

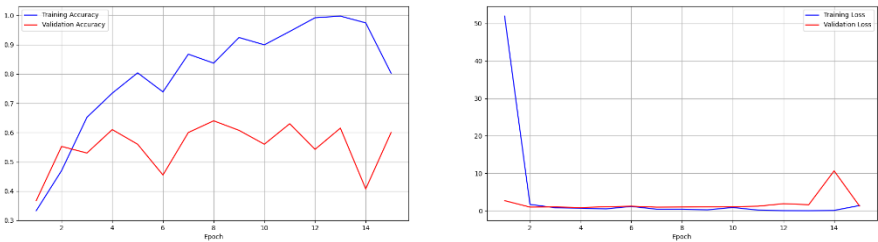
Team members:

Zexuan Li
Kat Pe Benito

basic_model:

Initial Network:		
Layer (type)	Output Shape	Param #
=====		
=====		
conv2d (Conv2D)	(None, 148, 148, 8)	224
max_pooling2d (MaxPooling2D)	(None, 74, 74, 8)	0
conv2d_1 (Conv2D)	(None, 72, 72, 16)	1168
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 16)	0
conv2d_2 (Conv2D)	(None, 34, 34, 32)	4640
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 64)	2367552
dense_1 (Dense)	(None, 3)	195
=====		
=====		

Graph:



Best learned model:

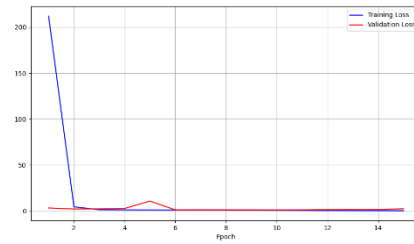
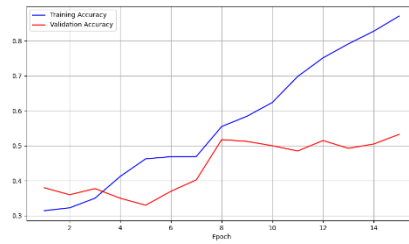
* Evaluating basic_model

30/30 [=====] - 3s 83ms/step - loss: 1.3211 - accuracy: 0.6042

dropout_model**With Dropout:**

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 8)	224
max_pooling2d (MaxPooling2D)	(None, 74, 74, 8)	0
dropout (Dropout)	(None, 74, 74, 8)	0
conv2d_1 (Conv2D)	(None, 72, 72, 16)	1168
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 16)	0
conv2d_2 (Conv2D)	(None, 34, 34, 32)	4640
flatten (Flatten)	(None, 36992)	0
dropout_1 (Dropout)	(None, 36992)	0
dense (Dense)	(None, 64)	2367552
dense_1 (Dense)	(None, 3)	195

Graph:



Best learned model:

* Evaluating dropout_model

30/30 [=====] - 3s 85ms/step - loss: 2.3392 - accuracy: 0.5482

merged_model:

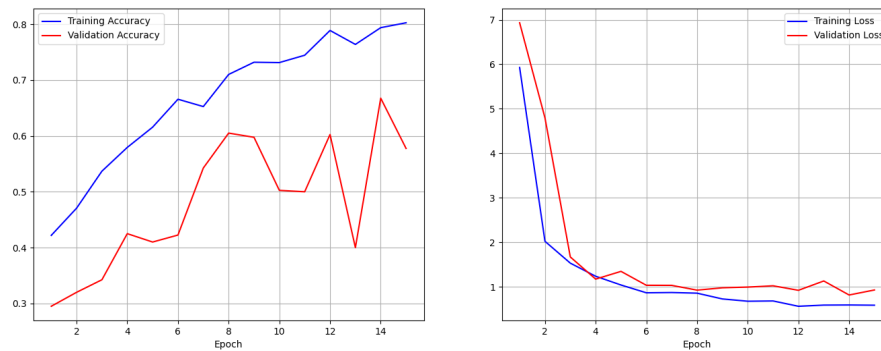
Layer (type)	Output Shape	Param #
=====		
=====		
conv2d (Conv2D)	(None, 148, 148, 8)	224
batch_normalization (Batch Normalization)	(None, 148, 148, 8)	32
max_pooling2d (MaxPooling2D)	(None, 74, 74, 8)	0
batch_normalization_1 (Batch Normalization)	(None, 74, 74, 8)	32
conv2d_1 (Conv2D)	(None, 72, 72, 16)	1168
batch_normalization_2 (Batch Normalization)	(None, 72, 72, 16)	64
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 16)	0
batch_normalization_3 (Batch Normalization)	(None, 36, 36, 16)	64
conv2d_2 (Conv2D)	(None, 34, 34, 32)	4640
batch_normalization_4 (Batch Normalization)	(None, 34, 34, 32)	128

flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 64)	2367552
dense_1 (Dense)	(None, 3)	195

=====

=====

Graph:



Best learned model:

* Evaluating merged_model

30/30 [=====] - 4s 116ms/step - loss: 0.8404 - accuracy: 0.6410

Play Game:

A trace of moves in that game:

```

| | | |
| | | |
| | | |

```

Turn: X

Player X took position (2, 2).

```

| | | |
| | | |
| | X |

```

Turn: O

reference:

row 0 is neutral.

row 1 is happy.

row 2 is surprise.

