Emotions Classification

Zekun Li
University of Electronic Science and Technology of China
Chengdu, China
kunkun0w0@std.uestc.edu.cn

Hongxin Song
University of California, San Diego
San Diego, United States
h3song@ucsd.edu

Haomiao Tang
Wuhan University
Wuhan, China
tanghm183@gmail.com

Yuebin Mao
University of Toronto
Toronto, Canada
yuebin.mao@mail.utoronto.ca

Tian Xia
University of California, San Diego
San Diego, United States
t3xia@ucsd.edu

Chenxin Yi
Tongji University
Shanghai,China
1951739@tongji.edu.cn

Abstract

Facial expression is an essential part of communication in human life. Automating the process of facial expression recognition, especially in natural conditions, has been a significant and challenging task. In this paper, we explore the problem of facial expression recognition through the FER-2013 dataset, which consists of over 30000 48x48 pixel grayscale images of faces cropped from various real-life pictures. The goal is to sort these pictures correctly into seven basic emotion classes. We address this task by Convolutional Neural Network, using both a shallow CNN model of our own as well as deeper models such as ResNet, VGG, and Inception. We fine-tuned these models and compared their performances on the dataset. Then we ensembled them to reach a better performance. As a result, we obtained a final model that reaches 70. 24% accuracy on the test set, beating two baselines (human recognition rate and null model accuracy) proposed by previous works.

1. Introduction

Facial expression is a significant indicator of a person's physical and psychological conditions. It plays a fundamental role in human communication. The ability to do Facial Expression Recognition (FER) automatically via computer vision enables novel applications in numerous fields such as human-computer interaction and mental healthcare [1, 2, 5]. For years, the potential of FER has driven countless researchers to dive into the task of classifying facial expressions in pictures, and significant progress has been made. In fact, recognizing basic expressions under controlled conditions (e.g. frontal faces and posed expressions) can now be considered a solved problem, with state-of-the-art accuracy reaching 98.9%. In real life, the situation is not as ideal. Pictures containing facial expressions often vary in head pose, illumination, and occlusions. Therefore a FER program needs to develop the ability to distinguish basic expressions under natural conditions. This, however, remains to be a challenging task. As shown in paper [14, 19], reliable FER under naturalistic conditions is still an unsolved problem.

Convolutional Neural Networks (CNN) seems to be by far the most promising type of network for this task. Several recent works on FER successfully utilize CNNs for feature extraction and inference [24, 26]. Among all the datasets used for benchmarking these CNN models, FER-2013 [6] is the most common image dataset in CNN-based FER and one of the largest publicly available datasets in this field. It originates from image data on Google and well represents the naturalistic scenarios where we may apply FER in real life.

This paper aims to propose a CNN model of our design that achieves an acceptable performance on the FER-2013 dataset. The acceptable performance means that the accuracy of our model beats some of the well-known baselines. We will introduce these baselines in section two. We scope the project within the range of this course, tuning our model with basic techniques such as data augmentation, dropout, early stopping, etc. We also experimented with transfer learning from several prominent deep CNN networks such as VGG, ResNet, and Inception. We make a comparison

between these models and ensemble them with our shallow CNN model to reached a higher accuracy.

We present our work in the following structure:

- In Sec. 2, we review the previous works on FER-2013 and state-of-the-art researches.
- In Sec. 3, we make a detailed description of the models we use and approaches to improve their performance, *i.e.*, ensemble learning.
- In Sec. 4, we evaluate and analyze our trained models not only on benchmark dataset but also in real world by our real-time facial expression recognition demo.
- In Secs. 5 and 6, we summarize our work and indicate further improvements that can be made.

2. Related Work

2.1. Baselines of FER

By far the most prevalent benchmark for natural-conditioned FER is the Facial Expression Recognition 2013 (FER-2013). It was introduced by Pierre Luc Carrier and Aaron Courville. They cropped facial expression images out of raw pictures collected by Google image search API. The images were examined by human labelers before being resized to 48×48 pixels and converted to grayscale. In 2013, the Kaggle contest on FER [6] presented a subset of this image collection, and mapped the fine-grained emotion keywords into seven broad categories. This dataset is then known as the FER-2013 dataset. Due to its internet-origin nature, FER-2013 is very close to real-life FER scenarios and serves as a fairly good test ground for the natural-conditioned FER task.

Shortly after the release of the FER-2013 dataset, Ian Goodfellow proposed a baseline for FER-2013 based on some small-scale experiments to estimate the human performance on this task. His findings suggested that human accuracy on FER-2013 was 65±5% [6]. Another baseline was proposed by James Bergstra [3] regarding the best performance of a "null" model, which is a model that consists of a convolutional network with no learning except in the final classifier layer. Using the TPE hyperparameter optimization algorithm, he found that the best such convolutional network obtains an accuracy of 60%. Using an ensemble of such models, he obtained an accuracy of 65.5%. In order to beat the two baselines mentioned above, a model needs to achieve an accuracy higher than 66% on the FER-2013 dataset.

