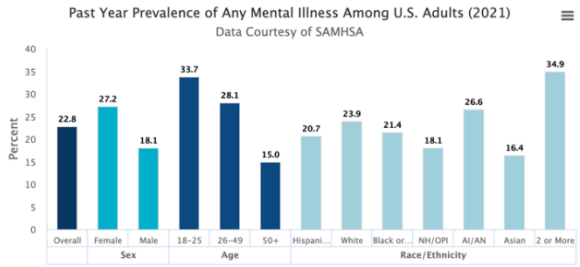
MoodVision: A Machine Learning model to detect the facial emotion of people and track their mental health

Group 15: Huiming Jiao, Luo Yang, Nisin Saj, Russell Quah, Xiao Yawen
GitHub Link: github /A-Machine-Learning-model-to-detect-facial-emotion-

1. Introduction

In recent years, mental health concerns are growing due to the fast-paced globalized lifestyle people are now experiencing. According to a study done by National Institute of Mental Health, 22.8% of adults in US are facing any mental illness. Therefore, there has been an increasing interest in the use of this technology to track the mental health of individuals.



Facial emotion recognition is an emerging technology that uses machine learning algorithms to detect and interpret human facial expressions. By analyzing facial expressions, researchers can gain insight into a person's emotional state and use this information to identify potential mental health issues. This approach has the potential to revolutionize mental health care by providing early detection and intervention for those in need. In this context, a machine learning model can be trained to detect and classify facial emotions accurately, leading to a more comprehensive understanding of mental health.

1.1 Use Cases

The machine learning model for facial emotion recognition has many potential use cases.: Some are below.

- Mental Health Screening: The model can be used in clinical settings to screen individuals for potential mental health issues by analyzing their facial expressions and detecting any signs of emotional distress.
- Remote Monitoring: The model can be used to remotely monitor the emotional state of individuals, such as patients with chronic illnesses or elderly individuals living alone, to detect any changes in their mental health and provide timely interventions.
- Education and Training: The model can be used to train mental health professionals, teachers, and caregivers to recognize and respond to emotional cues and expressions in their clients, students, or patients.
- Customer Service: The model can be integrated into customer service applications to identify customer emotions and provide tailored responses or interventions based on their emotional state.
- Gaming and Entertainment: The model can be used in the gaming industry to create more immersive and interactive experiences by detecting and responding to players' emotional states.

Our focus of the project will be on tracking the mental health of the people. Hence the rest of the report will just be focusing on the particular use case.

1.2 Business Value

From a user perspective, the improved mental health care provided by the model would offer several benefits. Firstly, early detection and intervention would mean that users can receive treatment for mental health issues earlier than they would have otherwise, which can improve their outcomes and reduce the severity of their symptoms. Additionally, users may feel more satisfied with the care they receive, as the model would help mental health professionals to provide more personalized and effective treatment. Finally, the reduced healthcare costs associated with early detection and intervention may mean that users pay less for their mental health care.

From a business perspective, the improved mental health care provided by the model can have significant benefits for mental health professionals and healthcare organizations. Firstly, early detection and intervention can improve patient outcomes and reduce the need for costly and time-consuming treatments. This can result in reduced healthcare costs for organizations and increased efficiency in treating mental health issues. Additionally, the use of the model can help mental health professionals to provide more personalized and effective treatment, which can result in increased patient satisfaction and retention. Finally, the adoption of the model can help organizations to differentiate themselves in a crowded market and attract new patients who are looking for advanced mental health care options.

Additionally, the use of the model would result in the generation of a large amount of data that could potentially be used for various analyses to identify patterns and trends in lifestyle or treatment that influence mental health.

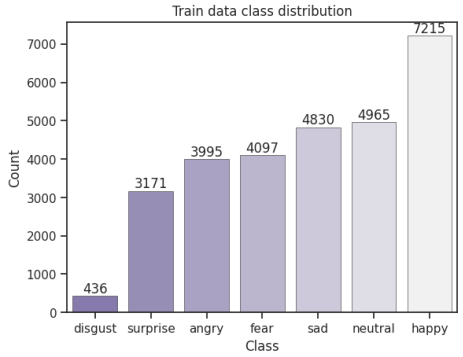
However, it is important to emphasize that the use of this data must be done ethically and with appropriate regulations in place to protect the privacy and confidentiality of patients. Organizations must ensure that they adhere to relevant laws and guidelines and that they have robust data protection policies in place to protect patient information. Only with these considerations in mind can the data generated by the model be used for valuable insights that can further improve mental health.

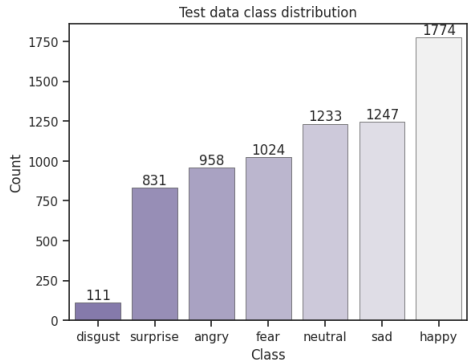
2. Facial Expression Recognition

2.1 Data Description

The dataset used for model training and evaluation was sourced from the [1]Facial Expression Recognition 2013 (FER-2013). The dataset consists of 35,887 grayscale images of faces, each of size 48x48 pixels, with each image labeled with one of seven different emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral as seen in the chart below:

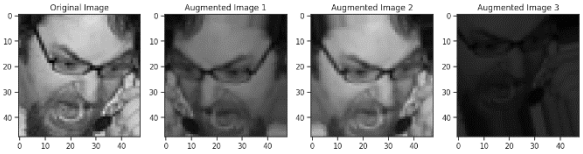
The 35,887 samples are further split into 28,709 samples for training and 7,178 samples for testing.





There is a heavy class imbalance in the training and testing dataset, where images of the "disgust" class are heavily underrepresented while the "happy" class is heavily overrepresented. This imbalance extends to the test dataset as well, as seen in the figure below.

