Facial Expression Recognition

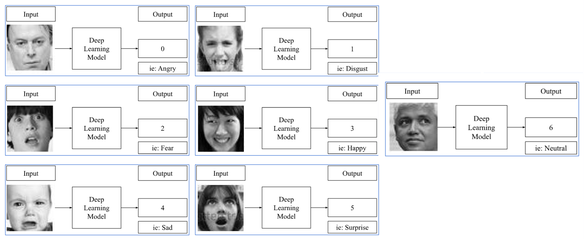
Malak Gaballa & Masa Tantawy

The American University in Cairo

Introduction

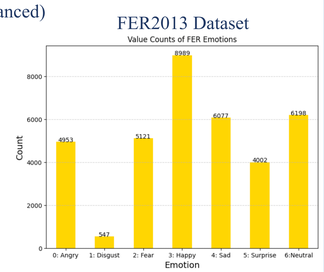
Given images of human faces showing different expressions, the model should be able to **categorize each image into one of 7 categories.**
0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral.

- Model input: image - vector of pixels for a 48x48 pixel grayscale image
- Model output: A number from 0 to 6 indicating the facial expression
- Evaluation: **weighted accuracy metric** to account for class imbalance in the data.



Datasets

- Selected Dataset: **FER2013**
  - 35,887 facial grayscale images (48x48 pixels), 63 MBs
  - 7 categories (highly imbalanced)
  - Has a test-train split.



- Other Datasets

AffectNet
  - 12,809 images, 5 GBs
  - 8 categories
  - Slightly balanced

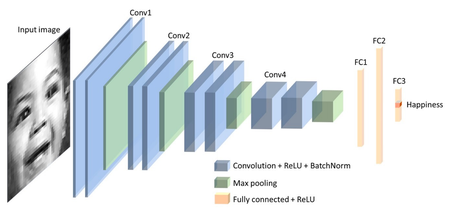
ExpW

Expression in-the-Wild Dataset
  - 91,793 images, 8 GBs
  - 7 categories

Baseline Model

The **VGGNet model**, short for Visual Geometry Group Network was **trained on FER2013** dataset achieving an **accuracy of 73.28%.** The framework of the model is **PyTorch**.

It is a classical CNN consisting of 4 convolutional stages (for feature extraction, dimension reduction, and non-linearity) and 3 fully connected layers (for classification).



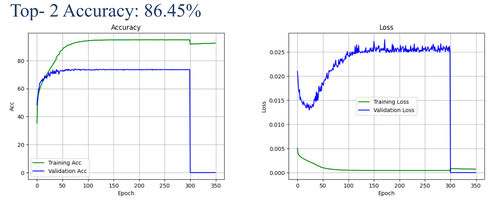
- Convolutional stage: 2 convolutional blocks & a max-pooling layer
- Convolution block: consists of a convolutional layer, a ReLU activation, and a batch normalization layer.
- The first 2 fully connected layers are followed by a ReLU activation. The third fully connected layer is for classification.

Baseline Model - Performance

- PyTorch:

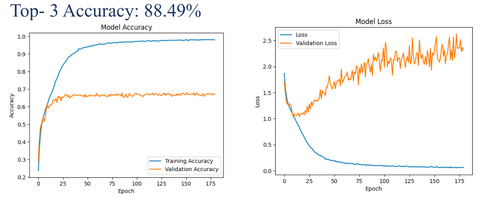
Original model, but the hyperparameters cannot be modified.

Epochs = 350
  - Top- 1 Accuracy: 73.27%
  - Top- 2 Accuracy: 86.45%



- Keras TensorFlow:

Using epochs = 180 instead of 350 due to GPU limit
  - Top-1 Accuracy: 65.76%
  - Top- 2 Accuracy: 79.91%
  - Top- 3 Accuracy: 88.49%



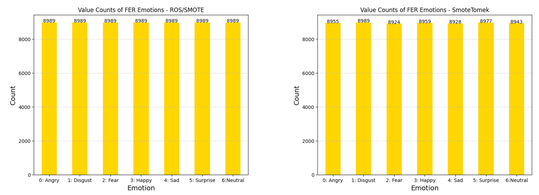
Methodology

- Hyperparameters Tuning

Different regularizers and optimizers with varying learning rates were experimented. The final modifications to the baseline model were:
  - **Adam Learning Rate = 0.0001** instead of 0.001
  - **Early Stopping** with patience = 10

- Data Imbalance Handling

The oversampling techniques - **Random Over Sampling** and **SMOTE** were used in addition to oversampling followed by undersampling techniques - **Smote + Tomek** and **Smote + ENN** were used.



ROS and SMOTE
A total of 62,923 images

SmoteTomek
A total of 62,675 images

Before being added to the model, each dataset was split in the same ratio as the original data (80% train ,10% test ,10% validation).

