Final Evaluation

Occluded Facial Expression Detection

Predicting Facial Expressions In Real Time

Group Members:
Abdul Mannan K-163620
Murtaza Multanwala K-163618
Muhammad Shahryar K-163750

Supervisor: Ms Javeria Farooq
Co-Supervisor: Ms Syeda Rubab Jaffar

Introduction

- Occluded facial expressions detection
- Studied existing approaches
- Found out the gap
- Presented solution to fill the gap

Occlusions

Existing work

Name	Approach	Problem
Octavio [1]	Trained neural network from scratch	Real time, frontal face, not on occlusion
Farhad [2]	Uses 3d face reconstruction technique	Real time, high computational power, single face
Duncan [3]	Transfer learning on VGG S model	Real time, frontal face, not on occlusion
Schwan [4]	Transfer learning on VGG Face model	Real time, single frontal face, not on occlusion
Enrique [5]	Transfer learning approach	Real time, single frontal face

Problem Definition

Although numerous facial expression identification methods have been proposed and built, majority of them are built in controlled surroundings. Mostly, the research is either on frontal faces or either without occlusion. Despite numerous researches [1,4] on different datasets, recognition of facial expression is still challenging because of partially occluded faces.

Objective

1. To investigate a technique of facial expression detection in real time environment.

2. To develop and validate the proposed facial recognition approach on occluded and non occluded students in class rooms.

Problem Scope

This project focuses on building a prototype for generation of automatic feedback on the basis of facial expressions. It can be deployed where the feedback is prioritized such as classrooms, cafes or meeting rooms.

Methodology

1) Camera Feed

Face Detection

2) Detected Faces

Preprocessing

Expression Classification Model

3) Final Output

Methodology(Face detection)

Stage 1:

1. Pass in image
2. Feed images into P-Net
3. Gather P-Net output
4. Delete bounding boxes with low confidence
5. Reshape bounding boxes to square

Stage 2:

1. Pad out-of-bound boxes
2. Feed images into R-Net
3. Gather R-Net output
4. Delete bounding boxes with low confidence
5. Non-Maximum Suppression for all boxes
6. Reshape bounding boxes to square

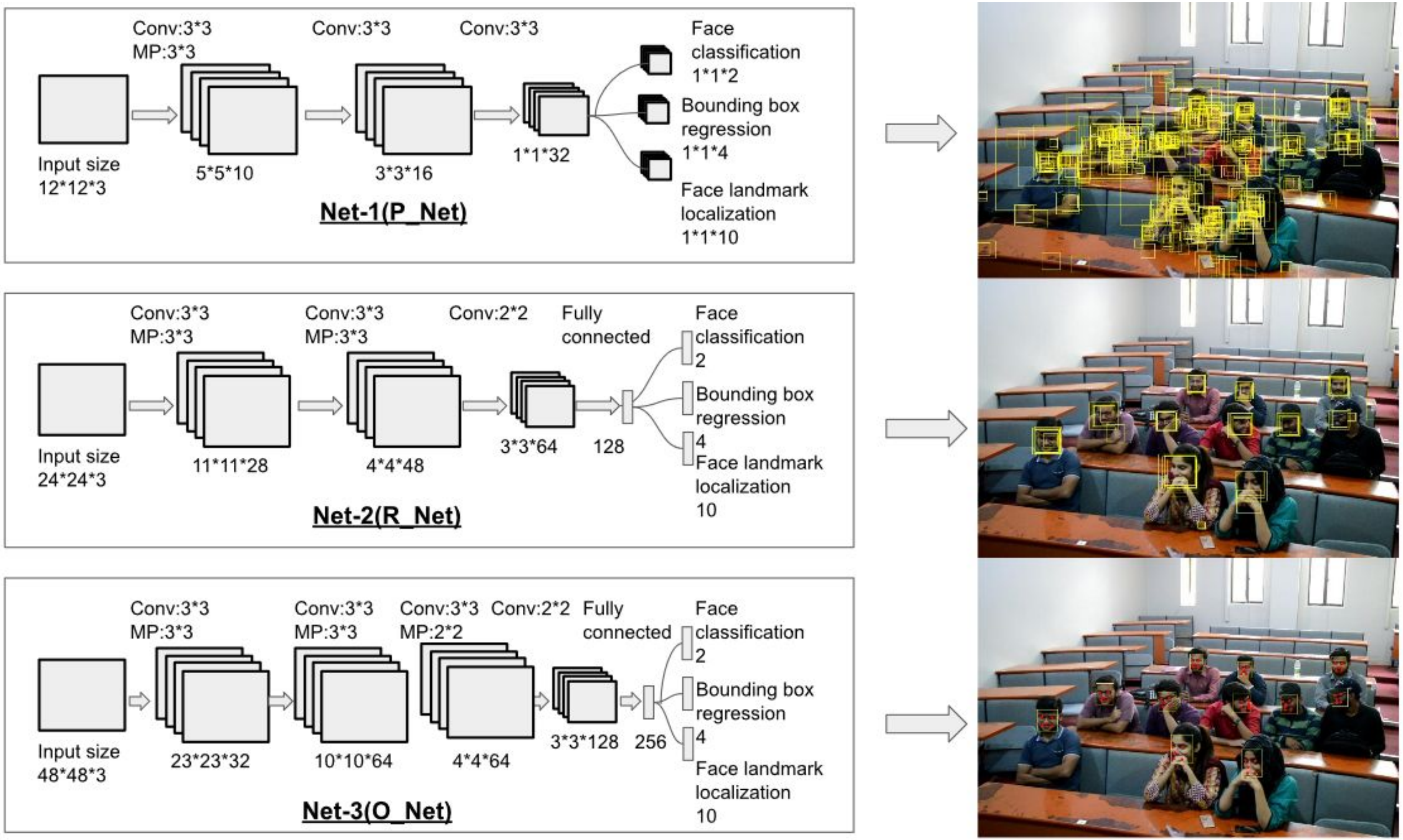
Stage 3:

