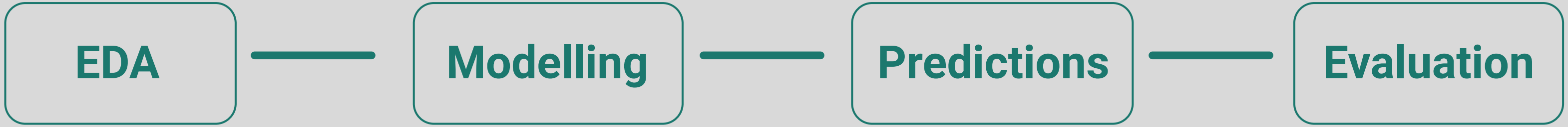
PROJECT CAPSTONE
Facial Expression Recognition

Lynette Ng, DSI20

problem statement

- In an increasingly digitalized world, there has been a decreasing focus on soft skills
- Can be an issue in the aspect of Customer Service, in this case in Cafes
- What if we can use facial expression recognition to classify emotions and use this as a measure of Customer Satisfaction?

methodology



EDA

Modelling

Predictions

Evaluation

exploratory
data
analysis

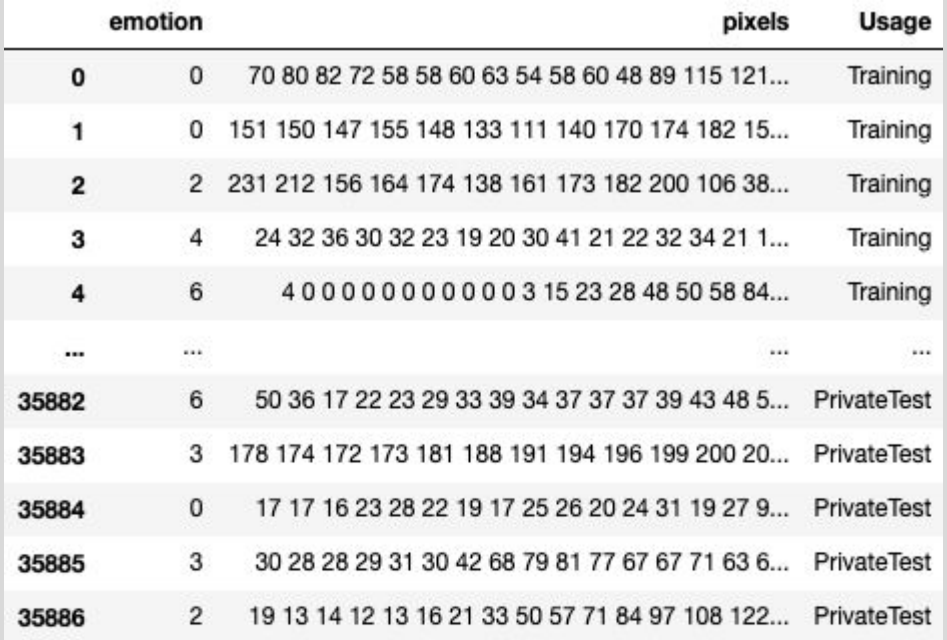
exploratory data analysis - dataset

- 48x48 pixels, b/w
- Pixels were already in csv string format

exploratory data analysis - dataset FER2013

exploratory data analysis - dataset

- 35,000 images,
- 7 classes
- Labelled
- Pre-split into train & test
- Dropped usage column did my own train, test & split

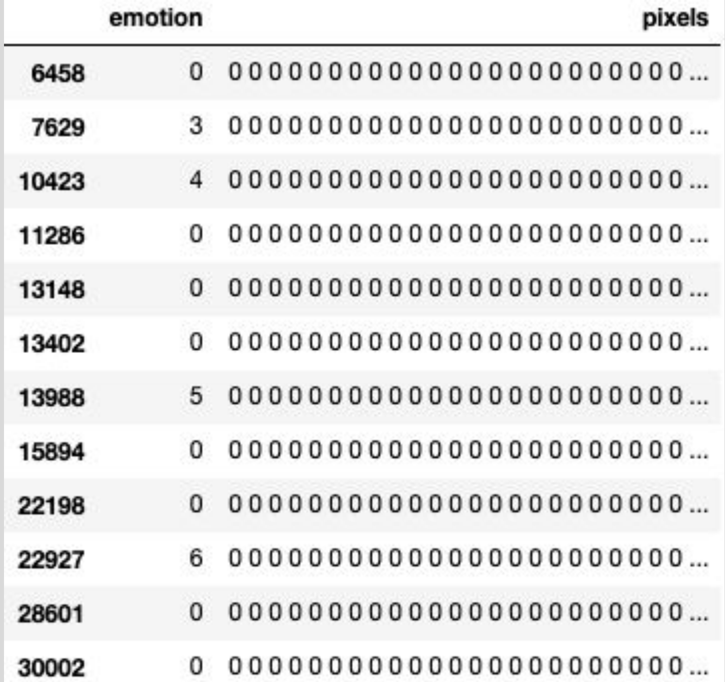


	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training
...	...	...	...
35882	6	50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5...	PrivateTest
35883	3	178 174 172 173 181 188 191 194 196 199 200 20...	PrivateTest
35884	0	17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9...	PrivateTest
35885	3	30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6...	PrivateTest
35886	2	19 13 14 12 13 16 21 33 50 57 71 84 97 108 122...	PrivateTest

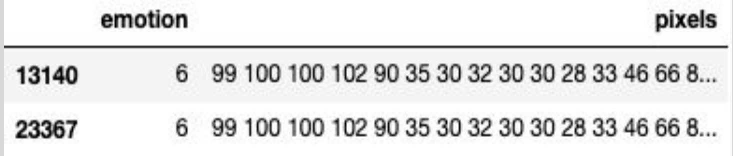
exploratory data analysis - dataset

- Checked that the ‘0’ pixels were entirely zeros and removed duplicates

- Checked non-zero rows as well, to make sure classified label was the same



	emotion	pixels
6458	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
7629	3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
10423	4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
11286	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
13148	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
13402	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
13988	5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
15894	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
22198	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
22927	6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
28601	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
30002	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...

