Scientific Research Publishing

Journal of Computer and Communications, 2022, 10, 57-71
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Deep Learning-Based Emotion Detection

Yuwei Chen, Jianyu He

Dublin City University, Dublin, Ireland
Email: yuwei.chen36@mail.dcu.ie, jianyu.he2@mail.dcu.ie

How to cite this paper: Chen, Y.W. and He, J.Y. (2022) Deep Learning-Based Emotion Detection. *Journal of Computer and Communications*, **10**, 57-71.
https://doi.org/10.4236/jcc.2022.102005

Received: January 9, 2022
Accepted: February 25, 2022
Published: February 28, 2022

Abstract

In order to make artificial intelligence smarter by detecting user emotions, this project analyzes and determines the current type of human emotions through computer vision, semantic recognition and audio feature classification. In facial expression recognition, for the problems of large number of parameters and poor real-time performance of expression recognition methods based on deep learning, Wang Weimin and Tang Yang Z. *et al.* proposed a face expression recognition method based on multilayer feature fusion with lightweight convolutional networks, which uses an improved inverted residual network as the basic unit to build a lightweight convolutional network model. Based on this method, this experiment optimizes the traditional CNN MobileNet model and finally constructs a new model framework ms_model_M, which has about 5% of the number of parameters of the traditional CNN MobileNet model. ms_model_M is tested on two commonly used real expression datasets, FER-2013 and AffectNet, the accuracy of ms_model_M is 74.35% and 56.67%, respectively, and the accuracy of the traditional MovbliNet model is 74.11% and 56.48% in the tests of these two datasets. This network structure well balances the recognition accuracy and recognition speed of the model. For semantic emotion detection and audio emotion detection, the existing models and APIs are used in this experiment.

Keywords

Expression Recognition, CNN, Face Recognition, Semantic Recognition, Feature Fusion, Inverted Residual

1. Introduction

At present, chatbots are not really considered as “artificial intelligence” [1], they have great limitations, they can only predict the response to the user’s conversation based on big data, so the “intelligence” of chatbots need to be improved, in order to improve the “intelligence” of chatbots, chatbots need to analyze various factors, the most important of which is the analysis of the user's emotion, so it's important to study the facial emotion recognition and semantic emotion recognition to improve the intelligence of bots.

This project is a program written in Pycharm based on the Python environment that can analyze the current emotions through the current camera and microphone and with appropriate graphs. There are 7 types of computer vision and audio feature classification [2] in this project, namely anger, happy, scared, surprise, neutral, sad and disgust, and 4 types of semantic recognition [3], namely positive, neutral, negative and compound. Facial expressions are important information reflecting human emotions, and psychologists have found that the emotional information conveyed by facial expressions in human communication activities accounts for 55% of the overall information [4]. Therefore, this paper will focus on the study of face expression recognition, and in order to balance the recognition accuracy and recognition speed of the model, this paper build a lightweight FER model based on CNN multi-level feature fusion. A lightweight convolutional network model is built using a modified inverted residual network as the basic unit, and the deep and shallow information of CNN is fused on this basis to improve the expression recognition accuracy. In addition, for the problem that the fusion of deep and shallow features using fully connected layers directly generates a large number of parameters, the method of first filtering several shallow features in the convolutional network and then fusing them with deep features is adopted. Experiments were conducted on two commonly used expression datasets to verify the effectiveness of the proposed method. In this paper, various machine learning algorithms in audio emotion recognition have been tried, trained and evaluated on an emotion audio dataset, and finally found that XGBoost and MLP Classifier have higher accuracy, and chose to use vader library for emotion semantic recognition, and achieved more satisfactory results.

2. Literature Review

With big data and deep learning networks, artificial intelligence is getting smarter and smarter. Many people say that the least intelligent part of AI is because it has no emotion. If it is able to analyze the user's voice, facial expressions and other behaviors to give the user's emotions back to the robot, so that the robot knows the user's emotions, it will make the robot appear more intelligent. This can be used in the chatbot, if the chatbot knows the user's emotions, then does it become more humane to chat with the user? The key to the problem is how to use deep learning to train an efficient network model based on the data, one of the purposes of the experiments in this paper is also to investigate how to build a more efficient neural network to train the model.

The traditional FER method includes two steps of feature extraction and feature classification: the extracted features can be local binary patterns (LBP) [5], histogram of orientation gradients (HOG) [6] and wavelet features [7], etc.; the feature classification methods mainly include K-nearest neighbor method [8], support vector machine [9], Adaboost classifier [10] and neural network, etc. Although the traditional methods perform well on the laboratory pose data set, they are less effective in real complex and variable scenes. In recent years, with the rise of deep learning techniques, especially the wide application of convolutional neural networks (CNNs) in computer vision, many studies in traditional vision fields have started to adopt deep learning methods.FER, as a classical pattern recognition problem, has also seen a large number of studies on deep learning. Yu *et al.* [11] [12] both adopted the method of integrating multiple CNN models to improve the The performance of FER model was improved; Li *et al.* [13] proposed a Boosted-CNN method, which first trained a CNN model, based on which a random sampling method was used for unbalanced learning of each expression, and obtained better results. Mollahosseini *et al.* [14] improved the learning ability of the FER model by adding the Inception module to the network and achieved a high recognition rate with several data sets; Hongying Zhang *et al.* [15] proposed a two-channel convolutional neural network with one channel input to the LBP map and the other channel input to the gradient map to combine the two features for expression recognition, and achieved a high recognition rate with several The test results on several datasets showed that the recognition accuracy was improved. Several of the above methods used complex network structures to improve the accuracy of the FER model, and the recognition speed was limited. To improve the recognition speed, Arriaga *et al.* [16] combined the residual module and depth-separable convolution to design a streamlined network structure, which could recognize expressions in real time, but the structure was too simple and could only achieve baseline accuracy on the FER2013 dataset. Lv Zhan *et al.* [17] first used a face segmentation network to obtain the most relevant regions for expression recognition, and then recognized the segmented images by the FER network, both of which used lightweight networks and achieved good recognition accuracy and recognition speed. However, the method is complicated, firstly, a face segmentation dataset has to be constructed on the expression dataset, and then the face segmentation network and expression recognition network have to be trained, which is a tedious step. In summary, the existing FER method based on deep learning still cannot achieve the balance of recognition accuracy and recognition speed.