1. Pad out-of-bound boxes
2. Feed images into O-Net
3. Gather O-Net output
4. Delete bounding boxes with low confidence
5. Non-Maximum Suppression for all boxes

Delivering Results:

1. Package all coordinates and confidence levels into a dictionary
2. Return the dictionary

Architecture face detector



Methodology (Facial Expression Recognition Models)

1. Approach A
    - **Train all the weights of the model from start.**
2. Approach B
    - **Transfer learning (finetune pretrained weights).**

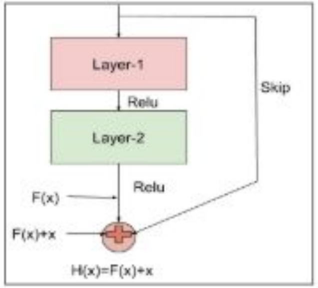
Approach A Architecture

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 46, 46, 64)	640
conv2d_2 (Conv2D)	(None, 46, 46, 64)	36928
batch_normalization_1 (Batch	(None, 46, 46, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 64)	0
dropout_1 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 23, 23, 128)	73856
batch_normalization_2 (Batch	(None, 23, 23, 128)	512
conv2d_4 (Conv2D)	(None, 23, 23, 128)	147584
batch_normalization_3 (Batch	(None, 23, 23, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 11, 11, 128)	0
dropout_2 (Dropout)	(None, 11, 11, 128)	0
conv2d_5 (Conv2D)	(None, 11, 11, 256)	295168
batch_normalization_4 (Batch	(None, 11, 11, 256)	1024
conv2d_6 (Conv2D)	(None, 11, 11, 256)	590080
batch_normalization_5 (Batch	(None, 11, 11, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 256)	0
dropout_3 (Dropout)	(None, 5, 5, 256)	0
conv2d_7 (Conv2D)	(None, 5, 5, 512)	1180160
batch_normalization_6 (Batch	(None, 5, 5, 512)	2048
conv2d_8 (Conv2D)	(None, 5, 5, 512)	2359808
batch_normalization_7 (Batch	(None, 5, 5, 512)	2048
max_pooling2d_4 (MaxPooling2	(None, 2, 2, 512)	0
dropout_4 (Dropout)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_6 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dropout_7 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 7)	903

