Spring 5-24-2020

Improving Micro-Expression Recognition with Shift Matrices and Database Combination

Yuqi Zhou

Improving Micro-Expression Recognition with Shift Matrices and Database Combination

by

Yuqi Zhou

In Partial Fulfillment of the Requirements

for the Degree of Bachelor of Computer Science

Rose-Hulman Institute of Technology

Terre Haute, Indiana

May 24, 2020

Acknowledgment

Thanks very much to anyone, especially Dr. Jason Yoder, who has helped me find out my interest in this object and gain this precious experience of researching.

Abstract

Micro-expressions are brief, subtle changes in facial expressions associated with emotional responses, and researchers have worked for decades on automatic recognition of them. As convolutional neural networks have been widely used in many areas of computer vision, such as image recognition and motion detection, it has also drawn the attention of scientists to use it for micro-expression recognition. However, none of them have been able to achieve an accuracy high enough for practical use. One of the biggest problems is the limited number of available datasets. The most popular datasets are SMIC, CASME, CASMEII, and SAMM. Most groups have worked on the datasets separately, but few have tried to combine them. In our approach, we combined the datasets and extracted the shared features. If new datasets under the same classifying rules (FACS) are created in the future, they can easily be combined using our approach. In addition to this novel approach for combining datasets, we use a new way of extracting the features instead of the Local Binary Pattern from Three Orthogonal Planes (LBP-TOP). To be more specific, we create shift matrices, the changing pattern of pixels, to keep the spatial information of the videos. Our highest recorded accuracy from 100 experiments was 88 percent, but we chose to report 72.5 percent. This is the median accuracy and a more convincing result though it’s a little bit lower than the best result to date. However, our f1 score is 72.3 percent and higher than the best result to date. Our paper presents an extendable approach to micro-expression recognition that should increase in accuracy as more datasets become available.

Personal Motivation

Researchers have been trying to imitate human understanding of the world, including computer vision and natural language processing. We humans can infer others' feelings from their facial expressions. For example, if people are smiling, then there is a high probability that they are happy. Therefore, I wondered if we can teach a robot to discriminate the emotions of people from their facial expressions. That was when I became interested in micro-expression recognition. After I read through some papers, it turned out that most researchers tested their approaches on the existing micro-expression datasets separately. However, all the datasets are so small that it is difficult to extract useful features. Therefore, I decided to pursue an extendable way of combining the datasets and set a standard for future work in this area as more datasets become available.

1. Introduction

1.1 Micro-expression

Facial expressions are an integral part of our daily communication. Compared to language, it may be easier for people to show their reactions with facial expressions only. For example, even in different cultures, smiles usually represent happiness [1]. However, normal facial expressions can be easily faked by people to hide their real emotions, which is one reason why facial expressions are not a reliable lie detector. Micro-expressions are a class of specific shifts in facial expression which are more reliable and harder to fake. There is also a classifying rule-set called FACS [2] used to identify microexpressions. In contrast to normal facial expressions, untrained people are unable to conceal their real emotions during microexpressions [3].

The reliable, automatic classification of micro-expressions could create excellent new opportunities in multiple domains. For example, as online courses become more popular, it is necessary for the teachers to receive responses from the students. Normally, students may show confusion on their faces if some of the material is not clear enough and an instructor can tell if more explanation is required on a given topic. However, this might not always be possible to do with online courses. Therefore, automatically detecting their microexpressions could be very helpful in that regard. Another example would be analyzing customer's emotions which "are what [drives] your audience to purchase." [4]. If one could capture micro-expressions of customers while navigating a website, it may enable the companies to realize what the audience would like to buy or may plan to buy in the future.

Recognizing micro-expressions is challenging, in part due to the lack of data and the complexity of video analysis. Unlike recognizing emotions from facial expressions that only require static images, detecting micro-expressions requires short videos from one-fourth to one-third of a second [3]. Normally, the detection of micro-expressions requires experts trained to do

so because the muscle movements are very difficult to observe. Therefore, learning to detect the microexpressions requires a long training time for humans, and even then, detecting the emotions is, likewise, very time-consuming, so scientists have been trying to find a way to automatically recognize them.

1.2 Convolutional Neural network

The convolutional neural network (CNN) is a kind of artificial neural network that is very effective in fields involving images such as image recognition and gesture detection. It is also widely used in many recognition areas using deep learning as a solution, such as NLP (natural language processing) and voice analysis. The concept was not new in the 2010s, however, as one of the most famous networks, LeNet, was first proposed in 1989 by Yann LeCun et al. [23]. However, CNN failed to draw the researchers' attention until 2012 because the computers in the early decades could not support networks big enough for optical performance. Things changed in 2012 when Alexnet [24] won the 2012 image recognition competition held by ImageNet (http://www.image-net.org/). Since then, CNNs (or deep learning) have become one of the hottest research areas.

When used for image processing (or recognition), the convolutional neural network automatically extracts the texture and color patterns by applying filters to the images in the convolutional layers. The filters record the information of one pixel of certain blocks (depending on the number of strides) by taking their neighbor pixels into account. For example, it can find the edge of the objects. After it finds the texture and color patterns, it learns what patterns to remember with a great number of training samples. The more often specific patterns appear in the correct sample of an object, the more relevant they are to the object. CNNs will find many different patterns and update their biases based on their relevance. When a new image of an object is tested, the network will try to extract certain features of a certain object or some possible objects with the constant filters and calculate the confidence score of the object(s). If

the confidence score is higher than a threshold, then it recognizes the object with the highest confidence score. The figure below is an example of a convolutional neural network.

