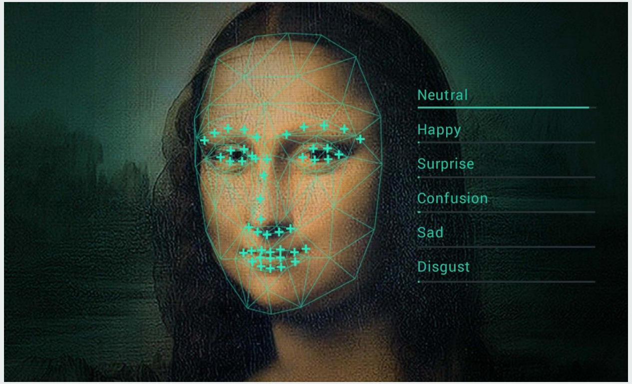
Facial Expression Recognition with Feature Visualization & Explainability

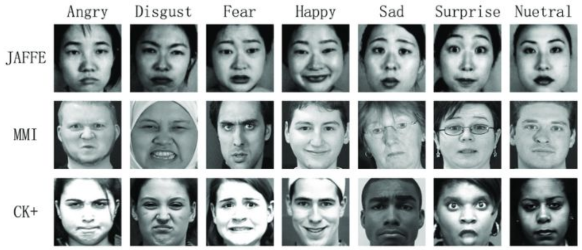
CS523 Final Project Team 2

Members:
Kaiyang Zhao (kyzhao@bu.edu)
Bingquan Cai (bqcai@bu.edu)
Bin Xu (xu842251@bu.edu)



Kaiyang Zhao contributed to experimental training part; Bingquan Cai contributed to model explainability part; Bin Xu contributed to feature visualization part; All participated in the concept and model establishment.

Facial Expression Recognition (FER)



- **What is FER trying to do**

  Recognizing facial expressions automatically enables applications in human-computer interaction and other areas.

- **Importance of FER**

  Being able to recognize facial expressions is key to nonverbal communication between humans. Facial expressions and other gestures convey nonverbal communication that play an important role in interpersonal relations. We can use algorithms to instantaneously detect faces, code facial expressions, and recognize emotional states.

BOSTON UNIVERSITY

Our project goal

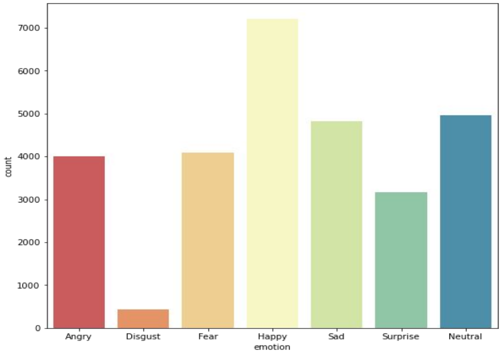
- Looking for FER related papers to reproduce the models and results
  we review the state of the art in image-based facial expression recognition using CNNs and find algorithmic differences and their performance impact.

- Feature visualization of different emotion classes
- Try to find some explainability of those FER models

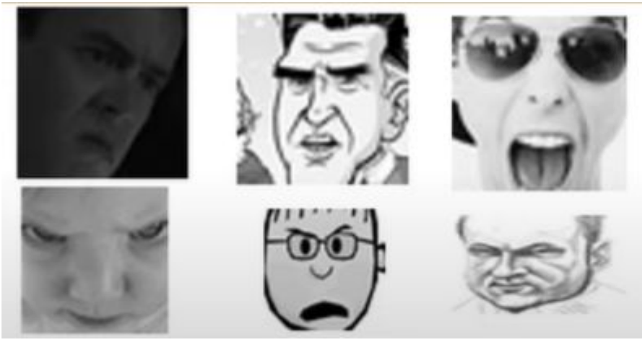
FER2013 dataset

- FER2013 dataset 48x48 pixel grayscale images of faces (35,887)
- Training (28,709), Validation (3,589), Testing (3,589)
- It has 7 categories of facial expressions (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)

- Problems:
  1. Data imbalance -> Sol: Data augmentation
  2. Intra-class variation -> Sol: Avoid overfitting



Angry



BOSTON UNIVERSITY

Main references

- Goodfellow I J, Erhan D, Carrier P L, et al. **Challenges in representation learning: A report on three machine learning contests**[C]//International conference on neural information processing. Springer, Berlin, Heidelberg, 2013: 117-124.