1/1 [=====] - 0s 86ms/step

[[0.00130237 0.98115206 0.01754556]] 1

Emotion detected as happy (row 1). Enter 'text' to use text input instead (0, 1 or 2).
Otherwise, press Enter to continue.

reference:

col 0 is neutral.

col 1 is happy.

col 2 is surprise.

1/1 [=====] - 0s 52ms/step

[[0.00144892 0.9945017 0.00404928]] 1

Emotion detected as happy (col 1). Enter 'text' to use text input instead (0, 1 or 2).
Otherwise, press Enter to continue.

Player O took position (1, 1).

```
|||  
||O|  
||X|
```

Turn: X

Player X took position (0, 0).

```
|X||  
||O|  
||X|
```

Turn: O

reference:

row 0 is neutral.

row 1 is happy.

row 2 is surprise.

1/1 [=====] - 0s 56ms/step

[[0.06910342 0.22442767 0.70646894]] 2

Emotion detected as surprise (row 2). Enter 'text' to use text input instead (0, 1 or 2).
Otherwise, press Enter to continue.

reference:

col 0 is neutral.

col 1 is happy.

col 2 is surprise.

1/1 [=====] - 0s 53ms/step

[[0.00858489 0.43127164 0.5601435]] 2

Emotion detected as surprise (col 2). Enter 'text' to use text input instead (0, 1 or 2).
Otherwise, press Enter to continue.

Position (2, 2) is already taken.

```
|X||
```

||O||

||X||

Turn: O

reference:

row 0 is neutral.

row 1 is happy.

row 2 is surprise.

WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x00000217A1B33910> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

1/1 [=====] - 0s 53ms/step

[[0.00462706 0.1362137 0.85915923]] 2

Emotion detected as surprise (row 2). Enter 'text' to use text input instead (0, 1 or 2). Otherwise, press Enter to continue.

reference:

col 0 is neutral.

col 1 is happy.

col 2 is surprise.

WARNING:tensorflow:6 out of the last 6 calls to <function Model.make_predict_function.<locals>.predict_function at 0x00000217A1B31AB0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

1/1 [=====] - 0s 54ms/step

[[0.01701618 0.07870506 0.90427876]] 2

Emotion detected as surprise (col 2). Enter 'text' to use text input instead (0, 1 or 2). Otherwise, press Enter to continue.

Position (2, 2) is already taken.

|X||

||O||

||X||

Turn: O

reference:

row 0 is neutral.

row 1 is happy.

row 2 is surprise.

1/1 [=====] - 0s 51ms/step

[[0.00149668 0.3401614 0.6583419]] 2

Emotion detected as surprise (row 2). Enter 'text' to use text input instead (0, 1 or 2).

Otherwise, press Enter to continue.

reference:

col 0 is neutral.

col 1 is happy.

col 2 is surprise.

1/1 [=====] - 0s 54ms/step

[[0.43038443 0.40881157 0.16080406]] 0

Emotion detected as neutral (col 0). Enter 'text' to use text input instead (0, 1 or 2).

Otherwise, press Enter to continue.

Player O took position (2, 0).

|X| | |

| |O| |

|O| |X|

Turn: X

Player X took position (2, 1).

|X| | |

| |O| |

|O|X|X|

Turn: O

reference:

row 0 is neutral.

row 1 is happy.

row 2 is surprise.

1/1 [=====] - 0s 52ms/step

[[0.5694582 0.17289548 0.25764626]] 0

Emotion detected as neutral (row 0). Enter 'text' to use text input instead (0, 1 or 2).

Otherwise, press Enter to continue.

reference:

col 0 is neutral.

col 1 is happy.

col 2 is surprise.

1/1 [=====] - 0s 55ms/step

[[0.42241728 0.08260878 0.4949739]] 2

Emotion detected as surprise (col 2). Enter 'text' to use text input instead (0, 1 or 2).

Otherwise, press Enter to continue.

Player O took position (0, 2).

```
|X| |O|  
| |O| |  
|O|X|X|
```

Player O has won!

- Answers to these questions:
 - How well did your interface work?

Can recognize facial expressions, but the accuracy needs to be improved.

- Did it recognize your facial expressions with the same accuracy as it achieved against the test set?

The probability of recognition is lower than the probability tested in training, especially for happy and surprise, my model is not very able to distinguish the difference between these two expressions, it always thinks surprise.

- If not, why not?

I think one of the reasons is that most of the pictures used for training are pure expressions, but there is a background behind me when I play the game, and the background may interfere with the judgment of the model.

Another reason I'm guessing is because I wear glasses and most of the data I see for training is facial expressions without glasses. Glasses may have affected the model's judgment of expressions.