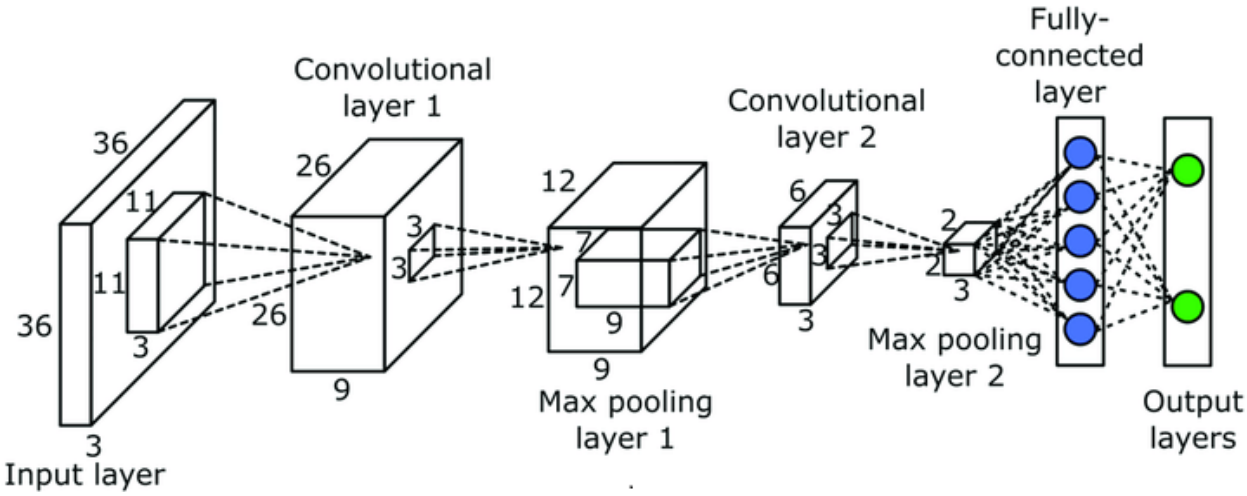

Fig.1 an example of a convolutional neural network
(https://www.researchgate.net/figure/Structure-of-the-convolutional-neural-network_fig3_323227084)

In the remainder of this paper, we will first summarize the key ideas in related works (section 2) that have resulted in the best results to date. After that, we will talk about the motivation and some early thoughts (section 3). Then we will talk about the databases we use (section 4). Next, we will introduce our novel approach using what we call shift matrices and describe our experiment setup and algorithm (section 5). Following this, we will report on our findings and analysis of results (section 6, 7) and make suggestions for the direction of future work (section 8). We will also discuss the implications and conclusions from our findings (section 9). Finally, I will discuss something I have done well, and some things I would do differently if I were to do it again. (section 10)

2. Related Work

In this section, we will first introduce some early and recent attempts at micro-expression recognition. Then we will discuss common approaches used for feature extraction and network training. Finally, we will describe the main challenges of micro-expression recognition in general and our approach to the challenges.

2.1 Early and recent attempts

Before deep learning became prominent, most researchers attempted to use “traditional” classification methods like gradient descriptors [5] or Gabor filters [6]. In recent years, however, as machine learning is becoming popular for many computer vision problems, researchers have started applying machine learning methods, such as convolutional neural networks, to micro-expression recognition. Usually, researchers separate the problem into two parts and find an optimal solution to each of them.

The first part is extracting features from the videos. The most popular way of keeping the changing pattern among the frames is called Local Binary Pattern from Three Orthogonal Planes (LBP-TOP). This method is known for its simplicity and efficiency in tracking object movement and has shown to be useful for recognizing emotions from facial expressions [7] [8]. Though useful, many groups are attempting to improve the LBP-TOP method and use it as a benchmark for comparison [9]. In this paper, we use a method similar to but in fact different from LBP-TOP.

The second part is training the network with the extracted features (since they may not be recognizable images anymore). Before convolutional neural networks (CNN) were widely used, researchers tended to use support vector machines (SVM) [10] or gradient descriptors [5]. However, due to the lack of data and the complexity of micro-expression videos, many of them could not receive a satisfying result. With CNNs, however, researchers decreased the difficulty of feature extraction by letting the network decide the features instead [11]. To solve the

problem of limited data, some groups even tried to combine the datasets and trained their networks with a bigger one [12]. These are the two main tasks for microexpression recognition in the future [9]. In this paper, we share our own approach to these two tasks and the results that it produced.

2.2 Major challenges

The muscle movements of micro-expressions are so tiny and complex that creating and labeling such datasets is very difficult. This is the reason why the existing datasets are much smaller than those of other deep learning problems. Due to the small size of the available datasets, the size and structure of the networks usually do not have a significant influence. Therefore, people focus more on image preprocessing and feature extraction.

2.2.1 Video(image) preprocessing

There are two popular ways to deal with the data. One is to use the whole face directly. Though there will be some noise in the background which is usually something unrelated to the face such as clothes and walls, it generally contains more useful features than the alternatives. For example, Liong et al. [12] used the whole face as the input to train their model. Another common approach is to divide the face into segmented landmarks [5] or blocks [9]. The micro-expressions in most widely used datasets were classified with the Facial Action Coding System (FACS) by Ekman et al. [2] and different emotions are associated with different action units (AUs). Fig. 2 shows an example of AUs distributed on a face. Action units are particular regions of the face located in different parts of the face. For example, AU6 refers to cheek raiser and AU12 refers to lip corner puller in which movements usually occur when people are happy. This is discussed further in the database section (Section III). Since AUs are always located in specific areas, those areas should contain more useful information about the micro-expression classification.

Fig.2. This is an example image from CASMEII [14] and the AUs are ActionUnits coded by FACS [2].