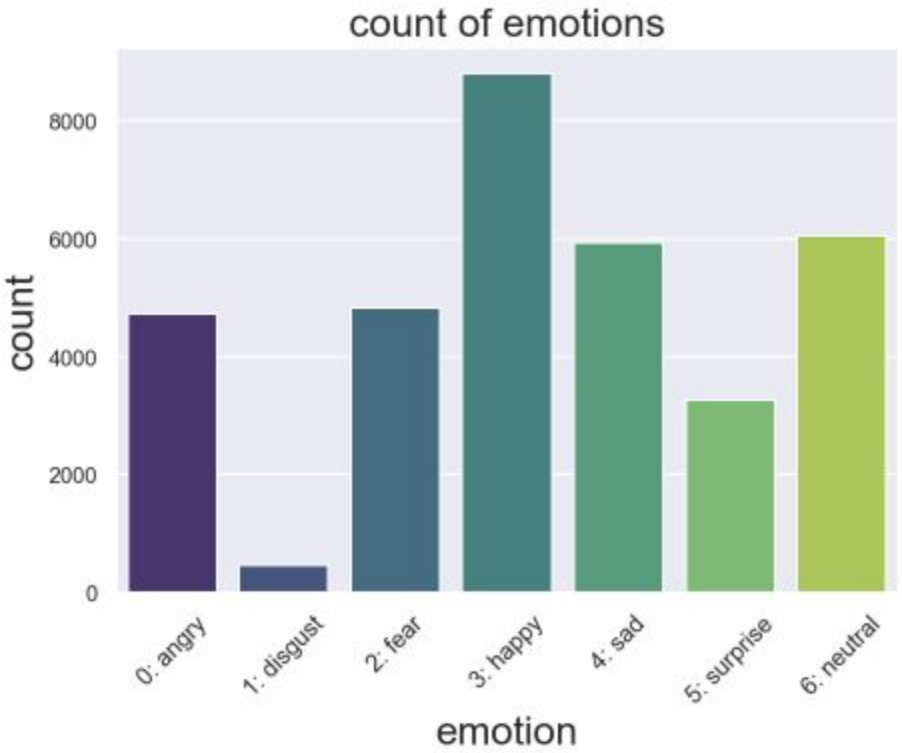


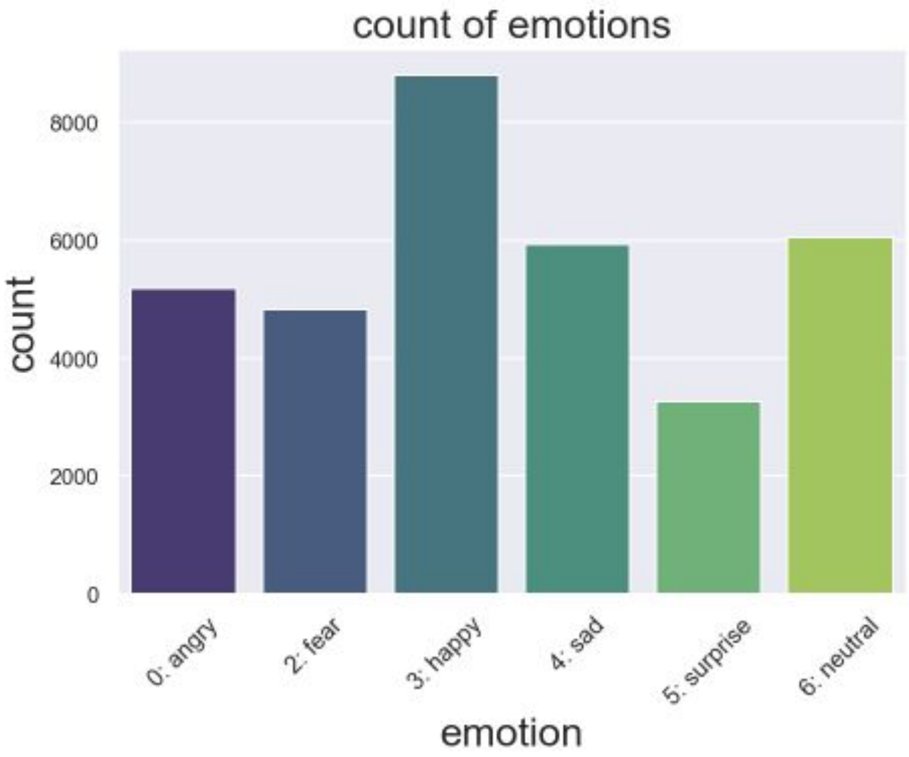
	emotion	pixels
13140	6	99 100 100 102 90 35 30 32 30 30 28 33 46 66 8...
23367	6	99 100 100 102 90 35 30 32 30 30 28 33 46 66 8...