Zhurakovskaya [18] and Oxana present and analyze in detail different emotion analysis tools and techniques including Ekmanns and Plutchiks emotion models, Word Embedding (GloVe), VADER [2] sentiment analysis, emoji features and a Random Forest classifier, and create a new emotion analysis tool, and finally the results of other emotion analysis tools their methods achieve a 10% improvement in accuracy.

Kumar, Akshi [19] proposed a hybrid deep learning model, which enhances the advantages of combining deep learning networks with machine learning for systems that can process text and visual images. They proposed ConvNet-SVM BoVW model with four modules, namely, discretization, text analysis, image analysis and decision module. Finally the proposed model achieves an accuracy close to 91% [2].

Zhang Jiayi [20] introduced a dual waveform sentiment recognition model for dialogues, they designed a waveform attention module that captures sentiment features from both source and synthesized waveforms and uses an effectiveness coefficient mechanism for fine-grained multimodal information fusion and finally proposed a new sentiment detection module for dialogues. Their approach has been applied on IEMOCAP and SEMAINE datasets with good performance.

In this paper, the project uses models and libraries with higher recognition rates by comparing audio and semantic recognition models. Since the deep learning methods used in current face recognition do not balance well between recognition rate and recognition speed, this experiment build a face expression recognition model based on multilayer feature fusion with lightweight convolutional networks.

3. Proposed Method

1) *Audio Feature Extraction*

About audio feature extraction librosa [21] is a very useful audio feature extraction tool, the text plots various audio features and corresponding moods, to compare the changes of different mood features, so as to select the features related to mood changes. The main audio features acquired for this project are chromagram, root mean square, 2nd-order polynomial, spectral centroid, zero-crossing rate, Mel frequency cepstral coefficients and so on, there are 12 audio features in total. These features were then made into a data frame using pandas with a size of 2556*183. **Figure 1** shows the data visualization of audio features, which supports the selection of audio features. The size of this data visualization is 7*7, where the 7 columns are 7 emotional charts, namely anger, disgust, fear, happy, neutral, sad and surprise, and the 7 rows are 7 audio feature charts, namely chroma_stft, spectrogram, MFCCs, root mean square, spectral centroid, spectral_band width and 2nd-order polynomial. The comparison of the individual emotion visualizations shows that there are significant differences in each emotion profile, so machine learning algorithms can be used to distinguish between these emotions.

2) *Semantic Recognition*

VADER [2] (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool, published in the AAAI conference in 2014, it does not need to use text data for training, after installation After installation, you can input the text you want to recognize for sentiment analysis. In this project, this library is used to recognize the semantics [22] of sentiment for the current words spoken by people. The result is the probability or degree of 4 emotions, which are positive, negative, neutral and compound. **Figure 2** shows a test of the effectiveness of the VADER [2] library. First at all, 30 sentences about 7 emotions are made, then a human judgment about the positive and negative of

