Facelook

Project Report

Group Members:

Wu Junxian

Wang Kedi

Ying Zhuozuo

Zong Jiulin

31/10/2021

Ⅰ Background and Objective

1.1 Background

Facial expression is one of the most crucial ways to transfer emotional information and coordinate interpersonal relationships within all kinds of emotional expressions. Facial expression recognition refers to the selection of facial expression states from static photos or video sequences to recognize people’s emotional and psychological changes. If we get someone’s emotional information, we can understand his psychological condition, analyze his real-time reaction, or roughly predict his behavior.

But sometimes expressions can be misleading. Humans are masters at hiding emotions. For social or psychological considerations, people will control their expressions. The concept of micro-expression was first proposed by Haggard and Isaacs in 1966. Unlike "traditional" facial expressions which can be consciously controlled, micro-expression is caused by psychological inhibition of unconscious contraction of lower muscles. The micro-expression has the characteristics of short duration (1/25 to 1/5s) and low action intensity, so it is difficult for humans to capture such information (recognition accuracy rate only 40%). But it is crucial that it can be used for lying detection, providing compelling evidence of lies. It can also be used in the treatment process by psychologists to accurately grasp a patient's emotions to make a better diagnosis.

1.2 Objective

Our objective is to recognize facial expression and micro-expression on videos and images.

For facial expression, we construct an emotion recognition model according to the training data. Predict selected test data and identify the state of the face from seven emotions. ['happy', 'angry', 'disgusted', 'neutral', 'fearful', 'sad', 'surprised’].

For micro-expression, we extract features of the frame of a video and classify micro-expression from four categories.
[‘disgust’, ‘happiness’, ‘repression’, ‘surprise’].

Ⅱ Usage of tools and Technology

2.1 Tools

2.1.1 Facial Expression Recognition

2.1.1.1 Paddlex

PaddleX integrates the tasks of image classification, object detection, semantic segmentation, and instance segmentation in the intelligent vision field and integrates the whole process of deep learning development from data preparation, model training, and optimization to multi-terminal deployment end-to-end. It also provides the unified task API interface and graphical development interface Demo. Developers do not need to install different kits, and they can quickly complete the whole process of development in the form of low code [1].

2.1.1.2 Paddleclas

PaddleClas is an image recognition toolset for industry and academia, helping users train better computer vision models and apply them in real scenarios. And it can use YAML to change config which is an easy and fast way to experiment [2].

2.1.1.3 albumentations

Albumentations is a Python library for fast and flexible image augmentations. Albumentations efficiently implements a rich variety of image transform operations that are optimized for performance and does so while providing a concise, yet powerful image augmentation interface for different computer vision tasks, including object classification, segmentation, and detection. Besides, it is easy to add the self-made data augmentation and function as a pytorch\keras plug-in [3].

2.1.1.4 EasyDL

Easydl is based on Baidu independent research and development of paddlepaddle which is combined with advanced engineering service technology. Besides, it can be a simple and easy-to-use model training and service platform [4].

2.1.1.5 PaddleDetection

PaddleDetection is an end-to-end object detection development kit based on PaddlePaddle, which aims to help developers in the whole development of constructing, training, optimizing, and deploying detection models in a faster and better way. It can easily adjust the config by YAML file [5].

2.1.2 Facial Micro-Expression Recognition

2.1.2.1 sklearn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project [6].

2.1.2.2 TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on the training and inference of deep neural networks [7].

Tensorflow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

TensorFlow has APIs available in several languages both for constructing and executing a TensorFlow graph. The Python API is at present the most complete and the easiest to use, but other language APIs may be easier to integrate into projects and may offer some performance advantages in graph execution [7].

2.1.3 Front-end Deployment

We use PyQt5 to design a simple user interface. UsersUsers can open local videos, images or open a webcam to capture live streams. Then we call the MediaPipecall the MediaPipe API to detect human faces and predict their expression in real-time.

PyQt5 is a set of Python bindings for the Qt5for the Qt5 application framework from Digia. Qt library is one of the most powerful GUI libraries.PyQt5 is implemented as a set of Python modules. It has over 620 classes and 6000 functions and methods [8].

Mediapipe is an open-source multimedia machine learning model application framework developed by Google research. It offers cross-platform, customizable ML solutions for live and streaming media [8]. We use its face detection module to implement fast real-time detection and recognition.

2.2 Technology

2.2.1 Facial Expression Recognition

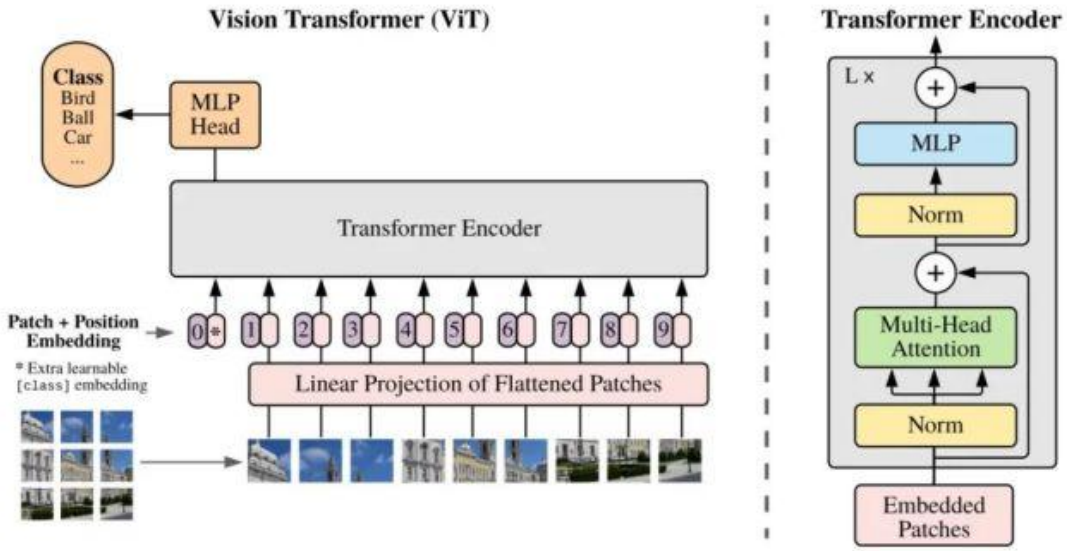
2.2.2.1 ResNet101V2

The difference between ResNet101V2between ResNet101V2 and ResNet50V2 lies in the depth of the model. The model is ResNet series with a depth of 101 layers. The model is implemented based on the model structure proposed in *Deep Residual Learning for Image Recognition*. Image classification tasks can be used, such as cat and dog classification, flower classification, and so on. The user provides a series of labeled data sets, and the algorithm will be loaded into the pre-training model on imagenet-1000 for migration learning on the user data set. The models generated after training can be directly deployed as online services or batch services on the ModelArts platform, and support reasoning using CPU, GPU, or Ascend 310 [9].

2.2.2.2 Vision Transformer

The classic transformer is divided into two modules: encoder and decoder, and the most important part is multi-head attention. The decoder part here will have an additional mask, which is mainly used to prevent the model from seeing future information during decoding because it is input together. For example, if you want to decode 10 words by inputting 8 words, you need to mask the next 9 words when decoding the first word [9].

However, Vision Transformer is a simple way to achieve SOTA in vision classification with only a single transformer encoder [10].



2.2.2.3 DarkNet53

Darkenet53 is part of the backbone of the yolov3 network. Darknet is an open-source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation [11].

Number	Block name	Type	(kernel, stride, padding)	Input size	Output size	output
	Conv Block	Conv	3, 1, 1	416, 416, 3	416, 416, 32	
		Batch Normalization	/			
		LeakyReLu	/			
		Conv	3, 2, 1	416, 416, 32	208, 208, 64	
×1	Residual Block	Conv	1, 1, 0	208, 208, 64	208, 208, 32	
		Batch Normalization	/			
		LeakyReLu	/			
		Conv	3, 1, 1	208, 208, 32	208, 208, 64	
		Batch Normalization	/			
		LeakyReLu	/			
		Block output	/		208, 208, 64	
		Conv	3, 2, 1	208, 208, 64	104, 104, 128	
×2	Residual Block	Conv	1, 1, 0	104, 104, 128	104, 104, 64	
		Batch Normalization	/			
		LeakyReLu	/			
		Conv	3, 1, 1	104, 104, 64	104, 104, 128	
		Batch Normalization	/			
		LeakyReLu	/			
		Block output	/		104, 104, 128	
		Conv	3, 2, 1	104, 104, 128	52, 52, 256	
×8	Residual Block	Conv	1, 1, 0	52, 52, 256	52, 52, 128	
		Batch Normalization				
		LeakyReLu				
		Conv	3, 1, 1	52, 52, 128	52, 52, 256	
		Batch Normalization				
		LeakyReLu				
		Block output	/		52, 52, 256	Output Feature
		Conv	3, 2, 1	52, 52, 256	26, 26, 512	
×8	Residual Block	Conv	1, 1, 0	26, 26, 512	26, 26, 256	
		Batch Normalization	/			
		LeakyReLu	/			
		Conv	3, 1, 1	26, 26, 256	26, 26, 512	
		Batch Normalization	/			
		LeakyReLu	/			
		Block output	/		26, 26, 512	Output Feature
		Conv	3, 2, 1	26, 26, 512	13, 13, 1024	
×4	Residual Block	Conv	1, 1, 0	13, 13, 1024	13, 13, 512	
		Batch Normalization	/			
		LeakyReLu	/			
		Conv	3, 1, 1	13, 13, 512	13, 13, 1024	
		Batch Normalization	/			
		LeakyReLu	/			
		Block output	/		13, 13, 1024	Output Feature

2.2.2.4 DenseNet264

Unlike ResNet, DenseNet does not stack features before they flow to the next layer but integrates features through the connection between layers. This connection seems to produce more parameters, but in fact, DenseNet requires fewer parameters than traditional neural networks. Because the network does not need to learn redundant characteristic graphs. In the traditional feedforward network, each layer of the network needs to obtain the input from the previous layer, and then transfer it to the next layer after transformation. ResNet explicitly passes the information to be retained into the subsequent layer through an identity transformation. However, because each layer of ResNet has its weight parameters, the total number of parameters is more. DenseNet makes a distinction between the information input into the network and the information to be retained. The layer of DenseNet is very narrow, and only a small part of the feature map is added to the common information in the network, and the other feature maps remain unchanged. The final classifier makes the final discrimination based on all the feature maps in the network [12].

Another advantage of DenseNet is optimized information and gradient flow. Because each layer is directly connected to the gradient of the loss function. More importantly, dense join has the function of regularization, which can reduce the overfitting of smaller data sets [12].

2.2.2.5 MobileNetV3_large_ssld

Mobilenetv3 is a network proposed based on mobilenetv2. It has less computation and fewer parameters. Compared with other lightweight networks, it has achieved good results. At the same time, resnetxt101_ 32x16d_ WSL is a teacher model, and mobilenetv3 is distilled by using ssld (simple semi-supervised label knowledge distillation)_ Large model as pre-training model; Compared with the original mobilenetv3 pre-training model, mobilenetv3_ The accuracy of ssld pre-training model on Imagenet dataset is improved by 3%, which helps users further improve the effect of model training on the user-defined dataset. This example shows how to train through paddlex on only a small data set [12] [13].

2.2.2.6 Xception71

Xception is another improvement of Inconcept V3 proposed by Google after Inconcept. It mainly uses depthwise separable revolution to replace the convolution operation in the original Inconcept V3 [13] [14].