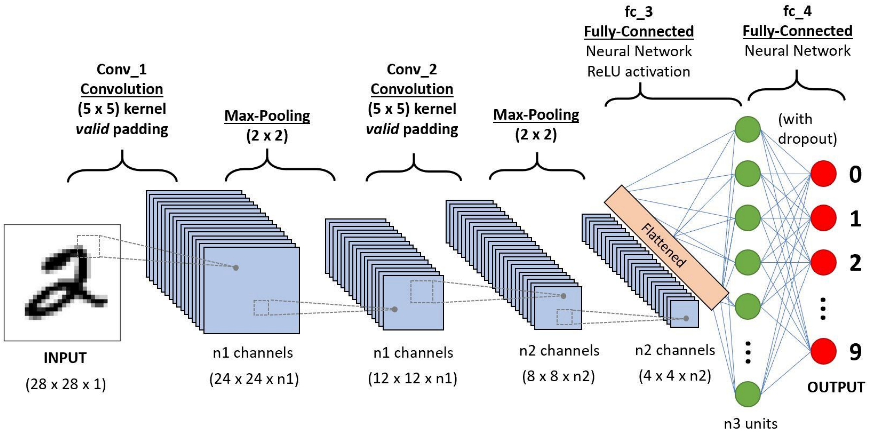
- Pramerdorfer C, Kampel M. **Facial expression recognition using convolutional neural networks: state of the art**[J]. arXiv preprint arXiv:1612.02903, 2016.

- Debnath, T., Reza, M., Rahman, A., Beheshti, A., Band, S. S., & Alinejad-Rokny, H. (2022). **Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity**. *Scientific Reports*, *12*(1), 1-18.

- Selvaraju R R, Cogswell M, Das A, et al. **Grad-cam: Visual explanations from deep networks via gradient-based localization**[C]//Proceedings of the IEEE international conference on computer vision. 2017: 618-626.

Methodology State-of-Art

- Networks: CNN
  - batch normalization
  - fully connected (fc) layer
  - Drop-out layer
  - Max-pooling & stochastic pooling
- Dataset Preprocessing: illumination Correction
  - normalizing the images
- CNN architecture:
  - VGG
  - Inception
  - ResNet
- CNN training and inference:
  - 300 epochs
  - learning rate = 0.1
  - batch size = 128
  - Data augmentation

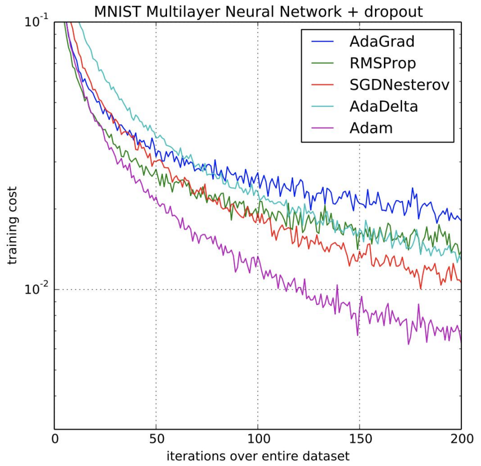


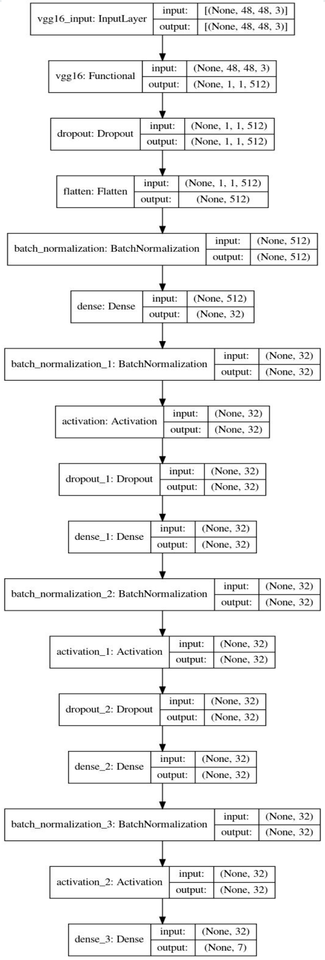
BOSTON UNIVERSITY

Methodology Our Model

- Transfer learning
- Additional layers
- Categorial Crossentropy
- Adam optimizer
- 300 Epochs