2.2 Data Augmentation



Data augmentation is a commonly used technique that can be employed to expand and diversify the training dataset for machine learning models. This involves generating new training samples with various transformations applied to the original images or data, resulting in new variations of data. This allows the model to learn form a larger and more diverse set of data, which can help to make it more robust and enable it to generalize better to new and unseen dataset. In our models, we experimented with several augmentation parameters and discovered that:

- Rescale: This involved recalling the pixel values of the images to a range of 0 to 1, which helps the numerical stability.
- Rotation range: We rotated the images to various degrees to help the model learn different features from various angles.
- Width and height shift range: By shifting the image vertically and horizontally, we aimed to simulate variations in the position of the object of interest within the image.
- Shear range: We applied shearing to the images to help the model recognize features that are not perfectly aligned.
- Flip: horizontal or vertical flipping. In our project, we set flipping horizontally to create a mirror image. This will help to increase the amount of data available for training.
- Brightness range: Given the photos submitted by users could be in varying lighting conditions, it was important to enable the model to identify features in different lighting scenarios to improve the generalization.

For the validation and test datasets used in our models, only the rescale transformation was applied, as data augmentation is not required for these datasets.

[1] Data Set

https://www.kaggle.com/datasets/msambare/fer2013

Although data augmentation can enhance the performance of machine learning model, it is important to be cautious not to overdo it. The selection of appropriate augmentation parameters should be based on the specific required of the use case, and experimentation with different parameter values is recommended to determine the optimal set of parameters that achieve the desired results. By doing so, a balance can be struck between increasing the diversity of the training dataset and preventing overfitting of the model to the augmented data.

Moreover, to overcome the issue of imbalanced datasets, we applied class weights by assigning a weight to each class based on its frequency in the dataset. The less frequent classes were given higher weights, while more frequent classes were given lower weights. This approach can help to improve the overall performance of models on the imbalanced dataset.

2.3 Modelling

The modelling part of the report involved the exploration and evaluation of various machine learning models, including convolutional neural networks (CNNs), VGG16, and ResNet50. Each model was trained and evaluated using appropriate features and data preprocessing steps, with performance measured using the categorical accuracy score.

Class weights were used to offset the imbalanced distribution of the dataset, by applying higher weights to the underrepresented classes like ‘disgust’ and ‘surprise’ and applying lower weights to overrepresented classes like ‘happy’. The aim is to try to prevent the model from having a heavy bias towards classes that appear more frequently in the training dataset.

Class weights:

Index	Class	Weight
0	Angry	1.03
1	Disgust	9.40
2	Fear	1.00
3	Happy	0.568
4	Neutral	0.826
5	Sad	0.849
6	surprise	1.29

2.3.1 CNN

CNNs use convolutional layers to detect spatial patterns and features within images, allowing them to classify objects within images with high accuracy. CNNs are deep learning model that is particularly well-suited for image and video processing tasks. The core concept behind a CNN is that it uses multiple layers of convolution and pooling operations to progressively learn more complex representations of the input data.

In a typical CNN architecture, the first few layers will perform simple operations such as edge detection, while the later layers will learn to identify more complex features such as shapes, patterns, and textures. These features are then fed into one or more fully connected layers, which are used to perform classification or regression tasks based on the learned representations.

The key advantage of a CNN is that it can automatically learn to identify relevant features in an image or video, without requiring explicit feature engineering. This makes it a powerful tool for a wide range of computer vision applications, including object detection, image recognition, and video analysis.

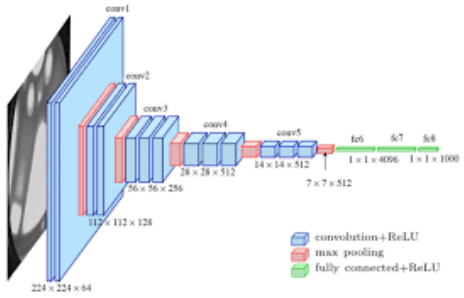
2.3.2 VGG16

VGG16 is a popular convolutional neural network architecture that was developed by the Visual Geometry Group at the University of Oxford. It consists of 16 convolutional and fully connected layers and is known for its deep architecture and high accuracy on image classification tasks.

Fine-tuning is a technique used in transfer learning, where a pre-trained model is used as a starting point for a new task. Fine-tuning involves taking a pre-trained model, such as VGG16, and retraining the last few layers on a new dataset. This allows the model to adapt to the new task while still benefiting from the learned representations in the pre-trained model. VGG aims to enhance the accuracy of its model by adding more layers, but if there are too many layers, such as over 20, the model may not converge due to a low learning rate, resulting in the inability to adjust its weights.

Dropout is a regularization technique used to prevent overfitting in deep neural networks. It works by randomly dropping out (setting to zero) a certain percentage of

neurons in a layer during training, forcing the network to learn more robust representations. When using dropout in a CNN, it is typically applied after the pooling layers to prevent overfitting to specific feature maps.



2.3.3 RESNET50

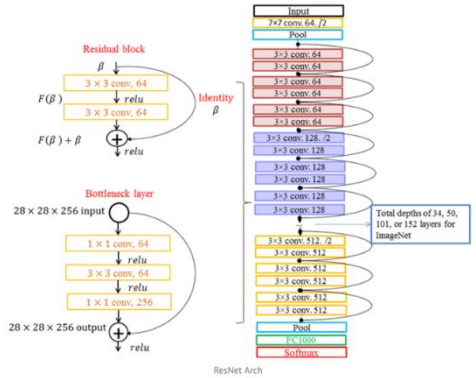
ResNet50 is a convolutional neural network architecture that was developed by Microsoft Research. It consists of 50 layers and is known for its deep architecture and its capability to address the vanishing gradient problem that can arise in deeper networks during training. This issue can be a significant challenge for deeper networks during the training. Despite having fewer filters and being less complex than VGG model, ResNet50 can still achieve impressive results in computer vision tasks. Furthermore, it has demonstrated better performance in reducing computational costs.

Fine-tuning is a transfer learning technique where a pre-trained model, such as ResNet50, is used as a starting point for a new task. The last few layers of the pre-trained model are replaced with new layers, and the entire network is trained on the new dataset. Fine-tuning allows the model to adapt to the new task while still leveraging the learned features from the pre-trained model.