2.2.2 Feature extraction

The problem of keeping spatial information on muscle movements while eliminating noise is a key problem of feature extraction. For example, using the whole face as input provides more information, but it will also contain significant noise. In contrast, using segmented landmarks or blocks can avoid some noise but at the same time loses important information. Most people choose to use or compare their method to the LBP-TOP method, such as LBP-SIP [7] and STLBP-IP [13]. This method can track the movement of objects and track Spatio-temporal information.

Another important part of feature extraction is the selection of frames to use. Micro-expressions are recognized from short videos, which contain too many frames to use. Therefore, most researchers are using frames from onset to offset (as labeled in the datasets such as CASMEII [14]) for detecting emotions. The onset and offset frames are frames where the micro-expression starts and ends respectively while the apex frames are the frames that contain the most drastic change during the time period. Many researchers focus primarily on

these frames [15]. Notably, Liong et al. [12] uses onset and apex frames as input into a convolutional neural network and produced the best results to date.

3. Exploration and early thoughts

I am going to share my thoughts on two parts, datasets and structure design, at the beginning of this program. The first one is how to choose the dataset and the other one is how to design the structure.

3.1 Datasets

As I was searching for the usable dataset, I paid attention to four major datasets that are very popular in other researches, CASME[16], CASMEII[14], SAMM[17], SMIC[19]. The first three datasets were coded by the same classifying rule FACS (which I will provide more details in the Databases section). Therefore, they share some of the emotion classes such as happiness and disgust while the features based on which they classify the emotions are also the same. To allow using new datasets with the same classifying rules in our model, I chose to use CASME, CASMEII, and SAMM.

3.2 Structure design

I designed two models that differ at the preprocessing level. One is to use the whole face and the other is to use different landmarks and concatenate them.

We found out that pixels near three landmarks, eye, nose, and mouth, have the most intensive changes, as shown in Fig.3. The green dots are pixels with an increase in grayscale while the red dots are pixels with a decrease in grayscale value. Only pixels with changes higher (absolute value) than a threshold are recorded. Therefore, the first model is the one shown in Fig.4. First, each landmark is inputted to one network and I concatenated them to output the result.

Fig.3 The green and red dots are pixels with the most intensive change. The green dots are pixels with the most increase in grayscale while the red dots are pixels with the most decrease in grayscale.

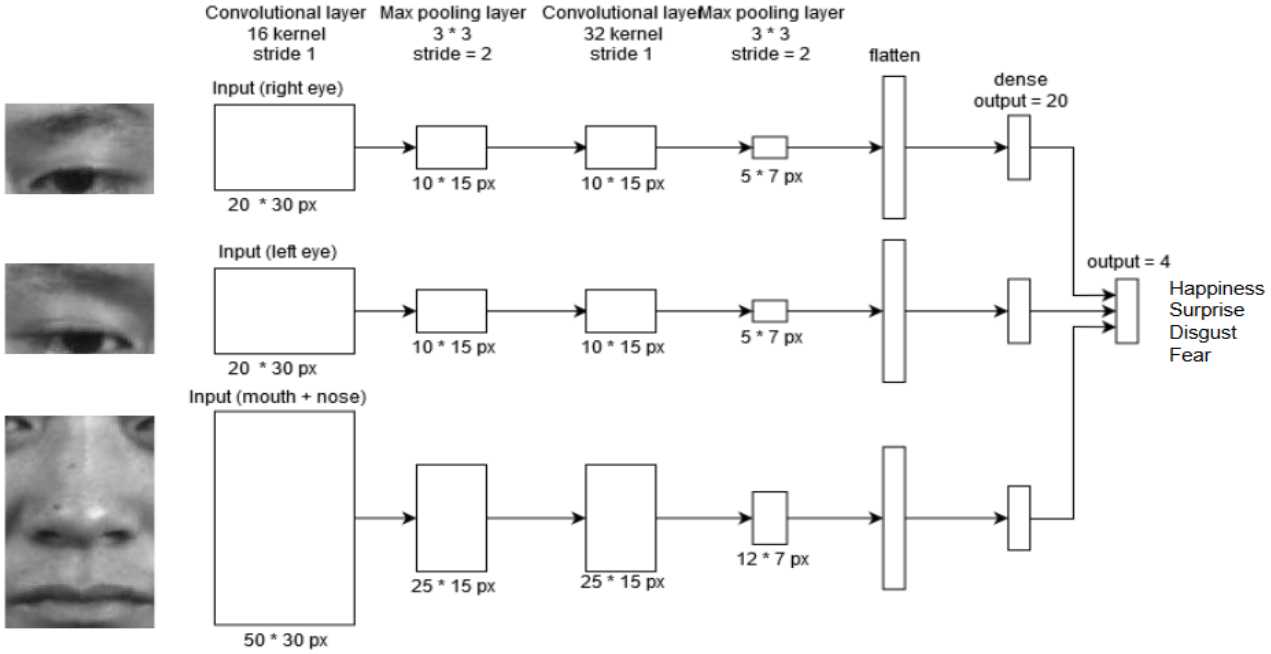

Fig. 4: the network of segmented landmarks. First, use each landmark as an input to one network and concatenate them.

4. Databases

There are several datasets such as USF-HD [18] and Polikovsky's database [5]. In recent years, however, scientists prefer to use SMIC [19], CASME [16], CASMEII [14], and SAMM [17]. Some important information about the four datasets is shown in Table 1. In general, the videos in these datasets are taken and labeled with the same classifying rule in similar ways, which allows the combination of datasets. By combining different datasets, scientists can have a larger and more reliable dataset for training and testing. Merghani et al. [9] combined CASMEII and SAMM with "a selective block-based feature fusion representation method." SAMM is very similar to CASMEII, because both were obtained in a very similar way. They both hired more than one coder for labeling the videos (CASMEII hired 2 while SAMM hired 3) with the FACS coding rule and recorded onset, apex, and offset frames. When determining the emotions, both groups also asked the participants to answer some questions such as their real emotions and reasons for their expressions. Liong et al. [12] combined SMIC, CASME, and CASMEII, and recategorized the videos of CASME and CASMEII to positive, negative, and surprise. In our approach, however, we combined CASME, CASMEII, and SAMM, and used videos of four classes: happiness, disgust, fear, and surprise. Though the three datasets have different resolutions and frame rates, they are coded under the same rule (FACS), which means their classifications should have a strong possibility of containing the same features. This is also the reason why we chose not to use SMIC.

