REAL-TIME EMOTION MONITORING

Student Name:	Lam Chi Wai
Student ID No.:	22067588D
Programme-Stream Code:	61431-SYF
Supervisor:	Dr LOU Wei

TABLE OF CONTENTS

01

CURRENT PROBLEM

02

METHODOLOGY

03

EXPERIMENT

&

DEMOSTRATION

04

CONCLUSION

01

CURRENT PROBLEM

PROBLEM

LACK OF EMOTION RECOGNITION METHOD

Current emotion recognition methods mainly focus on static facial emotions. This method ignores the changing process of expressions.

RESTRICTED EMOTIONAL ANALYSIS

Different well-known datasets have the problem of uneven distribution of facial expressions . The system can only perform well on specific emotions.

02

METHODOLOGY

DATASET SELECTION & METHOD

Through the following attempts:

1. FER-2013 dataset: Imbalance dataset → only perform well on specific emotions (happy and surprise)

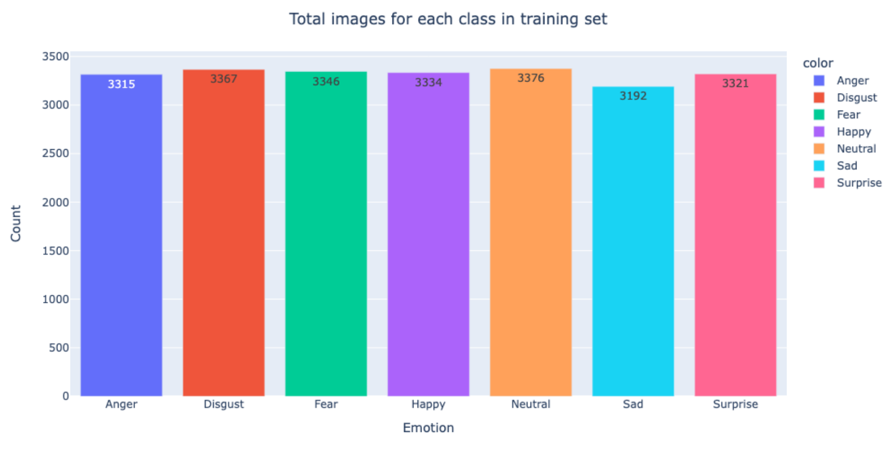
2. SMOTE: Newly generated images unreadable and could not pass through the system

3. ExpW dataset: Issues on inconsistent lighting conditions, bad photo quality and only western backgrounds → less reliable and having bias

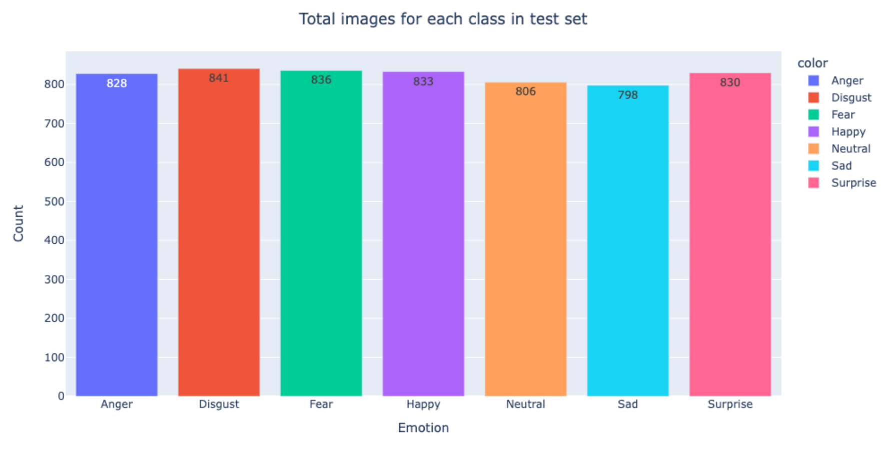
4. Clean and balance FER-2013 dataset: Solve the main problems of imbalance data and irrelevant images

DATA EXPLORATION

Training Data



Test Data

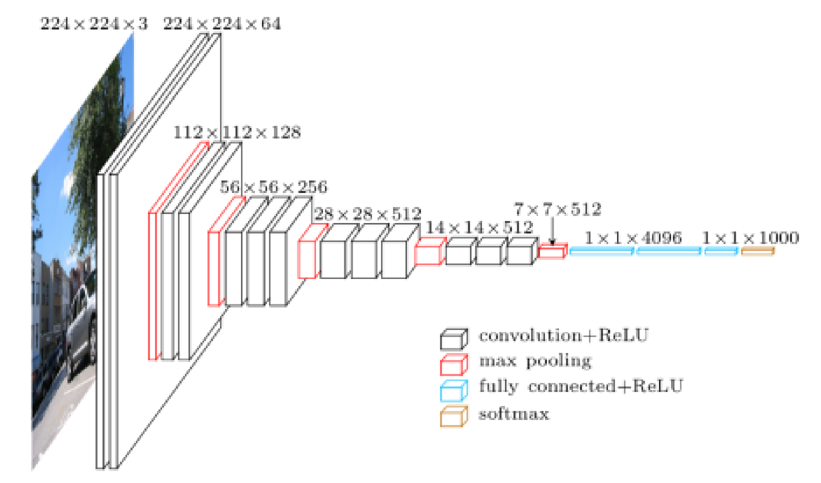


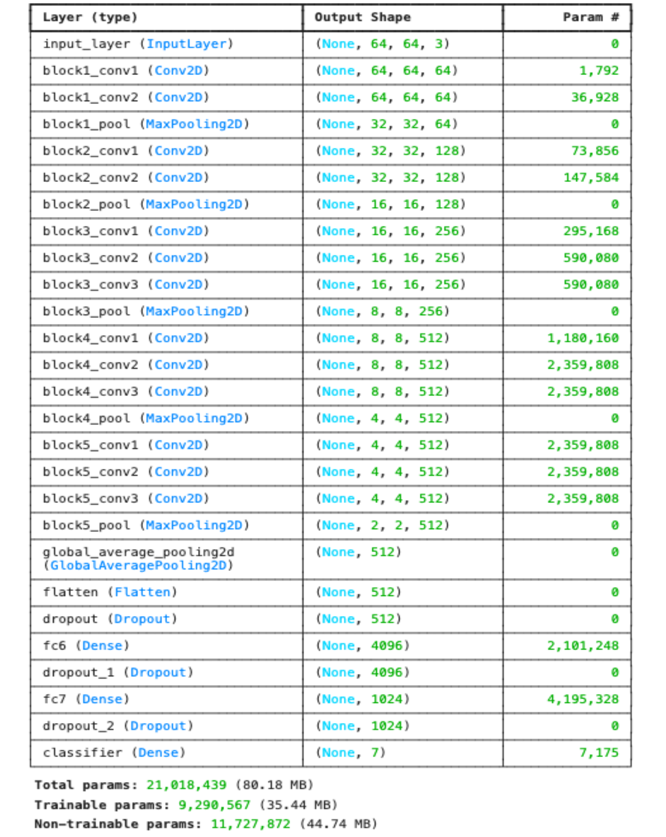
DATA GENERATOR & AUGMENTATION

These data generators efficiently load and preprocess the training, validation and test data in batches during model training. The generators automatically apply the specified data augmentation techniques to the training images, enabling the model to learn from diverse and augmented data samples.