Dropout is a regularization technique used to prevent overfitting in deep neural networks. In a ResNet50 model, dropout is typically applied after the fully connected layers to prevent overfitting to specific classes in the output layer. It works by randomly dropping out a certain percentage of neurons in a layer during training, forcing the network to learn more robust representations. ResNet50 is also pre-trained CNN model that have achieved state-of-the-art performance on the ImageNet dataset, which contains millions of labeled images. ResNet's architecture utilizes the idea of skipping connections. This enables the model to concentrate on learning new features after mastering a particular feature, resulting in a more efficient training process and improved accuracy. These pre-trained models have already been trained on a large amount of data and have learned to recognize many features, making them useful for transferring learning to a new task.

The VGG16 and ResNet50 pre-trained CNN models were fine-tuned to classify images related to mental health and detect potential issues using the augmented facial recognition dataset explained in the earlier part of the report. To achieve this, the last 4 layers of the models were trained while the rest of the layers were frozen. This approach is known as transfer learning, where the pre-trained model's learned features are used as a starting point for a new task.

During the fine-tuning process, the team experimented with adding dropout layers to the models to prevent overfitting. Dropout layers randomly deactivate a certain percentage of neurons during training, which can help the model generalize better to new data. The team also trained the models without a dropout layer to compare performance.



By fine-tuning pre-trained CNN models such as VGG16 and ResNet50, the project team was able to achieve high accuracy in classifying images based on facial expressions.

Ultimately, the report selected the best model based on the accuracy achieved, considering factors such as interpretability, ease of implementation, and computational resources required.

2.4 Model result

The CNN model used was based on the VGG architecture with 10 convolutional layers, 4 max pooling layers and 3 dense layers with dropout and normalization layers added to prevent overfitting. Additionally, max pooling layers of 2x2 were used to reduce the feature map (Figure 1 in Appendix).

For the variations in VGG19 (Figure 5 and Figure 9 in Appendix), the first variation used a softmax layer as the final output while the second variation added in additional Dropout layers with 3 further dense layers before the SoftMax activation layer.

Similarly, for the RESNET50 variations (Figure 13 and Figure 17 of the Appendix), a softmax output was used while the second variation had a deeper layer with dropouts.

In evaluating the performance of our model for facial emotion recognition, we used two key metrics: categorical accuracy and F1 score. The categorical accuracy measures the percentage of correctly predicted labels among all

samples in the test set. We chose this metric as it provides a simple and intuitive measure of overall model performance.

However, the categorical accuracy alone does not account for imbalances in the distribution of classes. In our dataset, some classes had significantly fewer samples than others, which could lead to misleading accuracy scores. To address this issue, we also used the F1 score, which is a harmonic mean of precision and recall.

The F1 score is a better metric to use when there are imbalances in the class distribution, as it considers both false positives and false negatives. This makes it a useful measure for evaluating the model's performance on each class, rather than just overall accuracy.

Table 1. Results of various machine learning models trained

Model Name	Categorical Accuracy Score	F1 Score
Customized CNN	0.63	0.64
VGG16+Softmax	0.27	0.32
VGG16+Dropout+3 Dense layer+Softmax	0.46	0.53
RESNET50+Softmax	0.27	0.27
RESNET50+Dropout+ Softmax	0.43	0.44

The customized CNN model outperforms the VGG16 and RESNET50 variations, with a categorical accuracy and F1 score of 0.63 and 0.64 respectively. Our team has two hypotheses on this.

Firstly, VGG and RESNET models were not pre-trained specifically for facial emotion recognition, but rather on a wide range of classification tasks. Their parameters may not be optimized to capture the subtle facial expressions and nuances that are crucial for accurate emotion recognition.

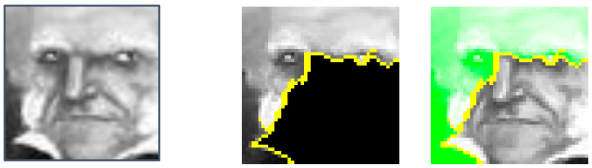
Secondly, our CNN model has all its parameters trained on facial emotion data, whereas the pre-trained models only had their last four layers fine-tuned with facial emotion data. This allowed our CNN model to better capture the unique features and characteristics of facial expressions that are critical for accurate emotion recognition.

2.5 Model Interpretation

To further understand how our CNN model makes its predictions, we applied Local Interpretable Model Explanation (LIME) method to explain two local data points. LIME works by perturbing the input data generating synthetic data, it trains an interpretable model on the synthetic data and the model's output for each synthetic instance.

Test image correctly detected as Happy.

The features learned by the model were everything around the forehead, which was highlighted in green. This indicates that the model associates happy emotions with features around the forehead, eyes, nose, mouth.



Test image (Sad) is wrongly detected as Angry

However, for the misclassified image, the CNN model learned features outside of the nose and mouth, particularly the eyes and forehead. It picked up angry eyes and frowning foreheads, which made it classify the photo as angry.

3. Mental Health Tracking

It is crucial and beneficial to identify the early signs of depression or other mental health conditions. Our mental health tracking system will provide real-time results in detecting changes in mental state and self-regulation and help people to self-control. In this section, we will be discussing the use of facial recognition technology for mental health tracking and exploring our deployment strategy and system architecture. Our focus will be on how the facial recognition model results will be utilized in decision-making, as well as the overall design of the system, including the model, web app, and other key components.

3.1 Model Deployment Strategy

Below is the step-by-step strategy for deploying the model and making it available to users

1. Host the machine learning model we developed on a cloud platform such as Amazon Web Services (AWS) or Microsoft Azure.
2. Build a web application using a framework such as Django or Flask.