2.2. High Performance Models on FER

Since the first occurrence of the FER-2013 dataset in 2013, there has been a wealth of existing research aiming to advance deep learning model performance on the FER-2013 dataset. S. Li and W. Deng made a comprehensive summary of the current state of deep-learning-based approaches to FER in their survey paper [14]. Among these approaches, CNN is definitely taking the lead. The top three teams in the 2013 Kaggle contest all used CNNs trained discriminatively with image transformations. The winner of the FER-2013 Kaggle competition combined CNN with a L2-SVM loss function for classification [24]. CNNs have shown great potential due to their computational efficiency and feature extraction capability. They are the most widely used deep model for FER task [9, 15, 25].

There are already some well-studied deep CNN models such as VGG [22], Inception [23], Resnet [7]. And successful attempts have been made to apply them to FER task [10, 18, 19]. In this paper we will also try to take advantage of these deep CNN models.

To the best of our knowledge, the state-of-the-art performance on the FER-2013 dataset was achieve by Khanzada et al. in 2020. they used ensembling with soft voting of seven models to obtain significant improvements in test accuracy [11]. The paper refered to a preceding state-of-the-art paper by Pramerdorfer and Kampel, in which the authors also ensembled six contemporary state-of-the-art papers to reach a test accuracy of 75.2% on FER-2013 [19]. For a single model, Zhang et al. achieved the highest accuracy of 75.1% by using auxiliary data and additional features that is a vector of HoG features were computed from face patches and processed by the first FC layer of the CNN (early fusion) [27]. They also processed the images for facial landmark registration. In the FER dataset, facial landmark extraction is inaccurate for about 15% of the images. The processing of facial landmarks will effectively improve this situation and increase the accuracy of facial landmark extraction. Another top-performance paper published by Kim et al. also did extensive processing of the images, including the utilization of face registration, data augmentation, additional features, and ensembling [12]. Some works [4, 8] also utilize bag-of-visual-words model [13] some handcraft feature to improve classification. Apparently, the key to high performance lies in data processing and ensembling multiple networks. Although auxiliary data is beyond the scope of this paper, ensemble learning is rather simple and we have implemented it in our model.

3. Approach

3.1. Shallow CNN Model

We first constructed a simple convolution neural network from scratch. The network architecture is illustrated by Fig. 1. It contains four 2D convolutions with batch normalization before activating function and 25% dropout layer after ReLU. The top of our model are 3 Dense Layer (256,512,7).

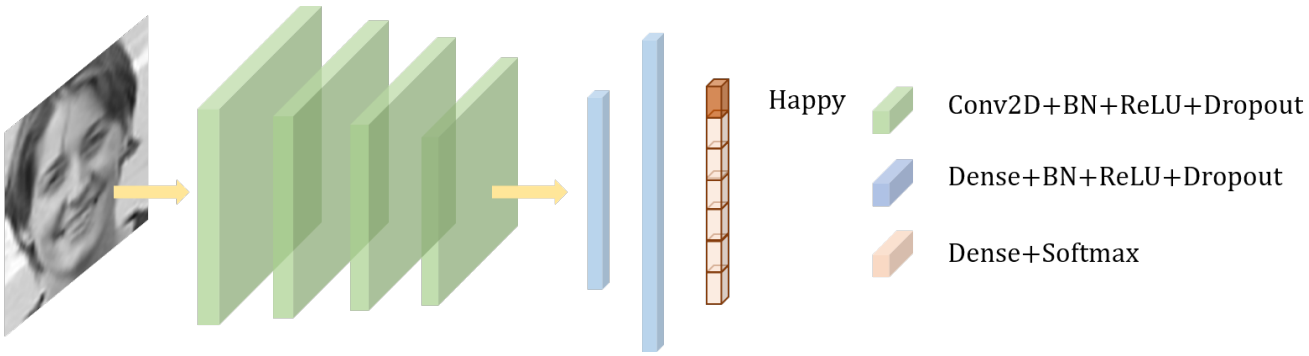


Figure 1. The shallow model containing 4 convolution blocks and 3 sets of fully connected layers.

We trained this model for 60 epochs on Google Colab. We did some careful fine tuning on the model. For training, we use Adam for optimizer. The learning rate is initially set to 0.001. And we use a reducing learning rate scheduler to stabilize the training process.

3.2. State-of-the-art Deep CNN Models and Transfer Learning

Apart from our own shallow CNN model, we also applied some state-of-the-art deep CNN models to the task, including VGG16, ResNet (both ResNet18 and ResNet50), and InceptionV3. We use these deep CNN models as our Backbone Architectures and add our own dense layers to the top. We train each model twice, once from scratch and once using transfer learning (freezing the weights of Convolutional layers and only training the parameters of the dense layers). We compare the performance of these models and choose the best one for the task.

VGG [22] To achieve better performance, we experimented with VGG, one of the most popular CNN backbone for representation learning. The original VGG16 model contains 16 layers, including 13 convolutional layers and 3 fully connected layers. The VGG16 network distributes a total of five pooling layers under different convolutional layers, and we use the maximum pooling layer. In VGG16, each convolutional layer contains 2 to 3 convolutional operations. The size of the convolution kernel is 3x3 with a convolution stride of 1, and the pooling kernel is 2x2 with a stride of 2. The most obvious improvement of VGG16 is to reduce the size of the convolution kernel and increase the number of convolution layers. This is because multilayer nonlinear layers can increase the network depth to ensure learning more complex patterns with less cost and fewer parameters. We incorporate VGG16 as a feature extraction tool by removing the last dense layer of the original model architecture and replacing it with a couple of new dense and dropout layers.The VGG16 is followed by one dropout layer with a rate of 0.5, a dense layer with 4096 nodes, another drop out layer with a rate of 0.5, and a dense layer with 1024 nodes. We experimented three appraoches, one trained from scratch with no pre-trained weights, one with pre-trained weights from ImageNet and one with pre-trained weights from VGGface [17]. For the two models with pre-loaded weights, we froze the VGG16 part of that model to perform transfer learning.