MACHINE LEARNING MODEL

The system used the customized VGG16 model

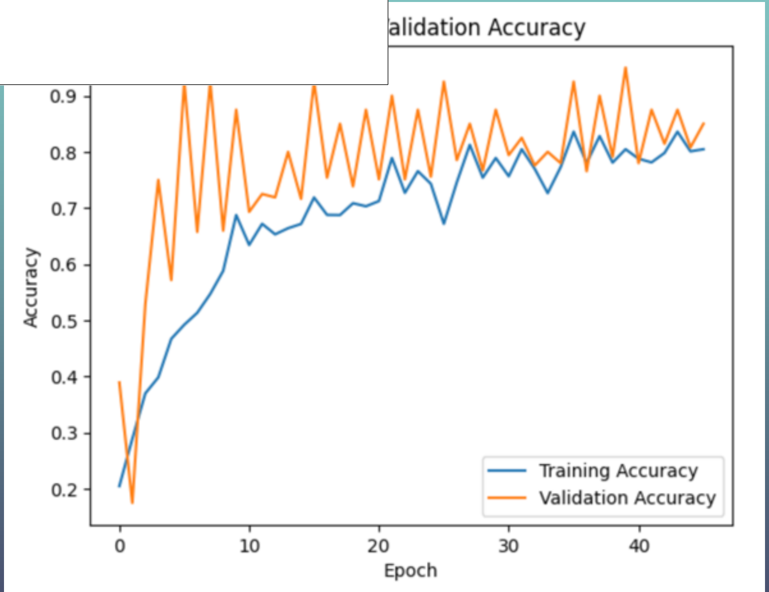


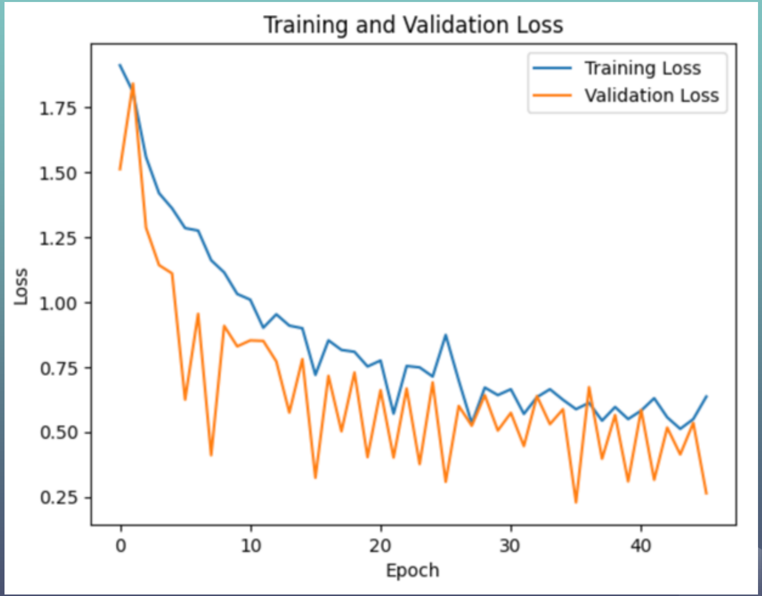


```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━┓
┃ Layer (type)                   ┃ Output Shape          ┃    Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━┩
│ input_layer (InputLayer)       │ (None, 64, 64, 3)     │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block1_conv1 (Conv2D)          │ (None, 64, 64, 64)    │      1,792 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block1_conv2 (Conv2D)          │ (None, 64, 64, 64)    │     36,928 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block1_pool (MaxPooling2D)     │ (None, 32, 32, 64)    │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block2_conv1 (Conv2D)          │ (None, 32, 32, 128)   │     73,856 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block2_conv2 (Conv2D)          │ (None, 32, 32, 128)   │    147,584 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block2_pool (MaxPooling2D)     │ (None, 16, 16, 128)   │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block3_conv1 (Conv2D)          │ (None, 16, 16, 256)   │    295,168 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block3_conv2 (Conv2D)          │ (None, 16, 16, 256)   │    590,080 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block3_conv3 (Conv2D)          │ (None, 16, 16, 256)   │    590,080 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block3_pool (MaxPooling2D)     │ (None, 8, 8, 256)     │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block4_conv1 (Conv2D)          │ (None, 8, 8, 512)     │  1,180,160 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block4_conv2 (Conv2D)          │ (None, 8, 8, 512)     │  2,359,808 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block4_conv3 (Conv2D)          │ (None, 8, 8, 512)     │  2,359,808 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block4_pool (MaxPooling2D)     │ (None, 4, 4, 512)     │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block5_conv1 (Conv2D)          │ (None, 4, 4, 512)     │  2,359,808 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block5_conv2 (Conv2D)          │ (None, 4, 4, 512)     │  2,359,808 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block5_conv3 (Conv2D)          │ (None, 4, 4, 512)     │  2,359,808 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ block5_pool (MaxPooling2D)     │ (None, 2, 2, 512)     │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ global_average_pooling2d       │ (None, 512)           │          0 │
│ (GlobalAveragePooling2D)       │                       │            │
├────────────────────────────────┼───────────────────────┼────────────┤
│ flatten (Flatten)              │ (None, 512)           │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ dropout (Dropout)              │ (None, 512)           │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ fc6 (Dense)                    │ (None, 4096)          │  2,101,248 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ dropout_1 (Dropout)            │ (None, 4096)          │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ fc7 (Dense)                    │ (None, 1024)          │  4,195,328 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ dropout_2 (Dropout)            │ (None, 1024)          │          0 │
├────────────────────────────────┼───────────────────────┼────────────┤
│ classifier (Dense)             │ (None, 7)             │      7,175 │
└────────────────────────────────┴───────────────────────┴────────────┘
 Total params: 21,018,439 (80.18 MB)
 Trainable params: 9,290,567 (35.44 MB)
 Non-trainable params: 11,727,872 (44.74 MB)
```

The picture can't be displayed.