In Xception, the depthwise separable revolution is mainly used, that is, the traditional convolution operation is divided into two steps. Assuming that the original convolution is 3 * 3, the depthwise separable revolution is to convolute the input M feature maps one-to-one with M 3 * 3 convolution cores, and generate m results without summation; Then n 1 * 1 convolution kernels are used to normally convolute the M results generated in front, sum them, and finally generate N results [13]. [14].

As an improvement of Inconcept v3, Xception mainly introduces the depthwise separable revolution based on Inconcept v3, which improves the effectiveness of the model without increasing the network complexity [13] [14].

2.2.2.7 ShuffleNetV2

In the ShuffleNetV1 module, 1x1 group convolution is widely used, which violates the principle that "excessive use of group convolution will increase memory access cost". In addition, V1 adopts a bottleneck layer similar to ResNet, with different input and output channels, which violates the principle of "minimizing memory access with the same channel size". At the same time, using too many groups also violates the principle that "network fragmentation will reduce parallelism". There is a lot of element level add operations in short-circuit connection, which violates the principle of "element level operations cannot be ignored" [14] [15].

To improve the defects of V1, V2 introduces a new operation: channel split. Specifically, in the beginning, the input characteristic graph is divided into two branches in the channel dimension: the number of channels is C 'and C-C', and C '= C / 2 in actual implementation. The left branch is mapped equally, and the right branch contains three consecutive convolutions, and the input and output channels are the same. Moreover, the two 1x1 convolutions are no longer group convolutions, and the other two branches are equivalent to having been divided into two groups. The output of the two branches is no longer the Add element, but concat together, followed by channel shuffle of the concat results of the two branches to ensure the exchange of information between the two branches. Concat and channel shuffle can be combined with the channel split of the next module unit to form an element-level operation [14] [15].

2.2.2 Facial Micro-Expression Recognition

2.2.2.1 HOOF

HOOF was originally proposed for motion recognition. Because motion recognition is based on linear dynamic systems, the extracted features belong to Euclidean geometry and are vulnerable to the influence of background and image size. However, the histogram does not belong to Euclidean geometry, so the feature extraction process can be improved by the histogram of oriented optical flow [16].

First, optical flow is computed at every frame of the window and the peripheral angle is divided into 2B regions. Each flow vector is binned into one region according to its primary angle from the horizontal axis and weighted according to its magnitude. Thus, all-optical flow vectors, $v = [x; y]^T$ with direction, $\theta = tan^{-1}(\frac{y}{x})$ in the range

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta \leq -\frac{\pi}{2} + \pi\frac{b}{B}$$

will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin b, $1 \leq b \leq B$, out of a total of B Bins. Finally, the histogram is normalized, to sum up to 1 [16].

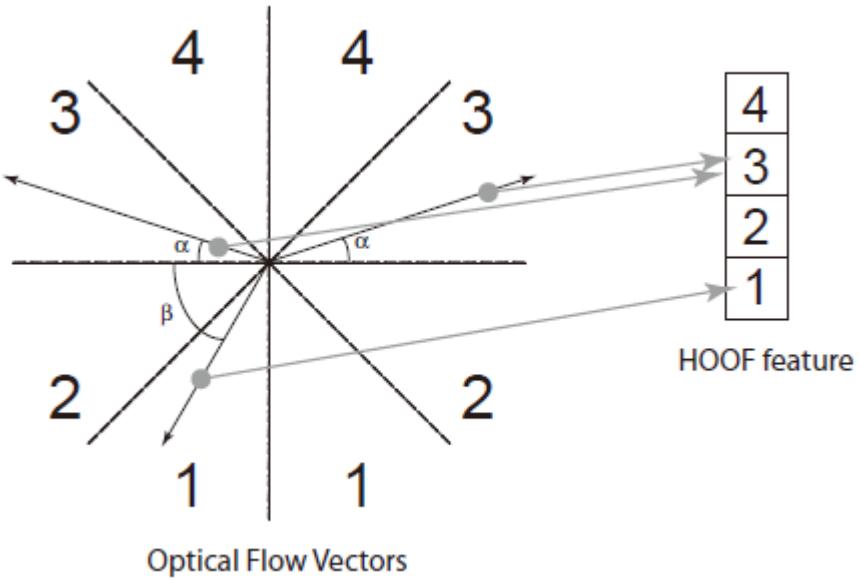


Figure 2. Histogram formation with four bins, $B = 4$

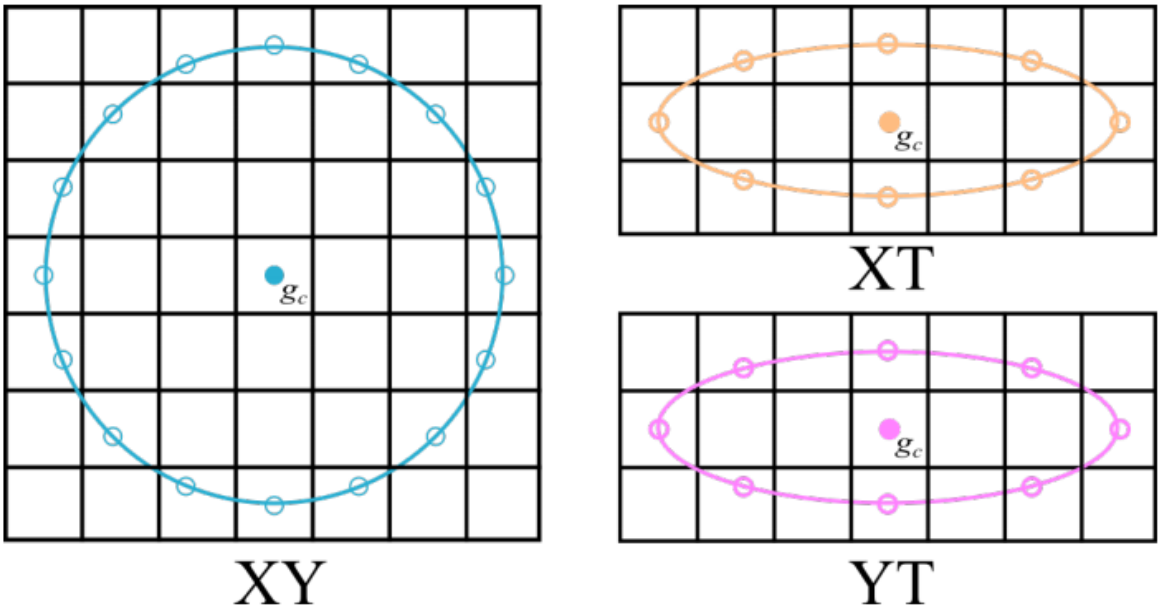
Binning according to the primary angle, the smallest signed angle between the horizontal axis and the vector, allows the histogram representation to be independent of the (left or right) direction of motion. Normalization makes the histogram representation scale-invariant [16].

Then this method can be utilized to recognize micro-expression. Amplitude and direction of motion can be computed in each frame and generate histograms. Calculate the mean of histograms and use this to represent video sequences.

2.2.2.2 LBP-TOP

Local Binary Pattern (LBP) is a descriptor originally proposed to describe the local appearance in an image. The key idea behind it is that the relative brightness of neighboring pixels can be used to describe the local appearance in a geometrically and luminously robust manner. The basic LBP feature extractor relies on two free parameters, called R and P. Uniformly sample the point P on the circle with the pixel as the center of the radius R, and obtain the brightness relative to the center pixel (the brightness is greater than or not greater than one bit of information), allowing the P bit number to characterize the neighborhood [17].

In micro-expression recognition, to encode the spatiotemporal symbiosis pattern, LBP-TOP (local binary pattern on three orthogonal planes) is used to extract the LBP features of the spatial (XY) plane, horizontal Spatio-temporal (XT) plane, and vertical Spatio-temporal (YT) planes in the image sequence. Neighborhood sampling is now performed on circles on the pure space plane and ellipses on the space-time plane [18].



$$LBP(M,R) = \sum_{m=0}^{M-1} s(g_m - g_c)2^m$$

2.2.2.3 SVM

In machine learning, support vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997[citation needed]) SVMs are one of the most robust prediction methods, is based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space to maximize the width of the gap between the two

categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

2.2.2.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.[1][2] Random decision forests correct for decision trees' habit of overfitting to their training set.[3]: 587–588 Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.[4][5]

2.2.2.5 PCA

The principal components of a collection of points in real coordinate space are a sequence of p unit vectors, where the i-th vector is the direction of a line that best fits the data while being orthogonal to the first i-1 vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest.

Ⅲ Design and model

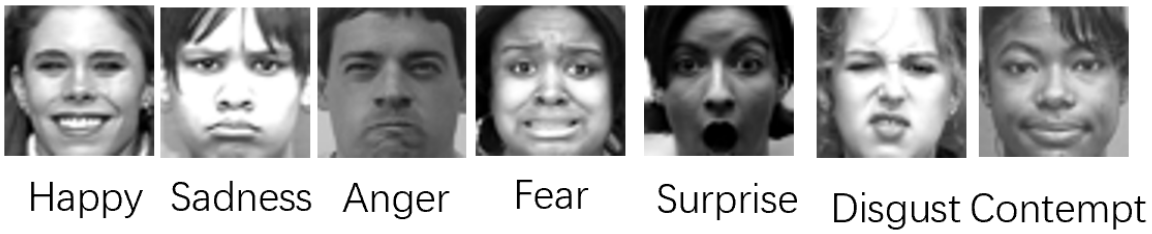
3.1 Facial Expression Recognition

3.1.1 Datasets

3.1.1.1 CK+48 dataset

The CK+ dataset is an extension of the CK dataset. It contains 327 labeled facial videos. We extracted the last three frames from each sequence in the CK+ dataset, which contains a total of 981 facial expressions. We use 10-fold Cross-validation in the experiment [19].

The number of sequences is increased by 22% and the number of subjects by 27%. The target expression for each sequence is fully FACS (Facial Actions Code System) coded and emotion labels have been revised and validated. In addition to this, non-posed sequences for several types of smiles and their associated metadata have been added [19].



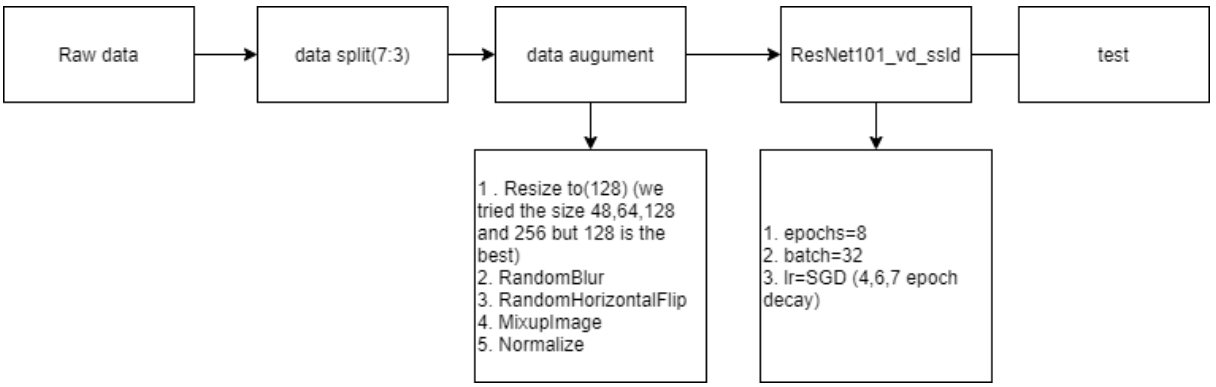
3.1.1.2 XunFei facial expression competition

Including training set and test set. The training set contains 28000 face images, and the test set contains 7000 face images. The size of all images is 48 * 48 pixels. The emotional tags included in the dataset include the following 7 categories: happy, sad, angry, fearful, surprised, disgusted, and neutral [20].