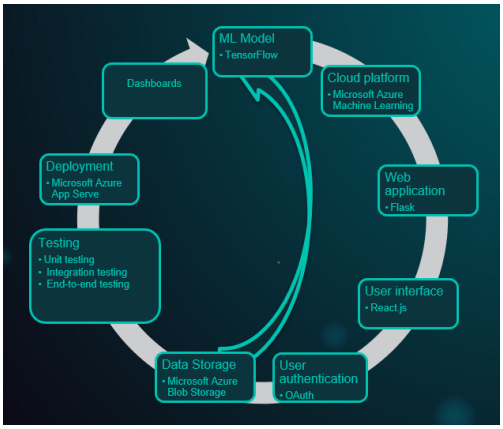
3. Integrate the hosted machine learning model with the web application to allow users to submit their selfies and get the predicted emotion.
4. Develop a user-friendly interface for the web application to make it easy for users to navigate and understand the results.
5. Enable user authentication to ensure that only authorized users can access their personalized reports.
6. Allow users to record their daily emotional state and track their mental health status over time.
7. Develop a dashboard for businesses to monitor the mental health status of their patients and track their progress.
8. Enable secure data storage to ensure that sensitive information is protected.
9. Conduct rigorous testing to ensure that the web application is bug-free and user-friendly.
10. Deploy the web application to a server, making it available for users and businesses to access.

3.2 System architecture

Based on the strategy explained earlier, we came up with a model deployment cycle. The process to draw a system architecture for the mentioned requirements involves several steps. The first step is to identify the different components required for the system architecture, such as the machine learning model, cloud platform, web application, user interface, user authentication, data storage, testing, and deployment. Here we already have a trained machine learning model which is customized to CNN model.

In the next step we created a diagram that shows how these different components are interconnected. The diagram provided an overview of the system architecture and how the components interact with each other.

After creating the initial diagram, more details are added to show how the different components interact with each other and what services are used. Security features are then added to the diagram to show how sensitive information is protected.



Finally, monitoring features are added to the diagram to show how businesses can monitor the mental health status of their patients and track their progress. Also, users themselves can monitor their mental health status and take timely decisions. In most cases mental health issues go undetected until the late stages and we hope the system that we are coming up with will help in those cases. This process helps to ensure that the system architecture is comprehensive and can meet the requirements mentioned in the project

We also developed the concept of the web application using HTML to get a feel of the solution that we going to deploy. Below is the picture showing the user interface of the web app concept.

The portal enables users to access their own history and receive relevant guidelines, without the need for external assistance.

3.3 Mental health scoring

To start with mental health monitoring, we need to understand the correlation between facial emotions and mental health. Research has shown that the monitoring changes in facial emotion over time may provide useful insight into a person's mental health status. For example, studies suggested that the reduction in intensity or frequency of smiles may indicate the decline in mental well-being. Different emotions can reflect and represent people's mental health to a different extent. Thus, we have created a comprehensive scoring system to measure the seven basic emotions. We have assigned score and weight to each emotion based on its importance in addressing mental health concerns. These weights range from 0 to 1, with a higher weight indicating greater impact on mental health status in negative direction.

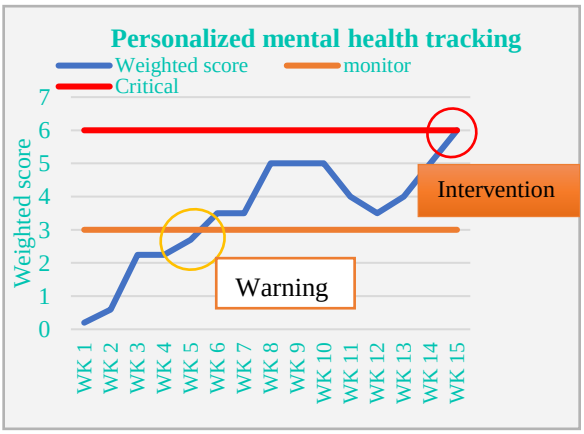
Based on the score and weight matrix, the weighted score can be calculated on a weekly, monthly, quarterly basis etc. By plotting this score over time, we can create a personalized mental health chart. Besides, we have identified 2 important thresholds which represent 2 key stages. The warning line which has a weighted score of 3 and intervention line which has a weighted score of 6. If the score exceeds the warning line, it will remind the user to take more care of their mental status while intervention line is a sign that the user should seek professional consultations.

Emotion	Score	Weight
Angry	8	0.15
Disgust	6	0.1
Fear	9	0.25
Happy	1	0.05
Sad	9	0.3
Neutral	2	0.05
Surprise	2	0.1

Table 2. Emotion scoring and Weightage

Below chart is a personalized mental health tracking chart that recorded the user's weekly mental score over 3 months period. When the user's mental score exceeds 3, the system will generate warnings and when the score exceeds 6, the system will highly recommend the user take intervention actions. By continuously monitoring the emotional responses of individuals, we can detect the abnormal mental status in an early stage.

Based on the scoring and weightage defined earlier ,below is a sample of the monitoring chart generated



4. Conclusion

Mental health tracking systems and machine learning models using facial recognition are important tools for improving mental health outcomes in today's world. They can provide valuable insights, early intervention, and personalized recommendations for treatment and self-care, using objective measures to overcome the limitations of traditional assessment methods.

Our team developed a machine learning model that can accurately predict emotions based on facial expressions. To ensure the model's accuracy, we utilized domain knowledge and expertise in the field of emotion recognition.

Moreover, we developed a scoring mechanism that links the model's output to a mental health tracking system, enabling users to gain insights into their mental health status based on the analysis of their facial expressions.

To make the model accessible to the public, we planned a deployment strategy that considers the practical challenges associated with implementing the system in real-world settings.

Finally, we created a user-friendly web application that demonstrates the functionality of the system using HTML. The application provides a simple and intuitive interface for users to upload their images and receive predictions and insights about their emotions and mental health status.,

5. Future works

Facial emotion recognition is indeed a difficult task to achieve, as expressions can vary greatly depending on the individual and their facial characteristics. To improve the accuracy and dynamism of the model, training it based on relevant data specific to the region of use can be beneficial. Training facial emotion recognition models on regional-specific data can help to improve their accuracy and dynamism, as well as mitigate potential biases in the model.

To improve the accuracy and stability of our face emotion detection models and make the model more generalizable , we can also explore other datasets such as FER plus, affectnet, CK plus, or real dataset available around the globe such as user-submitted images with meeting all regulations .

