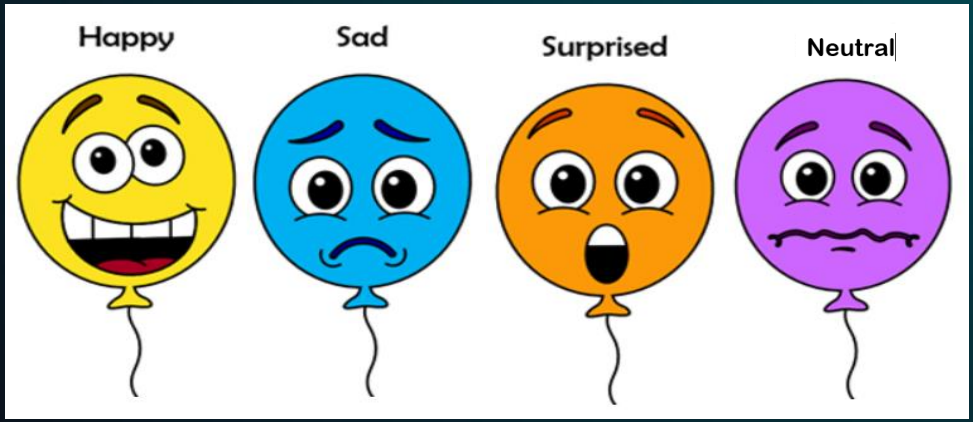
Facial Emotion Detection

Jeff Stearns

PROBLEM DEFINITION

- Can emotion be accurately detected by Convolutional Neural Networks (CNN)?
- Average natural human ability to detect emotions is around 90%, can we approach this level?

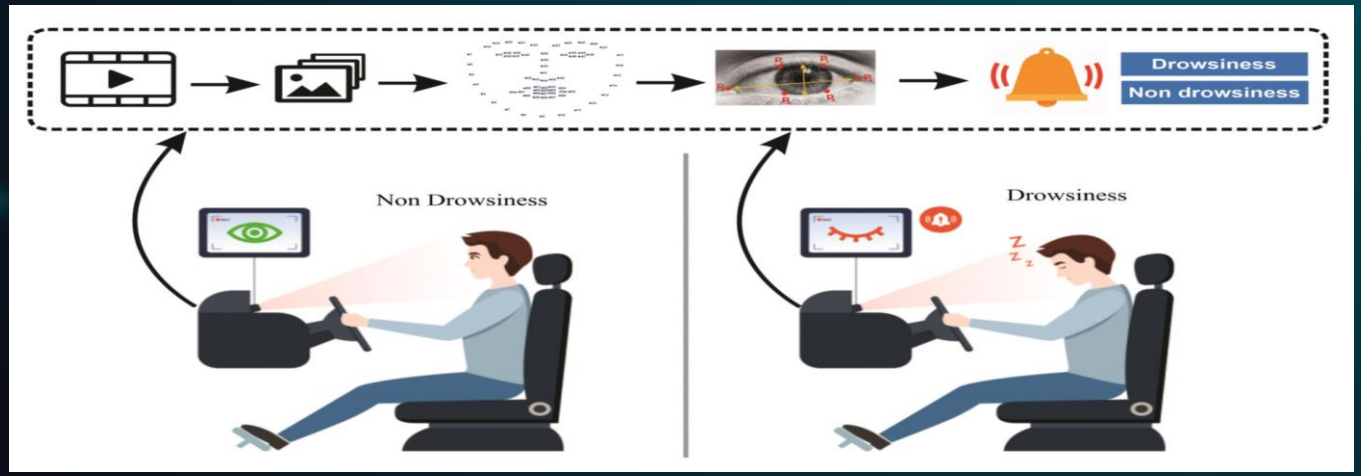


EMOTION DETECTION USE CASES

- Self driving car, detect drowsy or distracted.
- Potentially assist with security: Predict dangerous behaviors by reading emotions.
- Detect emotional response of customers to advertisement.

FACIAL EMOTION DETECTION USE CASE 1

Autopilot Automobiles : Driver alertness still required, sensors and software are not perfect.