MODEL EVALUATION

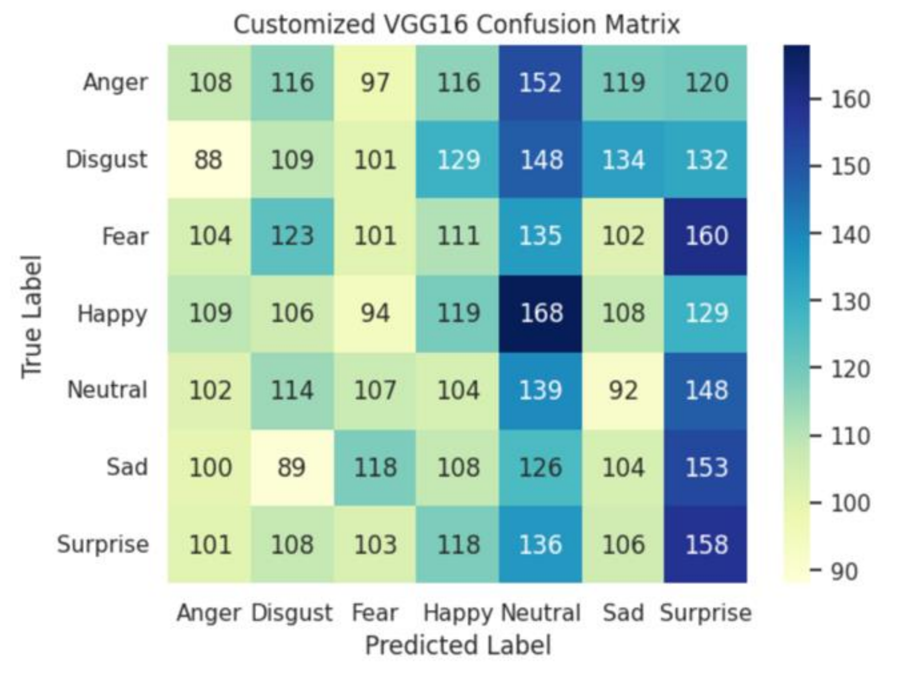




MODEL EVALUATION

Train Accuracy	Validation Accuracy	Test Accuracy
80.59%	79.13%	78.29%

MODEL EVALUATION

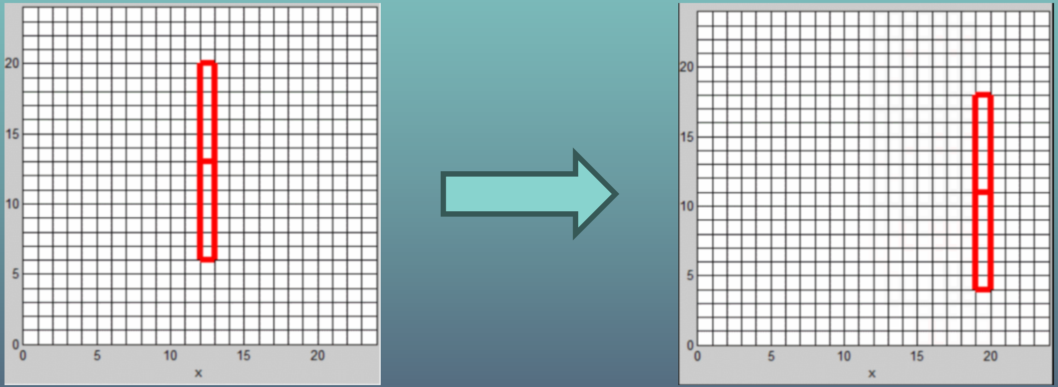


Customized VGG16 Confusion Matrix

True Label \ Predicted Label	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	108	116	97	116	152	119	120
Disgust	88	109	101	129	148	134	132
Fear	104	123	101	111	135	102	160
Happy	109	106	94	119	168	108	129
Neutral	102	114	107	104	139	92	148
Sad	100	89	118	108	126	104	153
Surprise	101	108	103	118	136	106	158

FACE DETECTION ALGORITHM

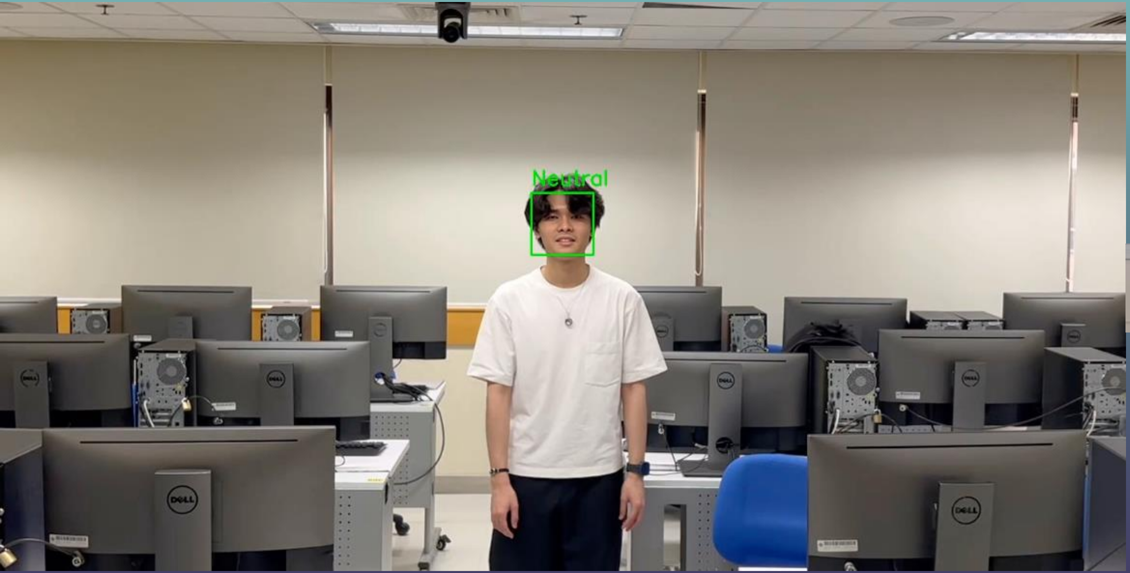
1. OpenCV Haar Cascade Classifier

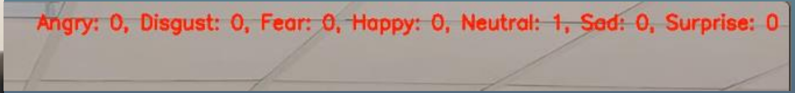


2. Dlib library with pre-trained model (mmod_human_face_detector)

EMOTION COUNTER

As this is a classroom analysis system, it is essential to have an emotion counter to tell the teachers how many students feel about that specific emotion.