Moreover, we can enrich our training dataset by acquiring more data sources, such as user-submitted images. This approach will provide a broader range of examples for the models to learn from, which may involve diverse sources, including different genders, ages or races. Expanding the dataset to consider a diverse population demographic can aid in preventing any potential biases towards specific groups, resulting in improved generalization and robustness. Ultimately, employing these strategies can help enhance the precision and effectiveness of our face emotion detection models.

We can also consider using an ensemble of modes to reduce the impact of error from individual models and improve the overall performance.

Accurate and reliable identification of stress or depression is crucial and requires a stable analysis and a valid experimental methodology framework to avoid misleading

results that could increase stress levels or exacerbate mental health problems. Our mental health tracking system can be improved over time to enhance the accuracy of detection, expand its capability to detect a broader range of mental health disorders and help healthcare professionals more accurately evaluate patients' symptoms. Furthermore, other companies can introduce this mental health tracking system generally to assist employees in monitoring their mental health regularly and help employers take better care of their employees' health. This can lead to a more productive and positive work environment in the long run.

References

Shiqing Zhang, *measuring depression severity based on facial expression and body movement using deep convolutional neural network*, Taizhou University, China, 2022.

Bhattacharya, A. (2022, October 26). *How to Explain Image Classifiers Using LIME*. Medium. https://towardsdatascience.com/how-to-explain-image-classifiers-using-lime-e364097335b4

Winastwan, R. (2021, January 20). Interpreting image classification model with lime. Medium. Retrieved April 22, 2023, from https://towardsdatascience.com/interpreting-image-classification-model-with-lime-1e7064a2f2e5

Mental illness (no date) National Institute of Mental Health. U.S. Department of Health and Human Services. Available at: https://www.nimh.nih.gov/health/statistics/mental-illness (Accessed: April 23, 2023).

Appendix

Figure 1: Customized CNN Model Summary

dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv3_1 (Conv2D)	(None, 12, 12, 256)	295168
batchnorm_5 (BatchNormalization)	(None, 12, 12, 256)	1024
conv3_2 (Conv2D)	(None, 12, 12, 256)	590080
batchnorm_6 (BatchNormalization)	(None, 12, 12, 256)	1024
conv3_3 (Conv2D)	(None, 12, 12, 256)	590080
batchnorm_7 (BatchNormalization)	(None, 12, 12, 256)	1024
m_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
dropout_3 (Dropout)	(None, 6, 6, 256)	0
conv4_1 (Conv2D)	(None, 6, 6, 512)	1180160
batchnorm_8 (BatchNormalization)	(None, 6, 6, 512)	2048
conv4_2 (Conv2D)	(None, 6, 6, 512)	2359808
batchnorm_9 (BatchNormalization)	(None, 6, 6, 512)	2048
conv4_3 (Conv2D)	(None, 6, 6, 512)	2359808
batchnorm_10 (BatchNormalization)	(None, 6, 6, 512)	2048
m_pooling2d_4 (MaxPooling2D)	(None, 3, 3, 512)	0
dropout_4 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
fc_3 (Dense)	(None, 256)	1179904
fc_4 (Dense)	(None, 128)	32896
dropout_5 (Dropout)	(None, 128)	0
output (Dense)	(None, 7)	903

Total params: 8,859,719
Trainable params: 8,854,343
Non-trainable params: 5,376

Figure 2: Customized CNN Categorical Accuracy and Model Loss

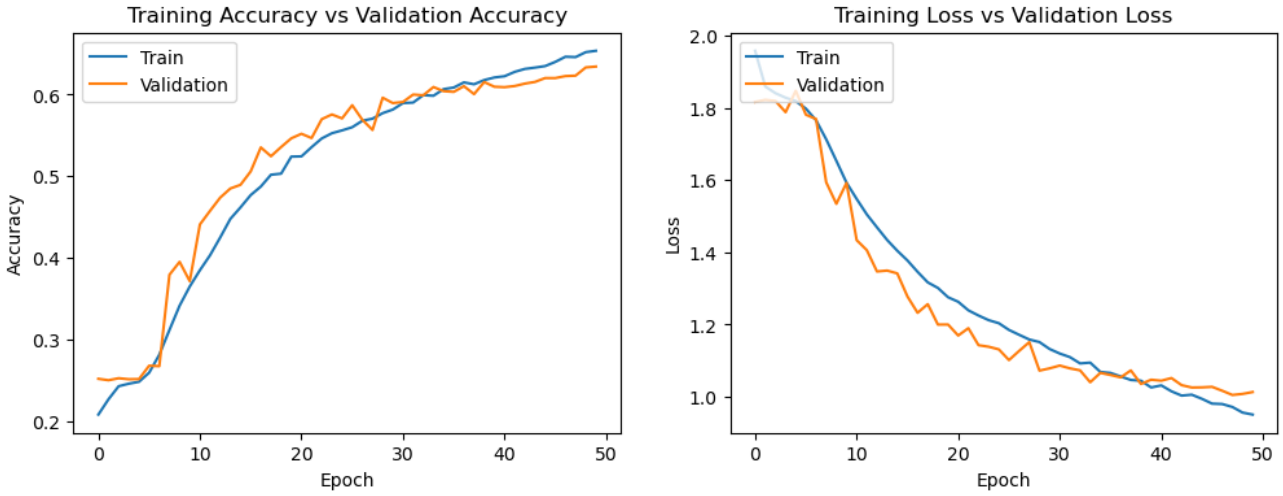


Figure 3: Customized CNN Classification Report

	precision	recall	f1-score	support
Angry	0.56	0.59	0.57	958
Disgust	0.53	0.40	0.45	111
Fear	0.52	0.28	0.36	1024
Happy	0.87	0.87	0.87	1774
Neutral	0.54	0.72	0.62	1233
Sad	0.51	0.57	0.54	1247
Surprise	0.80	0.71	0.75	831
accuracy			0.64	7178
macro avg	0.62	0.59	0.60	7178
weighted avg	0.65	0.64	0.64	7178