- Data Augmentation

A random balanced subset of the dataset has undergone different combinations (none, one, or multiple) of **horizontal flipping, rotation, and gaussian noise addition** with different ratios from given set ranges.

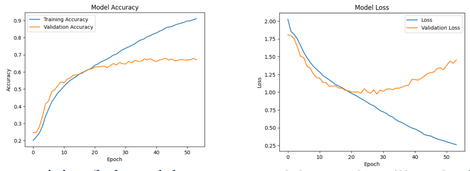
- Auxiliary Data

The **AffectNet Dataset** was used. It originally contained 8 categories of 96x96 coloured images. Before being added to the model, only the common 7 categories were selected, images were converted to grayscale and resized to 48x48.

Results

- Hyperparameters Tuning

Training stopped after 54 epochs.
  - Top-1 Accuracy: 66.15%
  - Top-2 Accuracy: 82.22%
  - Top-3 Accuracy: 90.89%



All extra training (balanced data, augmented data, and auxiliary data) was done on the hyper tuned model.

- Extra training on balanced data

	Random Over Sampling	SMOTE	SmoteTomek
Top-1 Accuracy	85.97%	87.18%	82.88%
Top-2 Accuracy	92.71%	93.88%	91.82%
Top-3 Accuracy	96.49%	96.84%	95.64%

- Extra training on augmented data

	Random Over Sampling	SMOTE	SmoteTomek
Top-1 Accuracy	59.74%	60.88%	58.29%
Top-2 Accuracy	79.91%	78.99%	77.46%
Top-3 Accuracy	88.63%	89.22%	86.77%

- Extra training on auxiliary data

	Random Over Sampling	SMOTE	SmoteTomek
Top-1 Accuracy	29.51%	30.31%	29.17%
Top-2 Accuracy	45.33%	44.58%	42.71%
Top-3 Accuracy	53.41%	54.25%	51.96%

None of the models showed overfitting.

Note: The model zoo for each model has been saved for next steps.

Discussion

Since the addition of auxiliary and augmented data declined the model's performance, an ensemble model of 3 distinct VGGNet models trained on balanced datasets generated through the different aforementioned resampling techniques was created.

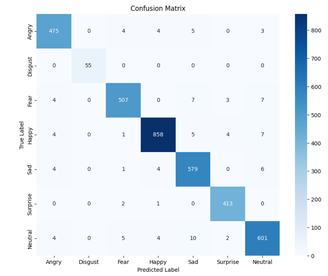
Each model was individually trained on the dataset, loaded as a trained model then defined as a component of the ensemble. By combining the outputs of these models and averaging them, ensemble's final output was generated.

Final model – Ensemble:

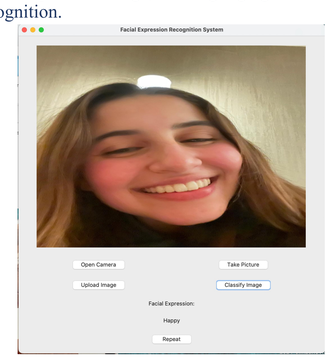
Facial Expression = Average of ROS, SMOTE, SmoteTomek

(augmented and auxiliary data excluded as they deteriorated the model)
  - Top-1 Accuracy: 97.18%
  - Top-2 Accuracy : 99.58%
  - Top-3 Accuracy : 99.72%

Final model – Ensemble: Confusion Matrix



A real-time application was developed allowing classification using the camera or from an uploaded image, offering high generalizability in facial expression recognition.



Conclusion & Future Work

Lessons Learnt

- Data imbalance handling significantly enhances the performance of the model.
- Constructing an ensemble model using multiple VGGNet models trained on balanced datasets further optimized performance.
- It is also concluded that extra training on auxiliary or augmented data may lead to worse performance of the model instead of enhancing it.
- The confusion matrix highlights Angry, Fear, and Neutral as the most challenging expressions to classify. This difficulty may arise from subtle facial differences or dataset imbalances.

Future Recommendations

- It is suggested to train the model on a more diverse and generalized database of facial expressions, such as Exp-W.
- Considering the inclusion of an 8th category of facial expression, such as *contempt* as seen in the AffectNet Dataset.
- Lastly, transitioning from grayscale to RGB images for input might yield better results.

References

1. FER2013 Dataset: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
2. AffectNet Dataset: https://www.kaggle.com/datasets/noamsegal/affectnet-training-data
3. VGGNet Repository: https://github.com/usef-kh/fer
4. VGGNet Research Paper: https://arxiv.org/pdf/2105.03588v1.pdf
5. Keras Tensorflow Model Source Code: https://colab.research.google.com/drive/1XiJ-sa5Kg324mpq_XG_JMWOlfj_DvZFv#scrollTo=FCly4_J8uwyv