Table1: data taken from, CASME [16], CASMEII [14], and SAMM [17].

	CASME	CASMEII	SAMM
Samples	195	247	159
Participants	35	35	32
Resolution	640 * 480 & 720 * 1280	640 * 480	2040 * 1088
Face resolution	150 * 190	280 * 340	400 * 400
Frame rate	60	200	200
FACS rated	Yes	Yes	Yes
Emotion classes (number)	Amusement(5) Sadness(6) Disgust(88) Surprise(20) Contempt(3) Fear(2) Repression(40) Tense(28)	Happiness(33) Disgust(60) Surprise(25) Repression(27) Other(102)	Contempt Disgust(12) Fear(4) Anger(8) Sadness(10) Happiness(69) Surprise(4)

5. Proposed Algorithm and Experimental Setup

5.1 Dataset combination and pattern keeping

Though LBP-TOP-like methods are popular and successful for facial expression and even micro-expression recognition, we decided to take a different approach. We chose to keep the changing pattern of each pixel after applying a max-pooling layer to the original image, as shown in Fig. 5.

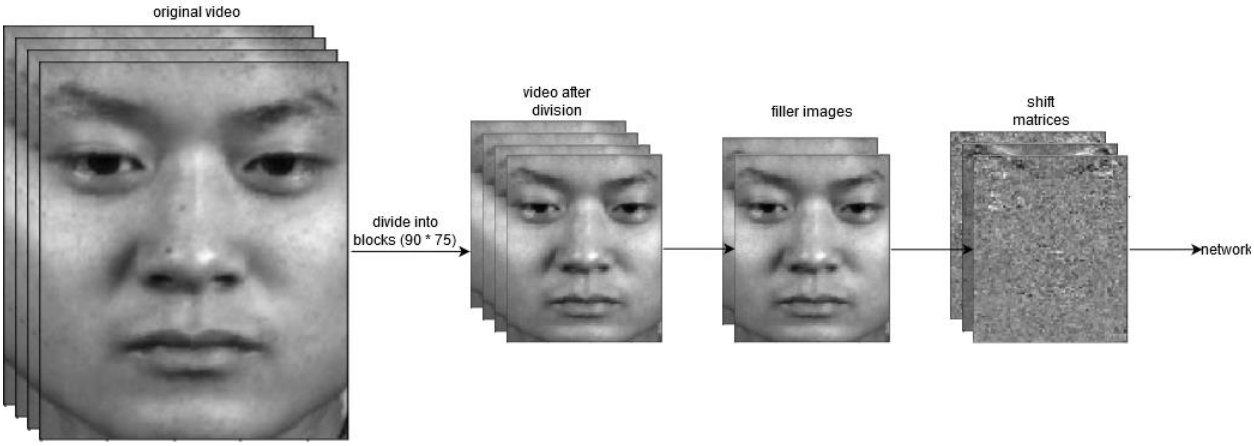

Fig.5: This is the preprocess part of our model. We first divide the images into blocks, and then we find the filler images and shift matrices. Finally, we input the shift matrices into the network.

To combine the CASME, CASMEII, and SAMM datasets and extract features at the same time, we first normalized the expressions of different people and created a constant number of sample frames among the datasets. To normalize the expressions of different people, we first cropped the faces from the images by using the dlib package of Python. Next, we divided the face into several uniform boxes where each contains certain features from a given area. Though the videos are of different people, the positions of their landmarks and other face regions do not vary much after being normalized in this way. However, the number of boxes is very important, because we cannot have too many or too few boxes. For example, if we have too many boxes and each box contains one or two pixels, then it cannot help with eliminating the pixels with irregular patterns among the samples. On the other hand, if we have too few boxes, each box will contain different parts of landmarks and muscle areas among the videos because of the variance of the landmark distributions among different people. For instance, in Fig. 6, the block contains different parts of the mouth. The example block of the left image contains half of the lower lip but that of the right image only contains a small part of the lower lip.