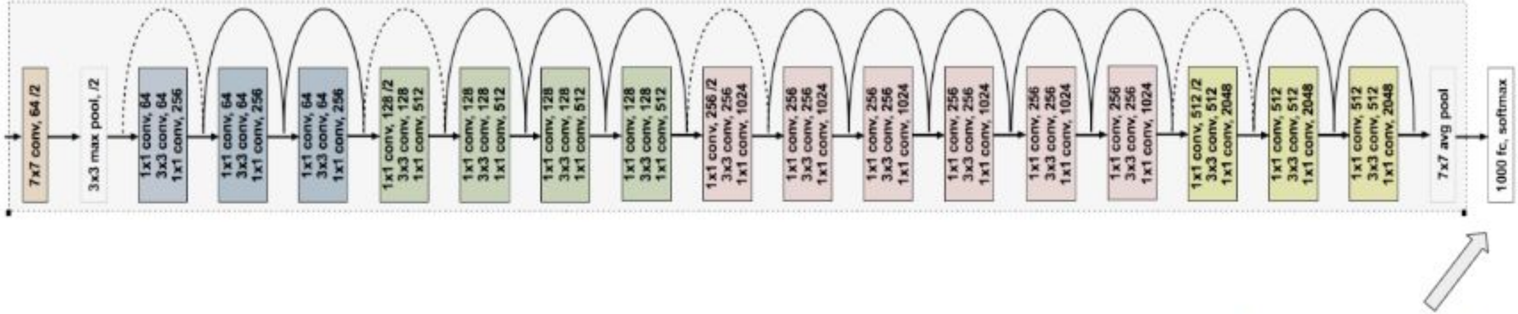
Total params: 5,905,863
Trainable params: 5,902,151
Non-trainable params: 3,712

Approach B Architecture



Residual Learning Block

ResNet 50 architecture



Fully Connected Layer

Approach A Results

Approach B Results

Dataset (FER2013)

Number of images	Emotion	Label
4593	Angry	0
547	Disgust	1
5121	Fear	2
8989	Happy	3
6077	Sad	4
4002	Surprise	5
6198	Neutral	6

Dataset Distribution

Emotion	Prototype label
Happy	Happy
Neutral	Neutral
Fear, Surprise	Surprised
Angry, Sad, Disgust	Bored

Prototype labels

FER2013 sample images

Total: 35,887 (48x48) gray scaled images

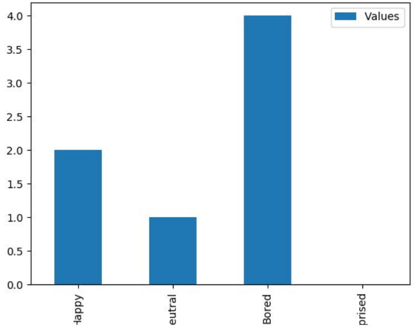
Sample Results

FYP-1

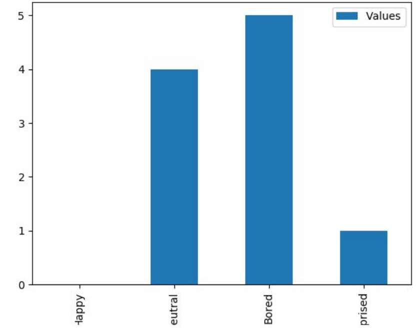
Deliverables Achieved

FYP-1

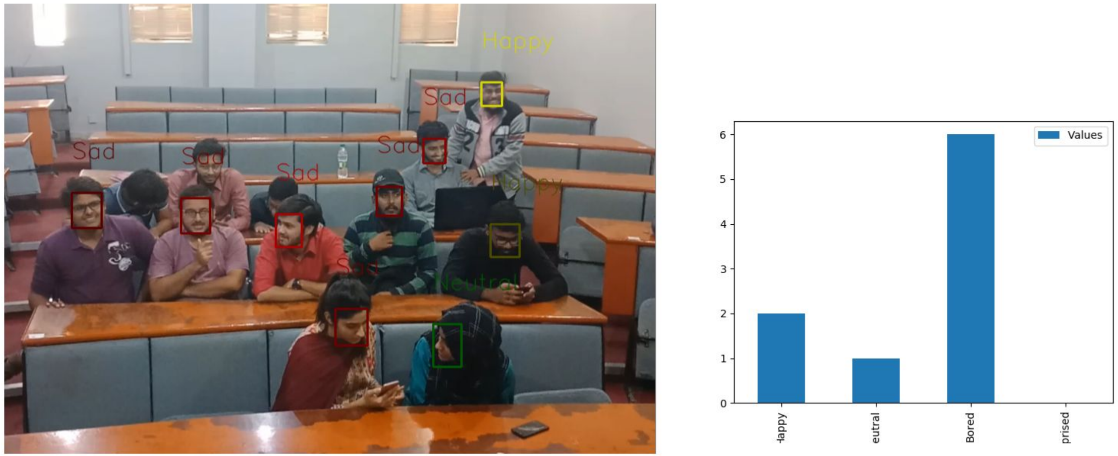
- ➔ Prototype ✓
- ➔ Multiple faces detection in real time ✓
- ➔ Predict facial expressions in real time ✓
- ➔ Dataset creation ✓
- ➔ Feedback generation(Jury recommendation) ✓