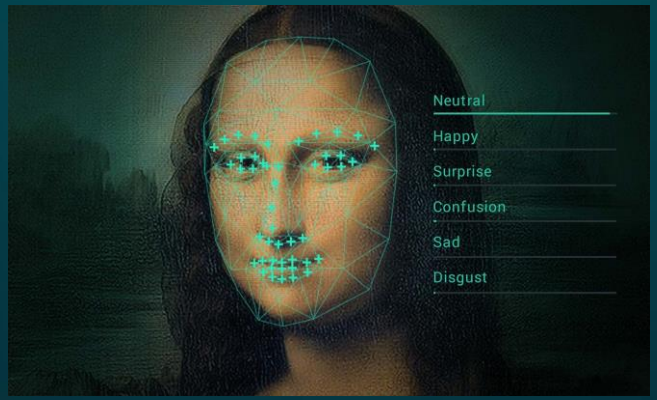


FACIAL EMOTION DETECTION USE CASE 2

Security alert, detect potentially dangerous individual through emotion recognition.

FACIAL EMOTION DETECTION USE CASE 3

- Detect emotional response of customers to advertisement.



DATA SET ANALYSIS

Facial_emotion_images.zip

- Black and White Images : 48 Pixel Squares
- Example Images

Happy

Neutral

Sad

Surprise

DATA SET ANALYSIS

- 20,214 Grayscale Images, 48 Pixel Square
- Very small TEST set : 60-80 / 10-20 / 10-20 standard ratios

TRAINING			VALIDATION			TEST		
Happy	3976	26.3%	Happy	3976	36.7%	Happy	32	25.0%
Sad	3982	26.4%	Sad	3982	22.9%	Sad	32	25.0%
Neutral	3978	26.3%	Neutral	3978	24.4%	Neutral	32	25.0%
Surprise	3173	21.0%	Surprise	3173	16.0%	Surprise	32	25.0%
15109		**74.7%**	**4977**		**24.6%**	**128**		**0.6%**

DATA SET ISSUES

No face, watermark, multiple face, cartoon, etc...

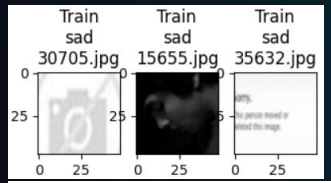
train\happy

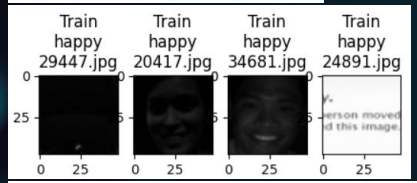
train\sad

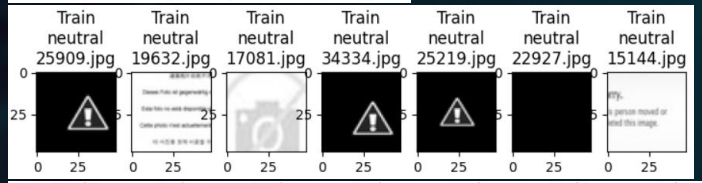
test\happy

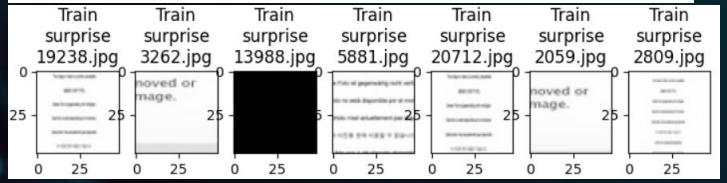
DATA SET IMAGES, NO FACE or TOO DARK

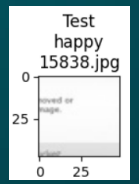
- Filter Tool : Reject Images Numpy.Average<20 or >235
- 25 Images : 21 Train, 3 Validation, 1 Test











CONVOLUTIONAL NEURAL NETWORK (CNN) MODELS

MODEL	LAYERS	PARAMETERS	COMMENT
Grayscale 1	5 : 3 Conv2d	605,060	Lowcode Model
Grayscale 2	6 : 3 Conv2d	389,604	BatchNormalization, Dense
VGG16	16 : 13 Conv2d	14,714,688	Transfer Model 1, RGB Images
ResNetV2	164 : 1000 Categories	42,658,176	Transfer Model 2, RGB Images
EfficientNet	B7 : 813	8,769,374	Transfer Model 3, RGB Images
Milestone 1	7 : 4 Conv2d	1,592,324	Kernel Size 3 & 2, Conv2D added
CapStone	8 : 5 Conv2d	2,973,700	Conv2D Layer Added