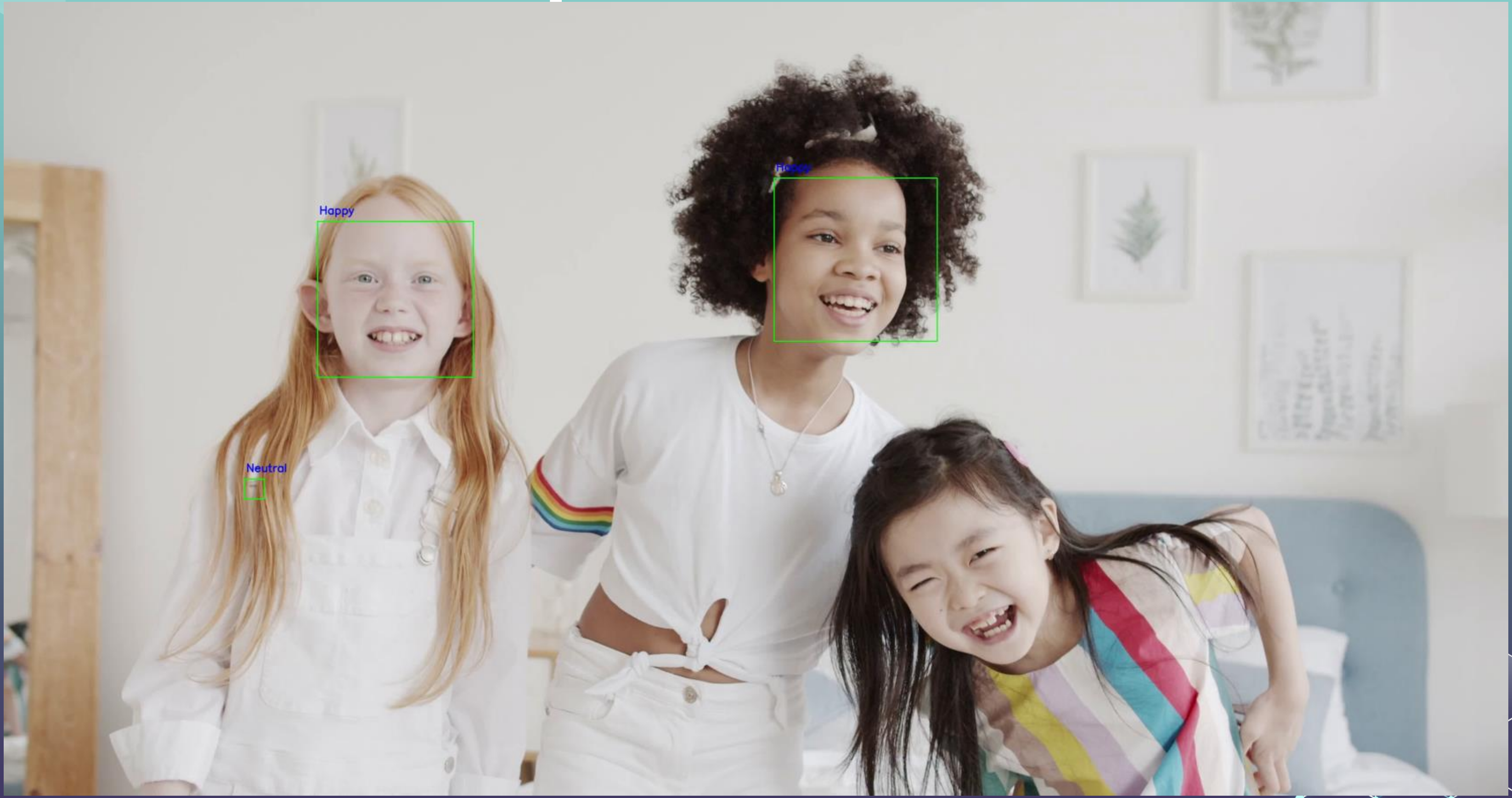


```
Angry: 0, Disgust: 0, Fear: 0, Happy: 0, Neutral: 1, Sad: 0, Surprise: 0
```

03

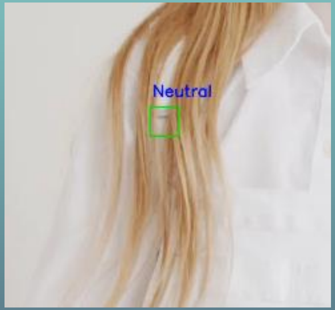
EXPERIMENT & DEMOSTRATION

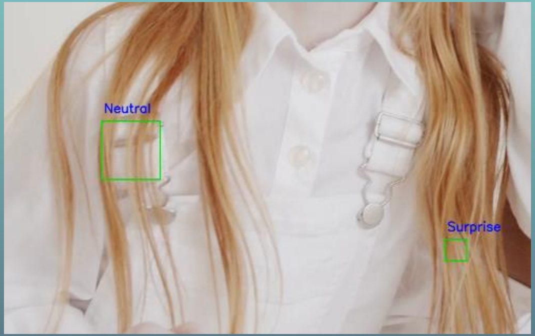
OpenCV Haar Cascade Classifier

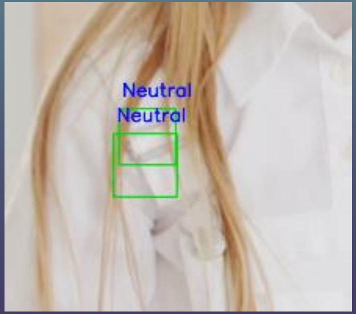


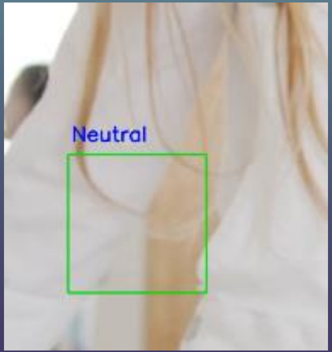
OpenCV Haar Cascade Classifier

Also, the algorithm detected not only the human faces, but also other objects.