ResNet [7] Another network architecture we applied in our model is ResNet, including ResNet-18 and ResNet-50. ResNet-18 is a 18-layer convolutional neural network with the first layer being a 7×7 convolutional layer and second layer being a max pooling layer. It has 12 3×3 convolutional layers in the middle. The first four layers of these 12 layers have 64 kernels, and the next four layers having twice the amount of kernels per layer, and vice versa, with the last four layers having 512 kernels each, followed by a flatten layer. ResNet-50 is a similar but much deeper and complex 50-layer network. The first two layers of ResNet-50 is the same as that of ResNet-18, but followed by 48 convolutional layers organized into 16 blocks, where each block contains three convolutional layers. Besides being connected by previous blocks, each block is connected directly to the input, known as ”shortcut connections.” We incorporated the network with three fully-connected layers with 4096, 1024, and 7 nodes each for classification. For ResNet-50, we pre-loaded weights from ImageNet and freeze the ResNet-50 part of the model to perform feature extraction. For ResNet-18, we trained it from scratch with no pre-loaded weights.

Inception-V3 Inception-v3 is a convolutional neural network architecture from the Inception family. Is is made up of 42 layers which is a bit higher than the previous inception V1 and V2 models, but Inception-V3 used tricks such as factorization into Smaller convolutions and auxiliary classifiers to improve its efficiency and performance. Still, it is a very big model and we had 34,397,991 parameters in total after combining it with our dense layers. The training

process is extremely slow and it often gets stuck in 30-40% accuracy. After trying many times, we found a fairly stable set of configuration. We use the SGD optimizer instead of Adam, initialize the learning rate to 0.01, decay to 0.0001, applying Nesterov Momentum with the momentum coefficient set to 0.9. We also find that the model learns better if we enable vertical flip in data augmentation, which is quite curious. Furthermore, the Inception-V3 network requires the input image size to be at least 75×75, so we had to set the target size of ImageDataGenerator to (75, 75). By now, We are able to steadily reach 60% accuracy in 70-80 epochs. Our final training lasted for 200 epochs before we can confirm that it was overfitting.

3.3. Ensemble

After generating different neural networks for emotion classification, we attempted to combine all applicable models to gain higher performance. We choose the three best models out of each category with individual accuracy scores of over 62%. The models we used include the shallow CNN model, the VGG16 with VGGface [17] weights, and the Resnet-18 model. We obtained prediction probability matrices of the test set using these three models and summed the probabilities together. Then we used argmax to find the best class for each image.

3.4. Data Augmentation

To improve the robustness of our model in facial expression recognition, we apply a series of data augmentation during training process, *i.e.*, scaling, horizontally flipping and rotating. Randomly scaling the images range from -10% to 10% compared with its original scale helps model detect emotion in different resolutions. Due to human faces are almost symmetrical and this augmentation will not damage the facial expression semantics, we apply a random horizontal flip with a probability of 50%. Randomly rotating input images range from -10 to 10 degrees can also enrich the training dataset. Additionally, the flip and rotation can also provide various perspectives for model to facial expression recognition, which represents the possibility to validate model in wild.

3.5. Visualization Technique

Saliency Map [21] In order to interpret the classification standard in neural networks, we provide a formulaic description. Given an image I and a class c, according to the score function $S_c(\cdot)$, we can rank the pixels in I based on their influence on the score $S_c(I)$ formulated as follow.

$$\arg\max_I S_c(I) \tag{1}$$

Based on Taylor Formula, we can use the first order approximation to rewrite the score function as Eq. (2).

$$S_c(I) \approx \omega^\top I + b \tag{2}$$

Through this way, we represents each pixel's contribution to the final score by calculating the gradient of $S_c(I)$ according to Eq. (3).

$$\omega = \left.\frac{\partial S_c}{\partial I}\right|_{I=I_0} \tag{3}$$

Grad-CAM [20] Different from saliency mapping, class activation mapping(CAM) can estimate which area of the image the model focuses on when making a particular prediction. It represents a smooth visualization compared with pixel estimation method, *i.e.*, saliency mapping. Given c categories for classification, the MLP output $z \in \mathbb{R}^c$ can be formulated as follow:

$$z = [z_1, z_2, ..., z_c] \tag{4}$$

where z_i is related to the possibility for predicting the sample as i-th class.