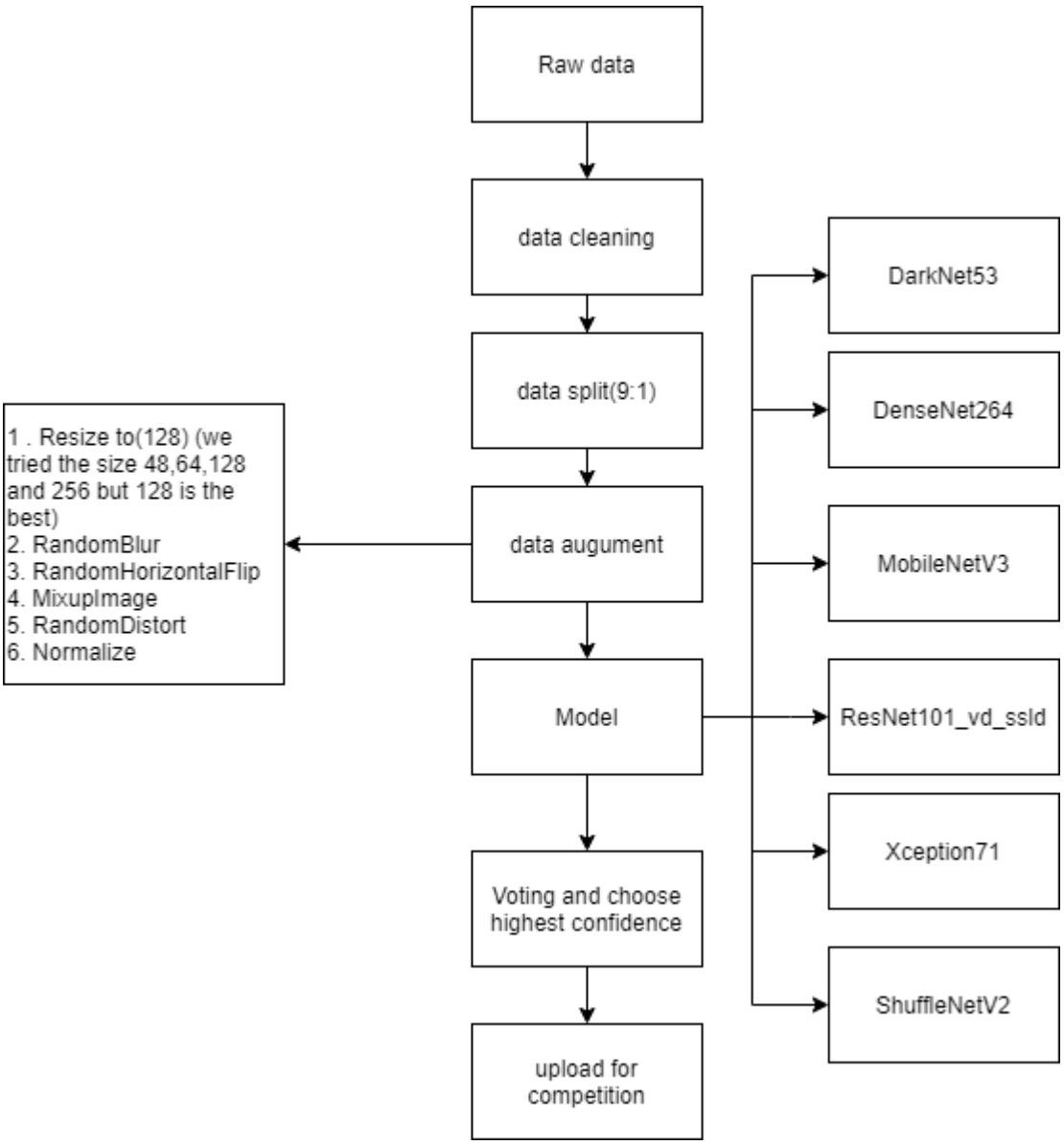
3.1.2 Model design

3.1.2.1 PaddleX pipeline

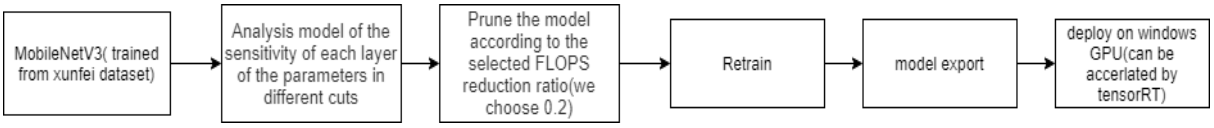
For CK+48 dataset:



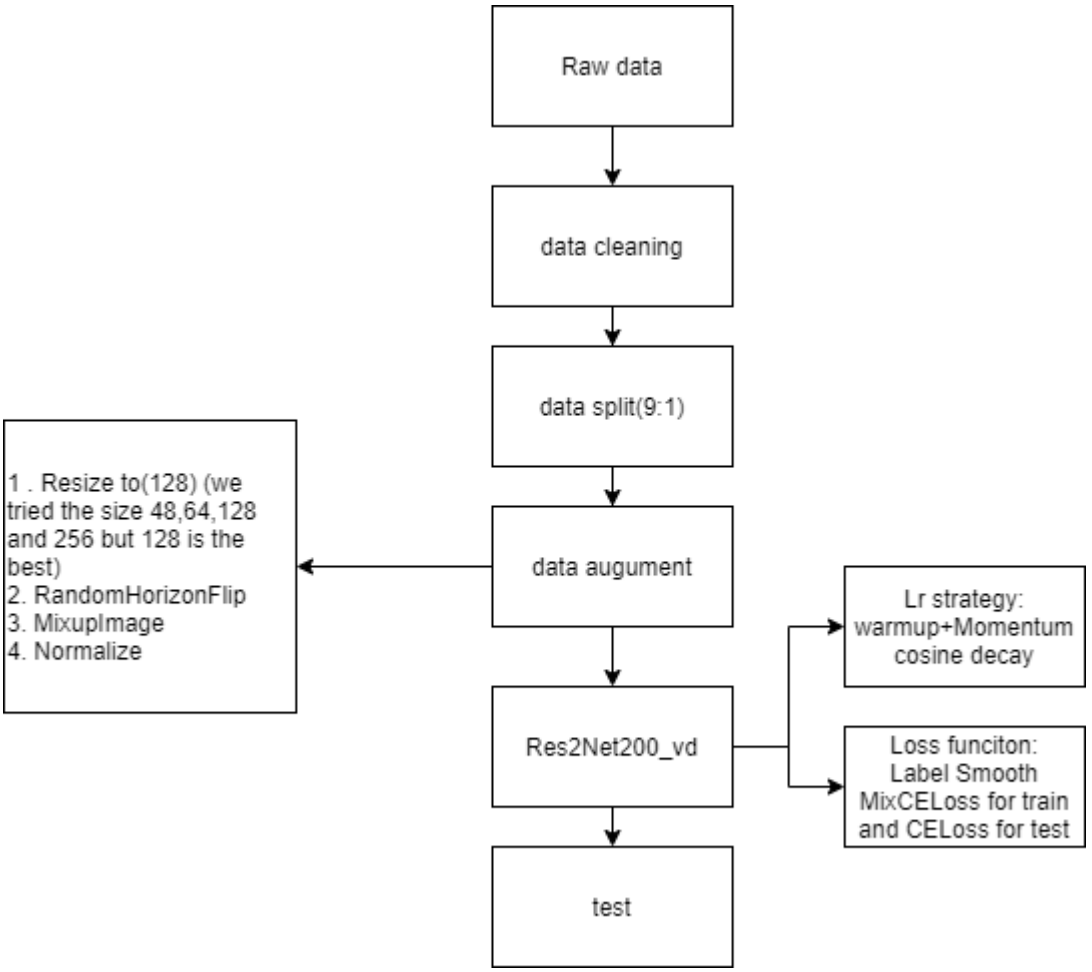
For xunfei emotion data set:



For real-time detection deployment:

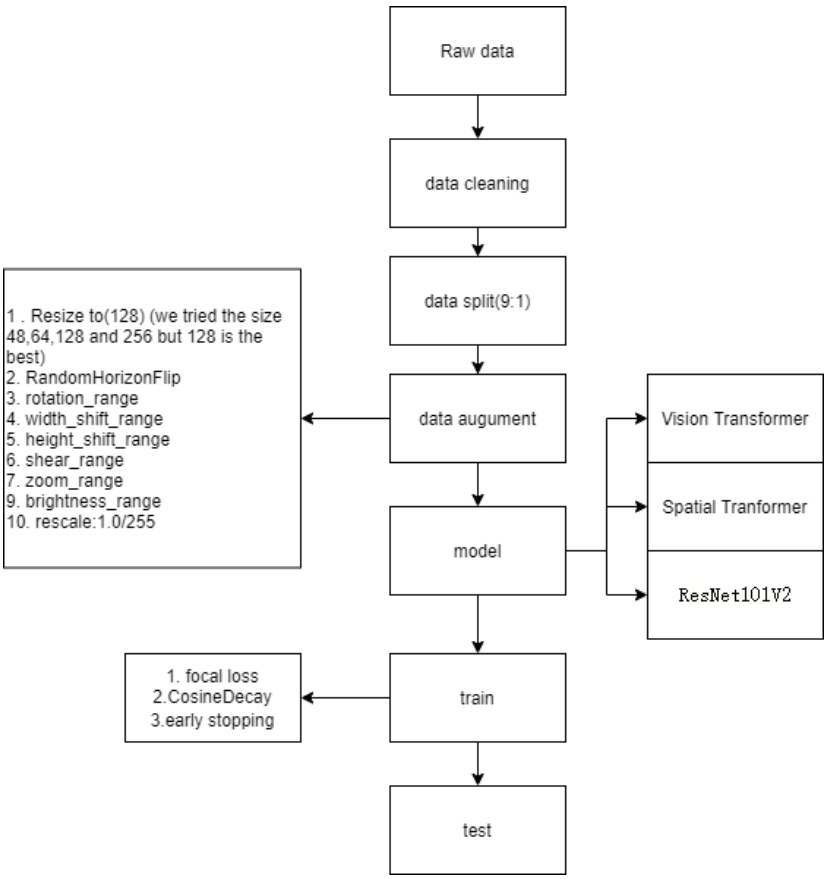


3.1.2.2 PaddleClas pipeline

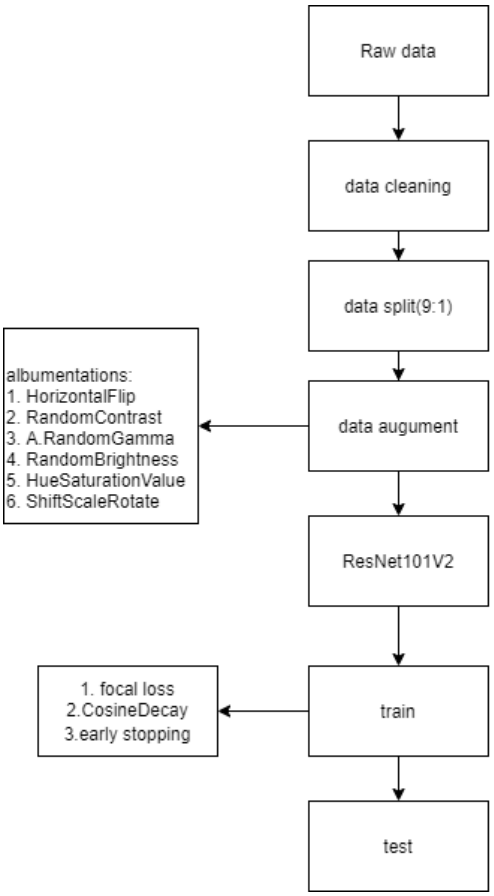


3.1.2.3 keras pipeline

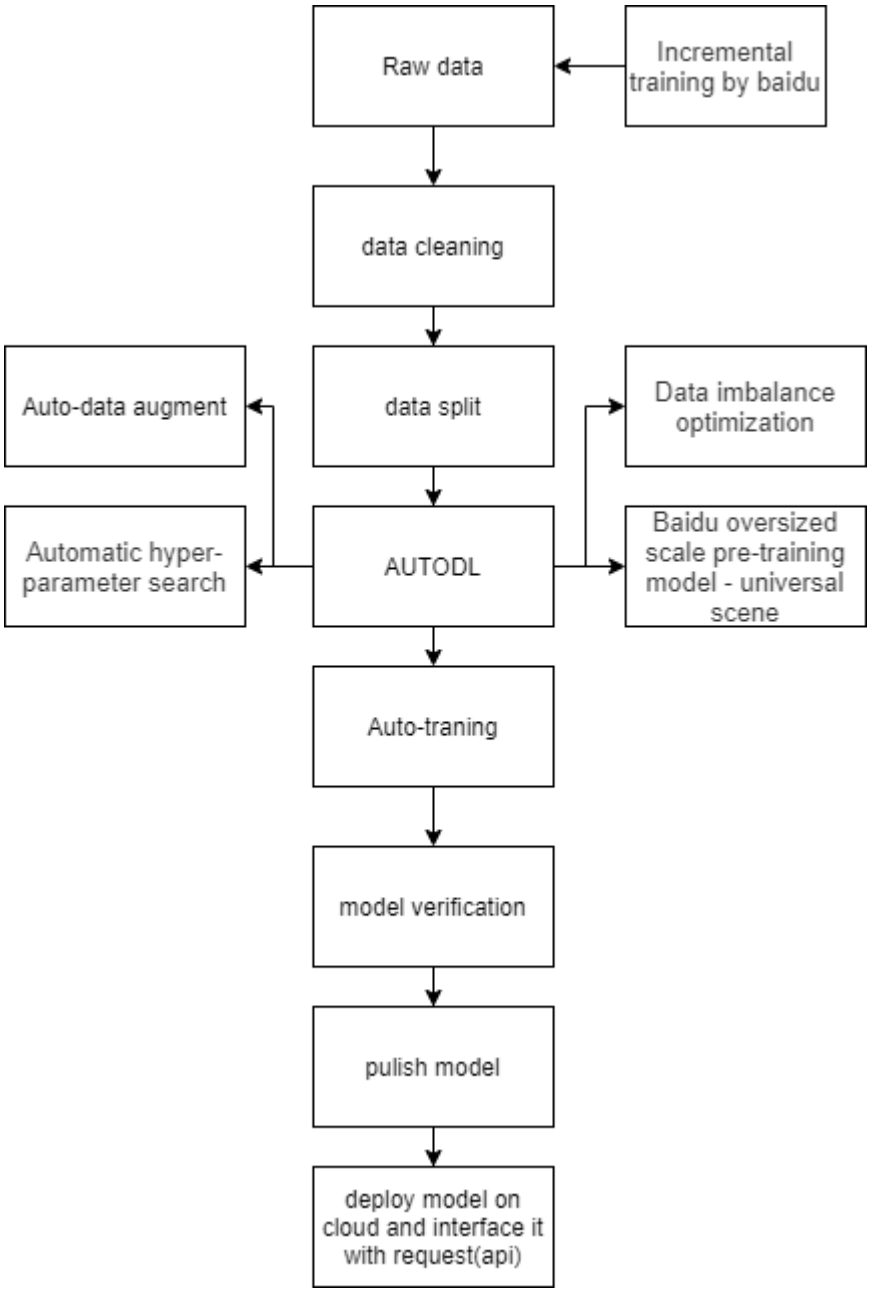
V1.0:


Raw data
data cleaning
data split(9:1)
1 . Resize to(128) (we tried the size 48,64,128 and 256 but 128 is the best)
2. RandomHorizonFlip
3. rotation_range
4. width_shift_range
5. height_shift_range
6. shear_range
7. zoom_range
9. brightness_range
10. rescale:1.0/255
data augument
Vision Transformer
Spatial Tranformer
model
ResNet101V2
1. focal loss
2.CosineDecay
3.early stopping
train
test

V2.0:


Raw data
data cleaning
data split(9:1)
albumentations:
1. HorizontalFlip
2. RandomContrast
3. A.RandomGamma
4. RandomBrightness
5. HueSaturationValue
6. ShiftScaleRotate
data augument
ResNet101V2
1. focal loss
2.CosineDecay
3.early stopping
train
test

3.1.2.4 easydl pipeline



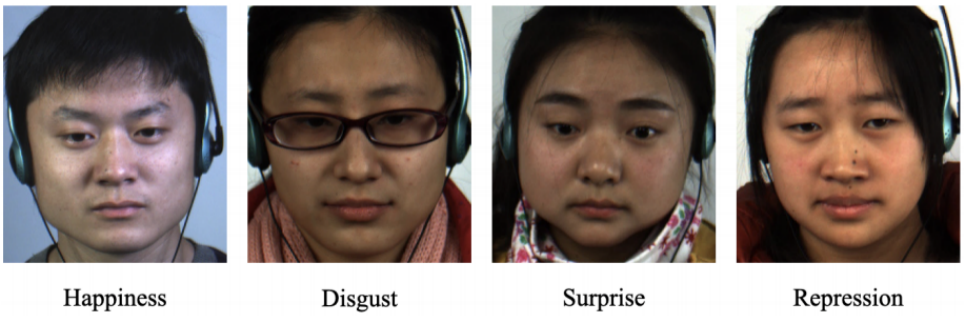
3.1.2.5 object detection pipeline

3.2 Facial Micro-Expression Recognition

3.2.1 Datasets

3.2.1.1 CASME II

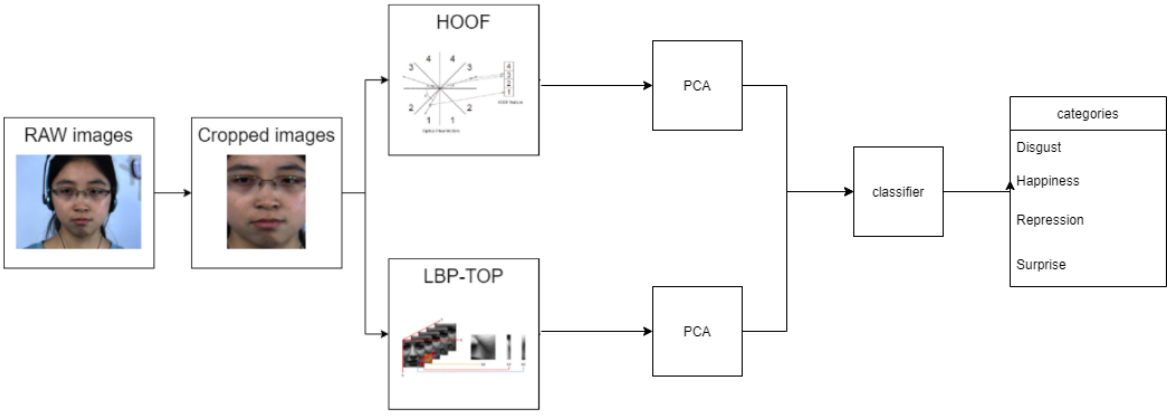
Chinese Academy of Sciences Micro-Expression II is an improved dataset version of casme. The sample has increased to 247 and 26 valid recipients. The average age is about 22 years old. The data are obtained under uniform illumination. Compared with casme, expressions are divided into five emotional categories, which are broader, including happiness (33 samples), disgust (60 samples), surprise (25 samples), depression (27 samples), and others (102 samples). The face size is cut to 280 pixels and 340 pixels [21].



CASME II also solved the lighting problems in previous data sets by providing a stable and high-intensity lighting environment. In addition, it has greater temporal and spatial resolution and a larger CASM sample size than CASME [21].

3.2.2 System Design

3.2.2.1 MLbased system



3.2.3 Details Design

3.2.3.1 Dataprepare

The experiment mainly uses the casme II data set and uses the pre-cut facial micro-expression frame sequence as the input. Firstly, the subjects 29 EP02 in the original data set are eliminated_ 07 and subject 9 ep02_ 02F after the two label-free samples, the only two "fear" label samples and the label "sadness" of four samples are eliminated. At the same time, 99 other label samples that cannot classify micro expressions are eliminated. Finally, a total of 150 samples are selected from the four categories of "disgust", "happiness", "repression" and "surprise". The maximum number of frames of the sample is 141 and the minimum number of frames is 24

3.2.3.2 Feature extraction

3.2.3.2.1 HOOF

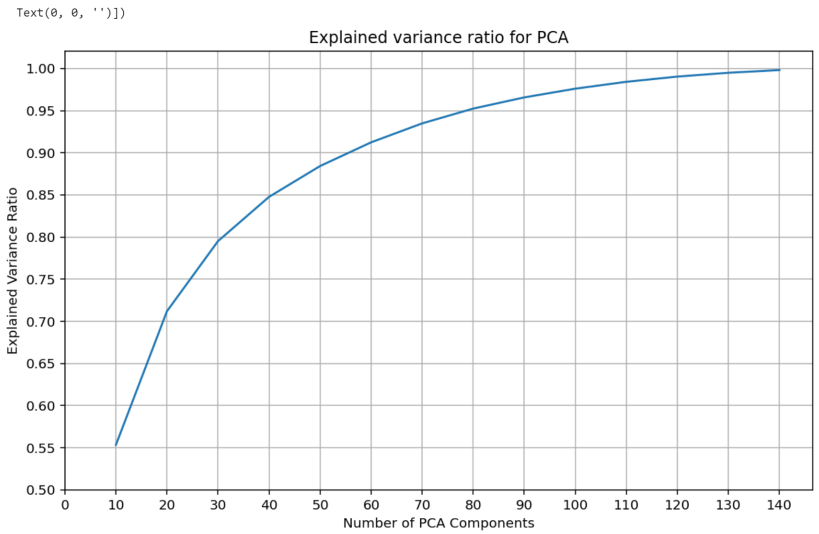
Because the size of video frames between different samples will be different, bilinear interpolation is used to cut to the same size (224 * 224). Because the number of hoof features extracted under different frame numbers is different, the optical flow frames are averaged to make the feature dimensions of hoof histograms of different samples consistent

3.2.3.2.2 LBP-TOP

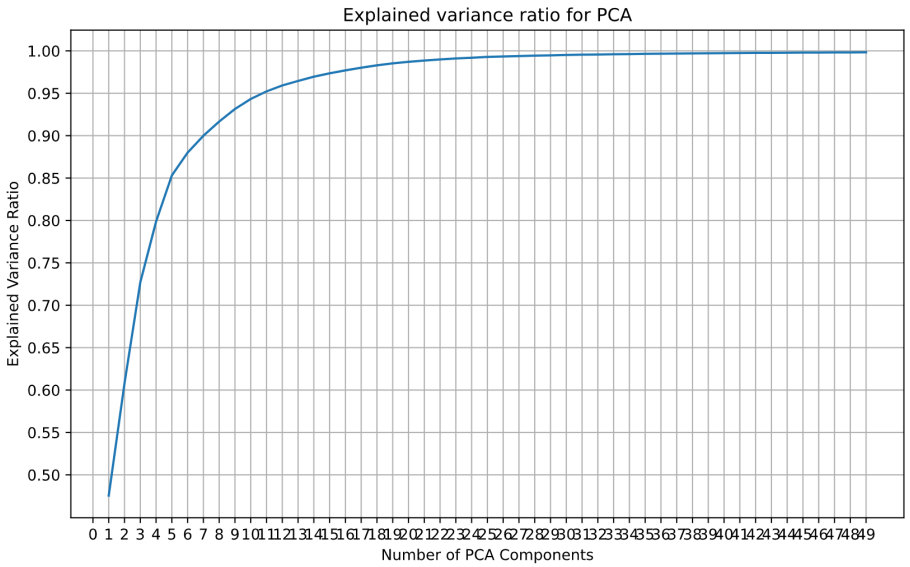
Extract LBP_ The top feature uses a similar treatment. Because the size of video frames between different samples will be different, bilinear interpolation is used to cut to the same size (224 * 224). Similarly, to maintain the consistency of the number of features, the extracted LBP-TOP histogram is normalized