DATASET LOADERS

TENSORFLOW TOOLS

- ImageDataGenerator : Image Augmentations Applied
  - rescale : Normalize pixel values 0-1
  - horizontal_flip : Teach model differing orientations
  - rotation_range : Handle random rotation of images
- flow_from_directory method
  - Read data, labels from directory structure
  - Easier to load and process data in batches

CNN MODEL 1

Low Code Notebook Model

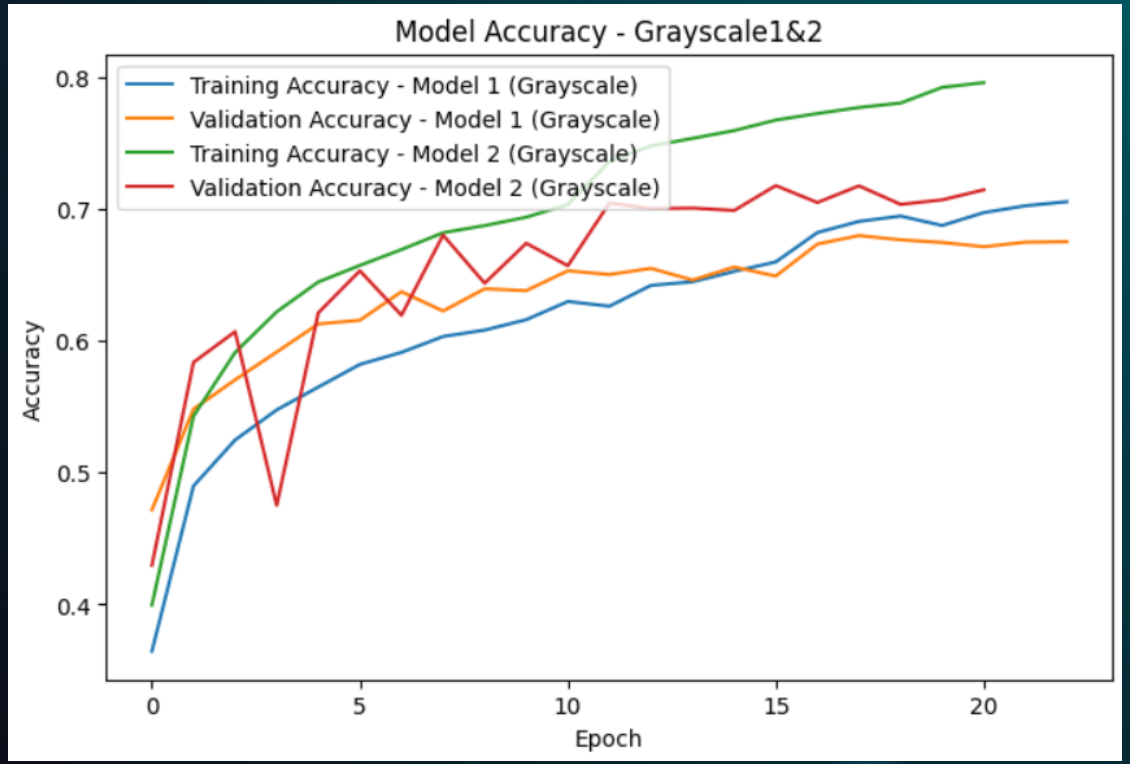
- We want our Base Neural Network architecture to have 3 convolutional blocks.
- Each block contains one Conv2D layer, maxpooling layer and Dropout layers.
- Add first Conv2D layer with **64 filters** and a **kernel size of 2**, 'same' padding **input shape = (48, 48, 1) if you're using 'grayscale' colormode**. Use **'relu' activation**.
- Add a second Conv2D layer with **32 filters** and a **kernel size of 2, 'same' padding** and **'relu' activation.**
- Add a third Conv2D layer with **32 filters** and a **kernel size of 2, 'same' padding** and **'relu' activation.**
- After adding your convolutional blocks, add your Flatten layer.
- Add your first Dense layer with **512 neurons**. Use **'relu' activation function**.
- Add a Dropout layer with dropout ratio of 0.4.
- Add your final Dense Layer with 4 neurons and **'softmax' activation function**
- Print your model summary