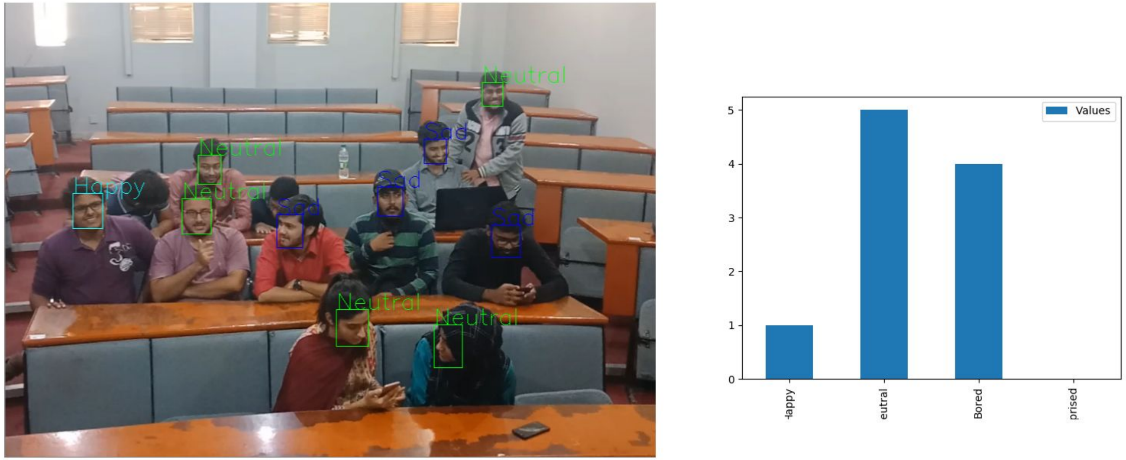
Approach A



Approach B



Approach A



Approach B

FYP-2(Improvements)

Deliverables Achieved

FYP-2

- ➔ Improve accuracy of FYP-1 ✔
- ➔ Dataset merge(Jury recommendation) ✔
- ➔ Creation of Expression Classification Model on merged dataset(Jury recommendation) ✔
- ➔ Identify facial features(Jury recommendation) ✔
- ➔ Define Expressions/Emotions (Jury recommendation) ✔
- ➔ Comparison between Approach A and B(Jury recommendation) ✔
- ➔ Plotting of segmented images of interest region(Jury recommendation) ✔