exploratory data analysis

exploratory data analysis

before >>> after





modelling

modelling

1. Custom CNN
2. VGGFace - Base
3. VGGFace - Fine-tuned
4. EfficientNetB0 - Base
5. EfficientNetB0 - Fine-tuned

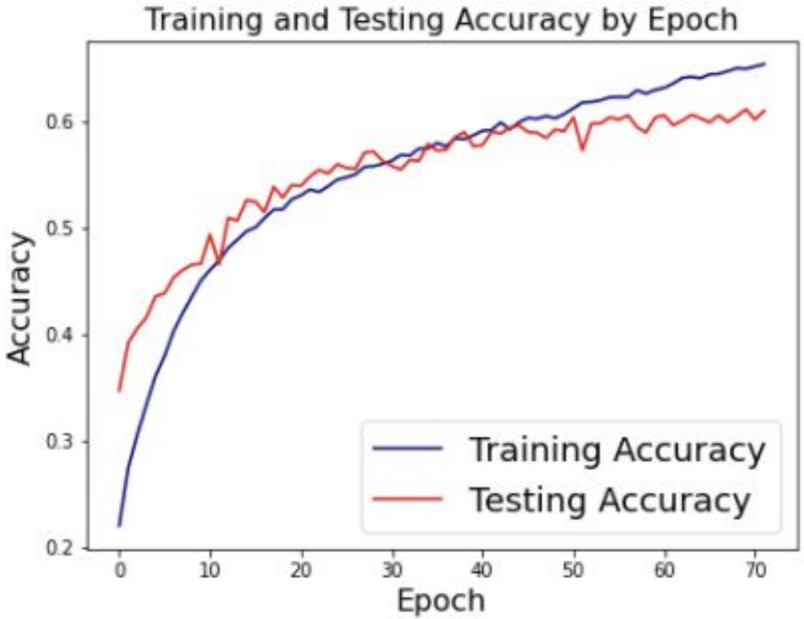
modelling - custom CNN

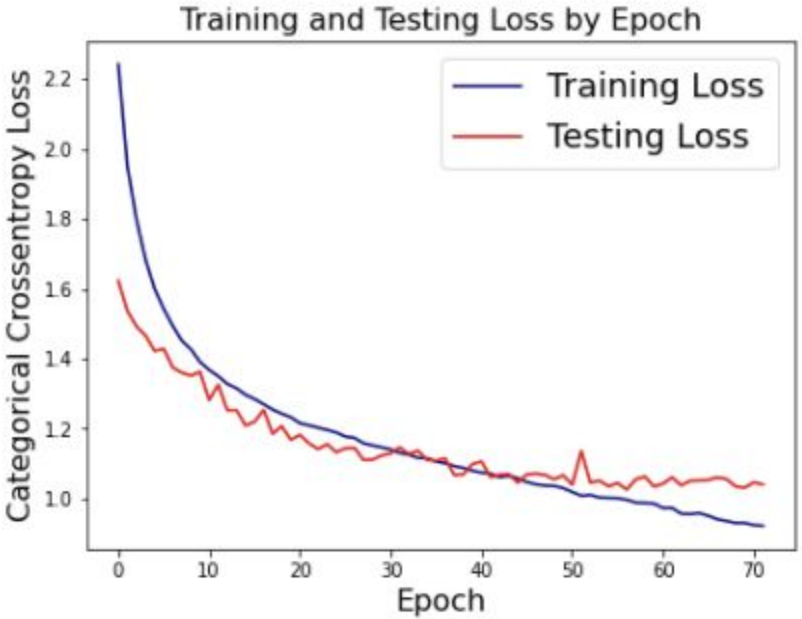
- Augmentation
- 3 Conv + 2 Dense + 1 Output
- ~ 2mil params
- BatchNormalization at every layer
- MaxPooling 2D at every conv layer
- Dropout on later conv layers & all dense layers
- LR = 0.0001

modelling - custom CNN

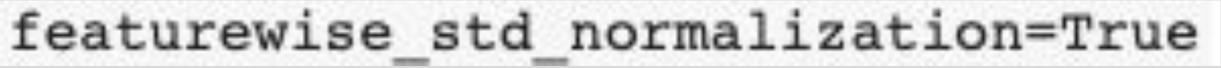
Validation accuracy - 60.65

Validation loss - 1.03





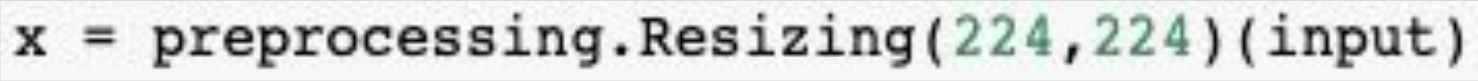
modelling - VGG & EffNetB0 - preprocessing

- Converted to RGB format
- Image Augmentation:
    - No scaling
    - 

      ```
      featurewise_std_normalization=True
      ```

modelling - VGG & Effnet Base

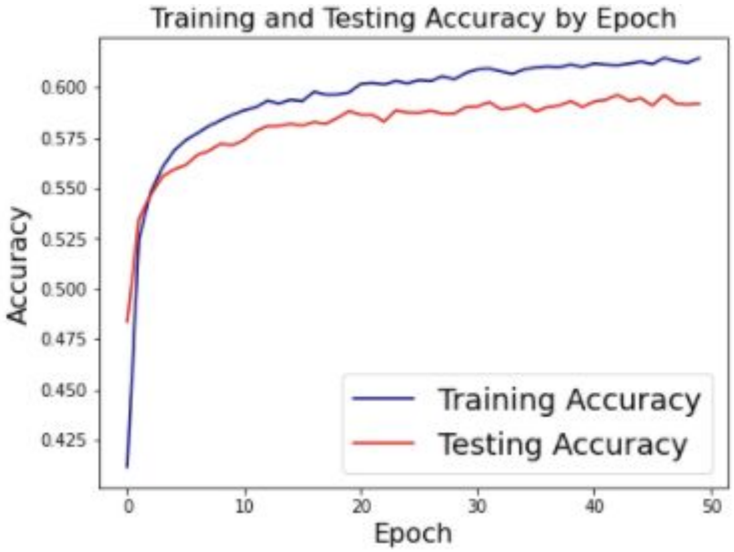
- Froze all layers
- Replaced output layer with number of classes
- Input requirements different



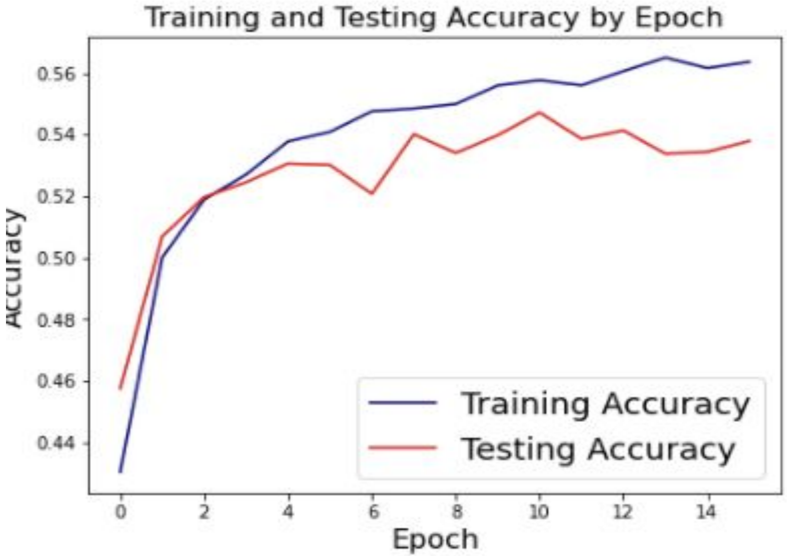
```
x = preprocessing.Resizing(224,224)(input)
```