CNN MODEL 2

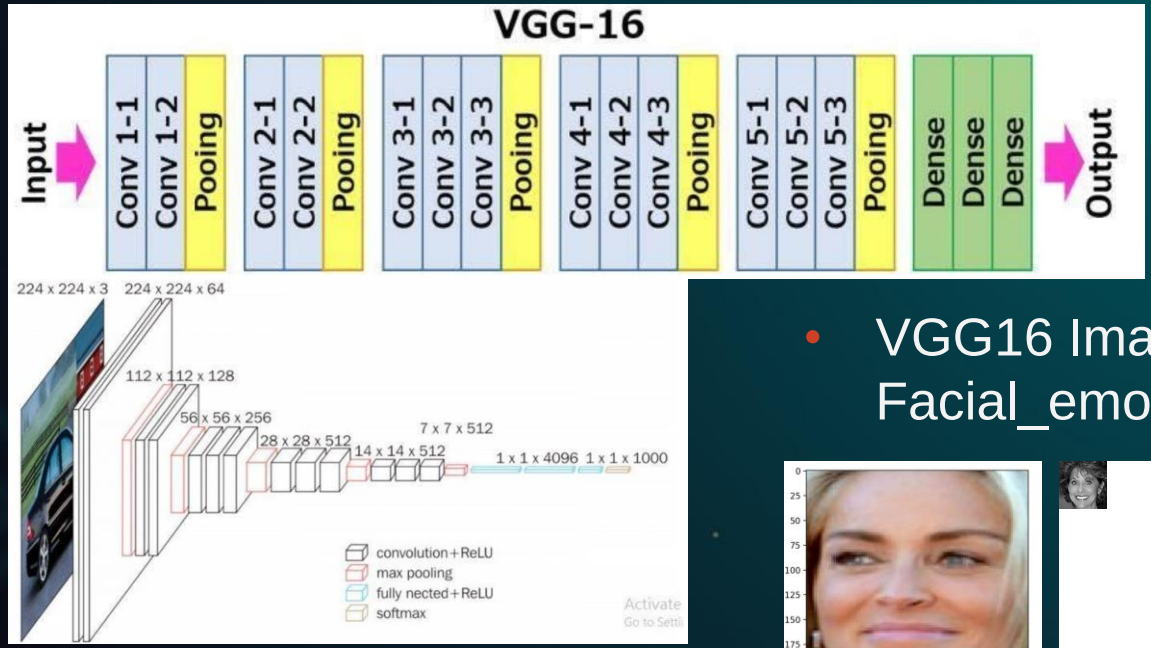
Enhanced Low Code Notebook Model

- Conv2D Layer Added
- Dense Layer Added
- More Neurons per Conv2D Layer
- BatchNormalization Added