Figure 4: Customized CNN Confusion Matrix (Normalized)

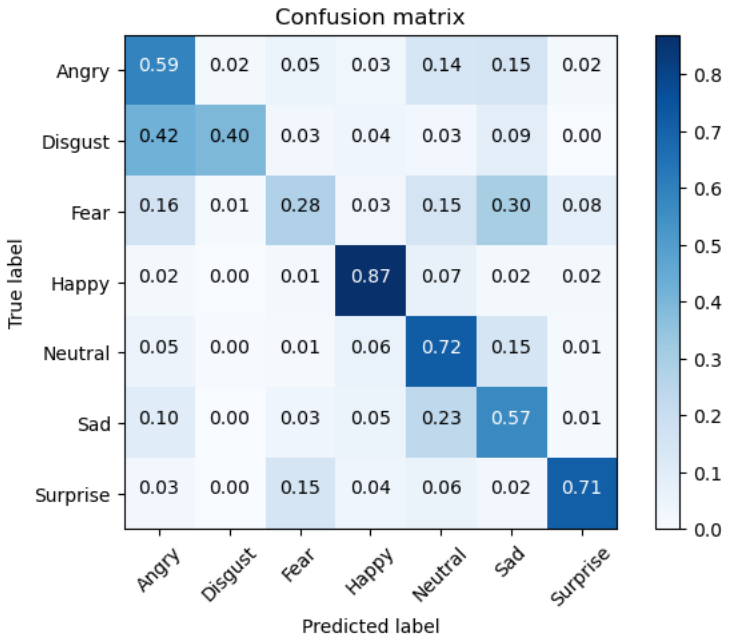


Figure 5: VGG16+Softmax Model Summary

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 48, 48, 3)]       0

 vgg16 (Functional)          (None, 1, 1, 512)         14714688

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 7)                 3591

=================================================================
Total params: 14,718,279
Trainable params: 3,591
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 6: VGG16+Softmax Categorical Accuracy and Model Loss

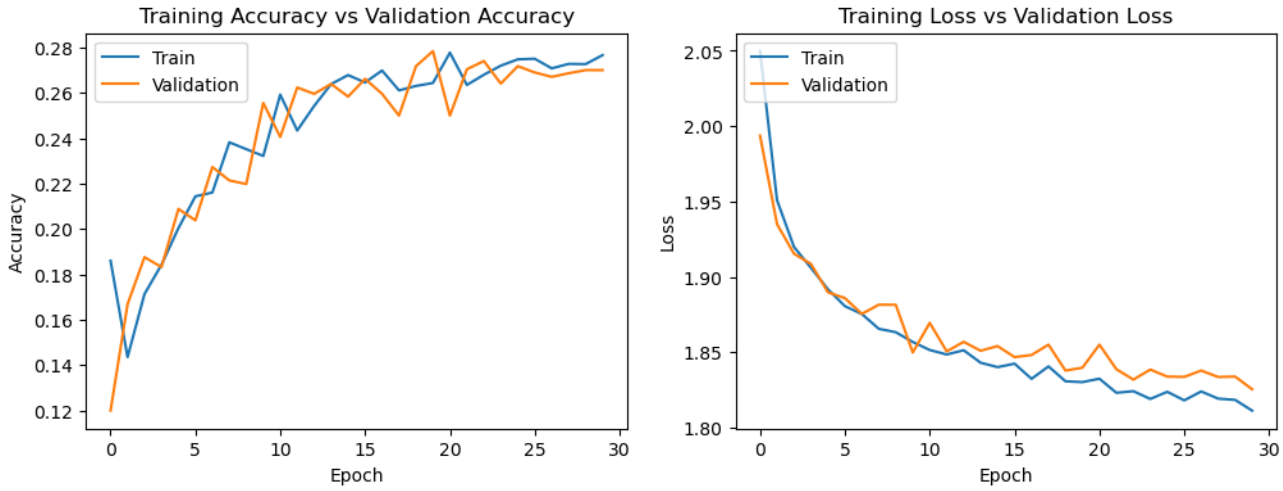


Figure 7: VGG16+Softmax Classification Report

	precision	recall	f1-score	support
Angry	0.27	0.16	0.20	958
Disgust	0.06	0.18	0.09	111
Fear	0.25	0.14	0.18	1024
Happy	0.41	0.47	0.44	1774
Neutral	0.30	0.33	0.32	1233
Sad	0.41	0.17	0.24	1247
Surprise	0.30	0.65	0.41	831
accuracy			0.32	7178
macro avg	0.29	0.30	0.27	7178
weighted avg	0.33	0.32	0.30	7178

Figure 8: VGG16+Softmax Confusion Matrix

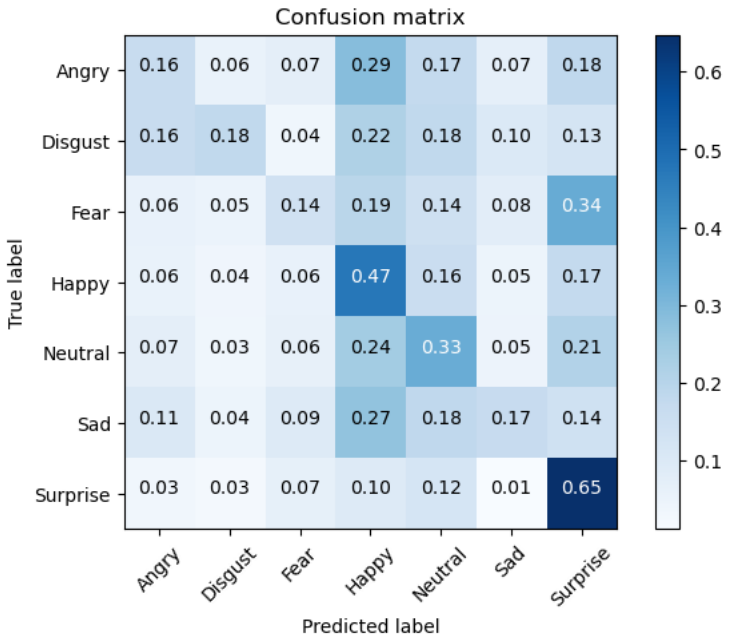


Figure 9: VGG16+Dropout+Dense+Softmax Model Summary

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 48, 48, 3)]	0
vgg16 (Functional)	(None, 1, 1, 512)	14714688
flatten_1 (Flatten)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 4096)	2101248
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 7)	28679