modelling - VGG & Effnet base

VGG Base
val acc: 58.88
val loss: 1.08



EfficientNetB0
val acc: 54.52
val loss: 1.20



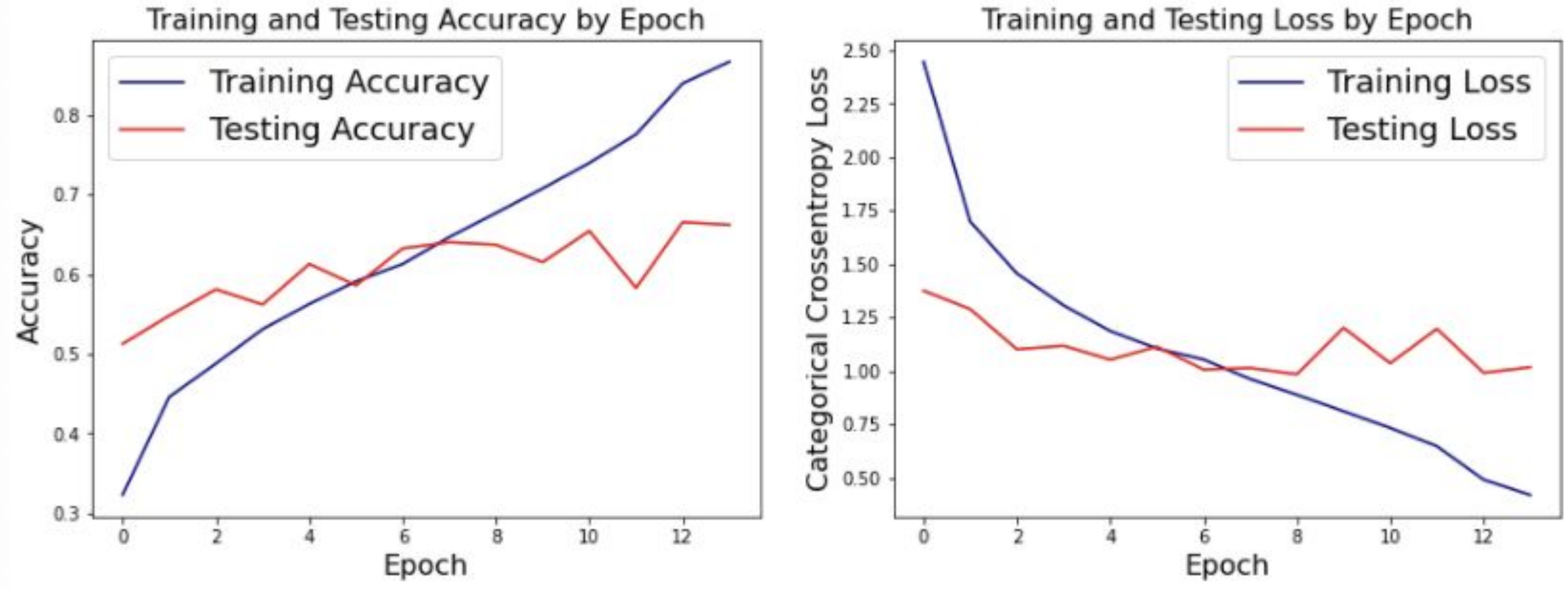
modelling - VGG fine tuned

- Froze all layers before the 5th conv layer
- Added 2 dense layers
- Dropout Rate 0.7
- ~7m trainable params
- LR 0.0001