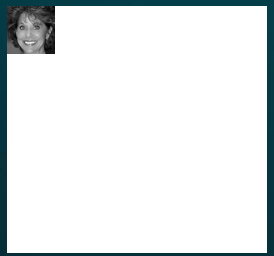
CNN MODELS 1 & 2 GRAPH



VGG-16 ARCHITECTURE



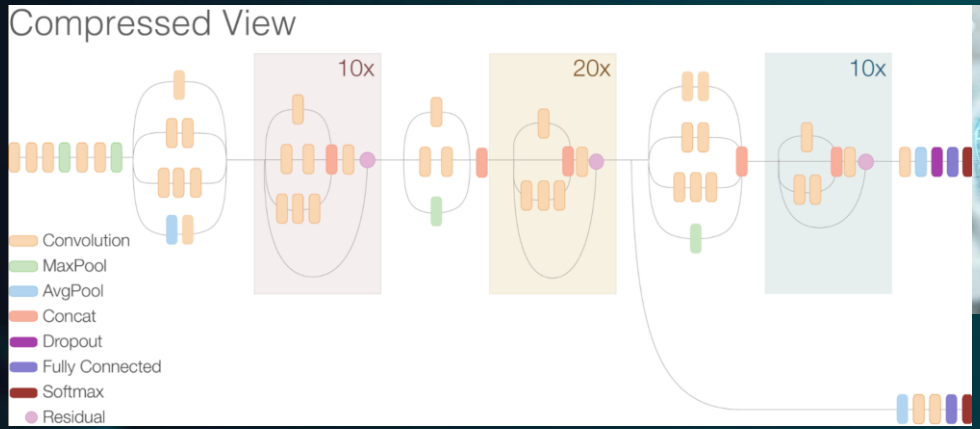
- VGG16 Image vs Facial_emotion_images

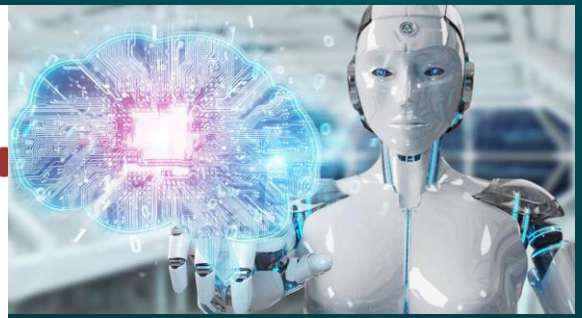


- 65 to 1 Image Data

ResNetV2 ARCHITECTURE

- 42,658,176 Parameters

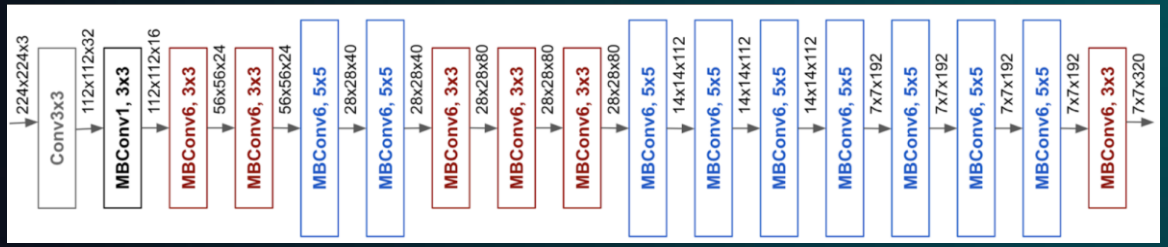




- Poor performance on Dataset : ~25%

EFFICENTNET ARCHITECTURE

- 8,769,374 Parameters

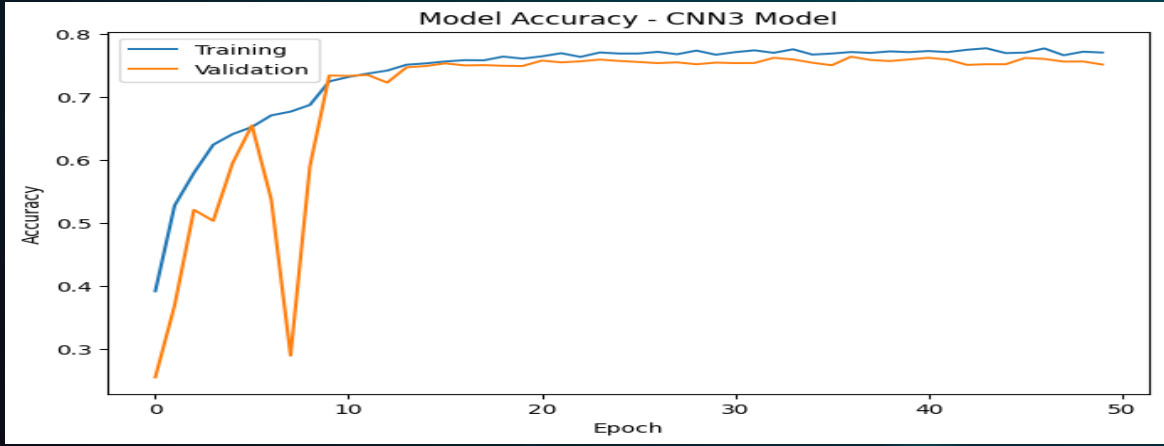