Total params: 33,625,927
Trainable params: 25,990,663
Non-trainable params: 7,635,264

Figure 10: VGG16+Dropout+Dense+Softmax Categorical Accuracy and Model Loss

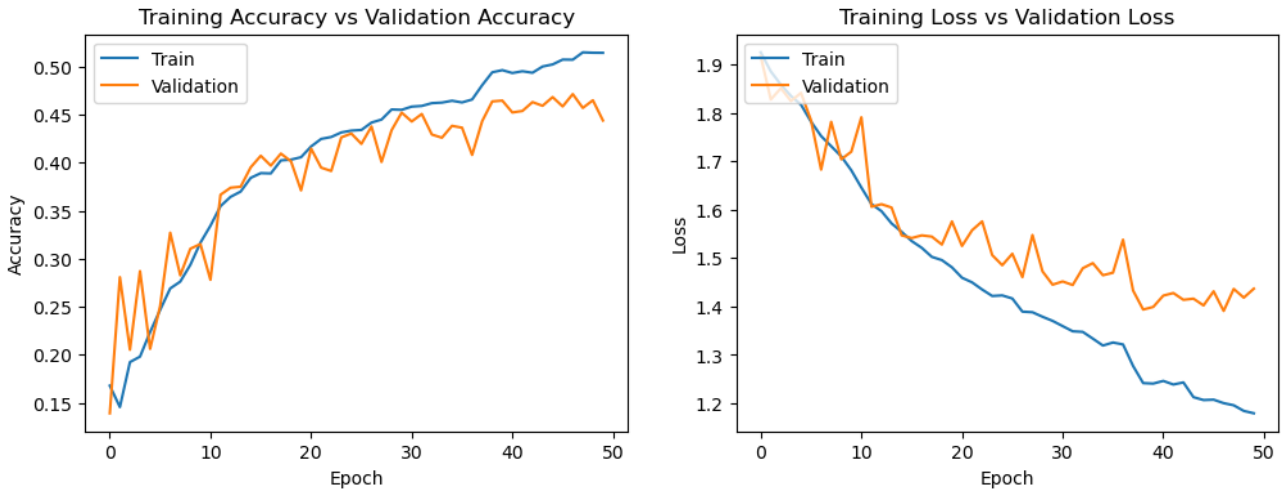


Figure 11: VGG16+Dropout+Dense+Softmax Classification Report

	precision	recall	f1-score	support
Angry	0.46	0.42	0.44	958
Disgust	0.19	0.61	0.29	111
Fear	0.43	0.29	0.35	1024
Happy	0.78	0.65	0.71	1774
Neutral	0.42	0.62	0.50	1233
Sad	0.49	0.36	0.42	1247
Surprise	0.62	0.76	0.68	831
accuracy			0.53	7178
macro avg	0.48	0.53	0.48	7178
weighted avg	0.55	0.53	0.53	7178

Figure 12: VGG16+Dropout+Dense+Softmax Confusion Matrix (Normalized)

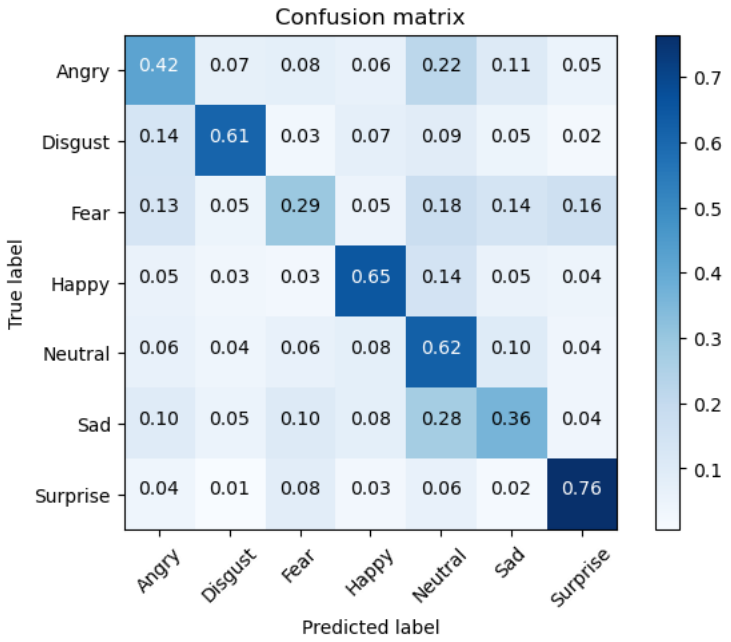

Confusion matrix
Angry 0.42 0.07 0.08 0.06 0.22 0.11 0.05
Disgust 0.14 0.61 0.03 0.07 0.09 0.05 0.02
Fear 0.13 0.05 0.29 0.05 0.18 0.14 0.16
Happy 0.05 0.03 0.03 0.65 0.14 0.05 0.04
Neutral 0.06 0.04 0.06 0.08 0.62 0.10 0.04
Sad 0.10 0.05 0.10 0.08 0.28 0.36 0.04
Surprise 0.04 0.01 0.08 0.03 0.06 0.02 0.76
True label
Angry Disgust Fear Happy Neutral Sad Surprise
Predicted label
0.7 0.6 0.5 0.4 0.3 0.2 0.1

Figure 13: RESNET50+Softmax Model Summary

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2, 2, 2048)	23587712
flatten_1 (Flatten)	(None, 8192)	0
batch_normalization (BatchNormalization)	(None, 8192)	32768
dense_2 (Dense)	(None, 32)	262176
batch_normalization_1 (BatchNormalization)	(None, 32)	128
activation (Activation)	(None, 32)	0
dense_3 (Dense)	(None, 7)	231