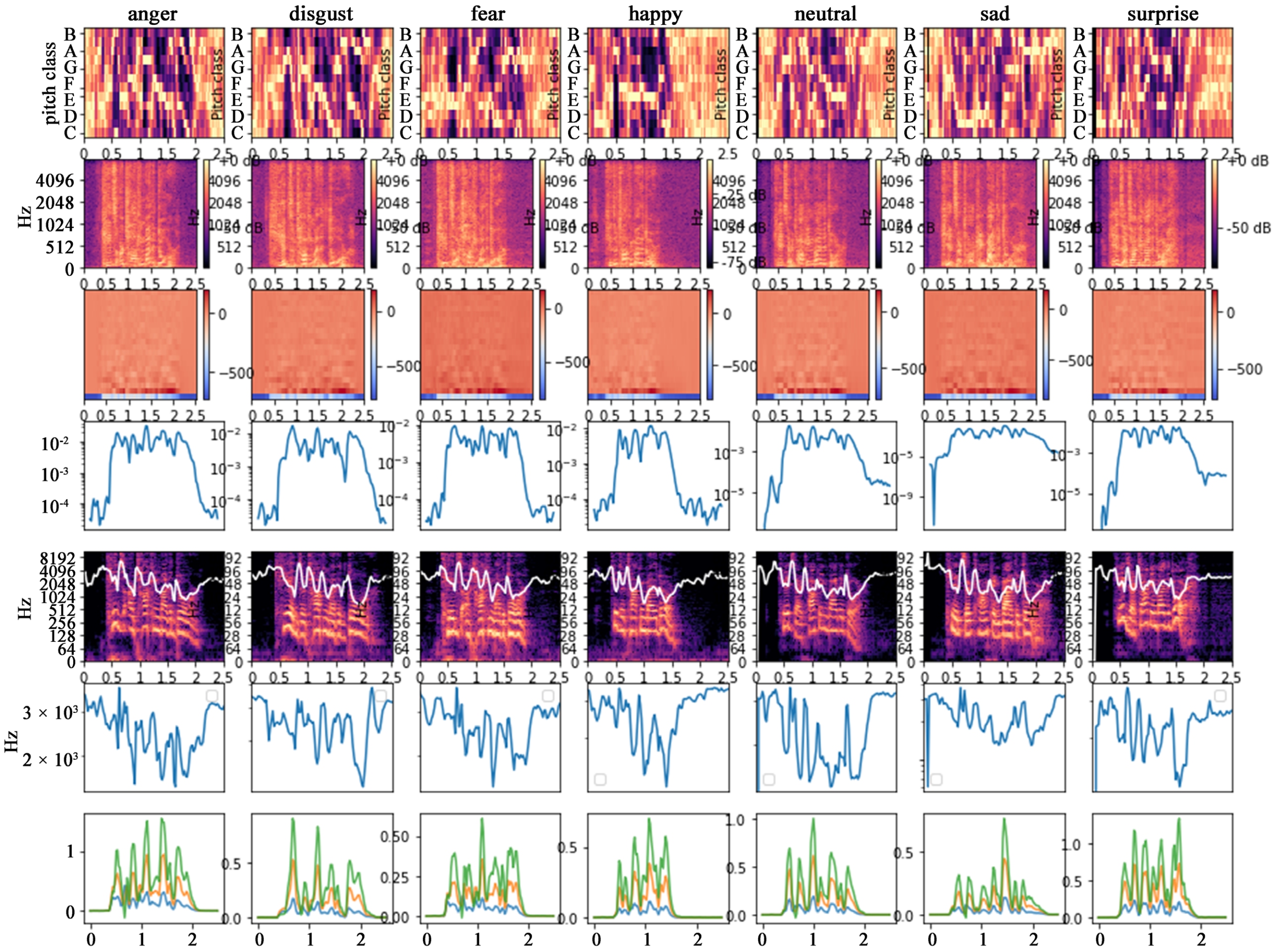
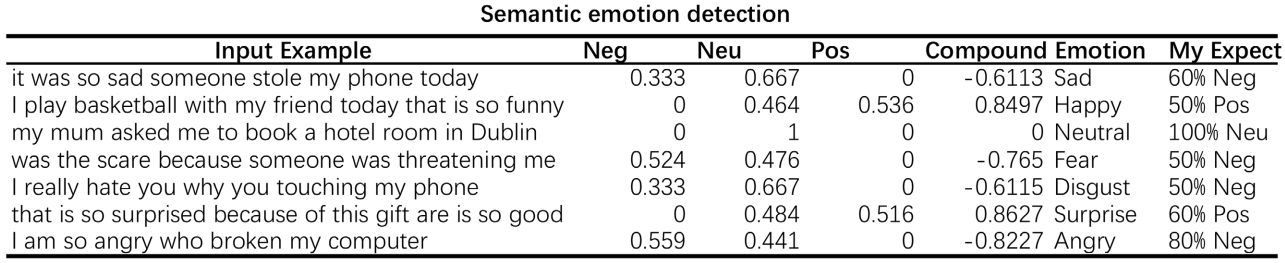


Figure 1. Data visualization of audio features.



Semantic emotion detection

Input Example	Neg	Neu	Pos	Compound Emotion		My Expect
it was so sad someone stole my phone today	0.333	0.667	0	-0.6113	Sad	60% Neg
I play basketball with my friend today that is so funny	0	0.464	0.536	0.8497	Happy	50% Pos
my mum asked me to book a hotel room in Dublin	0	1	0	0	Neutral	100% Neu
was the scare because someone was threatening me	0.524	0.476	0	-0.765	Fear	50% Neg
I really hate you why you touching my phone	0.333	0.667	0	-0.6115	Disgust	50% Neg
that is so surprised because of this gift are is so good	0	0.484	0.516	0.8627	Surprise	60% Pos
I am so angry who broken my computer	0.559	0.441	0	-0.8227	Angry	80% Neg

Figure 2. Text emotion recognition test.

these sentences also are made, then VADER [2] is used to judge these sentences, and finally the output results were made a table with the most representative 7 sentences. Figure 2 shows that the effectiveness of VADER [2] is about 70% by comparing the human judgment result with the output result of VADER [2], so VADER [2] can meet the needs.

3) *Facial expression recognition.*

a) *Overall network architecture design*

In the CNN model, different convolutional layers, due to different convolutional depths, have smaller relative receptive fields in the feature maps of the shallow layers, which are more sensitive to local feature information, and larger relative receptive fields in the feature maps of the deep layers, which are more sensitive to overall contour information. The studies in the literature [23] [24] used fully connected layers to fuse the features of the deep and shallow layers of the CNN model, which improved the model performance but also increased the model parameters substantially. In order to reduce the number of parameters generated by fusion, the shallow features are first filtered, and then the deep and shallow features of the convolutional network are fused. According to WANG Weimin and TANG Yang Z *et al.* [25], a multilayer feature fusion method for face expression recognition based on lightweight convolutional networks, the overall architecture of the convolutional network fusing multi-layer features as shown in **Figure 3** is designed.