- Poor performance on Dataset : ~25%

MILESTONE 1 MODEL

Enhanced Version 3 CNN Model

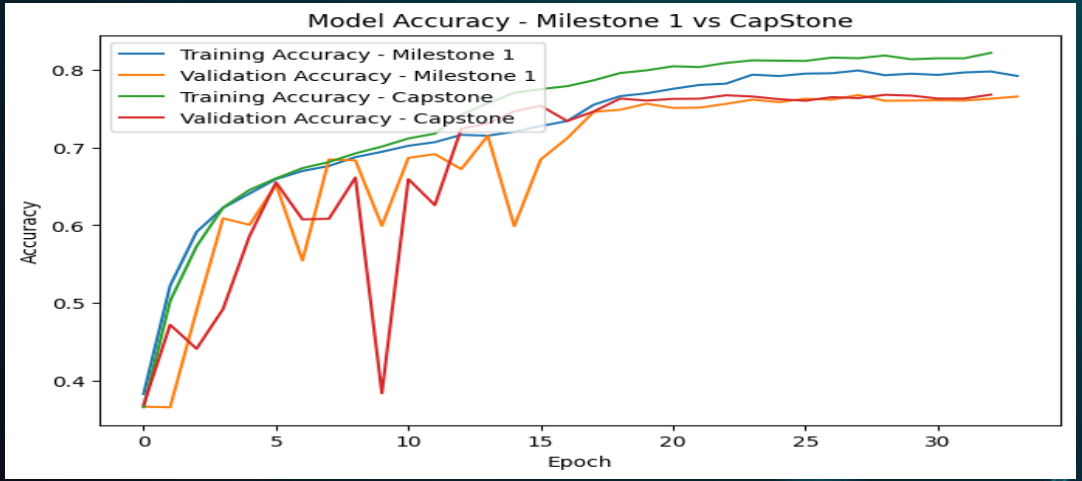
- Conv2D Layer Added
- More Neurons per Conv2D Layer
- Data Loader Augmentation Hyperparameter adjustments
- Validation accuracy dips to in Epoch 8?



CAPSTONE MODEL

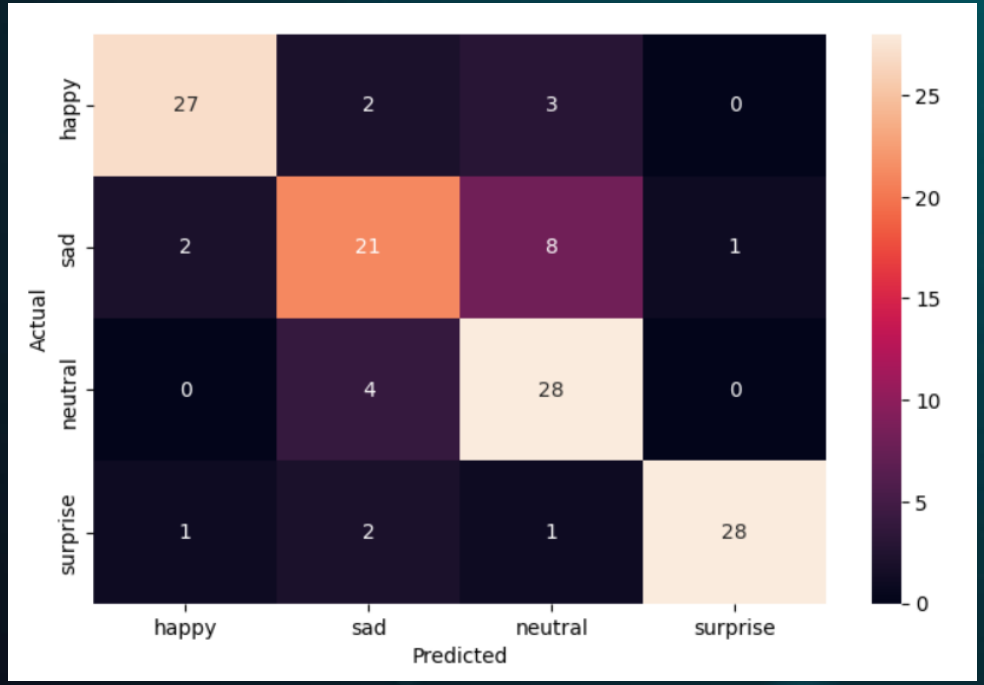
Final CapStone Model Version

- Conv2D Layer Added
- More Neurons per Conv2D Layer
- More parameters : 2,973,700