Through back-propagation, we obtain the gradient $G \in \mathbb{R}^{k\times h\times w}$ the feature map $A \in \mathbb{R}^{k\times h\times w}$ output by the convolution backbone as Eq. (5), where $G^k_{i,j}$ represents the gradient of the pixel located at (x, y) in the s-th channel of A.

$$G^s_{x,y} = \frac{\partial z_i}{\partial A^k_{x,y}} \tag{5}$$

Applying global average pooling on G and get the weight vector $\alpha \in \mathbb{R}^k$ which is defined as follow:

$$\alpha = [\alpha_1, \alpha_2, ..., \alpha_k] \tag{6}$$

where α_i is calculated as Eq. (7).

$$\alpha_s = \frac{1}{h \times w}\sum_x^h \sum_y^h G^s_{x,y} \tag{7}$$

Utilizing the weight vector α to average A, the 2D activation map computed by Eq. (8) represented the contribution for the specific category in different region.

$$GradCAM = ReLU(\sum_s \alpha_s A^s) \tag{8}$$

4. Experiment Results

4.1. Dataset Inspection

We test our method on FER2013 (Facial Expression Recognition 2013 Dataset) [6] which is first introduced by the ICML 2013 Representation Learning Workshop's face expression recognition challenge. This dataset contains approximately 30,000 facial RGB images of different expressions with size restricted to 48×48, and there are 7 types of emotions, *i.e.*, Angry, Disgust, Fear, Happy, Neutral, Sad,

	Angry	Disgust	Fear	Happy
train	3995	436	4097	7215
test	958	111	1024	1774
	Neutral	Sad	Surprise	
train	4965	4830	4965	
test	1233	1247	831	

Table 1. The amount of samples in each category.

Surprise. Tab. 1 shows the detail of FER2013. Additionally, we visualize the distribution of the dataset in Fig. 2 illustrates the unbalanced quantity in different classes, especially Disgust which only has 436 for training and 111 for testing.

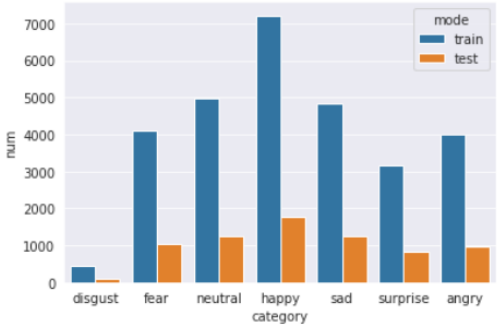


Figure 2. The data distribution shows the class imbalance in FER2013.

4.2. Model Comparison

The validation accuracies of all our models are shown in Tab. 2. It did not surprise us to see our Shallow CNN outperforming all the deeper and larger networks, as the Shallow CNN is much easier to train and we did a lot more fine-tuning on it. In fact, as illustrated in Fig. 3, our shallow CNN model has not yet reached its full potential and may achieve even better accuracy given the same time and epochs as the large networks.

Among the deeper networks, VGG16 with pre-trained weights of VGGFace [17] has the highest accuracy in all the models using transfer learning, perhaps because VGGface is originally trained on a face dataset which has much similarity to our task. Only two of the models beat the human performance and null model baseline. Our results suggest shallow networks perform better than deeper ones. We assigned sufficient epochs to all the deeper networks, so this phenomenon is more likely due to the nature of the dataset: 48×48 images might be too small for deep models like ResNet50 and InceptionV3, because the features break into meaningless pieces as they travel through many layers of convolution.

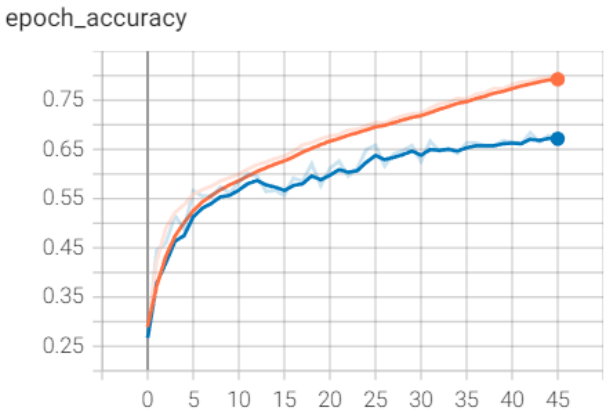


Figure 3. The train history plot of our shallow CNN model, showing both the training accuracy and validation accuracy

Model	Test Accuracy
Shallow CNN	**68.39%**
ResNet-50	48.53%
ResNet-18	64.25%
VGG16-Not Pretrained	57.89%
VGG16-ImageNet	67.22%
VGG16-VGGface [17]	67.65%
InceptionV3	62.58%
Human Performance	65%±5%
Null Model Performace [3]	65.5%

Table 2. Comparison of Model accuracies

4.3. Ensemble Neural Networks

Though the deep models didn't work as charming as expected in our hands, we can still take the advantage of them by ensembling them with our own network. The model after ensemble achieve a final accuracy of 70.24% which is 2% higher than the shallow CNN model, 2.5% higher than the VGG model and 6% higher than the ResNet-18 model as shown in Tab. 3. This indicates the effectiveness of ensemble learning in this condition.

Model	Test Accuracy
Shallow CNN	68.39%
ResNet-18	64.25%
VGG-16(VGGface [17])	67.65%
Average(3-Models)	**70.24%**

Table 3. Ensembled model and original models

4.4. Model Interpretation

Deep learning models are known as having the 'black box' nature, meaning it is difficult to directly interpret how models process the input image. Therefore, instinctive ways

to analyze models' performances are helpful for researchers to 'demystify' deep learning networks, and visualization has been proven to be an effective technique. In this project, we have applied several visualization methods for our deep learning models, including Gradient-weighted Class Activation Mapping (Grad-CAM), saliency map, and confusion matrix to help us understand the model's performance.