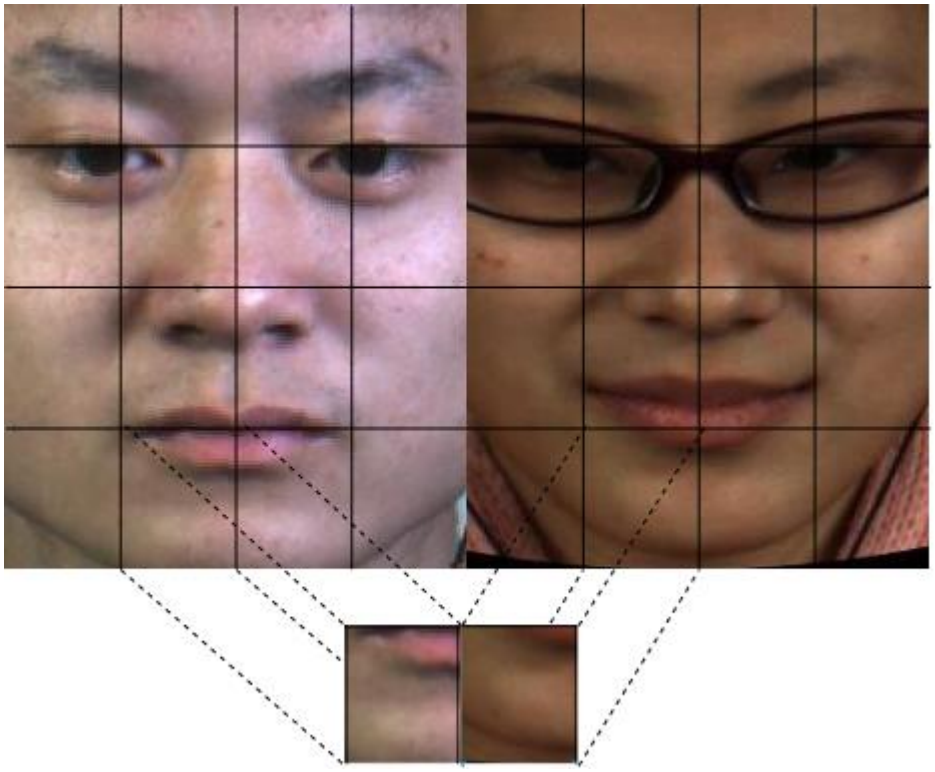

Fig.6: The blocks of different images will contain different parts of landmarks if the size of the block is too big.

When normalizing and preprocessing the images, we focused on the most intensely changing features and used only the frames between the onset frame and the apex frame of the videos, where the apex frame is the frame with the most dramatic change. For each emotion, the related muscle movements have constant features among all the videos in all the datasets as they are labeled by FACS. Since the features are consistent among the videos, we recorded the changing pattern of each pixel in the new videos created by applying a max-pooling filter, like what is shown in Fig. 5.

To be more specific, with our approach we are creating new matrices, shift matrices, by comparing relative shifts among frames. Fig. 5 shows the flow of our preprocessing procedure.

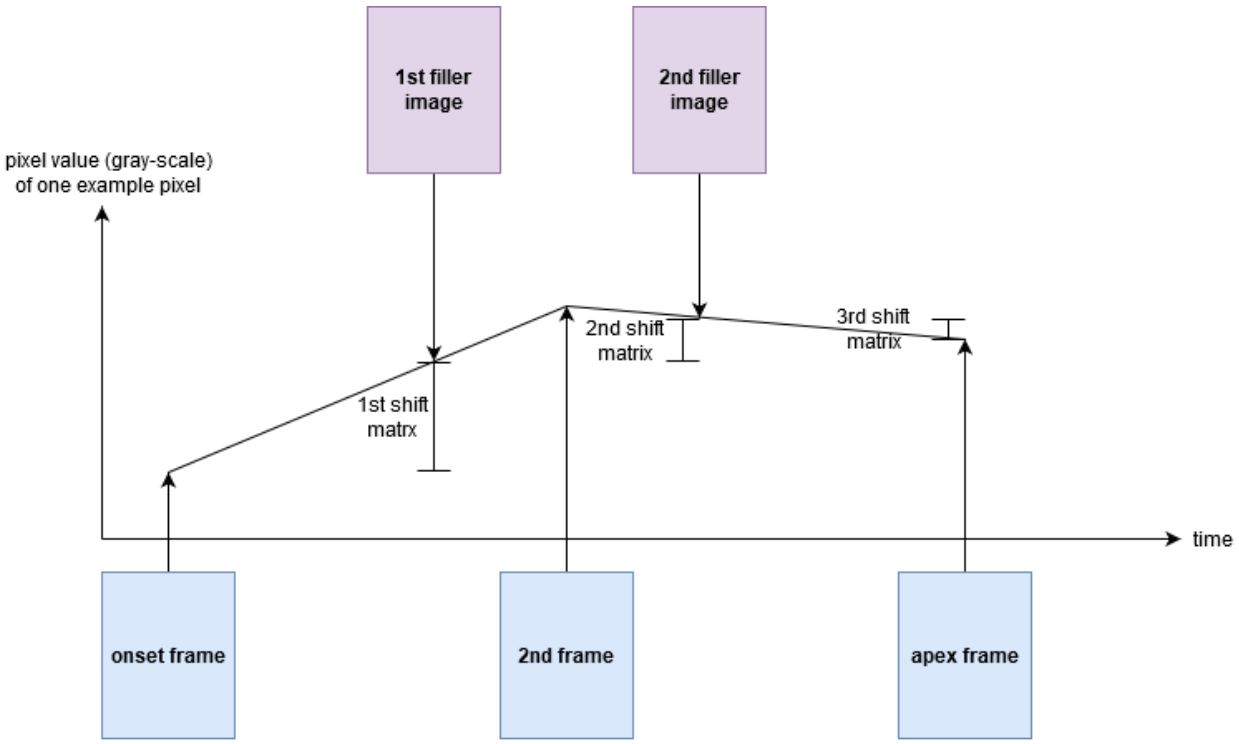

Fig.7: This shows the steps of creating shift matrices. First create the filler images based on the original frames, and then create shift matrices.

Fig.7 shows the steps of creating shift matrices. The shift matrices normalize the videos of different databases by eliminating the influence of light, resolution, and individual people. Without shift matrices, the distinct environments (lighting, background, etc.) where the videos were created (varying between databases) will lead to different color ranges and time information. To obtain the shift matrices, we need a constant number of frames between onset and apex frames. Therefore, we will need to create placeholders when a frame does not exist at the desired moment in time. We call these place-holding images filler images (*fis*). We define the frames of the original video to be *f1*, *f2*, . . . , *fn*. There will be m-1 filler images *fi_1*, *fi_2*, . . . , $fi\_(m-1)$, where m is the number of shift matrices and is a parameter we varied to conduct experiments.