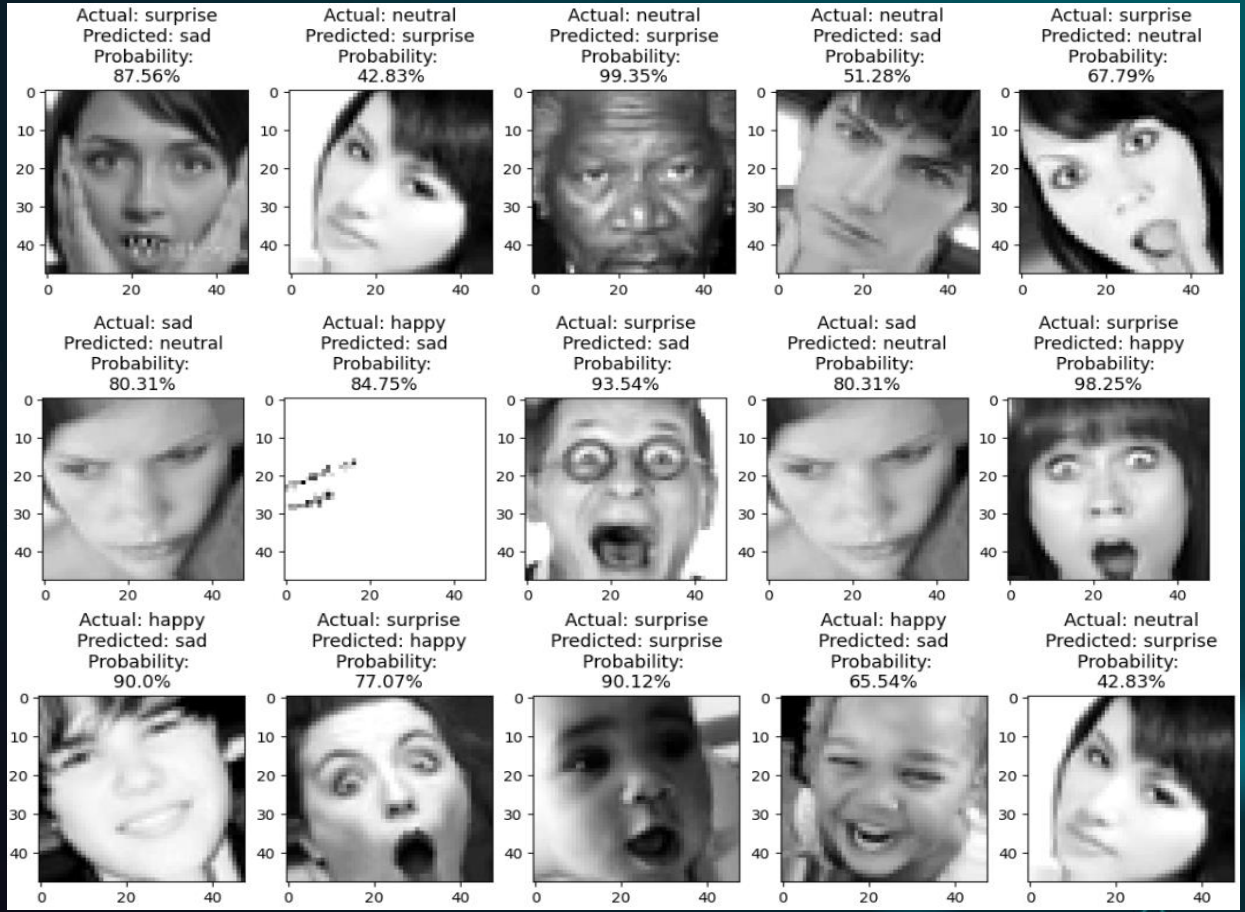


FINAL MODEL HEATMAP

PREDICTED RESULTS vs ACTUAL (Folder)



FINAL MODEL MISMATCH IMAGES



FINAL CNN MODEL ANALYSIS

	Parameters	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
Model 1: Grayscale 1	605,060	0.706	0.725	0.68	0.79	0.62	0.81
Model 2: Grayscale 2	389,604	0.796	0.496	0.715	0.74	0.68	0.68
Model 3: VGG16	14,714,688	0.72	0.687	0.675	0.815	0.72	0.71
Model 4: ResNet V2	42,658,176	0.27	1.38	0.36	1.37	0.25	1.40
Model 5: EfficientNet	8,769,374	0.27	1.38	0.37	1.35	0.25	1.41
Model 6: Grayscale 3	1,592,324	0.80	0.51	0.77	0.60	0.805	0.543
Model 7: Grayscale 4	2,973,700	0.83	0.44	0.77	0.65	0.77	0.60

CONCLUSIONS

- Less complex models work better with Facial_emotion_images
- Transfer learning models mediocre to poor on dataset

CONCLUSIONS

- How did the models come out Mr Bean?