PARAMETER OF Haar Cascade Classifier

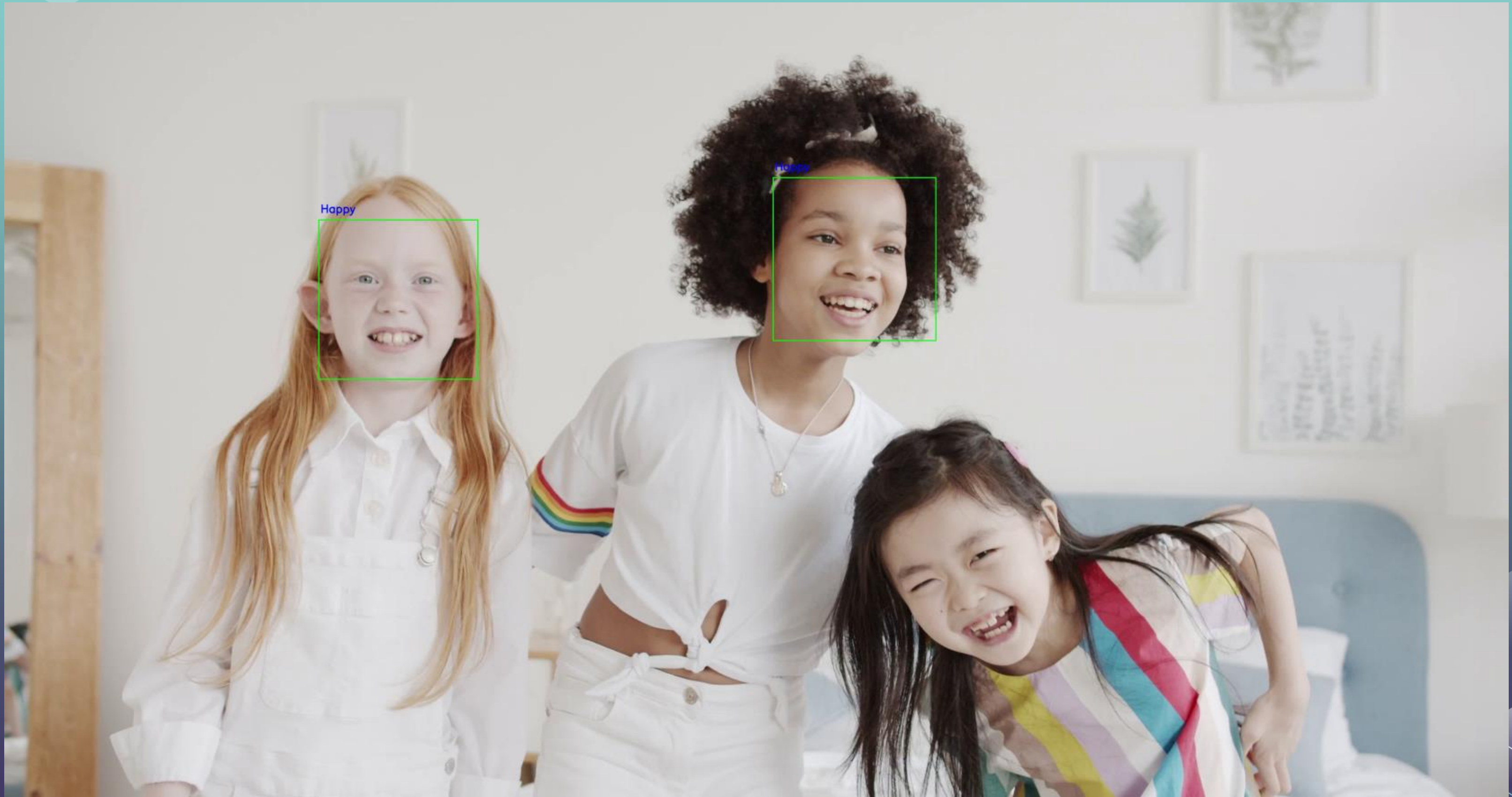
Through using the function of “detectMultiScal” in OpenCV library and set the follow parameters:

“”scaleFactor = 1.29,minNeighbors = 6,minSize = (50,50)””

It solved the following problem

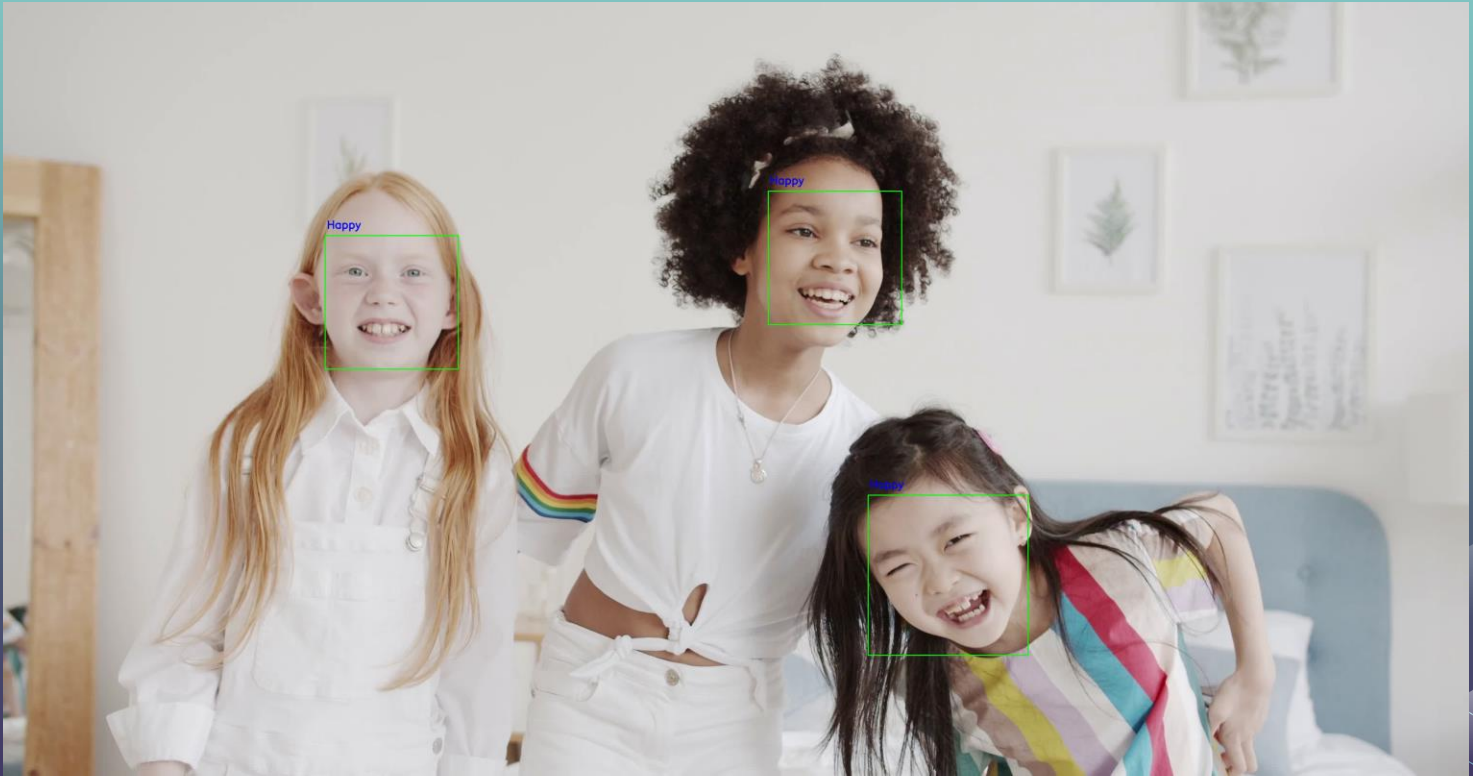
Problem: Detected objects that are not related to human face.