The first step in this process is to locate each filler image *fi*. Filler images are inserted into the original video and the time gap between every two filler images of the same video (including the onset and apex frame) is constant. Thus, the frames from the original video are used to create the filler images while the filler images are used to create the shift matrices.

$$fpfi = \frac{n-1}{m+1}$$

With the frames per filler image *fpfi* we can calculate the desired moment (or location) *filoc* of each filler image. *fi_k* is the kth filler image and *fi_kloc* is the desired moment of *fi_k*.

$$fi\_kloc = (k + 1) * fpfi$$

The index of the frames before and after the desired moment of the filler image *fi_k*, *index_before_k* and *index_after_k*, are

$$index\_before\_k = \text{ceil}(fi\_kloc)$$

$$index\_after\_k = \text{floor}(fi\_kloc)$$

With the frames of the original video *f_beforek* and *f_afterk* at *index_before_k* and *index_after_k* respectively, we can calculate filler images *fi_k*.

$$fi\_k = f\_beforek + (f\_afterk - f\_beforek) * (fi\_kloc - \text{ceil}(fi\_kloc))$$

We also define current and previous frames, *c's* and *p's*, each of which can be a filler image, the onset frame, or the apex frame. Except for the last current frame cm, which is the apex frame fn, we calculate the other current frames based on the equation above. The first previous frame *p1* is the onset frame *f1* while the previous frames *p2* ... *pm* are the current frames *c1*...*c(m−1)*, respectively. With the current frame *ck* and the previous frame *pk*, *gk* is the difference of them.

$$gk = ck - pk$$

Fig. 5 is an example of shift matrices of size 3. The number of frames from the original video can be any size.

5.2 Convolutional neural network

The network we use is much simpler than the other convolutional neural networks such as the GoogleNet [20] because the size of our database (even after combination) is small. We have two convolutional layers, each connected by a max-pooling layer, followed by two fully connected layers, and an output layer. Fig. 8 is the structure.

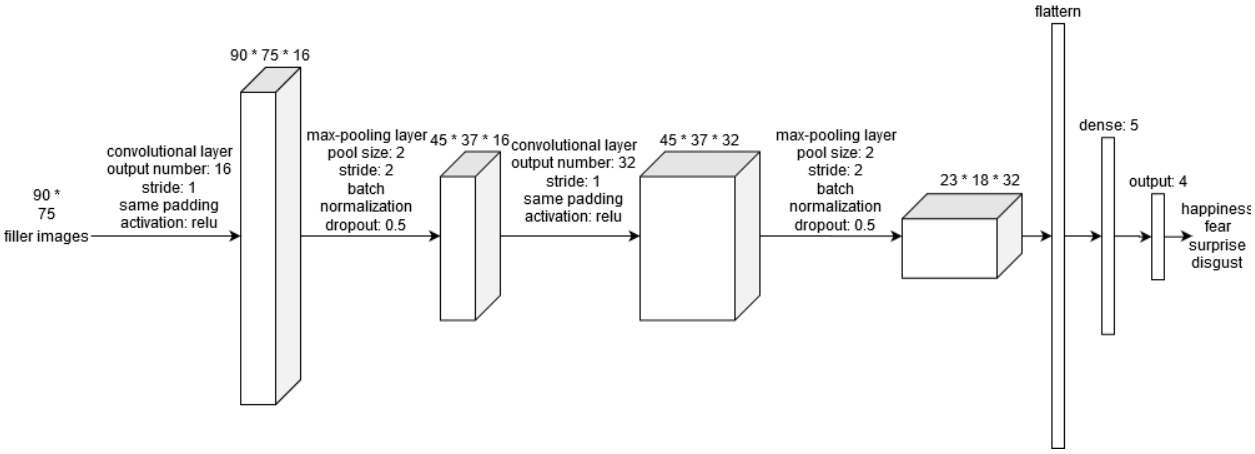

Fig. 8: This is our convolutional neural network. We have two convolutional layers, each followed by one max-pooling layer. At last, we have a flatten layer, followed by a dense layer and the output layer.

5.3 Experiment Setup

We experimented on the number of shift matrices, m, from 1 to 4. For each group, we created shift matrices based on the equations above and ran the experiment 100 times for each group. For each experiment, we randomly divided the shift matrices into training and testing sets (4: 1), trained the model with the training sets, and tested the model with the testing sets. When

training the model, we used 80 percent to train and 20 percent to validate. Finally, we recorded the accuracy and f1 score from the results of the testing set.

6. Results (and those of failed results)

6.1 Result

In Fig. 9, we show the results of the 400 experiments, 100 for each group delineated by the number of shift matrices used (G1, G2, G3 referring to one, two three shift matrices respectively). Results indicate that the G1 works the best. We calculated the median accuracy and f1 score and compared them to other published results. We report the median score instead of mean or max because there are some extreme values such as the lowest value of the f1 score of G1. The accuracy and f1 score are calculated with the equations below while the definitions of true/false positive/negative are in Table II. The expected outputs are the labels provided by the dataset authors while the actual outputs are the emotions predicted by our model.