3.2.3.3 Dimensionality reduction

Due to a large number of feature dimensions (hoof is 9180 dimensions and LBP is 768 dimensions), the dimensionality reduction method of PCA is used to remove the redundant dimensions. We select the feature dimension 80 with a reduction rate of more than 95%, and finally, reduce hoof features to 80 dimensions and LBP features to 11 dimensions



HOOF PCA



LBP PCA

3.2.3.4 Classifier

3.2.3.4.1 Random forest

Using grid parameter search, the number of trees is finally determined to be 7000. ((Here dimensionality reduction is directly reduced by random forest, that is, dimensionality reduction is not performed in advance through PCA))

3.2.3.4.2 Support Vector Machine

Use support vector machines for classification. To maximize tTENSOR data, the leave-one-out method is used here for testing and training

Ⅳ System Performance

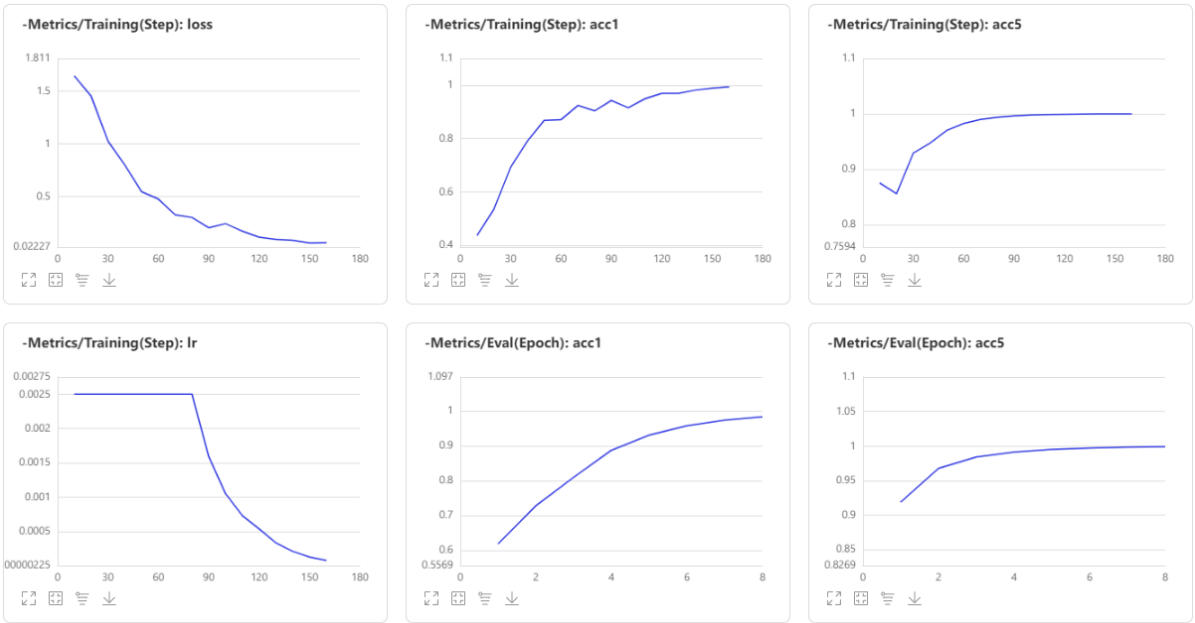
4.1 Facial Expression Recognition

4.1.1 Xunfei Competition Results

9	facelook	0.7314

4.1.2 CK+48 model training

4.1.2.1 ResNetv101_vd_ssld



4.1.3 XunFei model training

We choose the best 6 models to do ensemble learning and achieve the highest accuracy of 0.7314. The following is the training process.

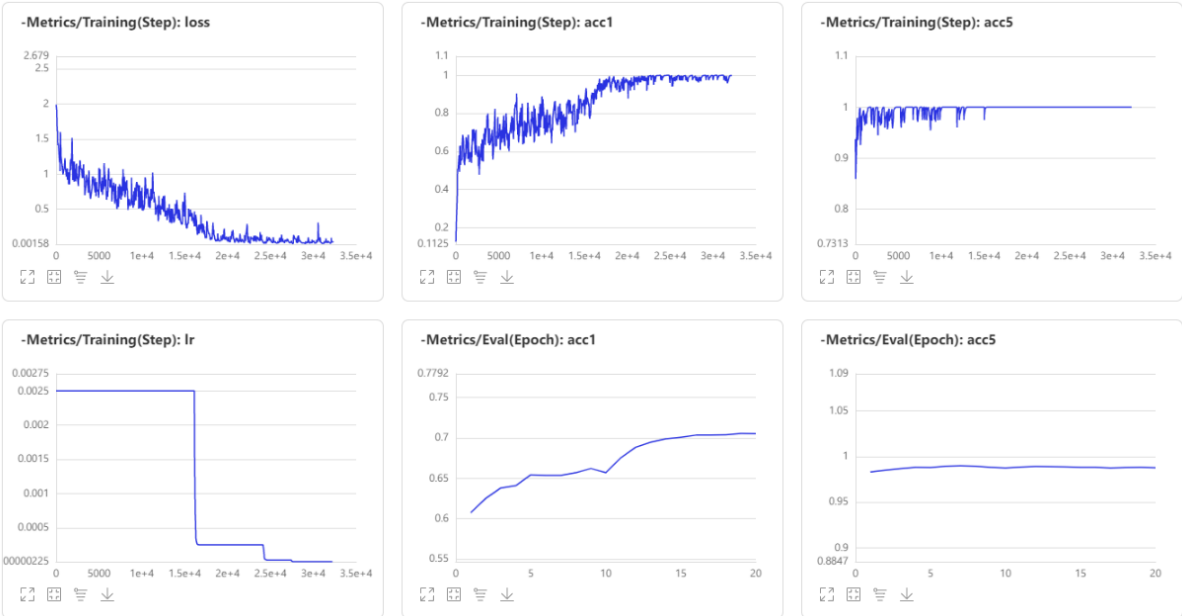
4.1.3.1 ImageNet pre-trained model information

The GPU assessment is running 500 epochs under the FP32 + TENSORT configuration (remove the top 10 Warmup times)

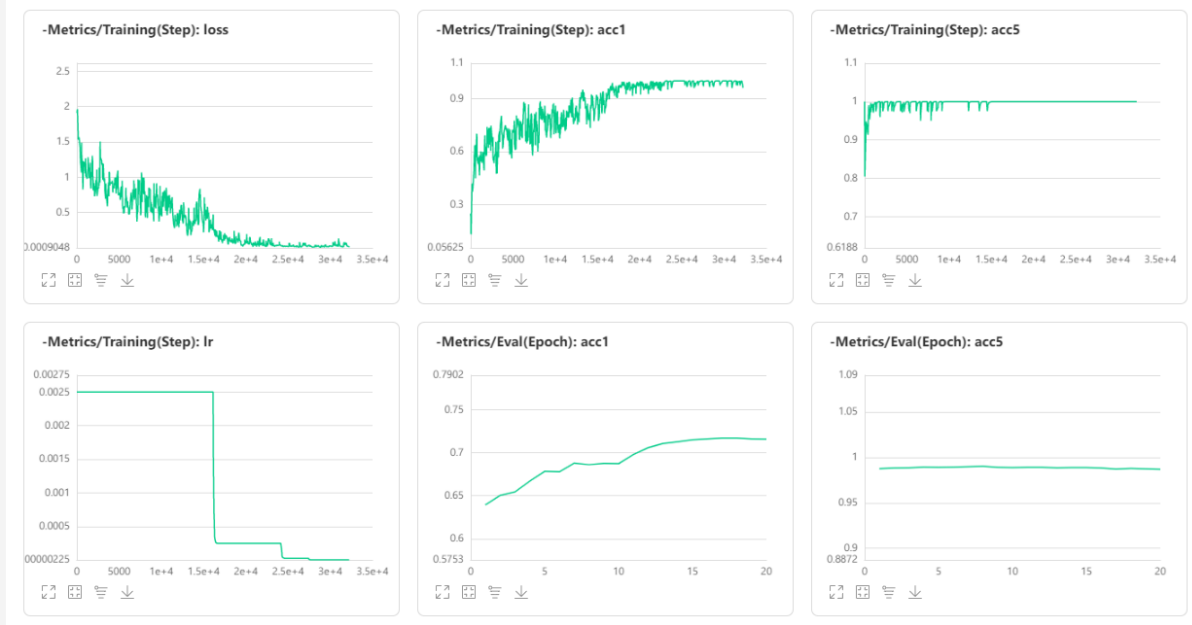
Model	Top-1 Acc	Top-5 Acc	SD855 time(ms) bs=1	Flops(G)	Params(M)	Model storage size(M)
MobileN	0.7896	0.9448	19.30835	0.45	5.47	21

etV3_large						
ShuffleNetV2	0.7315	0.912	34.770149	1.12	7.32	28
ResNet101_vd	0.8373	0.9669	6.11704	13.76222	16.1	44.57
DarkNet53	0.78	0.941	4.10829	12.1714	18.58	41.6
DenseNet264	0.7796	0.9385	12.14722	26.27707	11.54	33.37
Xception71	0.8111	0.9545	8.72457	31.55549	31.77	37.28