Neutral
Happiness

Bored
Happy
Happy
Surprised
Happy
Happy
Bored
Bored
Happy

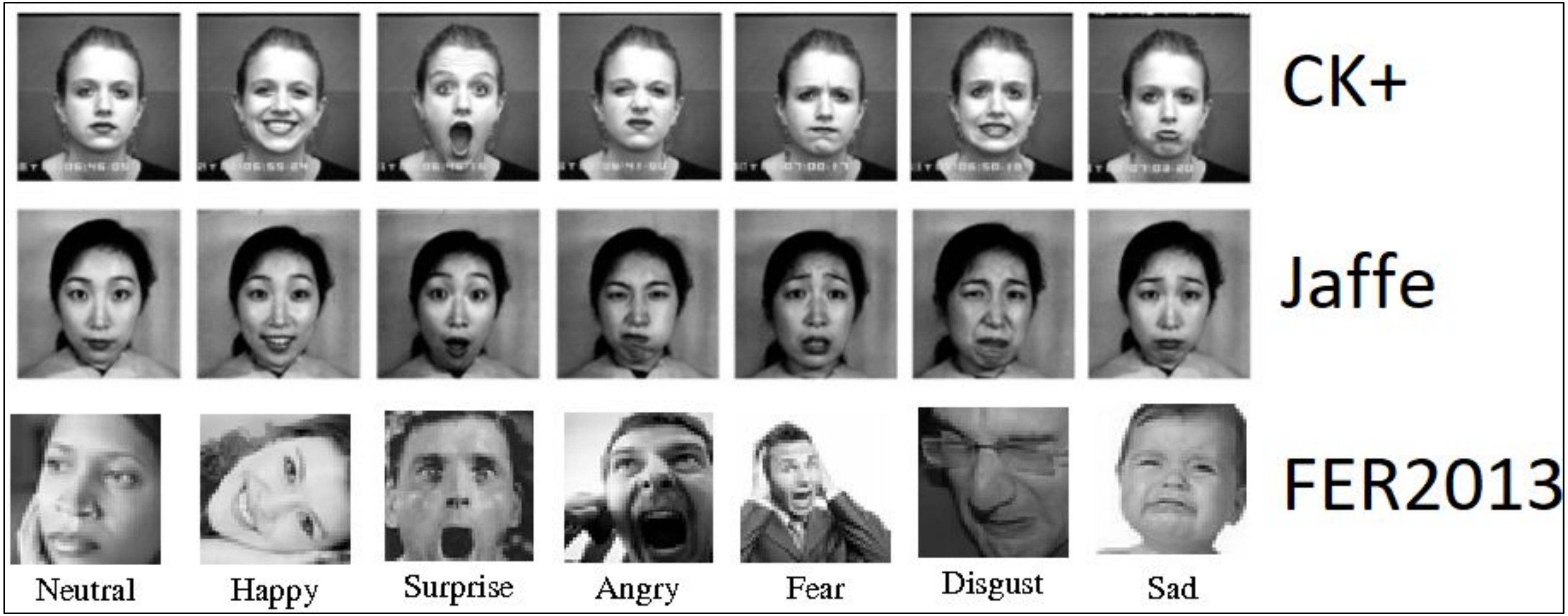

Neutral
Neutral
Bored
Neutral
Bored
Bored
Bored
Happy
Bored
Bored
Happy


Neutral
Neutral
Bored
Happy
Happy
Surprised
Bored
Happy
Neutral

Happy
Happy
Happy
Happy
Happy
Happy
Happy
Happy
Happy
Happy
Bored
Surprised
Happy
Happy
Neutral
Bored
Surprised
Neutral
Happy

Merged Dataset

CK+	FER2013	JaFFE
• 5,876 images	• 35,887 images	• 213 images



Model Accuracy

Validation split of 0.3 and epoch = 30

```
Epoch 30/30
247/247 [==============================] - 0s 794us/step - loss: 0.0786 - acc: 0.9798


Test Loss: 4.182190297141908
Test accuracy: 0.5238095162406801
```

Validation split of 0.1 and epoch = 40

```
Epoch 40/40
318/318 [==============================] - 0s 815us/step - loss: 0.0204 - acc: 0.9906


Test Loss: 4.378694231548007
Test accuracy: 0.5079365003676641
```

Validation split of 0.01 and epoch = 20

```
350/350 [==============================] - 0s 725us/step - loss: 0.0076 - acc: 1.0000
Epoch 20/20
350/350 [==============================] - 0s 740us/step - loss: 0.0089 - acc: 1.0000


Test Loss: 4.627334564451187
Test accuracy: 0.5555555470406063
```