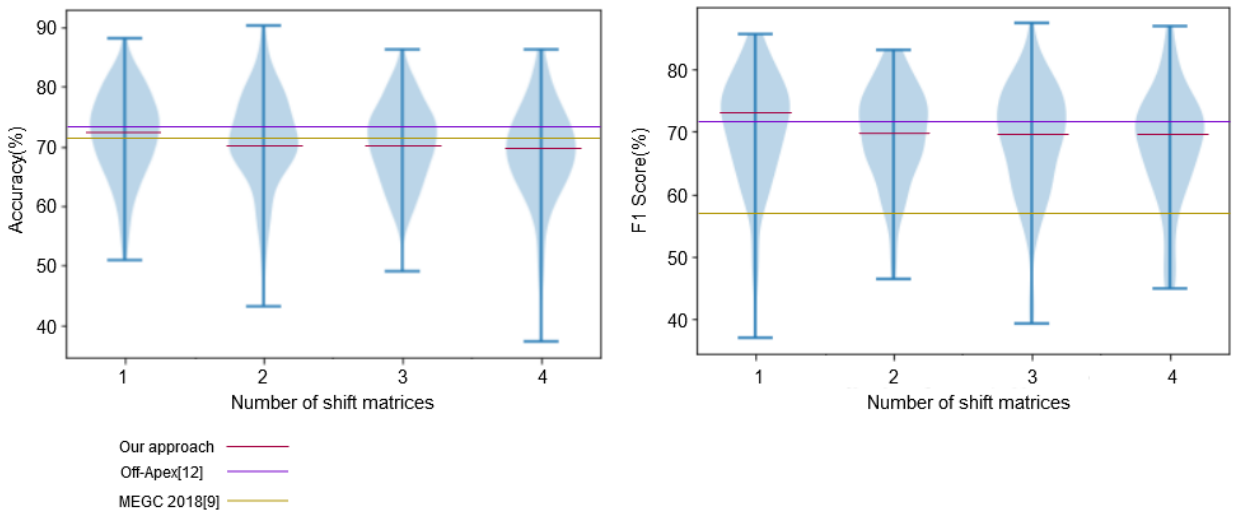

Fig.9: Violin plot of accuracies and f1 score for 100 experiments for each group of shift matrices. Our results are also compared with those of Liong etal.'s [12] and Merghani et al's. [9].

Table II: This is the definition of true/false positive/negative.

Expected output/actual output	true	false
true	True positive	False positive
false	True negative	False negative

$$accuracy = \frac{quantity\ of\ correctly\ classified\ samples}{quantity\ of\ samples}$$

$$precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative}$$

$$f1\ score = \frac{2 * precision * recall}{precision + recall}$$

In Table III, we compare our results to those of others.

Table III: Result and F1 score.

Approach	Classes	Databases	Accuracy	F1 score
OFF-ApexNet [12]	Positive Negative Surprise	SMIC[21] CASMEII SAMM	74.6%	71%
Evaluating Spatio-temporal features [9]	AU based classes	CASMEII SAMM	71.8%	57.9%
Our approach	Happiness Disgust Surprise Fear	CASME CASMEII SAMM	72.5%	72.3%

The results of experiments showed that the model of segmented landmarks did not work very well. We did not run it 400 times but 40 instead because it was very time-consuming. However, the general accuracy was only between 50% and 60%, though it was twice as fast as the model of the whole face. We also tried to use signs (+/-) instead of the actual differences

between each filler images as inputs, but it also did not work well, with accuracies between 40% and 60%.

7. Discussion

The exact reasons why G1 works the best remains unclear (as shown in Fig. 6), but we have some hypotheses. One way to think about it is that we are treating each frame as a node of the graph. As a result, more shift matrices mean more nodes on the graph of the videos and, thus, more potential features. However, the pixels that are not a part of the useful features will increase the noise as well. Due to the lack of data, an increase in noise may have a larger negative impact than the positive impact from the increase in features. We will need more videos with the same classification rules (FACS) to determine this.

Our approach is similar to previous work in several ways. Like others, we extract the features and train a network with them. However, our approach is more extendable since we classify the emotions with the given labels, and the datasets we use follow the same classification rules.

However, similar to others' approaches, we also have the problem of limited data, and this is why the results of each experiment vary so much, with a range of accuracy between 50.71% and 88.92%. The accuracy is much better than guessing (25%), which means our approach succeeds in extracting and training with the correct features in all experiments. Our accuracy is a little bit lower than the reported result of Liong et al. [12], however, we are reporting the median of the 100 experiments. Some of our experiments have a much better result than those to date, but we choose to report our median score because of the high variability and extreme values of accuracy and f1 in all G's.

8. Future work

We think one reason why G1 has the highest median accuracy and f1 score is that the influence of noise is higher than that of features. To solve this problem and further increase micro-expression recognition accuracy, more databases should be created. If more data becomes available, we will have more videos with consistent features. This should allow the features to be more influential than noise since the noise among the videos are generally inconsistent.

Another promising direction is to focus more on the bias of landmarks. As discussed in Section V, the areas around the landmarks (eyes, mouth, and nose) have the most dramatic change in intensity. One possible approach is to use different masks on and give different biases to different landmarks. Currently, each convolutional layer uses only one mask to loop through the whole face. However, since each landmark reveals a different feature, using one mask for each landmark should be more optimal.

An ambitious alternative to human experts manually labeling videos would be to use Generative Adversarial Networks (GAN) to swap faces and generate new videos based on current ones [22]. In pursuing this approach we expect an important goal would be to minimize the variance of results and increase the median accuracy. Fewer variations in the results will lead to more reliable features extracted. The higher the median accuracy is, the more convincing the model and result are.

9. Conclusion

The two main challenges for micro-expression recognition are the limited data and feature extraction, both of which we have addressed in this paper. Beyond these challenges,

there have been relatively few results reported in this area and often without much qualification of reproducibility. We have taken a more rigorous approach, offering more transparency of the variation of results when working with such limited datasets.

The first major challenge, the limited data, is additionally difficult in that currently available databases, including CASME, CASMEII, SAMM, and SMIC, are variant in many ways, such as the resolution and classification rules. Each database contains a relatively small amount of data that can be used to train the models. In our approach, we combined the databases by normalizing the videos from different databases. For example, to solve the problem of variant resolutions among the databases, we resized each image into the same resolution by dividing the images into the same amount of blocks.