4.1.3.2 DarkNet53

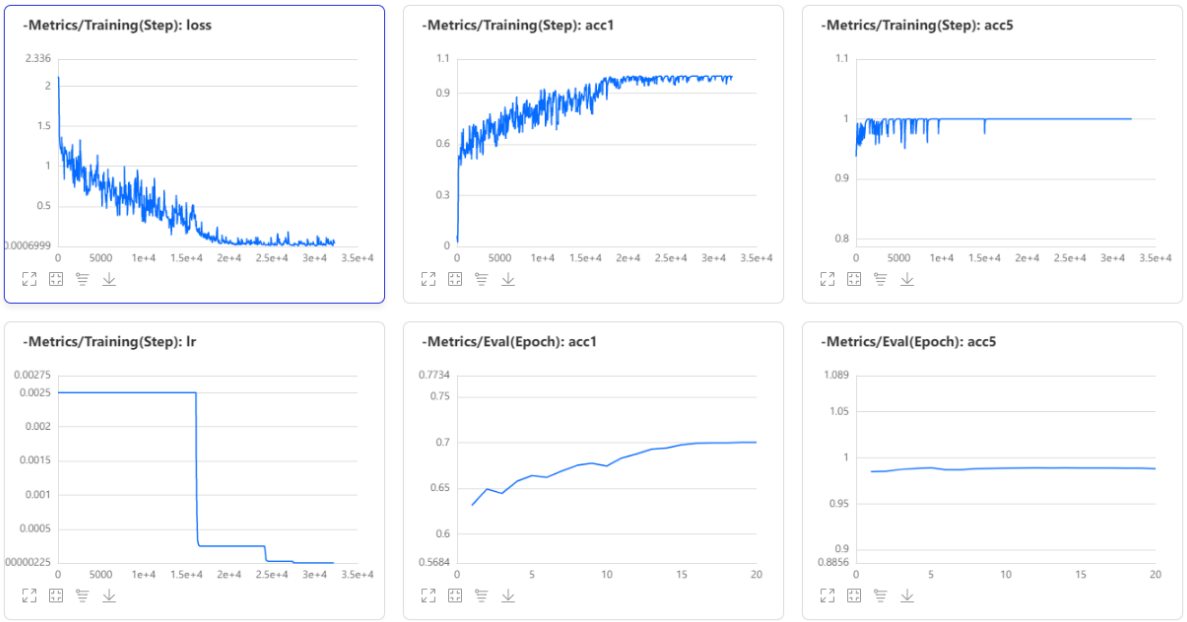


4.1.3.3 DenseNet264



4.1.3.4 MobileNetv3

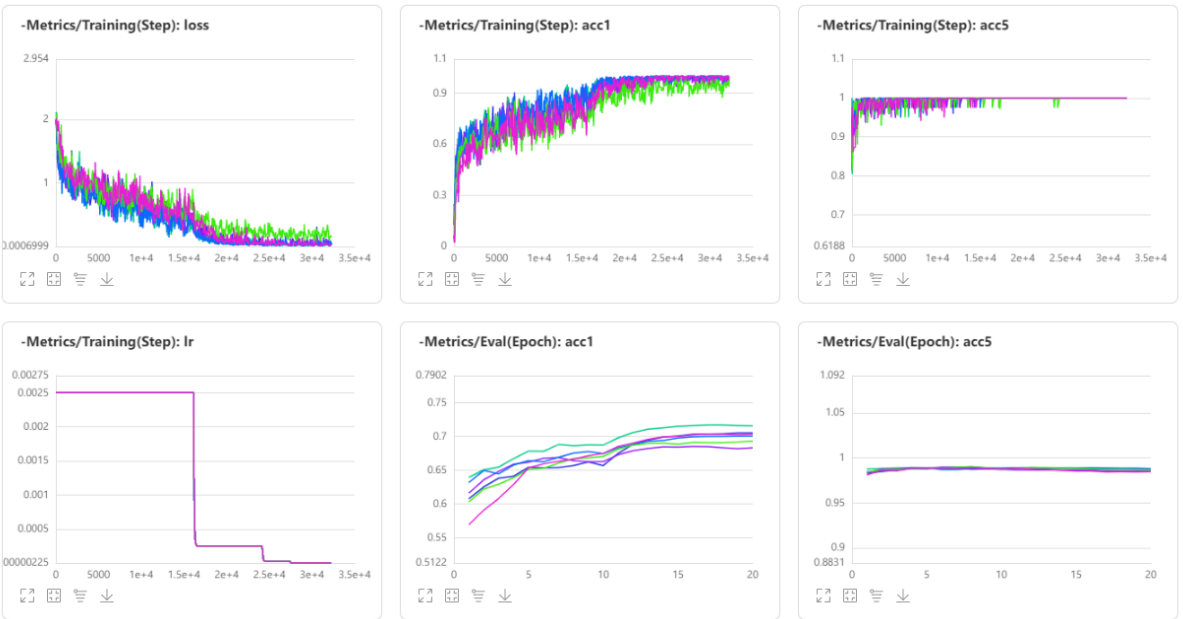
4.1.3.5 ResNetv101_vd_ssld



4.1.3.6 ShuffleNetv2

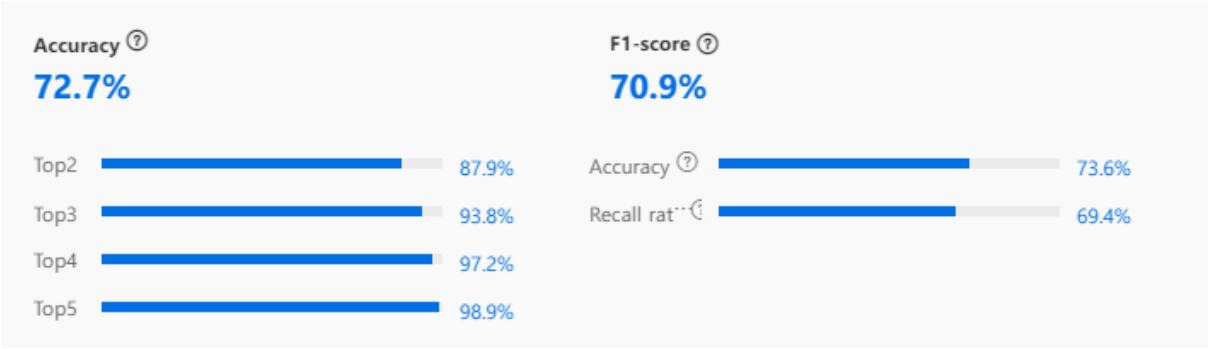
4.1.3.7 Xception71

4.1.3.8 Comparison



4.1.4 EasyDL

We also tried the tensor of the AutoDL model of Baidu and we achieved the top1-acc:72.73% and top5-acc:98.91%.



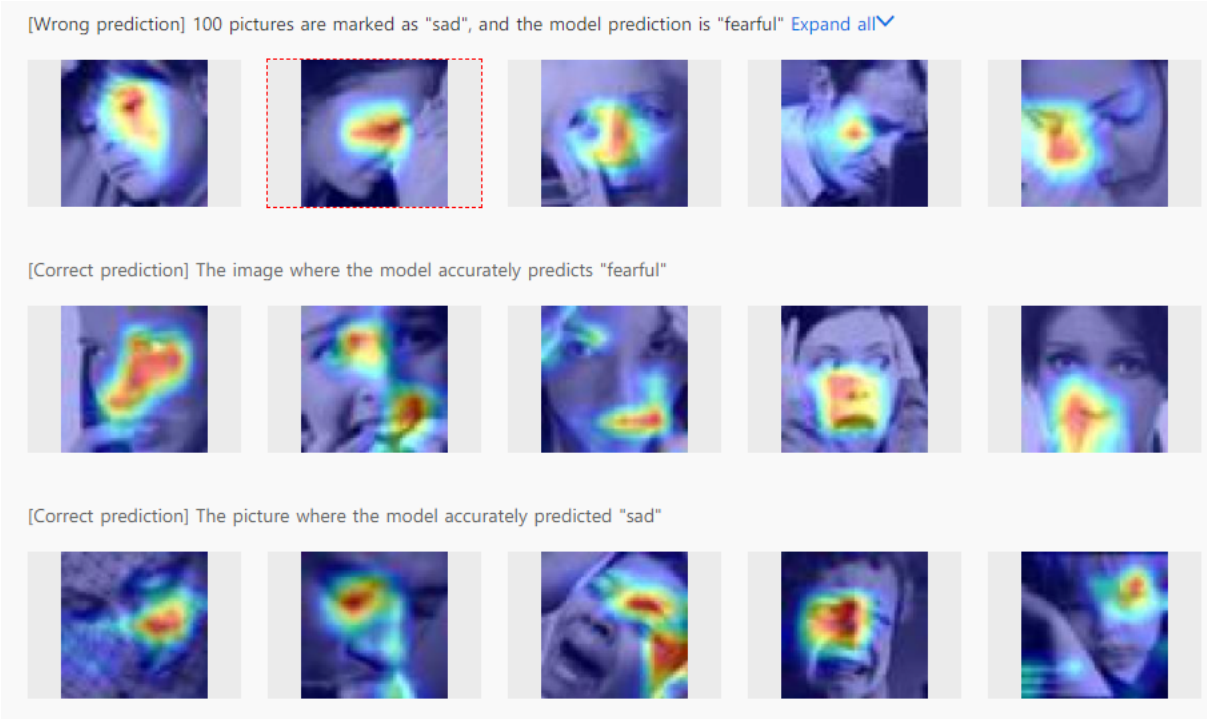
4.1.4.1 Matrix on test

Accuracy	Number of test sets	Recall rate	f1-score
63.5%	580	51%	57%
59.3%	682	66%	62%
67.7%	603	64%	66%
88.4%	64	59%	71%
68.9%	724	74%	72%

	surprised	sad	neutral	happy	fearful	disgusted	angry
surprised	299	5	13	21	33	0	11
sad	9	450	85	12	67	0	59
neutral	7	92	539	39	14	0	33
happy	11	14	41	938	6	0	7
fearful	47	121	42	12	297	0	61

disgusted	1	6	1	3	2	38	13
angry	10	71	61	21	49	5	386

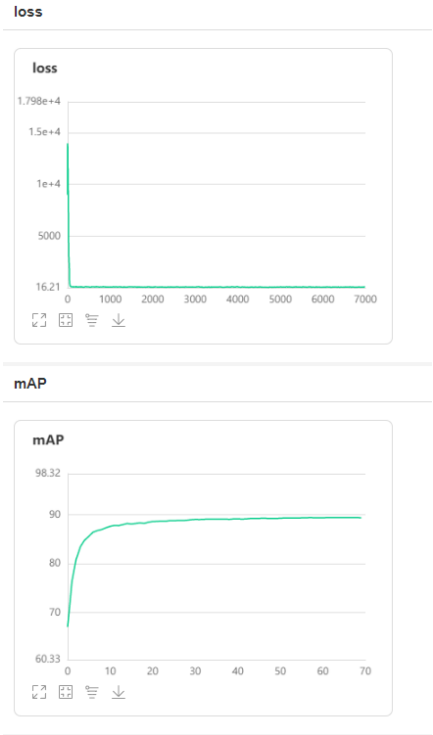
4.1.4.2 Prediction example with the heat map



4.1.5 Object detection for human head

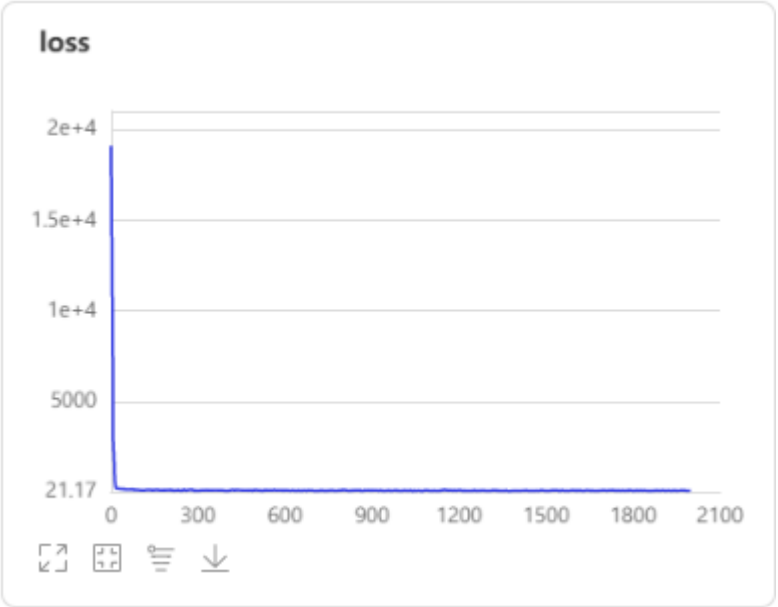
4.1.5.1 YOLOV4

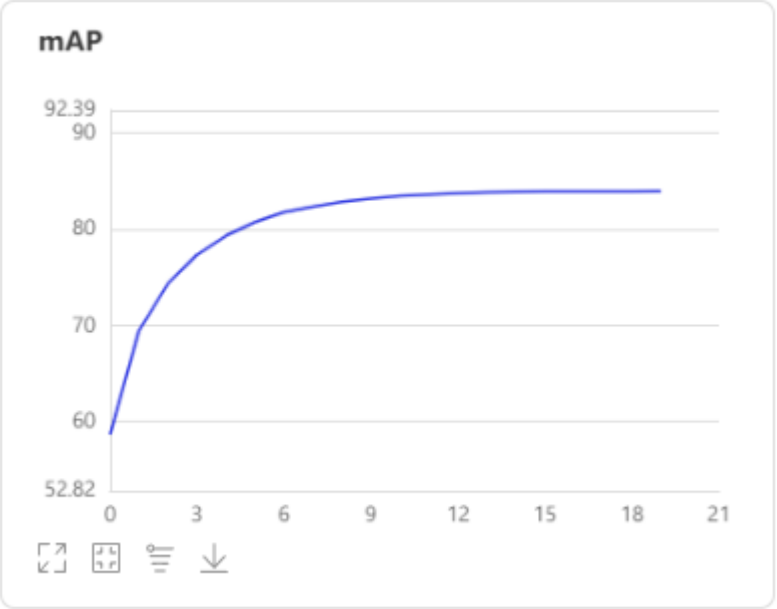
binary classification:mAP(0.50) = 89.31%