modelling - VGG fine tuned

Validation accuracy - 63.81

Validation loss - 0.98



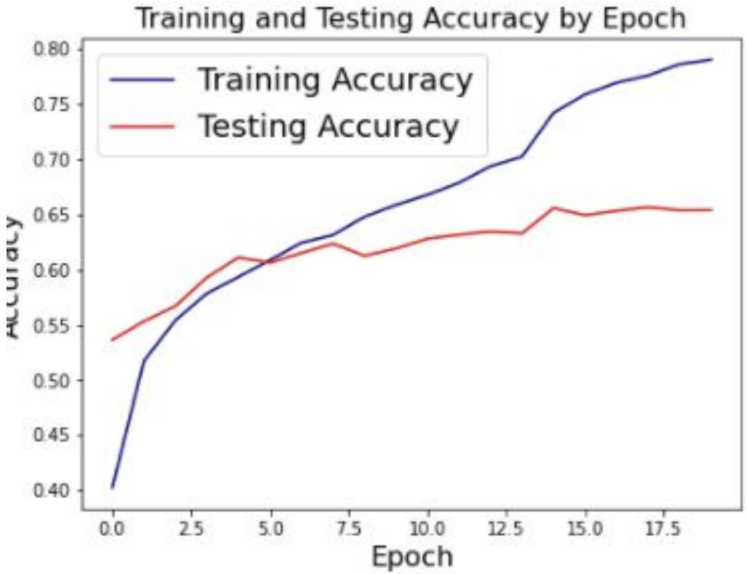
modelling - EfficientNetB0 fine tuned

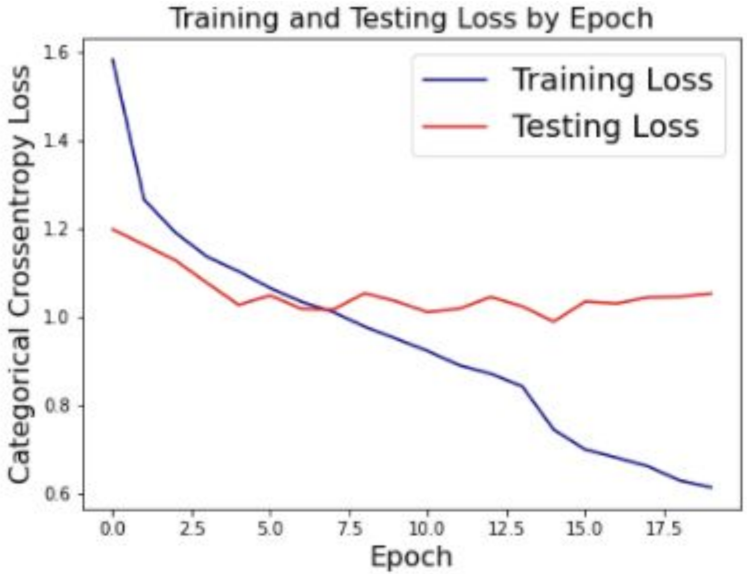
- Froze all BN Layers
- Froze up till 7th conv layer
- Added 2 dense layers
- Dropout 0.7