Optimized Haar Cascade Classifier



Dlib library

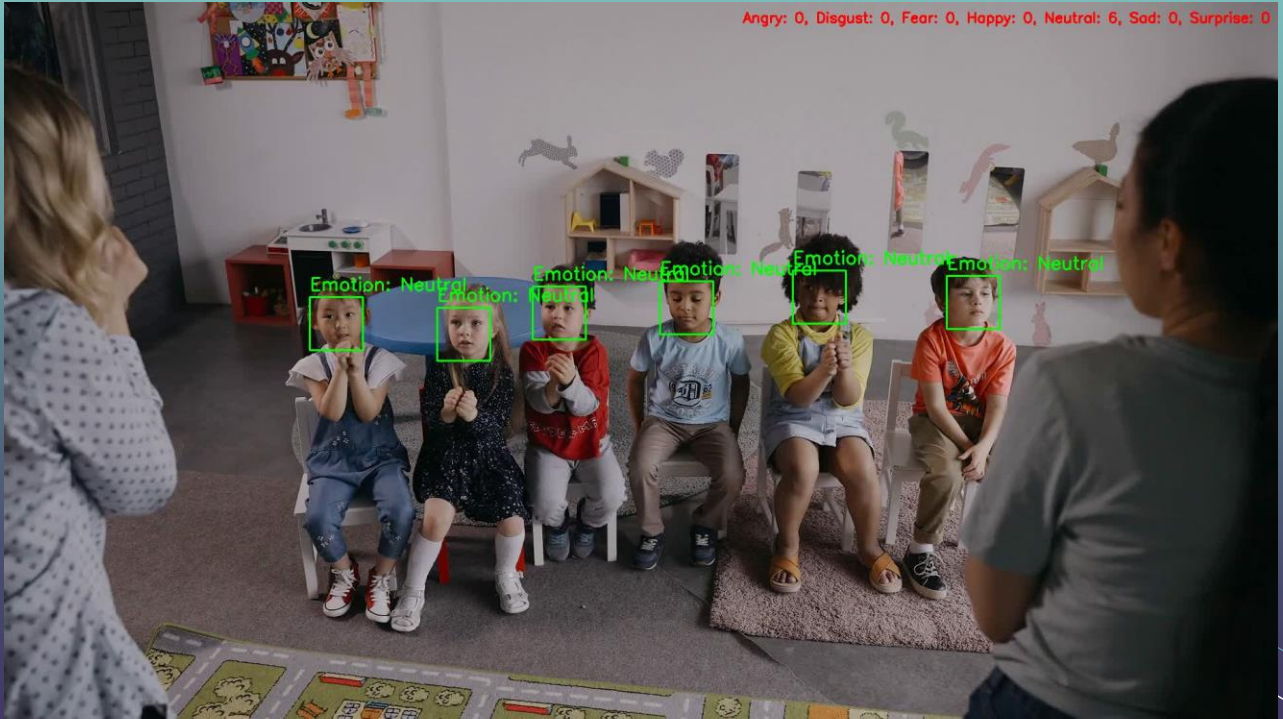
By using the function in Dlib library called ” get_frontal_face_detector” and here is the result.