4.1.5.2 PPYOLO

binary classification: mAP(0.50) = 83.99%





4.1.6 Model Prune and Deploy

4.1.6.1 API example(test with GPU cloud on Baiduemotionallabeled easydl)

```
1. read target image 'C:\Users\vemodalen\Desktop\PRS PROJECT\CK+48\anger\S010_004_00000017.png'
fill in 'image' after base63 code
2. ACCESS_TOKEN is NONE
new ACCESS_TOKEN: 24.ee7734922c9d9a4c612e02cac76812cd.2592000.1638283795.282335-25084163
3. send 'MODEL_API_URL' request
result:
{
    "log_id": 5603679461490077514,
    "results": [
        {
            "name": "angry",
            "score": 0.7659919857978821
        },
        {
            "name": "sad",
            "score": 0.12016067653894424
        },
        {
            "name": "neutral",
            "score": 0.043435972183942795
        },
        {
            "name": "fearful",
            "score": 0.03218459337949753
        },
        {
            "name": "happy",
            "score": 0.019223524257540703
        }
    ]
}
```

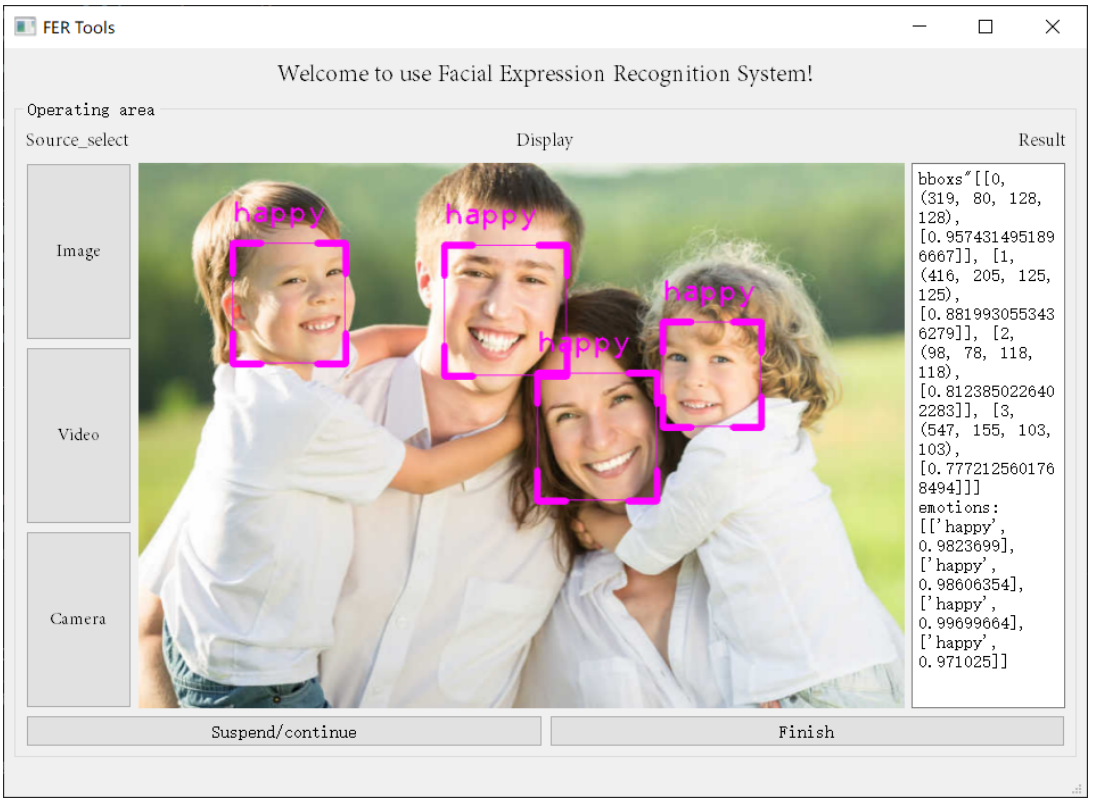
4.1.6.2 Local prune model(mobilenetv3) example（test with gtx1050 laptop)

```
------------------ Inference Time Info ----------------------
total_time(ms): 52.0, img_num: 0, batch_size: 0.0
average latency time(ms): 52.00, QPS: 19.230769
preprocess_time_per_im(ms): 1.00, inference_time_per_batch(ms): 51.00, postprocess_time_per_im(ms): 0.00
Predict Result:  angry 0.3517577
```

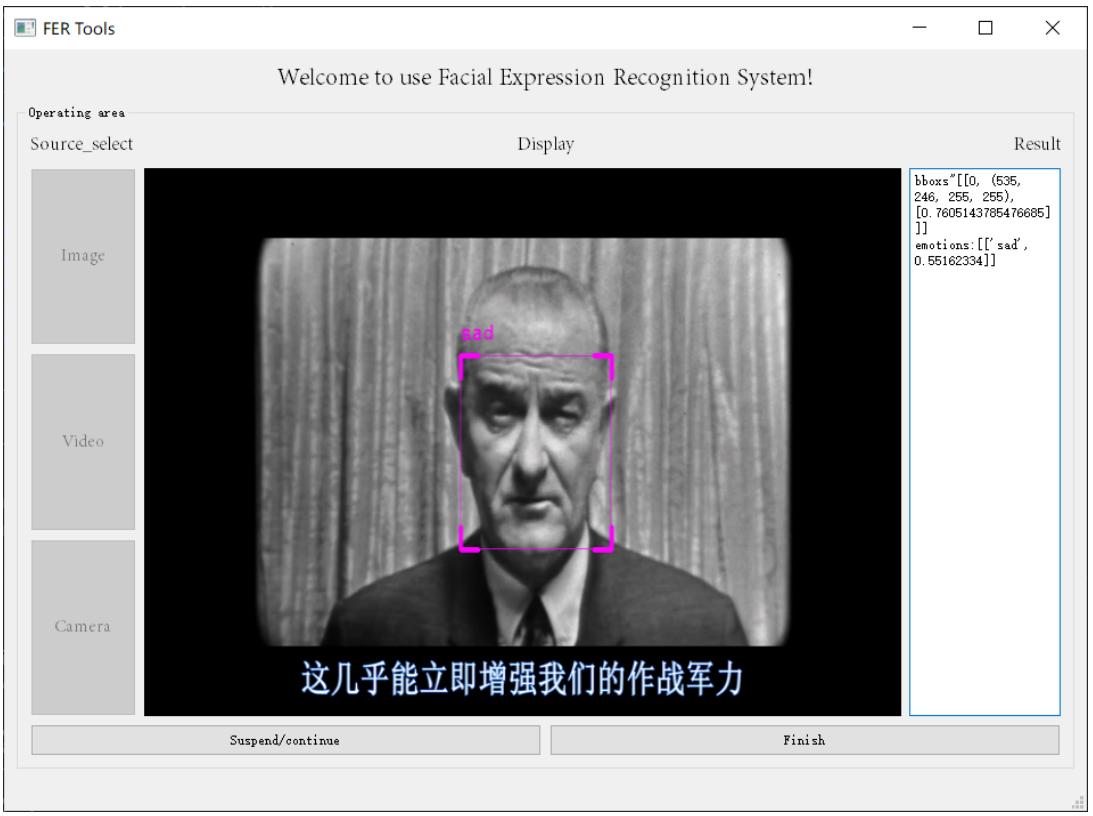
4.1.7 GUI

We can detect expressions in a variety of ways, including pictures, cameras, and videos.

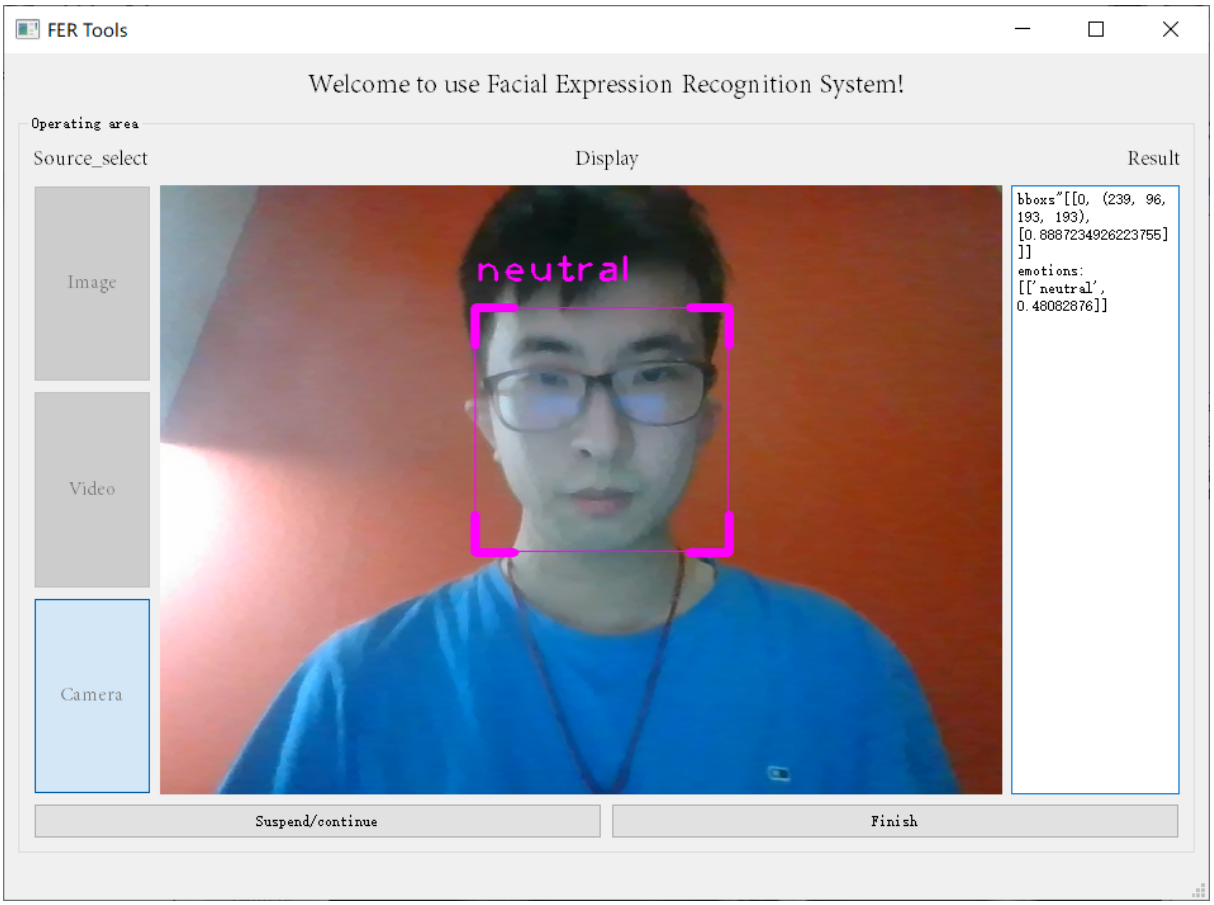
4.1.7.1 image



4.1.7.2 video



4.1.7.3 camera



4.2 Micro-Expression Recognition

4.2.1 Experimental results

We used the HOOF LBP_TOP feature combined with the SVM random forest classifier to conduct experiments. Finally, the two features were combined with the SVM classifier to achieve the best results, and the four classification accuracy rate was 57%. We also conducted an ablation experiment, which shows that the combination of the two features has the best classification performance.