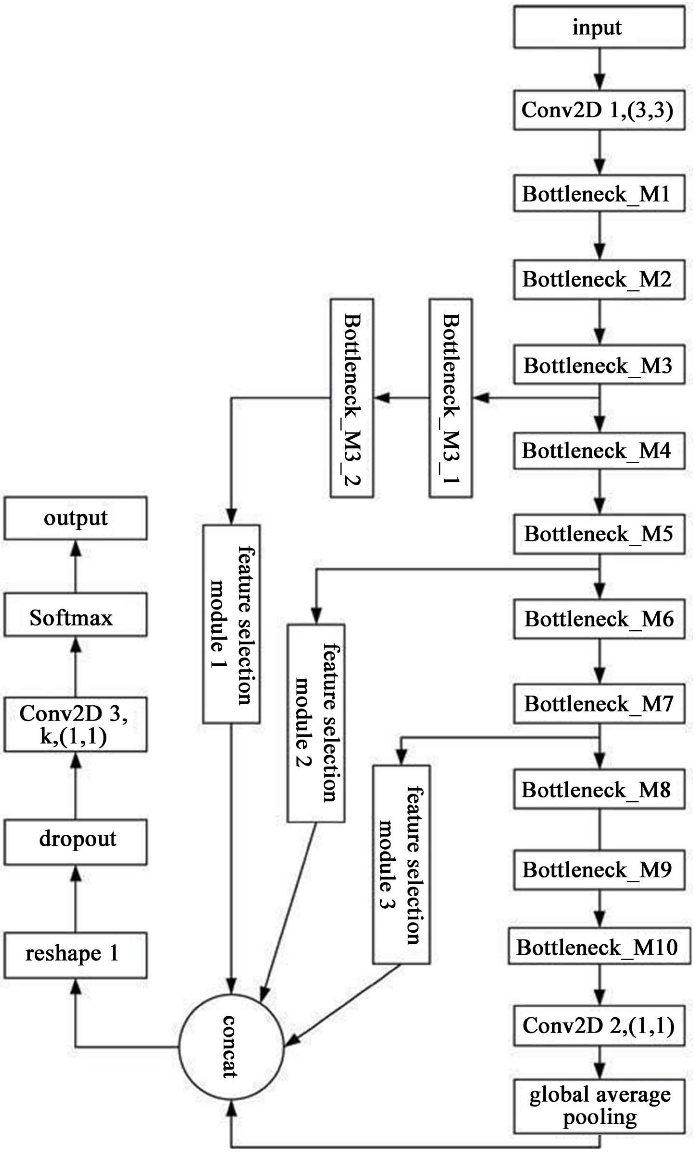


Figure 3. Overall architecture of designed convolutional networks incorporating multi-level features.

The network model is called ms_model, and the input image is firstly convolved with a normal convolution of size 3 × 3 (Conv2D 1) for initial feature extraction, and then connected with 10 modified inverted residual units (Bottleneck_M) for further feature extraction in a lighter way; then the feature maps of certain intermediate layers are filtered to obtain more refined intermediate layer features, These features are fused with the features output from the convolutional network backbone by the feature merge operation (concat) to achieve deep and shallow feature fusion and improve the model performance; then the size of the fused features is reshaped to 1 × 1 by the deformation operation (reshape 1), and some neurons are discarded with a certain probability using the discard operation to prevent overfitting, and 1 × 1 convolution (Convolution 1 × 1 (Conv2D 3) is used to change the number of feature maps to the same as the number of expression categories k. Finally, the probability vector of expression categories is output by Softmax function and deformation operation (reshape 2). The specific parameter settings are shown in Table 1, where c is the number of

Table 1. Parameters of the CNN.

Layer name	c	s	t	Output size
Conv2D 1 (3 × 3)	16	2		56 × 56 × 16
Bottleneck_M1	16	1	1	56 × 56 × 16
Bottleneck_M2	24	2	5	28 × 28 × 24
Bottleneck_M3	24	1	5	28 × 28 × 24
Bottleneck_M3_1	32	1	5	28 × 28 × 32
Bottleneck_M3_2	32	1	5	28 × 28 × 32
Feature selection module 1				32
Bottleneck_M4	32	2	5	14 × 14 × 32
Bottleneck_M5	32	1	5	14 × 14 × 32
Feature selection module 2				32
Bottleneck_M6	40	1	5	14 × 14 × 40
Bottleneck_M7	40	1	5	14 × 14 × 40
Feature selection module 3				40
Bottleneck_M8	40	1	5	14 × 14 × 40
Bottleneck_M9	48	2	5	7 × 7 × 48
Bottleneck_M10	64	1	5	7 × 7 × 64
Conv2D 2 (1 × 1)	64	1		7 × 7 × 64
Global average pooling				64
Concat				168
Reshape 1				1 × 1 × 168
Dropout				1 × 1 × 168
Conv2D 3 (1 × 1)	k			1 × 1 × k
Softmax				1 × 1 × k
Reshape 2				k

convolution kernels, s is the step size, and t is the expansion multiple of Bottleneck_M cell.

b) *Improved inverted residual cell Bottleneck_M*

The Bottleneck_M unit refers to the inverted residual structure in MobileNetV2 [26]. The basic principle of inverted residual structure is to expand the channel of the feature map first, use separable convolution to extract features when the number of channels of the feature map is large, and then compress the channel of the feature map at last. The experiments show that the inverted residual structure is beneficial to the extraction of image features by separable convolution when the number of channels in the feature map is large, and the number of parameters in the structure is also small because the 1 × 1 convolution is used for the channel change of the feature map.

The main improvement of the inverted residual structure is to use the Mish activation function [27] to replace the ReLU6 activation function in the original structure. Compared with ReLU6, the Mish activation function is smoother and slightly allows for negative values, which makes it easier to update the convolutional network parameters. **Figure 4** shows the structure of the modified inverted residual network. In this structure, the input feature map of the backbone part is firstly expanded by 1 × 1 convolution, then the features are extracted by 3 × 3 deep separable convolution, and finally the channels are compressed by 1 × 1 convolution; the input feature map of the branch part is directly output by 1 × 1 convolution operation to ensure the same number of channels as the backbone feature map, and the two outputs are superimposed at the output.

In **Table 1**, c in the Bottleneck_M cell is the number of convolution kernels of the 1 × 1 convolution with channel compression effect, and t is the expansion multiple of the 1 × 1 convolution with channel expansion effect on the input feature map.

c) *Multi-layer feature fusion design*

The feature selection module for filtering the feature maps output from the middle layer is shown **Figure 5**.