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output} \\ \text{size}}} y_i \cdot \log \hat{y}_i$$





Tools we need

- Tensorflow
  - Tensorflow.keras
- Pandas
- Numpy
- Google. Colab
- Pip
- Zipfile
- Matplotlib
- Seaborn
- skimage

```python
# dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
import skimage.io
import keras.backend as K
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten, Dropout,BatchNormalization ,Activation
from tensorflow.keras.models import Model, Sequential
from keras.applications.nasnet import NASNetLarge
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlyStopping
from tensorflow.keras.optimizers import Adam
```

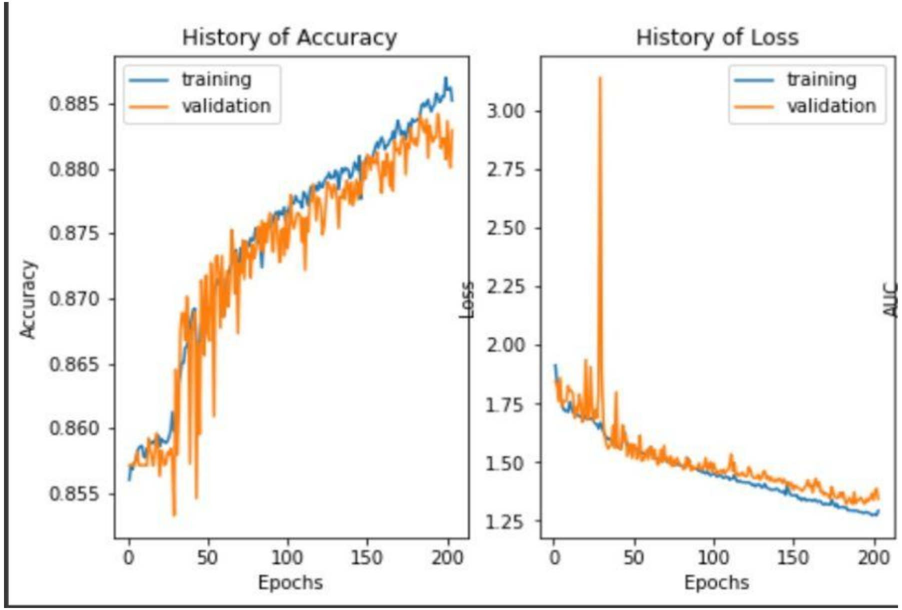
```
[6] ! kaggle datasets download msambare/fer2013

Downloading fer2013.zip to /content
 96% 58.0M/60.3M [00:00<00:00, 313MB/s]
100% 60.3M/60.3M [00:00<00:00, 302MB/s]
```

BOSTON UNIVERSITY

Training results

- Training Results of VGG16:
  - Early stoppage at epoch 203
  - Loss: 1.2933
  - Learning rate: 0.0005
  - Validation Accuracy: 0.8852



BOSTON UNIVERSITY

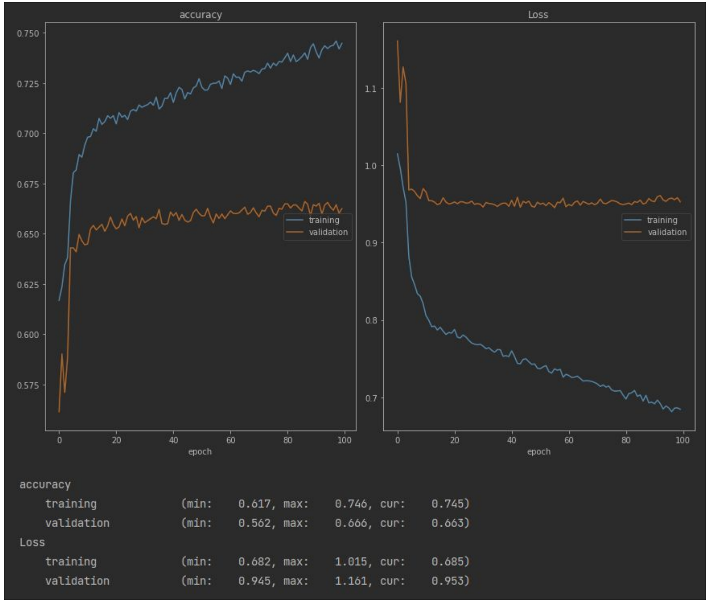
Training Results of other models

- 1st Show accuracies and plots for different methods we tried out

  Using CNN 5 layer model, **accuracy**: **0.6339**, then we increasing the epochs 1000 and using the lower learing_rate, **accuracy: 0.6682**