4.2.1.1 Confusion Matrix

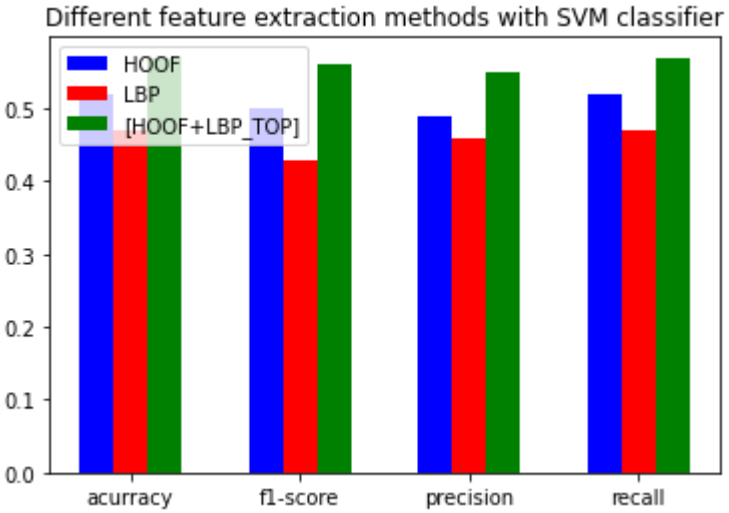
	Happiness	Disgust	Surprise	Repression
Happiness	49	5	5	4
Disgust	9	13	7	3
Surprise	6	9	9	3
Repression	9	2	3	14

4.2.1.2 Classification Report

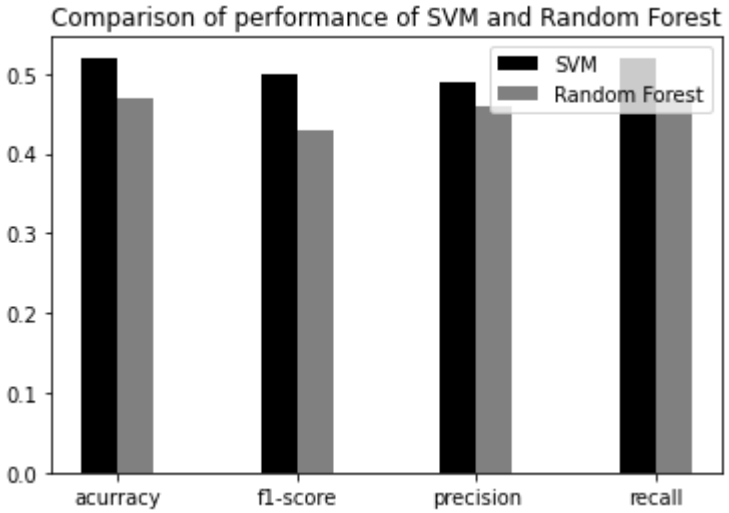
	precision	recall	f1-score	support
Happiness	0.67	0.78	0.72	63
Disgust	0.45	0.41	0.43	32
Surprise	0.38	0.33	0.35	27
Repression	0.58	0.50	0.54	28
accuracy			0.57	150
macro avg	0.52	0.50	0.51	150
weighted avg	0.55	0.57	0.56	150

4.2.2 Ablation experiment

4.2.2.1 Performance of different feature extraction methods with SVM classifier



4.2.2.2 Performance of different classifiers with [HOOF+LBP_TOP]



Ⅴ Findings and discussion

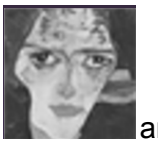
5.1 Facial Expression Recognition

5.1.1 dataset explore

1. The boundary between the different facial expressions is not so clear because the emotion state is not like a math problem and it is labeled by humans. So the labels in the dataset may not be so reliable.

For example

this is the angry label, this is the happy label
Solution: using label smooth or multi-feature like EEG.

2. we may counter some no-face images as datasets like  and
   Solution: add another label called others or use a confidence threshold to filter out the noise data

3. It is an imbalanced dataset maybe because of the different difficulties of collecting labels. So the training is hard.
   Solution: trying to do some effort on two sides. One is dealing with the data, trying some data oversample like Smote or just increasing the fewer data. Two is dealing with the loss function, we can try focal loss or weighted close to deal with this problem.

4. According to the heat map of data, the features of humans' eyes and noses seem to affect the result heavily. So a possible way to increase the accuracy is to fusion their features or add more weight when training.

5.1.2 training skills

1. data augment: according to the paper data augment, it is generally useful to use data augment like mixup, randomly rotating, random blur to simulate the real data condition and increase the robustness of the model.

2. the learning rate can try with adam,adamw, sgd with momentum. Besides, a warmup and cosine decay can be useful to converge the model. When increasing the batch size we should also change the LR.

3. A pre-trained model or transfer learning can be useful in most cases

4. Vision transformer needs a large dataset to converge

5.1.3 Deploy

1. If it is not a competition, it is a wise idea to prune and do knowledge distillation of the model to get a faster interference.

2. According to the device to choose different networks by their FLOP, also the platform to deploy should be concerned.

3. We can deploy our heavy model on the cloud GPU server and access it with API(request). Or just deploying a light model like MobileNet on the edge device.

4. To deal with real-time video recognition, it is possible to combine early/late fusion to make the results more stable.

5.2 Facial Micro-Expression Recognition

5.2.1 Findings

- In the case of small samples, the deep learning algorithm is very easy to overfit

- Combining different methods is a feasible way to get better results. In our experiments, usually, a combination of different deep learning models will help to improve the accuracy of expression recognition and a combination of different features will help to improve the accuracy of micro-expression recognition.

5.2.3 Challenge

1. Micro-expression recognition will be influenced heavily by background and noise. Compared with using raw images, using pre-cropped images help to get a better result.

2. The number of data sets is too small. The collection of micro-expression data sets often requires very large time and labor costs and must be collected under laboratory control conditions, and it is difficult to awaken certain emotions, so it is difficult to obtain a large number of data sets. Deep learning algorithms are difficult to perform. Therefore, how to improve the classification accuracy of small sample size is a huge challenge

3. Poor real-time performance. As you can see earlier, we did not set the GUI interface for micro-expression recognition, mainly due to LBP_ Top features and hoof features caused by the large computing time cost, while features with less computing resources are faced with the problem of reduced accuracy. Therefore, how to realize real-time and high accuracy micro-expression recognition has become a challenge

Ⅵ Future Work and Possible Application

6.1 Future Work

1. Increase the accuracy of expression recognition.
2. For micro-expression recognition, improve the classification accuracy under the condition of small samples
3. Improve the speed of micro-expression recognition and deploy it on the user interface. [Improve the real-time performance of classification as much as possible]

4. Combine expression recognition and micro-expression to recognize human emotion.

6.2 Possible Application

1. Combine emotion recognition models with a psychology chatbot to create a virtual psychological consultant. A chatbot can give some advice or get responses based on a patient's emotions.
2. Use emotion recognition models to predict the feelings of participants in classes. And help the teacher to adjust content or speaking speed.
3. Recognize people's reactions and feelings when watching a video or playing a game. Analysis of whether users like this product or not and which part users like best.

Ⅶ References

1. PaddlePaddle/PaddleX: PaddlePaddle End-to-End Development Toolkit. GitHub. (2021). Retrieved 31 October 2021, from https://github.com/PaddlePaddle/PaddleX
2. Gao, T. (2021). PaddlePaddle/PaddleClas: A treasure chest for visual recognition powered by PaddlePaddle. GitHub. Retrieved 31 October 2021, from https://github.com/PaddlePaddle/PaddleClas.
3. Druzhinin, M., Parinov, A., & Iglovikov, V. (2021). Albumentations. Albumentations.ai. Retrieved 31 October 2021, from https://albumentations.ai/.
4. EasyDL-Zero threshold AI development platform. Ai.baidu.com. (2021). Retrieved 31 October 2021, from https://ai.baidu.com/easydl/.
5. Qing, Q. (2021). PaddlePaddle/PaddleDetection: Object detection and instance segmentation toolkit based on PaddlePaddle. Retrieved 31 October 2021, from https://github.com/PaddlePaddle/PaddleDetection
6. scikit-learn - Wikipedia. En.wikipedia.org. (2021). Retrieved 31 October 2021, from https://en.wikipedia.org/wiki/Scikit-learn.
7. TensorFlow - Wikipedia. En.wikipedia.org. (2021). Retrieved 31 October 2021, from https://en.wikipedia.org/wiki/TensorFlow.
8. Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov,Karthik Raveendran, and Matthias Grundmann, "Blaze-face: Sub-millisecond neural face detection on mobilegpus," 2019
9. ResNet_ v2_ 101 (image classification). Support.huaweicloud.com. (2021). Retrieved 31 October 2021, from https://support.huaweicloud.com/aimarket-modelarts/modelarts_18_0019.html.
10. Vision Transformer. Zhihu (2021). Retrieved 31 October 2021, from https://zhuanlan.zhihu.com/p/273652295
11. Detailed explanation of parameters of each layer of darknet53 network. Jianshu.com. (2021). Retrieved 31 October 2021, from https://www.jianshu.com/p/ab8392ef3394.
12. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
13. PaddleX---MobileNetV3_ssld image classification. Blog.csdn.net. (2021). Retrieved 31 October 2021, from https://blog.csdn.net/weixin_43468161/article/details/105991766.
14. Detailed explanation of Xception algorithm - Road of AI. Blog.csdn.net. (2021). Retrieved 31 October 2021, from https://blog.csdn.net/u014380165/article/details/75142710.
15. ShuffleNetV2: the crown in lightweight CNN networks. Zhihu. (2021). Retrieved 31 October 2021, from https://zhuanlan.zhihu.com/p/48261931.
16. PaddleX---MobileNetV3_ssld image classification. Blog.csdn.net. (2021). Retrieved 31 October 2021, from https://blog.csdn.net/weixin_43468161/article/details/105991766.

17. Detailed explanation of Xception algorithm - Road of AI. Blog.csdn.net. (2021). Retrieved 31 October 2021, from https://blog.csdn.net/u014380165/article/details/75142710.
18. ShuffleNetV2: the crown in lightweight CNN networks. Zhihu. (2021). Retrieved 31 October 2021, from https://zhuanlan.zhihu.com/p/48261931.
19. R. Chaudhry, A. Ravichandran, G. Hager and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1932-1939, doi: 10.1109/CVPR.2009.5206821.
20. Zhang, Liangfei, and Ognjen Arandjelović. "Review of Automatic Microexpression Recognition in the Past Decade." Machine Learning and Knowledge Extraction 3.2 (2021): 414-434.
21. X. Hong, Y. Xu and G. Zhao, "LBP-TOP: A tensor unfolding revisit," in Anonymous Cham: Springer International Publishing, 2017, pp. 513-527.
22. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.
23. 2021 iFLYTEK A.I.开发者大赛-讯飞开放平台. (2021). Retrieved 28 August 2021, from https://challenge.xfyun.cn/topic/info?type=facial-emotion-recognition&invitaCode=8zCBfV
24. Yan, W. J., Li, X., Wang, S. J., Zhao, G., Liu, Y. J., Chen, Y. H., & Fu, X. (2014). CASME II: An improved spontaneous micro-expression database and the baseline evaluation. PloS one, 9(1), e86041.