Bottleneck_M3 outputs a feature map of 28 × 28 × 24. The input map is first passed through two Bottleneck_M modules to increase the depth of the feature

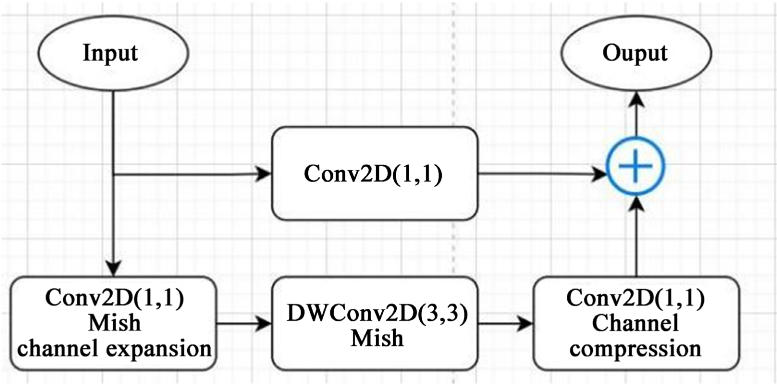


Figure 4. Bottleneck_M structure.

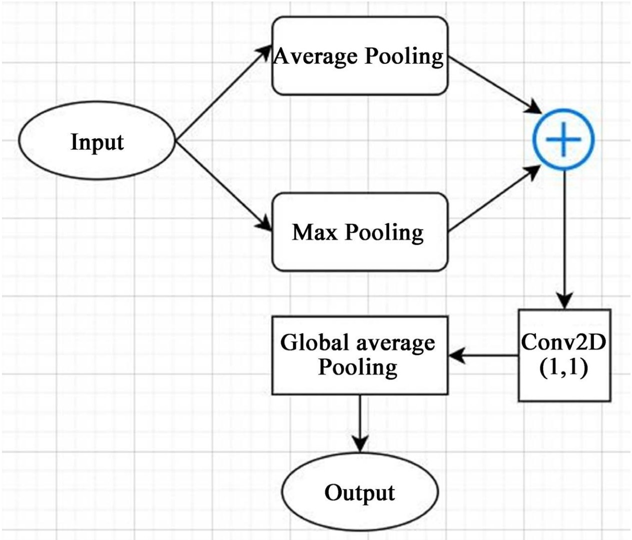


Figure 5. Process of feature selection module.

map, and the feature map size becomes 28 × 28 × 32; then the maximum pooling and average pooling with a pooling kernel of 4 × 4 and a step size of 4 are used to obtain two 7 × 7 × 32 feature maps, and the elements of the two feature maps are The two feature maps are superimposed to obtain a 7 × 7 × 32 feature map; then 1 × 1 convolution and global average pooling are performed to obtain a 32-dimensional feature vector. For Bottleneck_M5 (feature map size 14 × 14 × 32) and Bottleneck_M7 (feature map size 14 × 14 × 40), it is no need to increase the depth of the output feature map, and directly use the maximum pooling and average pooling operations with a pooling kernel of 2 × 2 and a step size of 2, and then perform element superposition, 1 × 1 convolution and global average pooling operations to obtain 32- and 40-dimensional feature vectors, respectively. The 32- and 40-dimensional feature vectors are obtained. After feature filtering, the feature maps with dimensions of 28 × 28 × 24, 14 × 14 × 32 and 14 × 14 × 40 are transformed into 32-dimensional, 32-dimensional and 40-dimensional feature vectors, respectively.

The multi-layer feature fusion design is shown in **Figure 3**. Let the output feature maps of Bottleneck_M3, _M5, _M7 and _M10 be X_3, X_5, X_7 and X_{10}, respectively, and the output of the backbone network be X_{10} (size 7 × 7 × 64). feature selection module) is set to G, then the feature map becomes:

$$O_1 = G_1(X_3) \tag{1}$$

$$O_2 = G_2(X_5) \tag{2}$$

$$O_3 = G_3(X_7) \tag{3}$$

where: O_1 and O_2 are both 32-dimensional vectors, and O_3 is a 40-dimensional vector. After fusion of these filtered features, then

$$O_{\text{fusion}} = [O_1, O_2, O_3, O_{10}] \tag{4}$$

where: [−] is the feature vector merging operation; O_{fusion} is the 168-dimensional vector.

If use the fully connected layer for deep and shallow feature fusion, it needs to first flatten the feature maps X_3 (size 28 × 28 × 24), X_5 (size 14 × 14 × 32), X_7 (size 14 × 14 × 40) and X_{10} (size 7 × 7 × 64) into vectors, and then stitch these four vectors together to finally obtain the dimension of the fused feature vector as 28 × 28 × 24 + 14 × 14 × 32+14 × 14 × 40 + 7 × 7 × 64 = 36,064, which is 214 times of the dimension of the feature vector obtained from the proposed ms_model_v1 model.

4) *Softmax classification*

The Softmax function outputs the probability of the expression category using the Softmax function, which converts each dimension of a multidimensional vector input to between [0,1], the expression is:

$$p_i = \frac{\mathrm{e}^{x_i}}{\sum_{i=1}^{k} \mathrm{e}^{x_i}}, \quad i = 1, 2, \cdots, k \tag{5}$$

where: x_i is the input of the Softmax function; p_i is the probability of the ith class.

The loss value of the convolutional network model is calculated using cross-entropy, and the formula is.

$$\mathrm{loss}(y, z) = -\sum_{i=1}^{k} z_i \log(y_i) \tag{6}$$

where: y_i is the predicted value of the input sample; z_i is the true value of the input sample; y and z are also the predicted and true values of the sample, respectively, but are r-dimensional vectors. The cross-entropy loss function is used as the objective function of the optimized convolutional network model.