- Compared to other methods or papers

  CPCPCPFFF: **accuracy: 0.5598**
  CCPCCPCCPFF: **accuracy: 0.5929**
  VGG16 : accuray: **accuracy: 0.6615**
  Ensemble Average: **accuracy: 0.6863**



Experimental Setup

1st experiment: CNN training and inference - State-of-Art:

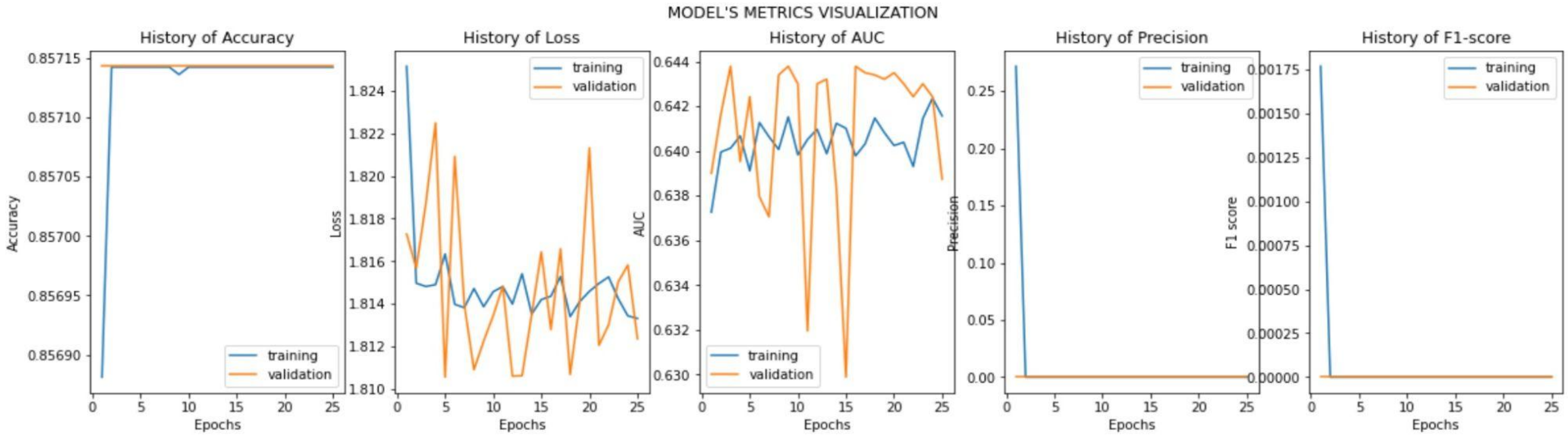
- 600 epochs
- learning rate = 0.1
- batch size = 128

2nd experiment: Networks: dataset preprocessing

- Illumination Correction

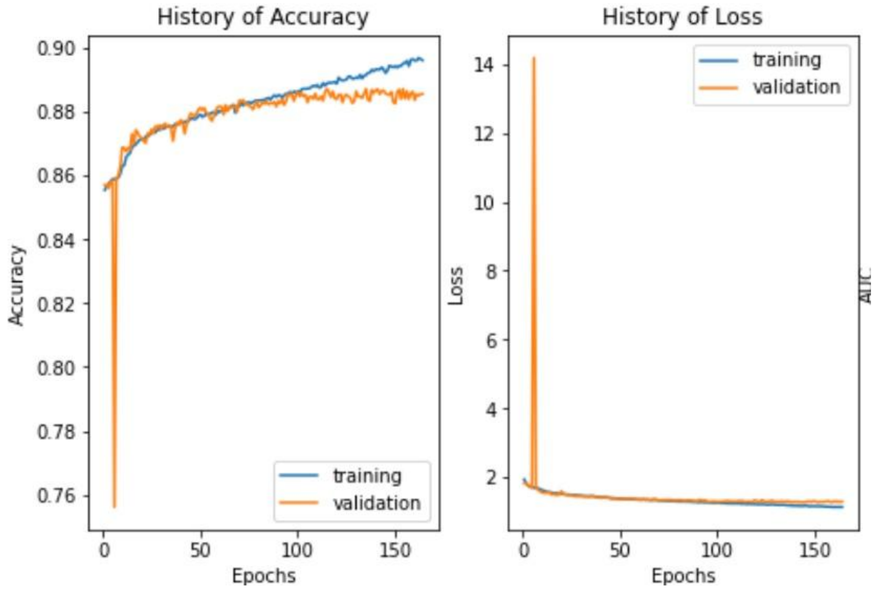
Experimental Results of

- 1ST EXPERIMENT
  - Early stop at epoch 25 & Model underfitting training dataset