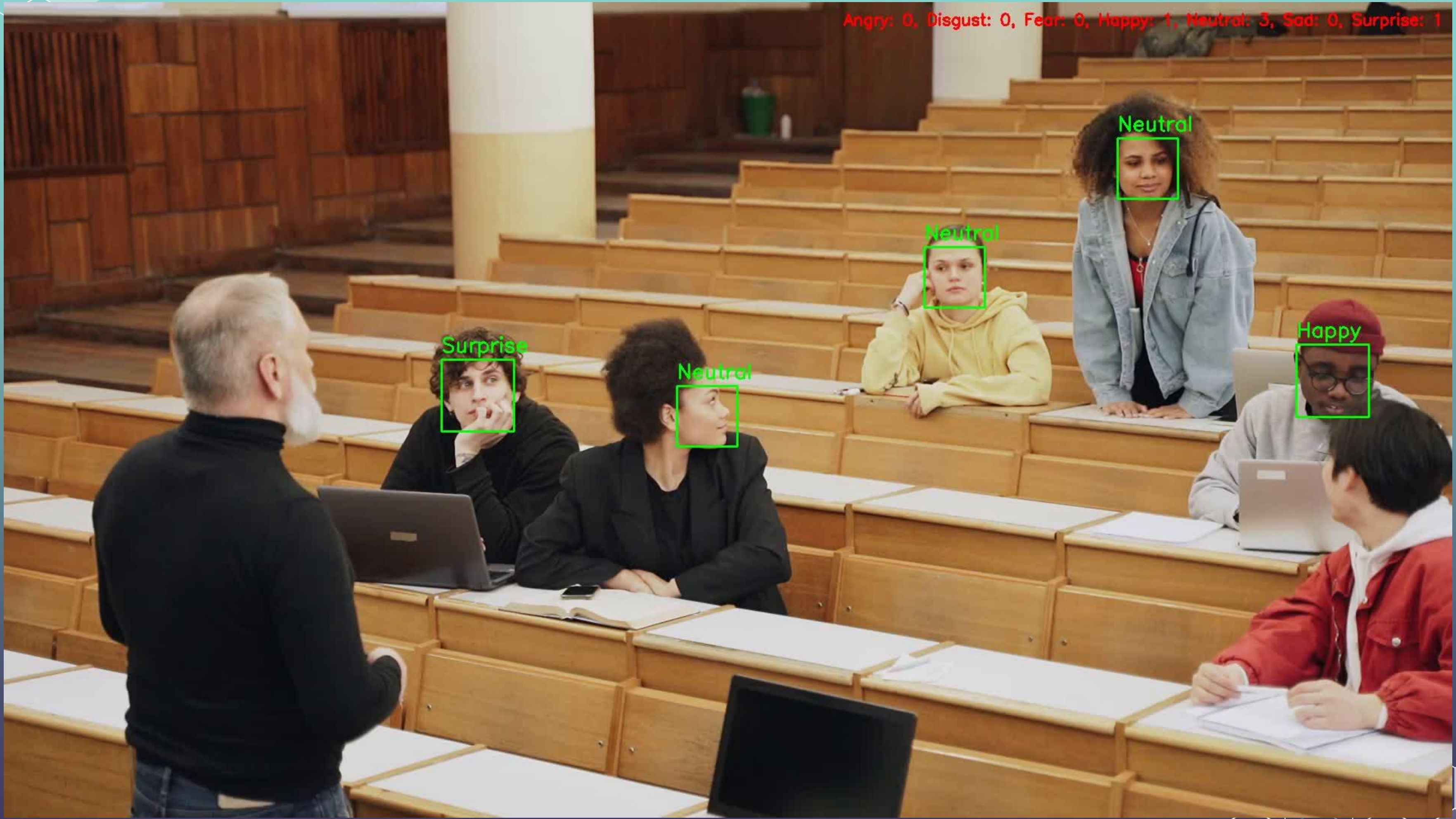
Dlib Pre-Trained Model

Let’s try to perform the task of detecting the small faces.

By combining the frontal face detection with the pre-trained model “mmod_human_face_detector” in the Dlib library



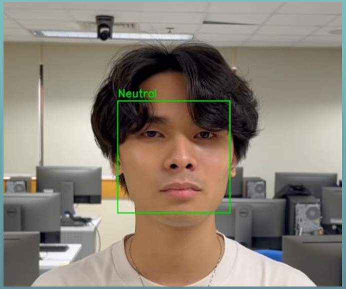
```
Angry: 0, Disgust: 0, Fear: 0, Happy: 0, Neutral: 6, Sad: 0, Surprise: 0
```



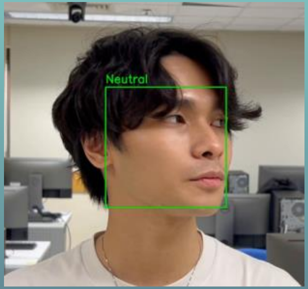
```
Angry: 0, Disgust: 0, Fear: 0, Happy: 1, Neutral: 3, Sad: 0, Surprise: 1
```