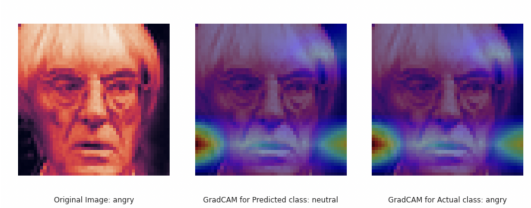
Grad-CAM Gradient-weighted Class Activation Mapping (Grad-CAM) is an effective approach to visualize and assess CNN performance. Grad-CAM propagates the gradients of the desired class (*e.g.*, "happy") to the convolution layers of the network, producing data that can be used to generate heat-map representing the important features of the image recognized by the model [20]. Using Grad-CAM allows us to visualize the regions of input images that deep learning models believe to be important, such as eyes, nose, and mouth with input images of human faces.

We have generated the Grad-CAM for our models for both correctly classified and incorrectly classified images, as shown in Fig. 4, where regions with higher importance have warmer color, while less important ones have colder color. The results indicate that in correctly classified images, important facial features (eyes and mouth) are properly recognized by our deep learning model, whereas in incorrectly classified images, such features are either incorrectly located or ignored. In the correctly classified example, the model detects the important feature (eyes and mouth), whereas in the incorrectly classified example, the model does not detect the important feature, which therefore fails to correctly predict the emotion in the image. Such information can help us understand reasons behind why images are correctly or incorrectly classified and improve the model accordingly.

Saliency Map With the Saliency Map, one of the significant visualization techniques in deep neural networks [21], we used it to show how the model processes and differentiates between different facial expressions. The Saliency Map can highlight the pixels that have the greatest impact on the loss value, so we can know the impact of each feature in the image on the final classification decision. Through saliency map, we can clearly see how the neural network classifies the important features of the picture, so that we can more clearly judge the reason why the neural network judges correctly or wrongly and adjust the network.

We generate the saliency map using our best performing network to understand how it classifies each emotion in the FER2013 dataset. Fig. 5 shows a saliency map for each emotion. Judging by all the images, our CNN can effectively capture most of the critical regions for each emotion. The model is placing a large importance on almost all facial features of the person in each image. This is most clearly

(a) A correctly classified angry face.



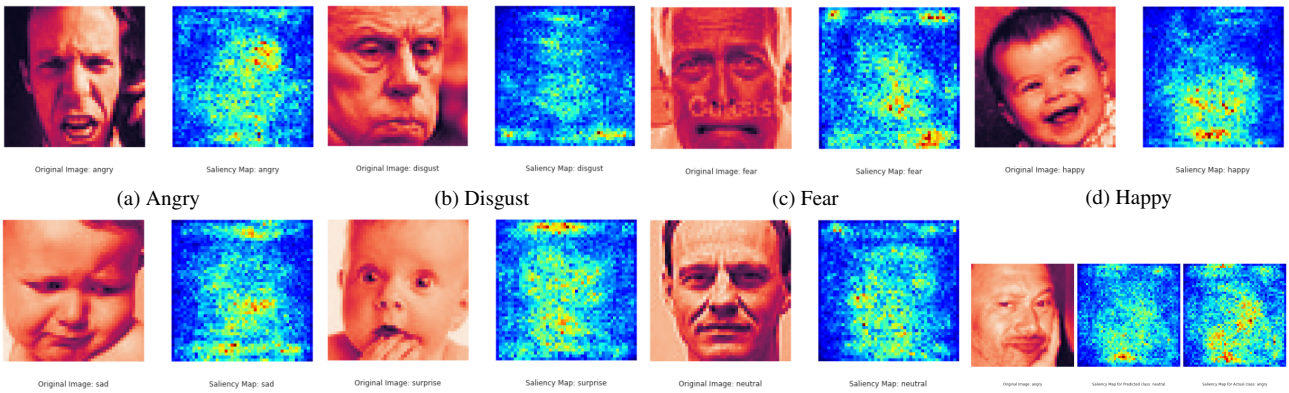
(b) Misclassify angry face as neutral face.

Figure 4. The Grad-CAM for correct case and failed case for angry face classification.

seen in Fig. 5g where the saliency map almost perfectly maps the entire face of the neutral man. Our model is also effectively dropping regions like the background in Figs. 5a, 5b and 5g, the hair in Figs. 5c, 5d and 5g, and the eyes and forehead in Fig. 5d (This is because for most of the times, we judge that if a person is happy through his smile, so the network just focus on the mouth and half of the face), all of which are not very informative when it comes to describing someone's emotion. There are some clear mistakes in the saliency maps, this can be seen in Figs. 5b, 5c and 5h where the model highlights some of the background pixels. We think that a model that can more effectively identify the facial features in an image and drop all useless information will perform better on this dataset.

Also, Fig. 5h shows one example about the angry emotion which is erroneous classified into neutral by the network. The saliency map shows that how the mistake come out: when the network works, it focus more on the face and the mouth pixels, but Ignore angry emotions in the eyes and nose pixels, so the classification is wrong. So we find that the saliency map is not that accurate, we should combine saliency map , gradcam map and confusion map to get a higher accuracy.