Experiment - Illumination Correction & VGG16

- Early stopping at Epoch 164
- Loss: 1.1461
- Accuracy: 0.8912
- No Overfitting nor Underfitting



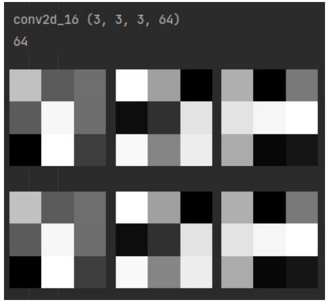
Feature visualization

- **What is feature map and filter visualization?**

  Visualizing the intermediate feature across different CNN layers to understand what happens inside CNN's to classify images.

- **Why do we need visualization?**
  1. Neural networks are like a black box, and learned features in a Neural Network are not interpretable. You pass an input image, and the model returns the results.
  2. Understanding the working of the model will help to know the reason for incorrect prediction that will lead to better fine tuning of the model and explain the decisions

Visualization details

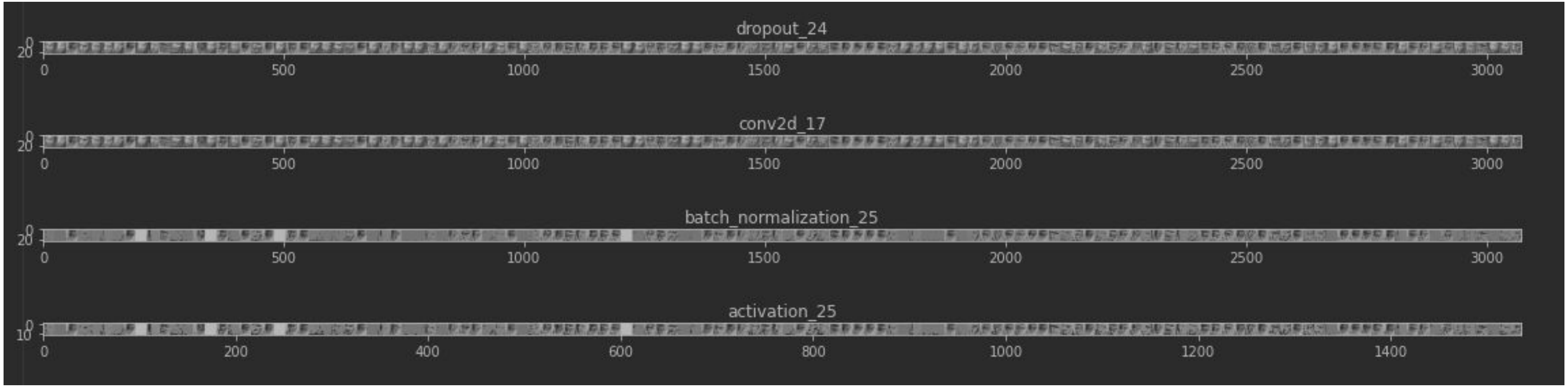


- **How we perform visualization?**
  - Visualizing Filters

    visualize the learned filters, used by CNN to convolve the feature maps
  - Visualizing feature maps

    called Activation Map, is obtained with the convolution operation, applied to the input data using the filter/kernel
- **Show steps and results**
  - using ***model.layers*** to iterate through all the layers
  - using ***get_weights()*** for that layer to extract the weights and bias values
  - Feature maps are generated by applying Filters or Feature detectors to the input image or the feature map output of the prior layers. Feature map visualization will provide insight into the internal representations for specific input for each of the Convolutional layers in the model.