Face Angle Test

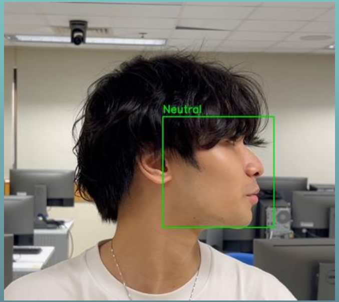
Right



0 degree

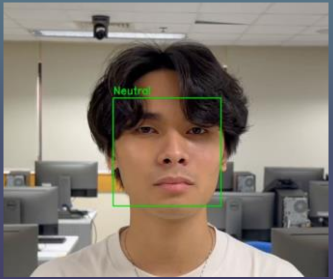


45 degree

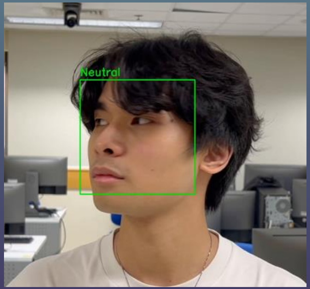


90 degree

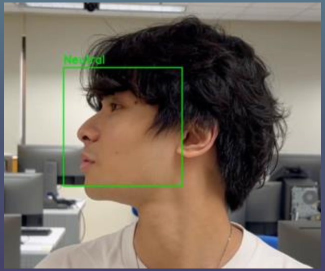
Left



0 degree

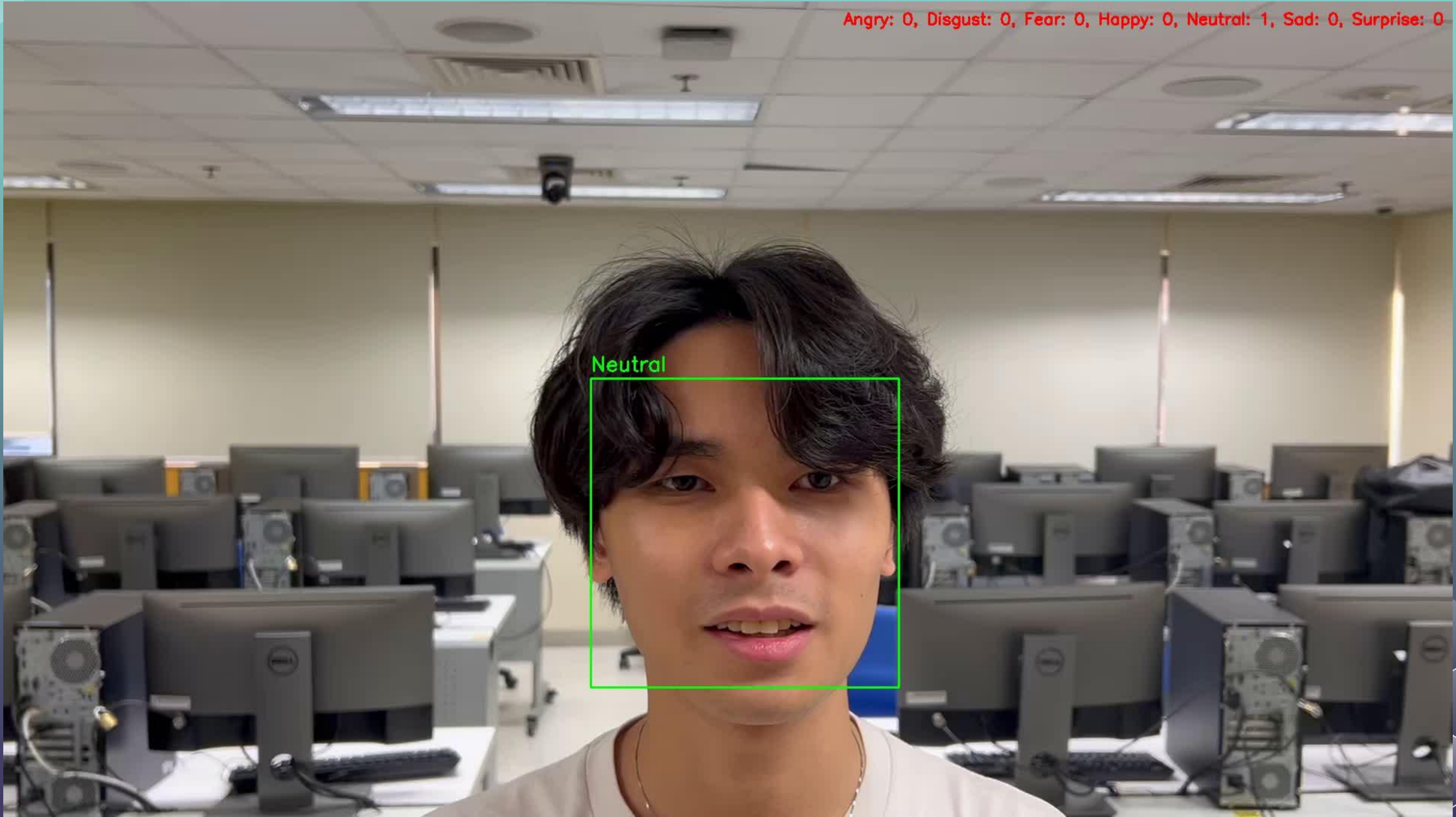


45 degree



90 degree

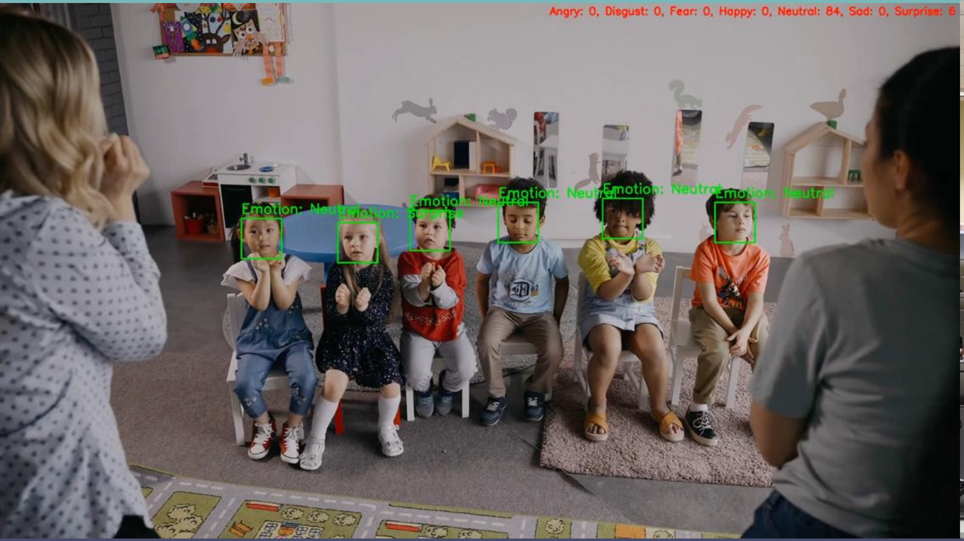
Distance Test



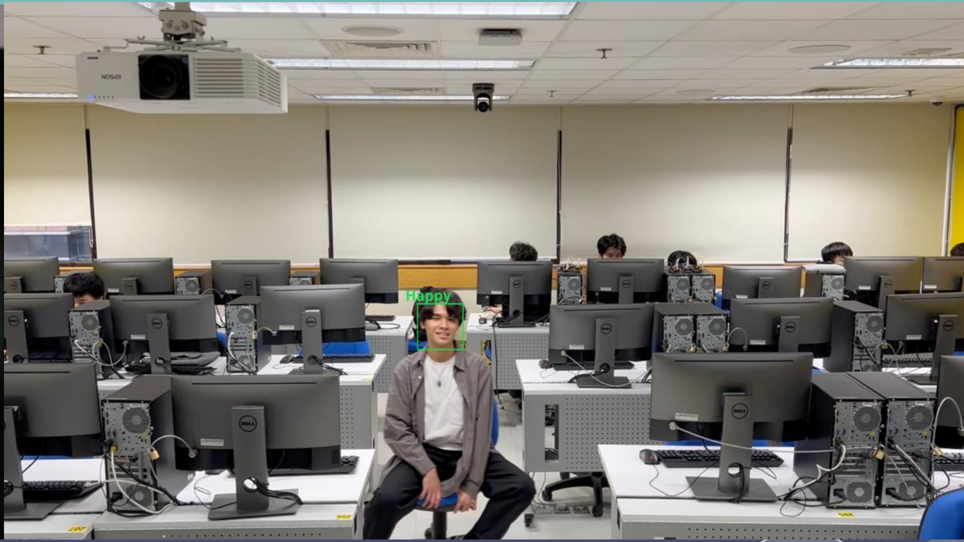
```
Angry: 0, Disgust: 0, Fear: 0, Happy: 0, Neutral: 1, Sad: 0, Surprise: 0
```

High-Angle Shot Test

I have tried to test the system with the camera that is placed higher than the normal field of view, which shooting from top to bottom. Here is the results:



```
Angry: 0, Disgust: 0, Fear: 0, Happy: 0, Neutral: 84, Sad: 0, Surprise: 6
```



03

CONCLUSION

Conclusion

Improved performance: By comparing different face detection algorithms, we observed that using the Dlib library's pre-trained model outperformed OpenCV's Haar Cascade algorithm for face detection. To ensure accurate facial expression recognition in small classrooms, I combined the Dlib model with my customized VGG16 model.

Optimized system: Through various tests including angle, high-angle shot, and distance tests, I fine-tuned the system's parameters to optimize face detection and emotion recognition. The system demonstrated excellent performance in these tests.

Implementation in small classrooms: This facial expression recognition system is suitable for small classrooms, enabling teachers to adjust their teaching methods and enhance the overall teaching experience.

THANK YOU