Confusion Matrix Confusion matrices are tables describing the performance of classification models. Its rows and columns represent actual classes and predicted classes, respectively. Fig. 6 shows the final model's confusion matrix on the FER2013 testing set. The model shows the best classification on the "happy" and "surprise" emotions. On the other hand, it makes the most mistakes when classifying be-



(a) Angry (b) Disgust (c) Fear (d) Happy

(e) Sad (f) Surprise (g) Neutral (h) Misclassify Angry as Neutral

Figure 5. Saliency maps for 7 correct cases and 1 failed case.

tween"neutral"and "sad". Next, the low classification accuracy in "fear" and "disgust" can be attributed to the fact that they have a lower number of samples in the original training set. The failure to distinguish "sad" and "fear" may be due to the inter-class similarities of the dataset. Also, from the confusion matrix, we find that the probability of correct classification is greater than 50%, up to 0.857, and the maximum probability of wrong classification is only 0.168, so we can judge the overall effect of the model from the matrix. However, due to different data samples and similar expressions, it is difficult to achieve high accuracy for all classifications, such as sadness, fear and anger, so we can manually identify and adjust, We train our model by increasing the amount of data of different similar expressions and correctly classified data. However, the matrix can only simply judge the correct or wrong classification of different expression labels using our model. It cannot judge how the neural network classifies and the reasons for the correct or wrong classification through the matrix. It cannot display the feature map. Therefore, we need to use salience map and Grad-CAM to visualize the features of different images.

Summary To better understand our network's behavior, we have employed various visualization methods including confusion matrix, saliency map, and Grad-CAM. By analyzing saliency map on correctly images in our model, we found that the network focus more on the whole face and the characteristics on the face instead of the environment variables like the background or the hair, in order to classify them correctly. However, with the grad-cam, we found that the network focus more on the details on the face such as nose, eyes, or part of mouths. In this way, the network can find the important features more easily, and compared to saliency map, we can get the importance of different pixels and features more quickly and clearly with grad-cam. Also,

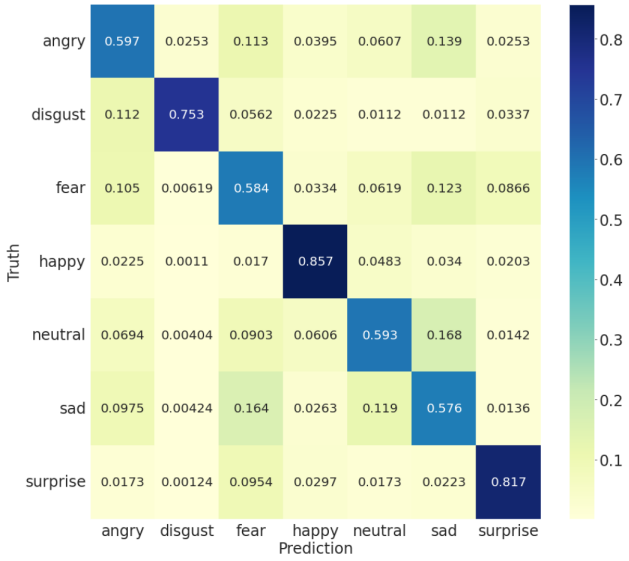


Figure 6. The confusion map validating on FER2013.

with confusion matrix, we can know which expression is more likely to be correctly recognized or incorrectly classified. Recognizing this as a class imbalance problem, we overcame it by collecting more error images for our model dataset. In all, three methods are all important for the result visulaization and better understanding of how the network works. Each method has its own characteristics and results, so we should combine the three methods to conduct comprehensive analysis, and adjust and improve the data and models according to the results to improve the accuracy and applicability.

4.5. Transfer Learning

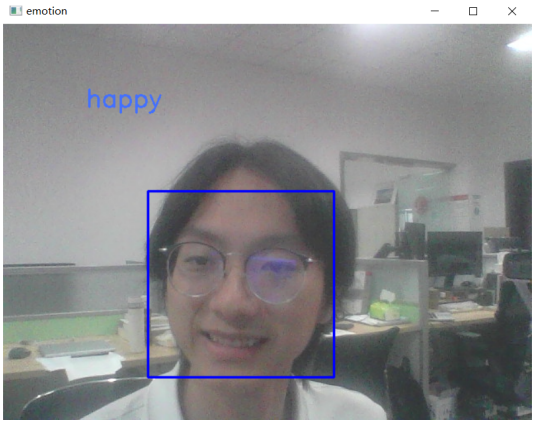
VGG [22] We experimented with three different approaches using the VGG16 model. First, we tried to utilize the VGG16 model structure directly and trained it from

scratch. The final test accuracy out of this approach is around 57.8% which is significantly lower than the shallow CNN model. Then we tried to load VGG16 with weights from ImageNet to extra important features. Since ImageNet is trained with a larger dataset that includes face images, we assumed that the model performance would improve by freezing the preloaded weights and only training the model with the other dropout and dense layers we added after the VGG16. The test accuracy for this approach is 67.2% which improved by about 9.4% compared to the last model. In order to further experiment with VGG16 transfer learning, we loaded the VGG16 model with weights from VGGfacenet [17]. Since the VGGfacenet is trained on a dataset that only contains numerous face images, it should have weights that work better with extracting facial features compared to that of ImageNet. We were able to obtain a final test accuracy of 67.7% which is 0.5% higher than that of the model with ImageNet weights. The models with pre-trained weights are better at predicting the emotion compared with the model with random weights. This shows the effectiveness of transfer learning and that using pre-loaded weights from task related network can help a lot with feature extraction.