Total params: 23,883,015
Trainable params: 1,333,575
Non-trainable params: 22,549,440

Figure 14: RESNET50+Softmax Categorical Accuracy and Model Loss

Figure 15: RESNET50+Softmax Classification Report

	precision	recall	f1-score	support
Angry	0.20	0.36	0.26	958
Disgust	0.03	0.31	0.06	111
Fear	0.25	0.08	0.12	1024
Happy	0.37	0.47	0.41	1774
Neutral	0.36	0.08	0.13	1233
Sad	0.32	0.13	0.19	1247
Surprise	0.35	0.45	0.39	831
accuracy			0.27	7178
macro avg	0.27	0.27	0.22	7178
weighted avg	0.31	0.27	0.26	7178

Figure 16: RESNET50+Softmax Confusion Matrix

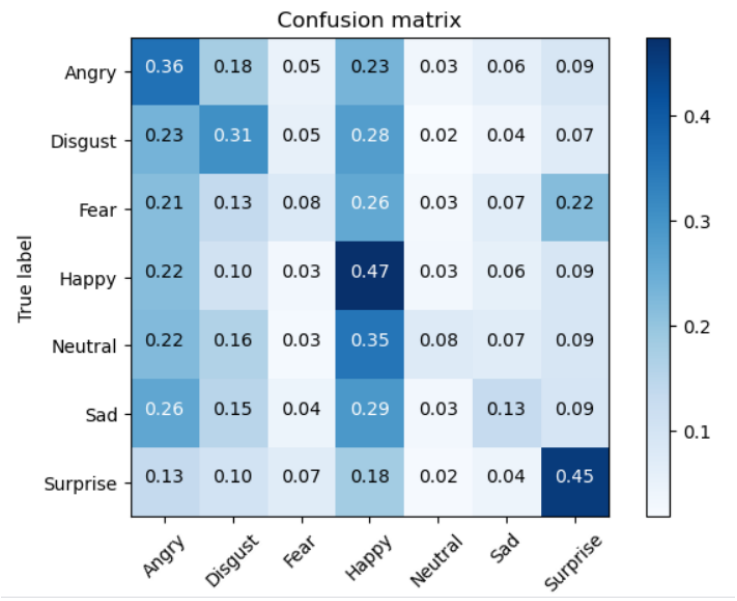


Figure 17: RESNET50+Dropout+Softmax Model Summary

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2, 2, 2048)	23587712
dropout_3 (Dropout)	(None, 2, 2, 2048)	0
flatten_3 (Flatten)	(None, 8192)	0
batch_normalization_6 (BatchNormalization)	(None, 8192)	32768
dense_8 (Dense)	(None, 32)	262176
batch_normalization_7 (BatchNormalization)	(None, 32)	128
activation_4 (Activation)	(None, 32)	0
dropout_4 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 32)	1056
batch_normalization_8 (BatchNormalization)	(None, 32)	128
activation_5 (Activation)	(None, 32)	0
dropout_5 (Dropout)	(None, 32)	0
dense_10 (Dense)	(None, 32)	1056
batch_normalization_9 (BatchNormalization)	(None, 32)	128
activation_6 (Activation)	(None, 32)	0
dense_11 (Dense)	(None, 7)	231

Total params: 23,885,383
Trainable params: 23,815,687
Non-trainable params: 69,696

Figure 18: RESNET50+Dropout+Softmax Categorical Accuracy and Model Loss

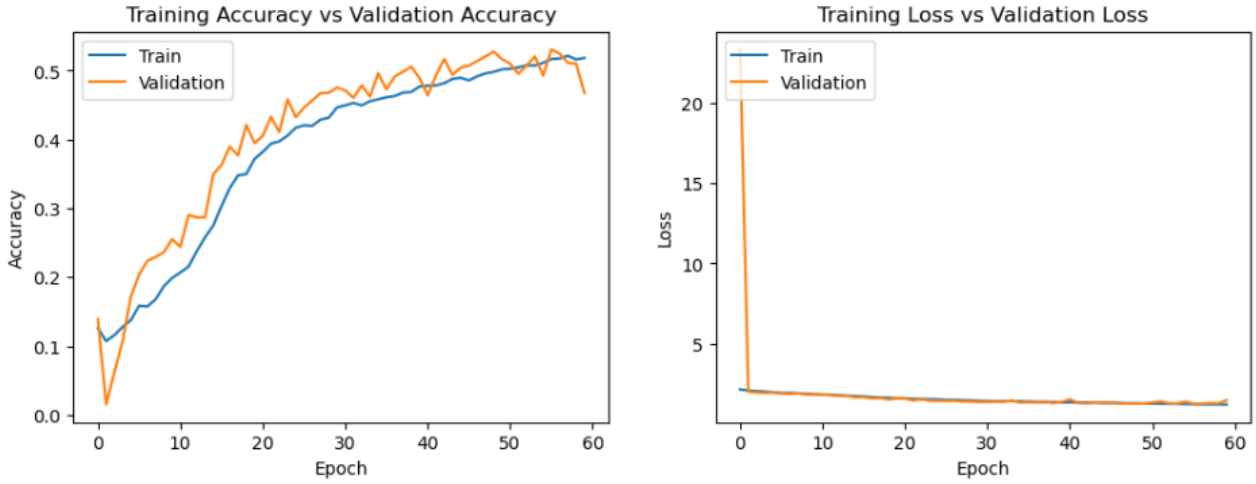


Figure 19: RESNET50+Dropout+Softmax Classification Report

	precision	recall	f1-score	support
Angry	0.26	0.47	0.33	958
Disgust	0.19	0.64	0.29	111
Fear	0.28	0.14	0.19	1024
Happy	0.97	0.44	0.61	1774
Neutral	0.49	0.30	0.37	1233
Sad	0.33	0.61	0.43	1247
Surprise	0.73	0.61	0.67	831
accuracy			0.43	7178
macro avg	0.46	0.46	0.41	7178
weighted avg	0.54	0.43	0.44	7178

Figure 20: RESNET50+Dropout+Softmax Confusion Matrix