modelling - VGG & Effnet base

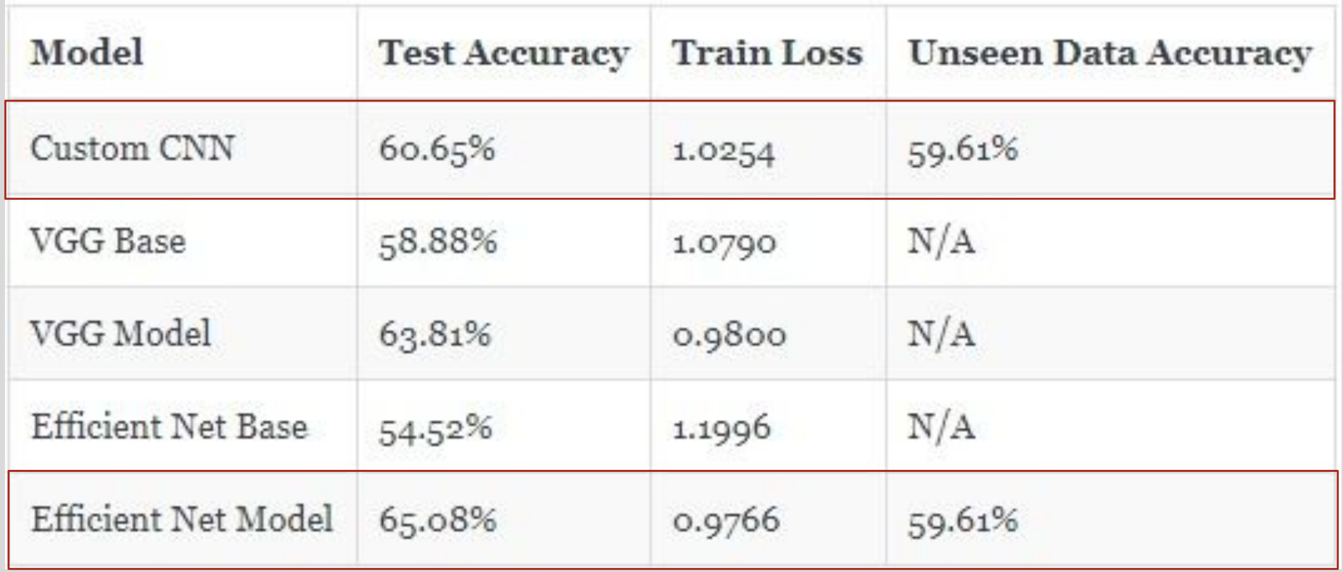
Validation accuracy - 65%

Validation loss - 0.98





modelling - summary

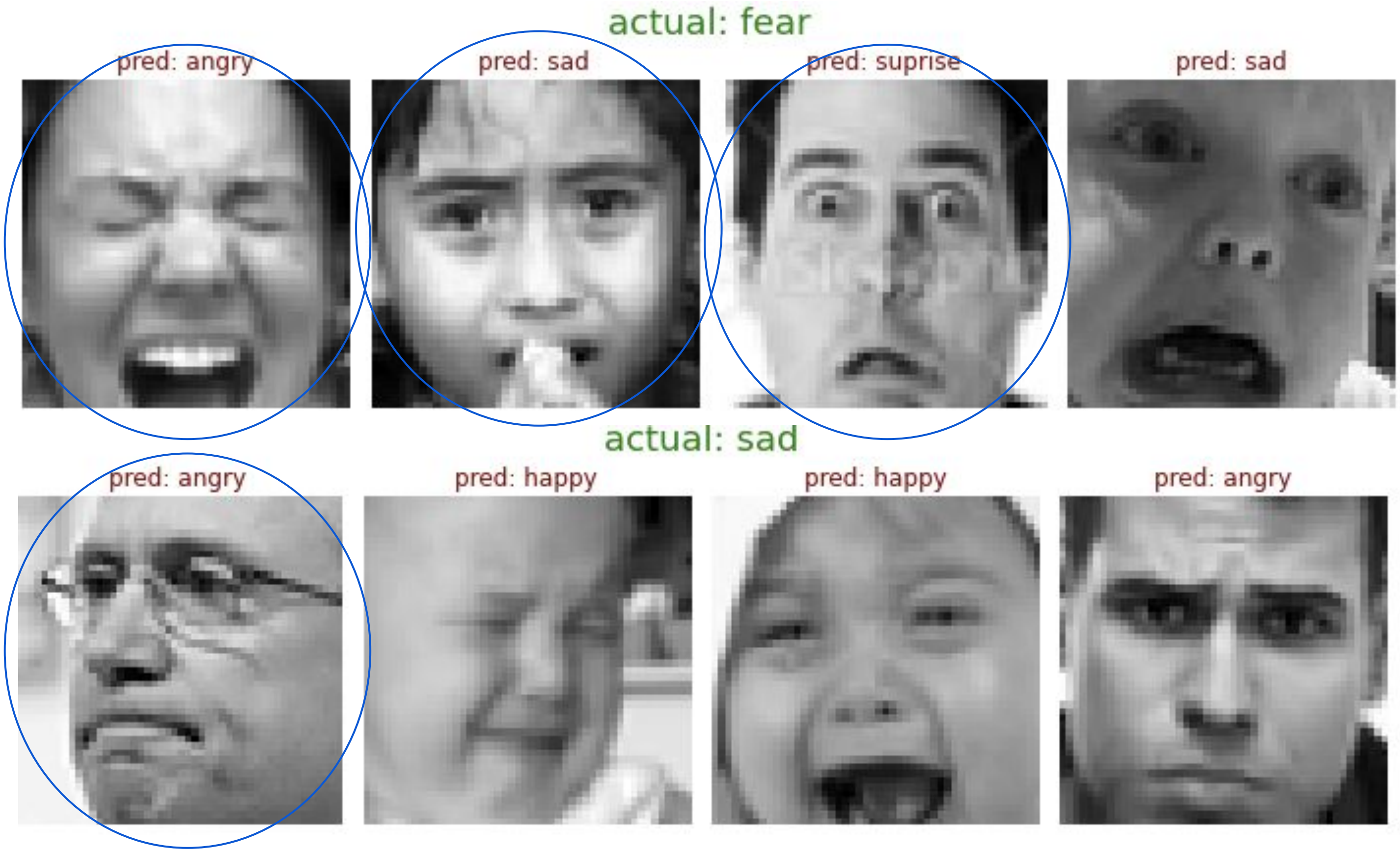


Model	Test Accuracy	Train Loss	Unseen Data Accuracy
Custom CNN	60.65%	1.0254	59.61%
VGG Base	58.88%	1.0790	N/A
VGG Model	63.81%	0.9800	N/A
Efficient Net Base	54.52%	1.1996	N/A
Efficient Net Model	65.08%	0.9766	59.61%

highest Kaggle score - ~71%

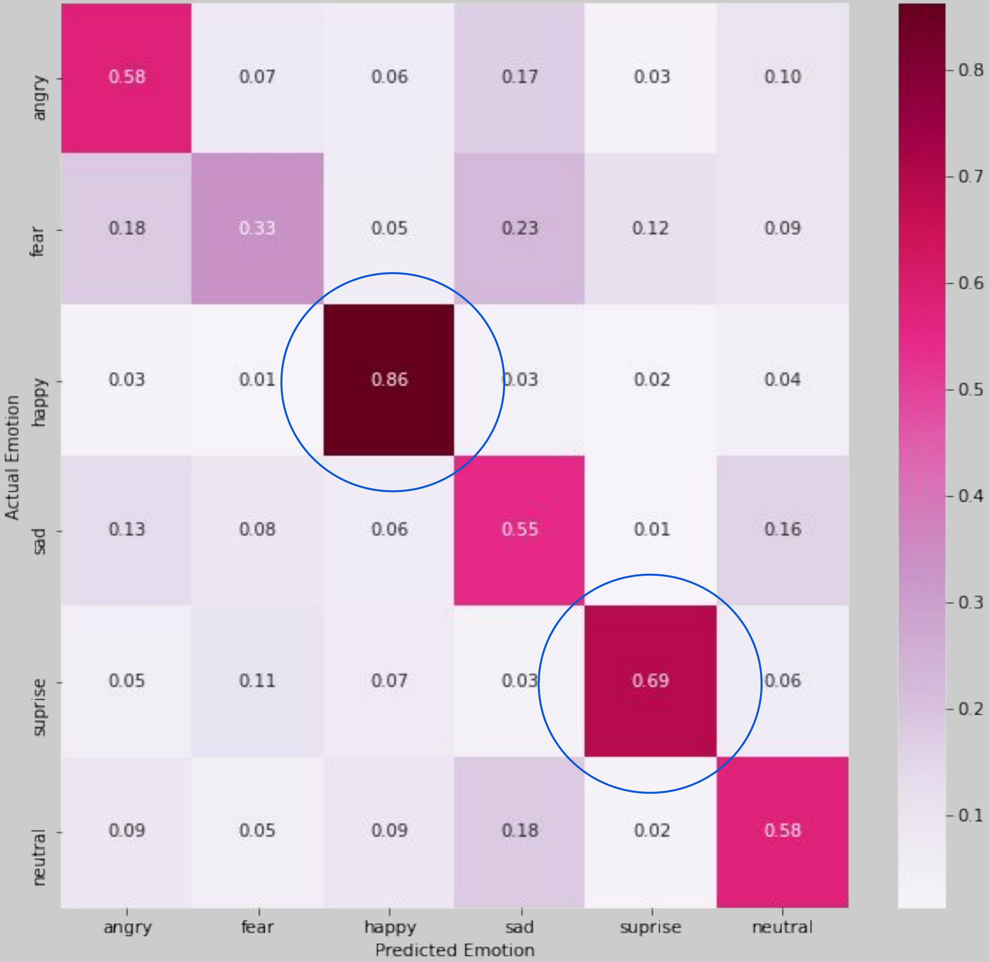
predictions

predictions - test set



predictions - test set

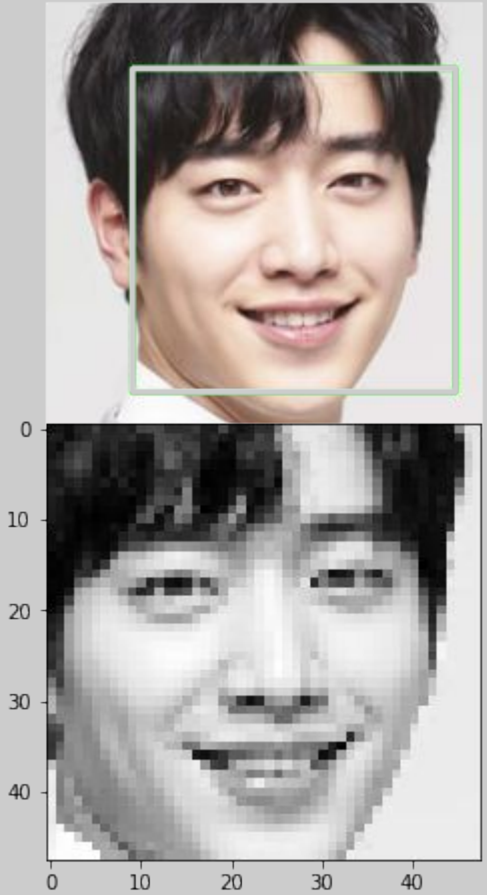
emotion	misclassified_count
fear	804
sad	668
neutral	643
angry	546
happy	304
surprise	252