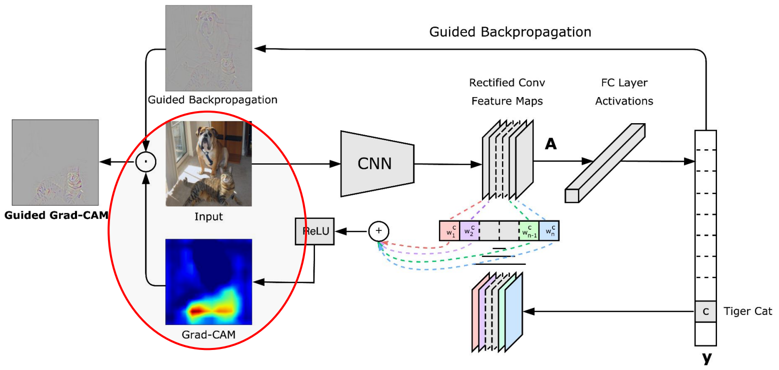
BOSTON UNIVERSITY



Explainability

- **Explainable AI** or **XAI** is an emerging field in machine learning that aims to address how the decisions of AI systems are made i.e. explain to humans how an AI system made a decision.
- Activations Visualization / Occlusion Sensitivity / …

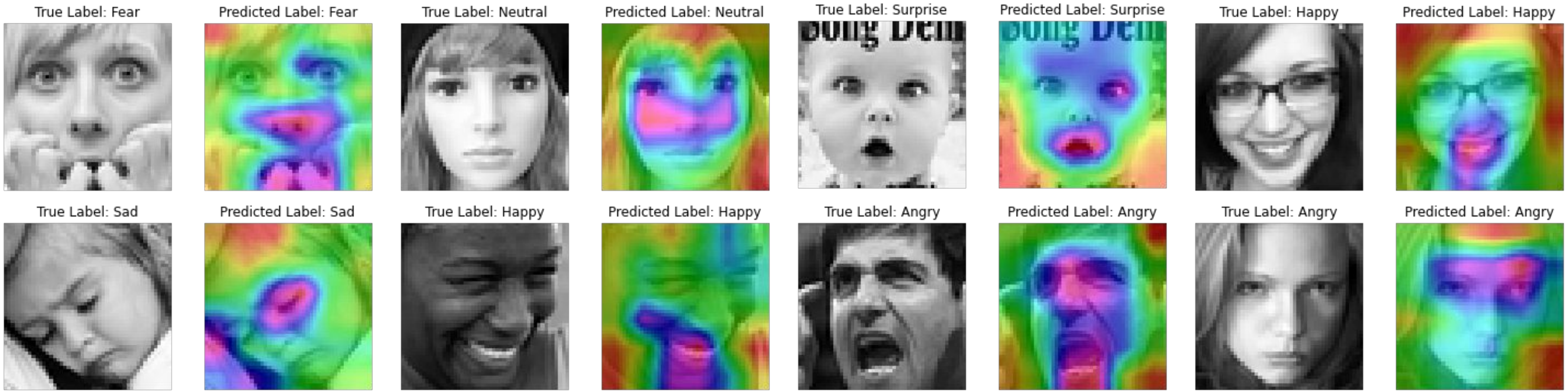
- **Gradient-Weighted Class Activation Mapping (Grad-CAM)**
    - Finding the final convolutional layer in the network
    - Examining the gradient information flowing into that layer
    - Computing an importance score based on the gradients to produce a heatmap
    - Highlighting the important regions within the image that resulted in a given class label



Interpretation details

- Correct predictions
- Mainly focusing on the eyes, the nose, the area under the eyes, and the mouth if the mouth is open or the teeth are exposed.

- I.e. happy (mouth), angry(eyebrows), surprise(eyes & mouth)