4. Evaluation and Testing

The experiments were conducted on a remote cloud server, Ubuntu 16.04 operating system, using Python programming language and Keras deep learning framework to build a convolutional network model. The computer hardware configuration is Intel E5430 CPU, 32G RAM, GPU model: NVIDIA Tesla V100-16G, number of GPUs: 1. Our emotion recognition system project is built on a local windows system using Pycharm based python environment.

1) *Results and evaluation of audio feature training*

a) *Datasets for audio expression recognition* The dataset for audio feature extraction is a dataset called “Speech Emotion Recognition” downloaded from kaggle, the size of this dataset is 1gb, and there are about 2559 wav audio files, they are assigned to 7 folders of basic emotions of human faces, The 7 emotions are neutral, happy, surprised, sad, angry, disgusted, and fear. Each emotion has approximately 436 wav files and includes both male and female voices.

b) *Training Results and Evaluation*

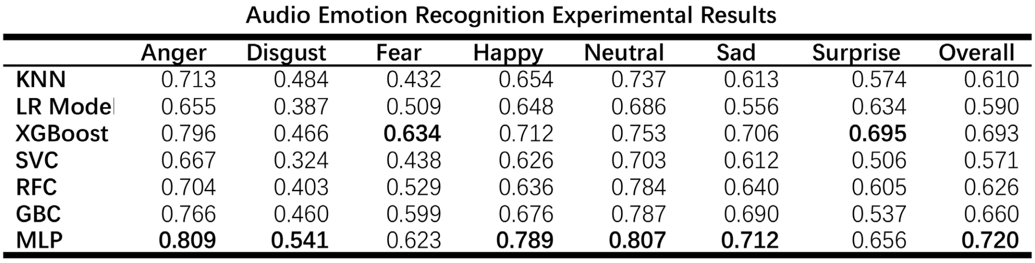
After processing the audio file, a data frame is obtained. Each line of data frame has a corresponding sentiment label, and then different machine learning algorithms need to be used and tested to come up with the best results. Sklearn [28] is a very useful machine learning library based on the Python environment, and this library can implement a machine learning algorithm with a few lines of code. This model use sklearn [28] to divide 75% of this dataset into the training set and the other 25% into the test set, and then use 7 different machine learning algorithms to train and predict the training and test sets. The seven machine learning algorithms are k-nearest neighbor, logistic regression model, XGBoost, support vector classifier, random forest classifier, gradient boosting classifier, and MLPC classifier. This experiment is based on changing the parameters of these functions to improve the accuracy of the algorithm. The change of the hidden layer size of MLPClassifier has a small improvement on the accuracy, the biggest improvement is when the hidden unit is 10,500 and the hidden layer is 1, and the accuracy decreases when the hidden unit is more or less than 10,500. **Figure 6** shows the F1 scores of each algorithm for each emotion, where MLP's accuracy is significantly higher than the other algorithms by a small margin, but XGBoost has some advantages in predicting emotions such as fear and surprise. But overall, MLP has the highest accuracy.

2) *Evaluation for facial expression recognition*

a) *Datasets for facial expression recognition* To verify the effectiveness of the proposed method, two public datasets of facial expressions, FER-2013 [29] and AffectNet [30], were used for testing. fer2013 [29], the Kaggle facial expression recognition challenge dataset, the facial expression dataset consists of 35,886 facial expression images, have 28,708 training image, PublicTest and PrivateTest 3589 each, each image is composed of a grayscale image with a fixed size of 48 × 48, with 7 expressions corresponding to the numerical labels 0 - 6, respectively, 1 disgust; 2 fear; 3 happy; 4 sad; 5 surprised; 6 normal.

AffectNet [30], this database contains more than 1,000,000 images from the Internet, which were obtained by searching on multiple search engines using emotional keywords. The data in AffectNet were labeled into eight expressions: expressionless, happy, sad, surprised, scared, disgusted, angry, and contemptuous, and included a manual labeling part and an automatic labeling part. The manual annotation part is relatively more reliable, so this experiment mainly uses this part of the dataset.

The proposed convolutional network model training input is a 112 × 112 face expression image, batch size is 32 and epoches number is 500, An adaptive moment estimation (Adam) optimizer was used with an initial learning rate of 0.001, and the learning rate was decayed by 0.5 when the loss value of the test set



Audio Emotion Recognition Experimental Results

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise	Overall
KNN	0.713	0.484	0.432	0.654	0.737	0.613	0.574	0.610
LR Model	0.655	0.387	0.509	0.648	0.686	0.556	0.634	0.590
XGBoost	0.796	0.466	**0.634**	0.712	0.753	0.706	**0.695**	0.693
SVC	0.667	0.324	0.438	0.626	0.703	0.612	0.506	0.571
RFC	0.704	0.403	0.529	0.636	0.784	0.640	0.605	0.626
GBC	0.766	0.460	0.599	0.676	0.787	0.690	0.537	0.660
MLP	**0.809**	**0.541**	0.623	**0.789**	**0.807**	**0.712**	0.656	**0.720**

Figure 6. Figure training results of model.

was no longer decreasing. In the training, Keras online data augmentation strategy was used to expand the data, mainly including random rotation, horizontal and vertical translation, shear transformation, random scaling, and random horizontal flipping of the expression data.

b) *Experimental results and analysis*

The metric measured in this experiment is the overall accuracy of the model on the test set, *i.e.*, the ratio of the number of correctly classified images to the overall number. The training results of different models are represented in the following Figure 7.