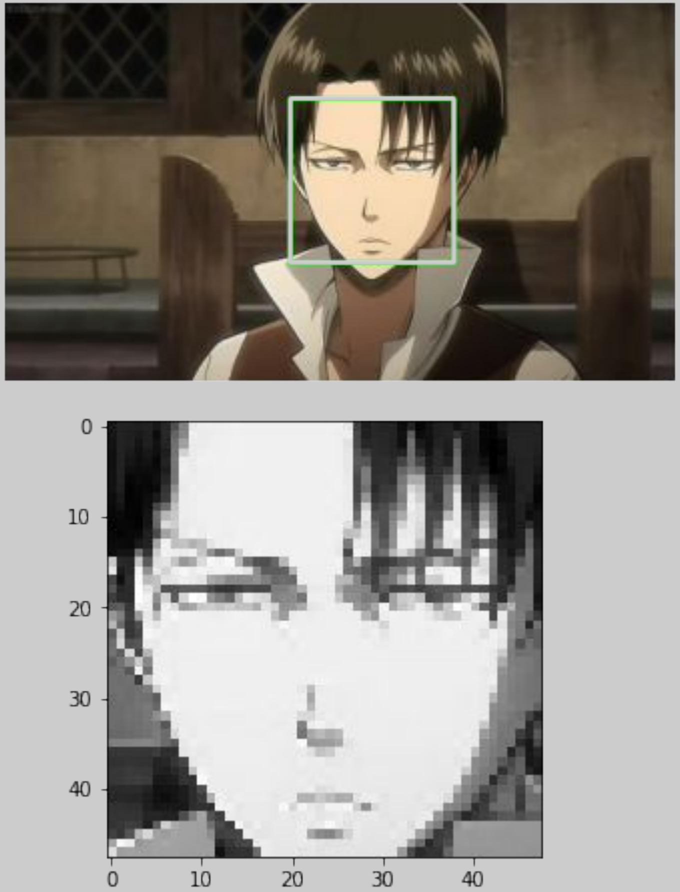


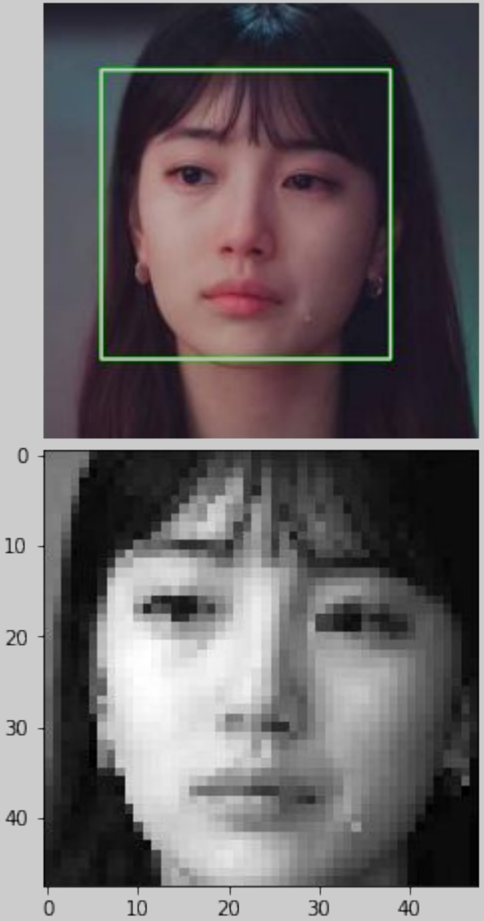
Actual Emotion \ Predicted Emotion	angry	fear	happy	sad	suprise	neutral
angry	0.58	0.07	0.06	0.17	0.03	0.10
fear	0.18	0.33	0.05	0.23	0.12	0.09
happy	0.03	0.01	0.86	0.03	0.02	0.04
sad	0.13	0.08	0.06	0.55	0.01	0.16
suprise	0.05	0.11	0.07	0.03	0.69	0.06
neutral	0.09	0.05	0.09	0.18	0.02	0.58