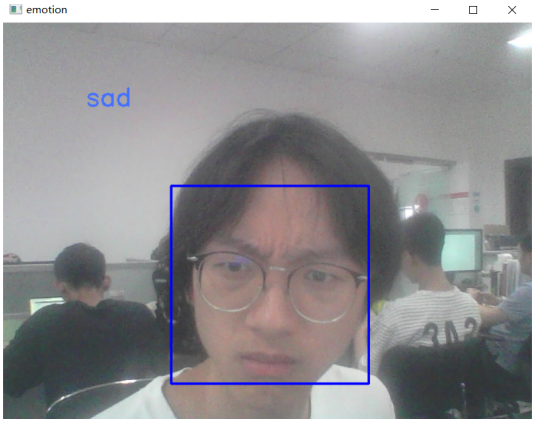
Resnet [7] Apart from VGG, we also examine deep neural networks model like Resnet. We first experimented with Resnet-50 with preloaded weights from ImageNet. Similar to the VGG transfer learning model, we removed the last dense layer of Resnet-50, followed by one dropout layer with a rate of 0.5, a dense layer with 4096 nodes, and another dropout layer with a rate of 0.5, and a dense layer with 1024 nodes. The test accuracy of Resnet-50 is 46.8%, the lowest out of all the models we have experimented with so far. The cause of this low accuracy is likely due to the large number of convolutional layers of this model. With a small input data size of 48X48, information is lost due to the deep model structure. Thus, we introduced Resnet-18, which has fewer layers compared with Resnet-50. Instead of loading pre-trained weights, we trained this model from scratch. The test accuracy for this model is 62.5%, significantly higher than the Resnet-50. This result confirms our theory of what caused the Resnet-50 to perform poorly. With a small input image size, there is no need to incorporate too many layers for the neural network.

4.6. Real World Testing

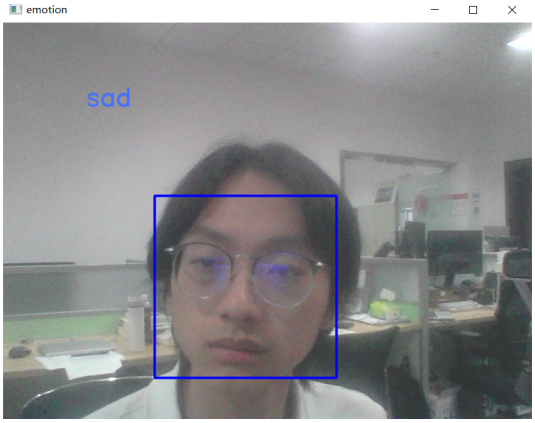
We utilize our model to classify the facial expression captured by OpenCV. In our demo, the emotion label represents the category of the human face in the bound box as Fig. 7 showed. Although we get an impressive performance in the benchmark dataset, we still find its incapability in real-world emotion classification. As the Fig. 7a, we find *Happy* is the easiest facial expression to identify for networks, which is consistent with the result of the confusion matrix in Fig. 6, *i.e.*, *Happy* get the highest classification accuracy. Additionally, we also represent some failed cases in Figs. 7b and 7c. Both the neutral face and the angry face are mistaken for sad expressions, which also validates *Neutral* and *Angry* are easily misclassified as *Sad* illustrated in



(a) A real happy face.



(b) A angry face is misclassified as sad.



(c) Maybe just nature facial expression, but neural network suggest I am sad.

Figure 7. User test results.

Fig. 6.

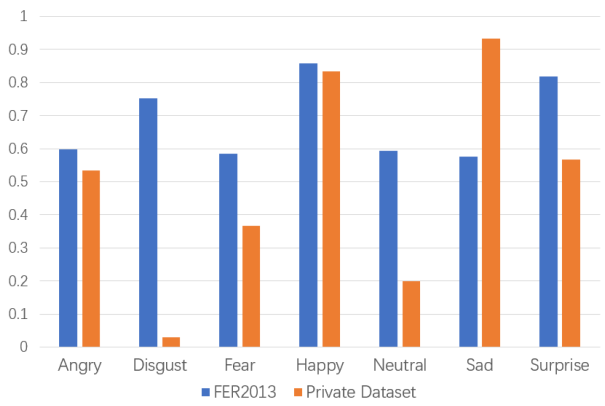


Figure 8. The shallow CNN performance comparison between FER2013 and our private dataset.

For quantitative analysis, we create a private test dataset that contains 30 facial images for each emotion category. The result in Fig. 8 shows that there is still a huge gap between real-world applications and benchmark validation. It is easy to notice that for some common expressions, *e.g.* *Happy* and *Sad*, the model can still distinguish with high accuracy. But for some rare negative expressions, *i.e.*, *Disjust*, which are highly personalized to each person. Some people will just frown but others will open their mouths, so it is difficult for the model to classify them accurately. Additionally, these features can often be mistaken for other common expressions such as *Sad* or *Angry*. However, there are few Asian faces in the FER2013 and we suggest it is also a reason why the model performs differently in our private dataset.