The second major challenge for micro-expression recognition, feature extraction, consists largely of trying to keep the spatial information of the videos despite each video having different numbers of useful frames. We developed a new way of monitoring the function of muscle movement of the microexpressions. For each video, we standardized the number of frames between the starting and ending frames, creating filler images as needed and kept the changing pattern of pixels over time in shift matrices.

With our approach, the model we trained using the shift matrices was able to obtain a high accuracy and the highest f1 score to date (even with reporting only our median value). To increase transparency, reproducibility, and fair comparison, instead of testing once and reporting a single final result, we ran the experiments 100 times and reported on that collection of results to show the range of possible scores.

In conclusion, we have provided an extendable approach to micro-expression recognition that can be easily understood and adjusted as more feature extraction methods are developed and more data with the same classification rules (FACS) become available.

10. Lessons learned

In this section, I will talk about what I have done well and what I should have done differently during the research.

I organized the paper and put them into different folders. For example, I put the papers of CASME, CASMEII, SAMM, and SMIC into the database folder and created folders, such as those for CNN, micro-expression and old image processing methods. In this way, I found it very easy to find papers I needed when I wanted to look deeper into the details. Marking some important contents in those papers were also very helpful, especially the terminologies and key points of the methods.

However, I should have done some things better, such as writing tests. I found it very hard to debug after writing functions based on other functions, such as the one for creating shift matrices. This function depends on the function for creating filler images, which is also run after the video normalizing functions. Therefore, when I found out that there was something wrong with the shift matrices function, it took me a very long time to find the root problem. To avoid this, I should have written more tests on each small function to make sure they were correct.

To store the results, I used configuration files (python configparser). It is very useful since I could easily and clearly store the parameters, time, and results. I first generated one configuration file for each experiment with parameters in them. Then when running each experiment, I read through the file, ran the parameters in it, and store the results back in the same file. One advantage of the configuration file over CSV files is its clarity. It is almost as easy as CSV files on searching through results and at the same time much cleaner.

11. References

[1] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. Journal of personality and social psychology, 17(2):124, 1971.

[2] Paul Ekman, Wallace V Friesen, and Joseph C Hager. Facs investigator's guide. A human face, page 96, 2002.

[3] Clancy W Martin et al. The philosophy of deception. Oxford University Press on Demand, 2009.

[4] Tom Shapiro. How emotion-detection technology will change marketing. https://blog.hubspot.com/marketing/emotion-detection-technologymarketing. Accessed: 2020-03-14.

[5] Senya Polikovsky, Yoshinari Kameda, and Yuichi Ohta. Facial microexpressions recognition using high speed camera and 3d-gradient descriptor. 2009.

[6] Qi Wu, Xunbing Shen, and Xiaolan Fu. The machine knows what you are hiding: an automatic micro-expression recognition system. In international conference on affective computing and intelligent Interaction, pages 152–162. Springer, 2011.

[7] Yandan Wang, John See, Raphael C-W Phan, and Yee-Hui Oh. Lbp with six intersection points: Reducing redundant information in lbp-top for micro-expression recognition. In Asian conference on computer vision, pages 525–537. Springer, 2014.

[8] Riccardo Mattivi and Ling Shao. Human action recognition using lbp-top as sparse spatio-temporal feature descriptor. In International Conference on Computer Analysis of Images and Patterns, pages 740– 747. Springer, 2009.

[9] Walied Merghani, Adrian Davison, and Moi Yap. Facial microexpressions grand challenge 2018: evaluating spatio-temporal features for classification of objective classes. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pages 662–666. IEEE, 2018.

[10] Anh Cat Le Ngo, John See, and Raphael C-W Phan. Sparsity in dynamics of spontaneous subtle emotions: Analysis and application. IEEE Transactions on Affective Computing, 8(3):396–411, 2016.

[11] Devangini Patel, Xiaopeng Hong, and Guoying Zhao. Selective deep features for micro-expression recognition. In 2016 23rd International Conference on Pattern Recognition (ICPR), pages 2258–2263. IEEE, 2016.

[12] YS Gan, Sze-Teng Liong, Wei-Chuen Yau, Yen-Chang Huang, and LitKen Tan. Off-apexnet on micro-expression recognition system. Signal Processing: Image Communication, 74:129–139, 2019.

[13] Xiaohua Huang, Su-Jing Wang, Guoying Zhao, and Matti Piteikainen. Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection. In Proceedings of the IEEE international conference on computer vision workshops, pages 1–9, 2015.

[14] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen, and Xiaolan Fu. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. PloS one, 9(1), 2014.

[15] Sze-Teng Liong and KokSheik Wong. Micro-expression recognition using apex frame with phase information. In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 534–537. IEEE, 2017.

[16] Wen-Jing Yan, Qi Wu, Yong-Jin Liu, Su-Jing Wang, and Xiaolan Fu. Casme database: a dataset of spontaneous micro-expressions collected from neutralized faces. In 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG), pages 1–7. IEEE, 2013.

[17] Adrian K Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and Moi Hoon Yap. Samm: A spontaneous micro-facial movement dataset. IEEE Transactions on Affective Computing, 9(1):116–129, 2016.

[18] Matthew Shreve, Sridhar Godavarthy, Dmitry Goldgof, and Sudeep Sarkar. Macro-and micro-expression spotting in long videos using spatio-temporal strain. In Face and Gesture 2011, pages 51–56. IEEE, 2011.

[19] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, and Matti Pietikainen. A spontaneous micro-expression database: Inducement, collection and baseline. In 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pages 1–6. IEEE, 2013.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1– 9, 2015.

[21] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, and Matti Pietikainen. A spontaneous micro-expression database: Inducement, collection and baseline. In 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pages 1–6. IEEE, 2013.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[23] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.

[24] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.