MobileNet is a traditional CNN model using fully connected layers, and ms_model_R and ms_model_M are the models for deep and shallow feature fusion on the basis of MobileNet. Even the accuracy of ms_model_M is higher than that of MovblieNet, which proves the effectiveness of the proposed method for deep and shallow feature fusion.

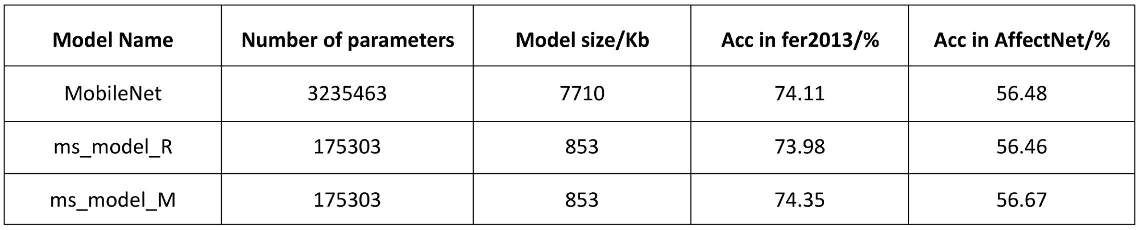
Both ms_model_R and ms_model_M perform the fusion of deep and shallow features, the difference is that ms_model_R is the baseline model built using the original inverted residual structure (using the ReLU6 activation function), while ms_model_M is the baseline model built using the modified inverted residual structure (using the Mish activation function). The accuracy of ms_model_M is a little higher than that of ms_model_R in both datasets, which proves the effectiveness of the proposed improved inverse residual structure. In summary, our new model framework ms_model_M (using shallow and deep feature fusion and using the improved inverted residual structure) has the best results.

The accuracy of the models on the FER-2013 dataset in other papers is generally between 65% and 78%, while the accuracy of the models on the AffectNet dataset is between 50% and 60%, probably due to the large amount of data and the complexity of the image types in the AffectNet dataset. The reasons for this will be investigated in future work.

5. Conclusions and Future Work

For speech recognition, this experiment achieved 72% accuracy in the kaggle dataset using MLP classifier, Although this dataset is 1 gb in size, it only has 2550 files, so the training results are not perfect. However, the model is able to predict the frequency of the user's voice on ui, and is able to meet our expectations.

For audio recognition, the experiment directly call VADER's emotion semantic recognition library to recognize text emotion, which has better emotion



Model Name	Number of parameters	Model size/Kb	Acc in fer2013/%	Acc in AffectNet/%
MobileNet	3235463	7710	74.11	56.48
ms_model_R	175303	853	73.98	56.46
ms_model_M	175303	853	74.35	56.67

Figure 7. Accuracy of different models.

recognition accuracy and can basically meet experiment needs.

In the face expression recognition, according to Wang Weimin and Tang Yang Z *et al.*, the experiment adopt the improved inverted residual structure to design the network model, so that the model has good feature extraction ability while being lightweight, and on this basis, this experiment fuse the deep and shallow features of the convolutional network, the accuracy of the model is further improved by integrating deep and shallow features of the convolutional network. The experimental results of these two models on two public datasets, FER-2013 and AffectNet, show that the performance of the proposed method is comparable with the direct fusion of fully connected layers, but the number of parameters is only 5% of the latter method, which achieves a better balance between recognition accuracy and recognition speed.

Finally, the project can be run on local desktop and local server based on Django, because this project designed 2 interfaces to better visualize the results of voice emotion recognition, audio emotion recognition and facial expression recognition.

The downside is that this project only displays the results of voice emotion recognition, audio emotion recognition and facial expression recognition separately, do not give them separate weights to output a total emotion result, which is the next step the project need to do. Besides, this project will focus on improving the accuracy of speech and audio recognition by training the models ourselves with different algorithms in the future.

Group Work

As a group, the team have own division of labour, with Yuwei focusing on the implementation and research of the facial expression recognition model and Jianyu focusing on the implementation and research of semantic recognition and audio recognition. Model for face recognition, and used it for the facial expression recognition part of the project. Jianyu searched for a dataset to train audio emotion recognition, and used this dataset to extract audio features and train and test algorithms to evaluate each emotion recognition algorithm, and chose one with better performance to predict emotions. Yuwei bought a remote server and trained our facial expression recognition model on the cloud GPU. It took us about six days to train a lot of models. Originally this project planned to use django to design the project interface, initially Jianyu used djano to design the web UI, but Yuwei felt that the desktop UI was easier to pass data between functions, so he created the desktop interface and Jianyu continued to design the web interface. However, the desktop interface did not turn out very well because when program tried to run the semantic recognition and audio recognition it caused a lag in the expression recognition, whereas the web interface did not have this problem, and in the end Yuwei and Jianyu kept own design.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Beck, J., Stern, M. and Haugsjaa, E. (1996) Applications of AI in Education. *XRDS: Crossroads, The ACM Magazine for Students*, **3**, 11-15. https://doi.org/10.1145/332148.332153

[2] Ye, J., Zhu, J. and Jiang, A. (2020) Facial Expression Recognition: A Survey. *Journal of Data Acquisition and Processing*, **35**, 21-34.

[3] Li, S. and Deng, W.H. (2020) Deep Facial Expression Recognition: A Survey. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/TAFFC.2020.2981446

[4] Mehrabian, A. and Russell, J.A. (1974) An Approach to Environmental Psychology. The MIT Press, Cambridge, 56-63.

[5] Shan, C.F., Gong, S.G. and McOwan, P.W. (2009) Facial Expression Recognition Based on Local Binary Patterns: A Comprehensive Study. *Image and Vision Computing*, **27**, 803-816. https://doi.org/10.1016/j.imavis.2008.08.005

[6] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*'05), San Diego, 20-26 June 2005, 886-893.