5. Conclusion

In this paper, we propose a shallow CNN for facial expression recognition and achieve much better than other deep CNN backbone. Additionally, we also adopt ensemble models to get a high classification accuracy up to 70.24%. Utilizing the techniques we learned in class, we complete abundant results analyses and model interpretation, *i.e.*, Grad-CAM, saliency mapping and confusion matrix,. Not only train a model from scratch, we attempt adopt transfer learning to get a higher accuracy. Although it failed due to the low resolution image is friendly to deep convolution neural network, but it still help us to improve our shallow CNN's performance by ensemble learning. Despite testing on FER2013, we use OpenCV to create a real-time facial expression recognition application for real world testing. Witnessing the huge gap in model performance between benchmark validation and real world testing, we convince that there still existing many issues in FER2013, such as class imbalance.

6. Limitaion and Future Work

Due to the limited time and resources, we are unable to construct a dataset with auxiliary data, auxiliary data has led to significant improvements in many researches, as shown in paper[ ]. So we expect that our model also has the potential of reaching a better accuracy if provided extra data for training, while applying different data augmentation techniques to address different camera brightness and angle issues. To further improve the accuracy of our models, we hope to utilize facial landmark detection and alignment [12], implement attentional CNNs [16], and retrain our network by occluding facial features irrelevant ton recognition [16]. We expect to carry out extra tuning on our model using Cosine Annealing and combine the training and validation datasets to further improve the classification accuracy. We expect to explore the effect of different activation functions on the classification accuracy of the model, including ReLU, L-ReLU, P-ReLU and Swish so on. For future work, we expect to explore different image processing techniques on FER2013 and investigate more ensembles of different deep learning architectures to further improve our performance in facial emotion recognition.

References

[1] F. Abdat, C. Maaoui, and A. Pruski. Human-computer interaction using emotion recognition from facial expression. In *Fifth Uksim European Symposium on Computer Modeling and Simulation*, 2012. 1

[2] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Conference on Computer Vision and Pattern Recognition Workshop*, 2003. 1

[3] James Bergstra and David D Cox. Hyperparameter optimization and boosting for classifying facial expressions: How good can a" null" model be? *arXiv preprint arXiv:1306.3476*, 2013. 2, 5

[4] Dawood Al Chanti and Alice Caplier. Improving bag-of-visual-words towards effective facial expressive image classification. *arXiv preprint arXiv:1810.00360*, 2018. 2

[5] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003. 1

[6] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*. Springer, 2013. 1, 2, 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2, 3, 8

[8] Radu Tudor Ionescu, Marius Popescu, and Cristian Grozea. Local learning to improve bag of visual words model for facial expression recognition. In *Workshop on challenges in representation learning*. ICML, 2013. 2

[9] Deepak Kumar Jain, Pourya Shamsolmoali, and Paramjit Sehdev. Extended deep neural network for facial emotion recognition. *Pattern Recognition Letters*, 120:69–74, 2019. 2

[10] Yousif Khaireddin and Zhuofa Chen. Facial emotion recognition: State of the art performance on fer2013. *arXiv preprint arXiv:2105.03588*, 2021. 2

[11] Amil Khanzada, Charles Bai, and Ferhat Turker Celepcikay. Facial expression recognition with deep learning. *arXiv preprint arXiv:2004.11823*, 2020. 2

[12] Bo-Kyeong Kim, Suh-Yeon Dong, Jihyeon Roh, Geonmin Kim, and Soo-Young Lee. Fusing aligned and non-aligned face information for automatic affect recognition in the wild: a deep learning approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016. 2, 9

[13] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2006. 2

[14] S. Li and W. Deng. Deep facial expression recognition: A survey. *arXiv preprint arXiv:1804.08348*, 2018. 1, 2

[15] Ninad Mehendale. Facial emotion recognition using convolutional neural networks (ferc). *SN Applied Sciences*, 2(3):1–8, 2020. 2

[16] Abdolrashidi A. Minaee S. Deep-emotion: Facial expression recognition using attentional convolutional network. 2019. 9

[17] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 3, 4, 5, 8

[18] Roberto Pecoraro, Valerio Basile, Viviana Bono, and Sara Gallo. Local multi-head channel self-attention for facial expression recognition. *arXiv preprint arXiv:2111.07224*, 2021. 2

[19] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*, 2016. 1, 2

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 2017. 4, 6

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations Workshop*, 2014. 4, 6

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2, 3, 7

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[24] Yichuan Tang. Deep learning using linear support vector machines. In *Workshop on Challenges in Representation Learning*. ICML, 2013. 1, 2

[25] Vedat Tümen, Ömer Faruk Söylemez, and Burhan Ergen. Facial emotion recognition on a dataset using convolutional neural network. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–5. IEEE, 2017. 2

[26] J Wang and M Mbuthia. Facenet: Facial expression recognition based on deep convolutional neural network. 2018. 1

[27] Zhanpeng Zhang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Learning social relation traits from face images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2