Validation split of 0.01 and epoch = 20

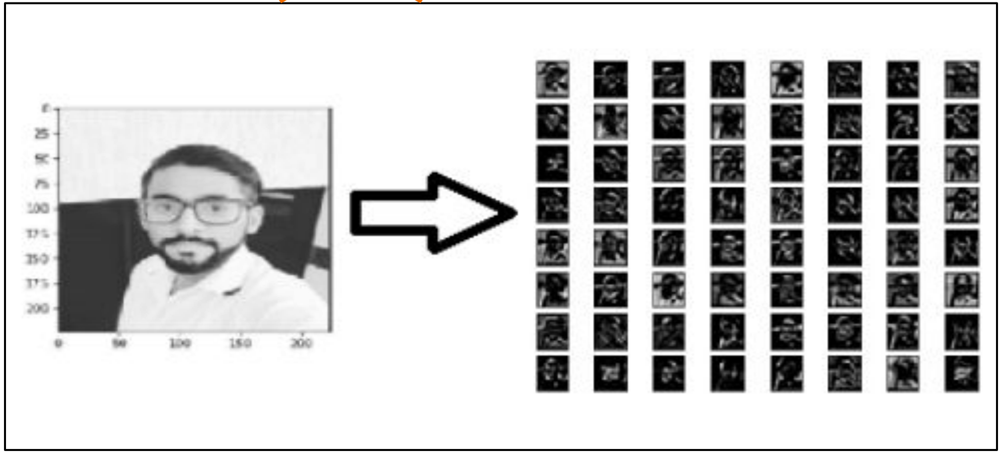
```
Epoch 20/20
350/350 [==============================] - 0s 705us/step - loss: 0.0134 - acc: 0.9886 - val_loss: 4.0296 - val_acc: 0.7500


Test Loss: 5.188326744806199
Test accuracy: 0.5555555621782938
```

- **Total Images = 417(CK+, FER2013, JAFFE)**
- **Training Images = 354 ⇒ Validation set (X% from training data)**
- **Testing Images = 63 (0.15% of training data)**

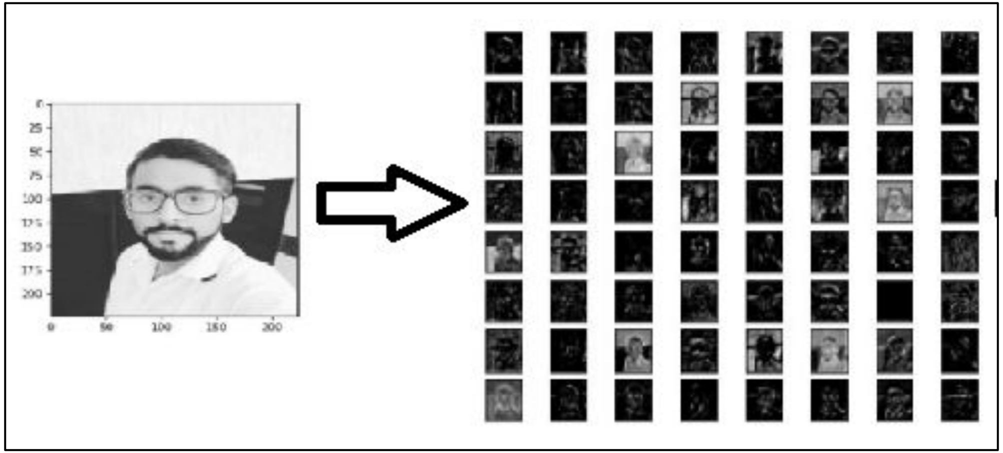
Model Summary

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 6)         456
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 16)        2416
_________________________________________________________________
activation_1 (Activation)    (None, 24, 24, 16)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 64)        9280
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               204928
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 217,983
Trainable params: 217,983
Non-trainable params: 0
```

Region of interest(ROI) visualization



Approach A
ROI
visualization



Approach B
ROI
visualization

Conclusion

1. Transfer learning models provide better results than scratch models
2. Scratch models accuracy are low.
3. Transfer learning models are best suited for our task
4. Using transfer learning technique, model takes less time to train

References

[1] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. "Real-time convolutional neural networks for emotion and gender classification". In: arXiv preprint arXiv:1710.07557 (2017).

[2] Farhad Goodarzi et al. "Real time facial expression recognition in the presence of rotation and partial occlusions". In: Journal of Theoretical and Applied Information Technology 96 (Feb. 2018), pp. 854–864.

[3] Dan Duncan, Gautam Shine, and Chris English. "Facial emotion recognition in real time". In: Stanford University (2016).

[4] Justus Schwan et al. "High-performance and lightweight real-time deep face emotion recognition". In: 2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). IEEE. 2017, pp. 76–79.

[5] Enrique Correa et al. "Emotion recognition using deep convolutional neural networks". In:Tech. Report IN4015 (2016)

[6] Kaipeng Zhang et al. "Joint face detection and alignment using multitask cascaded convolutional networks". In: IEEE Signal Processing Letters 23.10 (2016), pp. 1499–1503.

[7] Google images

Thank You