[7] Bashyal, S. and Venayagamoorthy, G. (2008) Recognition of Facial Expressions Using Gabor Wavelets and Learning Vector Quantization. *Engineering Applications of Artificial Intelligence*, **21**, 1056-1064. https://doi.org/10.1016/j.engappai.2007.11.010

[8] Dino, H.I. and Abdulrazzaq, M.B. (2019) Facial Expression Classification Based on SVM, KNN and MLP Classifiers. 2019 *International Conference on Advanced Science and Engineering* (*ICOASE*), Zakho-Duhok, 2-4 April 2019, 70-75. https://doi.org/10.1109/ICOASE.2019.8723728

[9] Zhong, W. and Huang, Y. (2017) A Facial Expression Recognition Algorithm Based on Feature Fusion and Hierarchical Decision Tree Technology. *Computer Engineering & Science*, **39**, 393-398.

[10] Wang, Y., Ai, H., Wu, B. and Huang, C. (2004) Real Time Facial Expression Recognition with Adaboost. *Proceedings of the* 17*th International Conference on Pattern Recognition* (*ICPR* 2004), Cambridge, 23-26 August 2004, 926-929.

[11] Yu, Z. and Zhang, C. (2015) Image Based Static Facial Expression Recognition with Multiple Deep Network Learning. *Proceedings of the* 2015 *ACM on International Conference on Multimodal Interaction*, Seattle, 9-13 November 2015, 435-442. https://doi.org/10.1145/2818346.2830595

[12] Zhou, T., Lü, X.Q., Ren, G.Y., *et al*. (2020) Facial Expression Classification Based on Ensemble Convolutional Neural Network. *Laser & Optoelectronics Progress*, **57**, Article ID: 141501. https://doi.org/10.3788/LOP57.141501

[13] Li, H. and Li, G. (2019) Research on Facial Expression Recognition Based on LBP and Deep Learning. 2019 *International Conference on Robots & Intelligent System* (*ICRIS*), Haikou, 15-16 June 2019, 94-97.

[14] Mollahosseini, A., Chan, D. and Mahoor, M.H. (2016) Going Deeper in Facial Expression Recognition Using Deep Neural Networks. 2016 *IEEE Winter Conference on Applications of Computer Vision* (*WACV*), Lake Placid, 7-10 March 2016, 1-10.

[15] Zhang, H. and Wang, H. (2020) Double-Channel Facial Expression Recognition Based on Local Binary Pattern and Gradient Features. *Laser & Optoelectronics Progress*, **57**, Article ID: 141005. https://doi.org/10.3788/LOP57.141005

[16] Arriaga, O., Valdenegro-Toro, M. and Plöger, P. (2017) Real-Time Convolutional Neural Networks for Emotion and Gender Classification. arXiv:1710.07557.

[17] Lü, H., Tong, Q. and Yuan, Z. (2020) Real Time Architecture for Facial Expression Recognition in Complex Scenes Based on Face Region Segmentation. *Computer Engineering and Applications*, **56**, 134-140.

[18] Haric, Z. and Hollen, O. (2020) Supplemental Material for People Respond with Different Moral Emotions to Violations in Different Relational Models: A Cross-Cultural Comparison. *Emotion*.

[19] Malhotra, Y. (2018) AI, Machine Learning & Deep Learning Risk Management & Controls: Beyond Deep Learning and Generative Adversarial Networks: Model Risk Management in AI, Machine Learning & Deep Learning. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3193693

[20] Qi, Q., Lin, L. and Zhang, R. (2021) Feature Extraction Network with Attention Mechanism for Data Enhancement and Recombination Fusion for Multimodal Sentiment Analysis. *Information*, **12**, 340-344. https://doi.org/10.3390/info12090342

[21] About Audio Feature Extraction Librosa Is a Very Useful Audio Feature Extraction Tool, the Text Plots Various Audio Features and Corresponding Moods.

[22] Batbaatar, E., Li, M. and Ryu, K.H. (2019) Semantic-Emotion Neural Network for Emotion Recognition from Text. *IEEE Access*, **7**, 111866-111878. https://doi.org/10.1109/ACCESS.2019.2934529

[23] Li, Y., Lin, X.Z. and Jiang, M.Y. (2018) Facial Expression Recognition with Cross-Connect LeNet-5 Network. *Acta Automatica Sinica*, **44**, 176-182.

[24] He, Z.H., Zhao, L.Z. and Chen, C. (2018) Convolution Neural Network with Multi-Resolution Feature Fusion for Facial Expression Recognition. *Laser & Optoelectronics Progress*, **55**, Article ID: 071503. https://doi.org/10.3788/LOP55.071503

[25] Wang, S.H., Govindaraj, V.V., Górriz, J.M., Zhang, X. and Zhang, Y.D. (2021) Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network. *Information Fusion*, **67**, 208-229. https://doi.org/10.1016/j.inffus.2020.10.004

[26] Sandler, M., Howard, A., Zhu, M.L., Zhmoginov, A. and Chen, L.C. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[27] Misra, D. (2019) Mish: A Self Regularized Non-Monotonic Neural Activation Function. arXiv:1908.08681.

[28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., *et al*. (2011) Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.

[29] Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., *et al*. (2013) Challenges in Representation Learning: A Report on Three Machine Learning Contests. *International Conference on Neural Information Processing*, Daegu, 3-7 November 2013, 117-124. https://doi.org/10.1007/978-3-642-42051-1_16

[30] Mollahosseini, A., Hasani, B. and Mahoor, M.H. (2019) Affectnet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, **10**, 18-31. https://doi.org/10.1109/TAFFC.2017.2740923