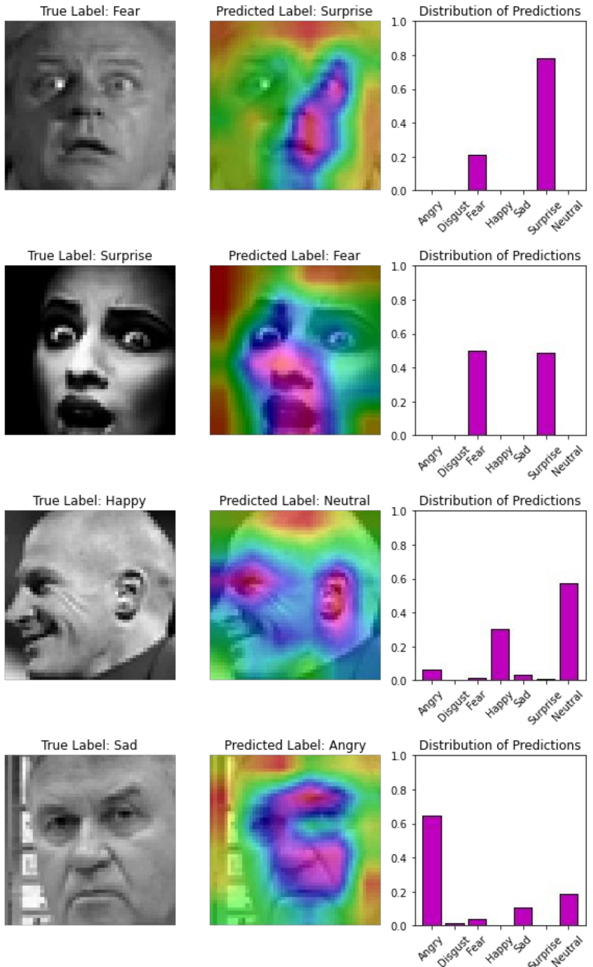


Interpretation details

- Wrong predictions and distributions

- Some faces have a mixed of many emotions
- Some faces don‘t show in the front
- Some emotions are hard to identify even for human being

- Using Grad-CAM to explain the model’s predictions also help us to understand why we can’t get a high accuracy on the FER2013 dataset (MNIST: >95%)



BOSTON UNIVERSITY

Conclusion

- Reproduce and fine-tune the model for FER2013 dataset
- Perform feature visualization on the model
- Show the explainability of the model
-
- Future steps:
    - Further improve accuracy
    - Study the bias that affect datasets related to FER problem
    - Evaluate the model on additional datasets (real-world questions)
    - ...

Github link

- https://github.com/BingquanCai/CS523_FinalProject.git

Thank you for listening!

Questions?

CS523 Final Project Team 2

Members:
Kaiyang Zhao (kyzhao@bu.edu)
Bingquan Cai (bqcai@bu.edu)
Bin Xu (xu842251@bu.edu)

Back-up slides following

References	Methods	Dataset	Accuracy
Zhou et al.[49]	CNN + MVFE-LightNet	FER2013	68.4%
Ziyang Yu et al.[50]	CNN + music algorithm	FER2013	62.1%
P. Ferandez et al.[51]	FERAtt	CK+	82.11%
N. Christou and N. Kanojiya[23]	CNN	FER2013	91%
F. Wang et al.[29]	EFDMs + EEG + CNN	EEG data on SEED	90.59%
		DEAP	82.84%
F. Nonis et al.[31]	3d approaches	BU-3DFE	60% to 90%
Ben Niu et al.[52]	SVM + LBP + ORB	JAFFE	88.5%
		CK+	93.2%
		MMI	79.8%
Ji-Hae Kim et al.[53]	LBP + deep neural network	CK+	96.46%
		JAFFE	91.27%
Sawardekar and Naik[54]	LBP + CNN	CK+	90%
Fei Wang et al.[28]	EEE based EFDMs	Cross datasets	82.84%
Hongli Zhang et al.[9]	CNN + image edge computing	FER2013 + LFW	88.56%
Ke Shan et al.[55]	KNN + CNN	JAFFE	76.74%
		CK+	80.30%
Pham and Quang[56]	CNN + FPGA	FER2013	66%
Guohang Zeng et al.[57]	Deep learning + handcrafted feature	CK+	97.35%
Shan Li and Deng[58]	Deep learning	All facial dataset	45% to 95%
Zuheng Ming et al.[59]	FaceLiveNet	FER2013	68.60%
		CK +	98%
Hussain and Balushi[35]	Deep learning	KDEF	88%
Smitha Rao et al.[32]	CNN + LSTM	CREMA-D	78.52%
		RAVDEES	63.35%
Khalid Bhatti et al.[60]	Deep features + extreme learning	JAFFE	92.4%
		CK	91.4%
		FER2013	62.7%
Proposed method	Fusion features (CNN + LBP + ORB) + ConvNet	FER2013	91.01%
		JAFFE	92.05%
		CK +	98.13%

BOSTON UNIVERSITY

BOSTON UNIVERSITY