predictions - external data

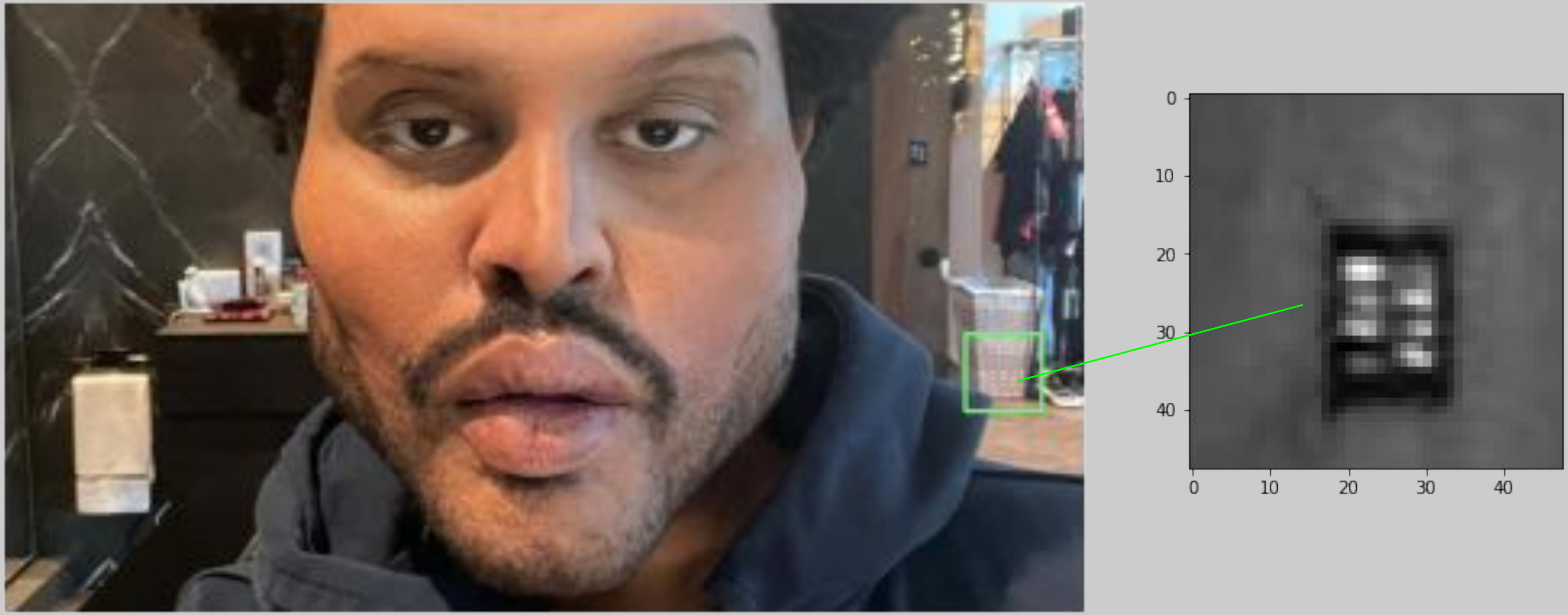
predictions - external data - crop



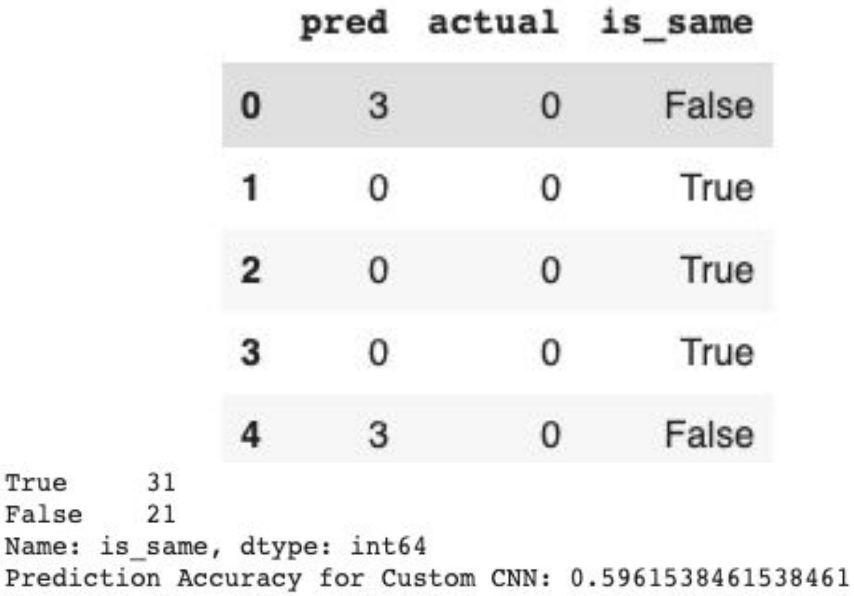




predictions - external data - crop

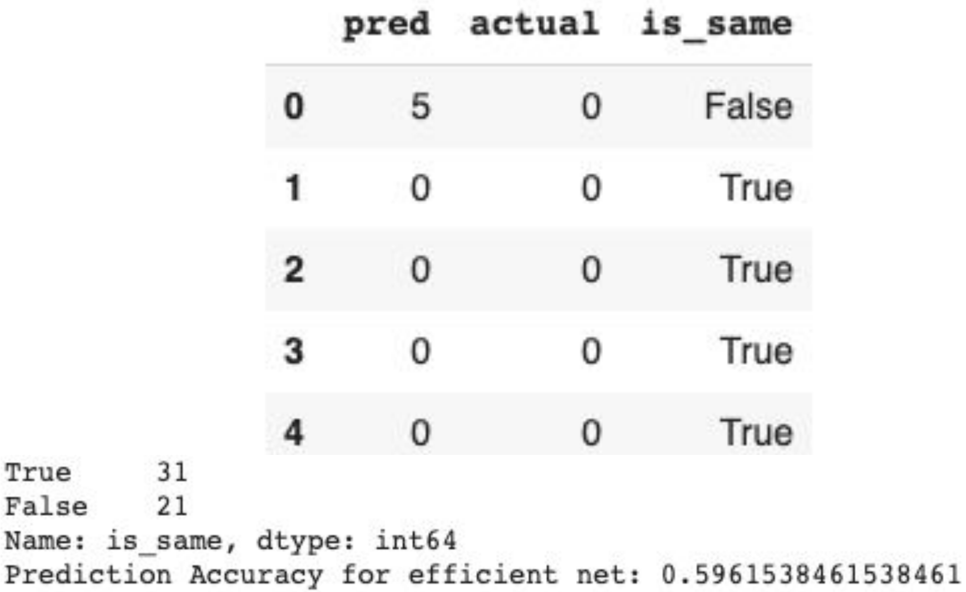


predictions - external data - crop



	pred	actual	is_same
0	3	0	False
1	0	0	True
2	0	0	True
3	0	0	True
4	3	0	False

```
True     31
False    21
Name: is_same, dtype: int64
Prediction Accuracy for Custom CNN: 0.5961538461538461
```



	pred	actual	is_same
0	5	0	False
1	0	0	True
2	0	0	True
3	0	0	True
4	0	0	True

```
True     31
False    21
Name: is_same, dtype: int64
Prediction Accuracy for efficient net: 0.5961538461538461
```

conclusions & recommendations

- make use of tuners to find optimal hyperparameters

- image augmentation was not very useful for this dataset

- explore more pretrained models

thank you

Q & A?