference Series,2021,2070(1).

[15] Song Yucong. Traffic sign recognition based on HOG feature extraction[J]. Journal of Measurements in Engineering,2021,9(3).

[16] Tantai Shuyu, Ma Xin. Interrelated Fusion CNN with Statistical Grouping among Multipatches for Occluded Facial Expression Recognition[C]. Proceedings of the 40th China Control Conference. 2021, 14:426-431.

[17] Atabansi C. A Real-Time Facial Expression Recognition System Using Normal Camera and Videos Based on Convolutional Neural Network Technique[D]. Southwest University,2021.

[18] Wang Sa, Zhengxin Cheng Xiaoming, Deng Liang, Chang Fuqing, Duan Ke Lu Leveraging. 3D blendshape for facial expression recognition using CNN[J].Science China(Information Sciences), 2020, 63 (02): 101-103.

[19] Cai Jun, Chang Quan, Tang Xianlun, Xue Can, Wei Chang. Facial Expression Recognition Method Based on Sparse Batch Normalization CNN[C]. Proceedings of the 37th China Control Conference, 2018:1034-1039.