COMP 6721 Applied Artificial Intelligence

Professor René Witte

Group Name:
NS 01

Group Members:
Dara Rahmat Samii (40281972)
Numan Salim Shaikh (40266934)
Shahab Amrollahibioki (40292670)

GitHub Link:
github.com/DaraSamii/A.I.ducation-Analytics

November 2023

Chapter 1

Data Pre-Processing

1.1 Dataset

1.1.1 Facial Expression Recognition(FER) 2013 Dataset

The primary dataset employed in this project is the FER2013 dataset[1], a pivotal resource in the realm of facial emotion recognition. This dataset was meticulously curated by Pierre-Luc Carrier and Aaron Courville, constituting a vital element within their ongoing research endeavor. The FER2013 dataset stands out due to its comprehensive nature and standardized structure.[1]

Comprising grayscale images, each measuring 48×48 pixels, the FER2013 dataset boasts an impressive total of **32,298** individual samples. This substantial dataset is a rich source for training and evaluating emotion recognition models.

Notably, the FER2013 dataset features meticulous image preprocessing. It ensures that each facial image is automatically aligned, meticulously centering the face and maintaining a consistent scale. This preprocessing greatly facilitates the extraction of essential facial features and enhances the efficiency of training emotion recognition models.

The FER2013 dataset categorizes emotions into seven distinct classes, each representing a unique spectrum of emotional expressions. These emotions encompass:

1. Angry
2. Disgust
3. Fear
4. Happy
5. Sad
6. Surprise
7. Neutral

1.1.2 FER+

In 2016 Emad Barsoum et al. in their paper "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution"[2] used Crowd-sourcing and 10 taggers to label each input image of FER2013, and compared four different approaches to utilizing the multiple labels.[2]

[1] dataset link: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
[2] Dataset link: https://github.com/Microsoft/FERPlus

Table 1.1: Sample of FER2013 Dataset

emotion	pixels
0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...
0	151 150 147 155 148 133 111 140 170 174 182 15...
2	231 212 156 164 174 138 161 173 182 200 106 38...
4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...
6	4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...

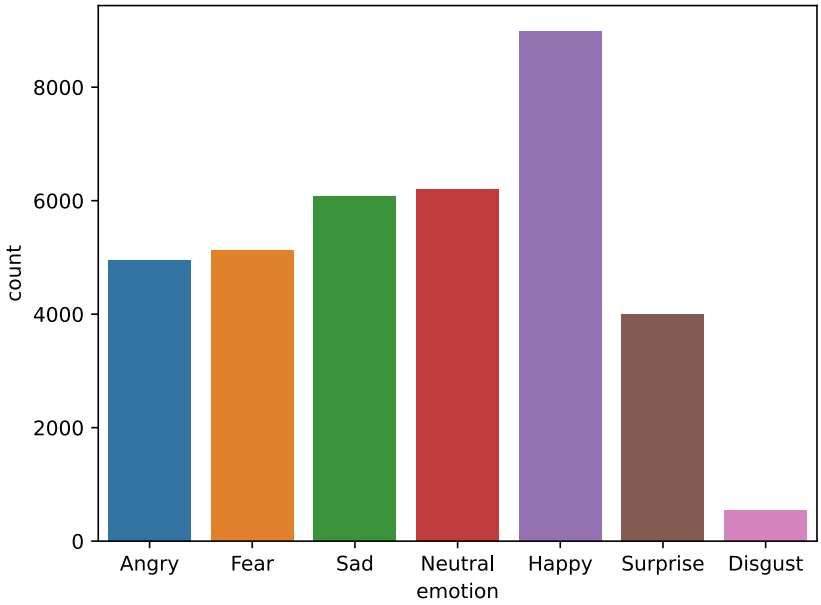


Figure 1.1: Number of images per each emotion in FER2013

Table 1.2: Sample of FER+ merged with FER2013

emotion	pixels	neutral	happiness	surprise	sadness	anger	disgust	fear	contempt	unknown	NF
Angry	70 80 ...	4	0	0	1	3	2	0	0	0	0
Angry	151 150 ...	6	0	1	1	0	0	0	0	2	0
Fear	231 212 8 ...	5	0	0	3	1	0	0	0	1	0
Sad	24 32 ...	4	0	0	4	1	0	0	0	1	0

The dataset contained the original number of samples, the difference was that emotions mentioned in the previous section in addition to "Not Face" and "Unknown" Labels were voted by 10 taggers and the score of each label had been recorded.

We merged two datasets to create more accurate dataset which it the intensity of each emotions can be measured by the number of votes each label had received by the taggers. For the project, Neutral and Angry labels can be used directly. Based on the scores for each emotion in FER+ and the ratio two other labels of 'Bored/Tired' and 'Engaged/Focused' can be extracted which the methodology will be described in the coming sections.

In the year 2016, Emad Barsoum and colleagues, in their paper titled "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution" [2], conducted a pioneering study that leveraged crowd-sourcing and engaged a team of ten taggers to label each input image within the FER2013 dataset.

This augmented dataset retained the original set of samples, with a notable enhancement: in addition to the emotions mentioned in the preceding section, namely "Not Face" and "Unknown," each of these emotions underwent a democratic evaluation by the ten taggers. The result was a meticulous recording of the score associated with each labeled emotion.

To enhance the dataset's accuracy, we combined the outcomes of two distinct datasets. The resulting amalgamation yielded a dataset wherein the intensity of each emotion could be precisely gauged based on the number of votes received by each label from the taggers.

Within the context of our project, the "Neutral" and "Angry" labels from this enhanced dataset can be directly employed. Moreover, by considering the scores attributed to each emotion within the FER+ dataset, we can derive two additional labels, specifically "Bored/Tired" and "Engaged/Focused." The subsequent sections will expound upon the methodology underpinning this extraction process in greater detail.

1.2 Data Cleaning

1.2.1 Eliminating Images Labeled as 'Not Face'

In the enhanced dataset, a subset of images was labeled as 'Not Face.' These images received various scores from the taggers, and their distribution is as follows:

- The number of images with a score $NF = 10$ is 176.
- The number of images with a score $NF = 4$ is 2.
- The number of images with a score $NF = 2$ is 4.
- The number of images with a score $NF = 1$ is 167.

A visual representation of some images labeled as 'Not Face' by the taggers can be observed in Fig. 1.2 .

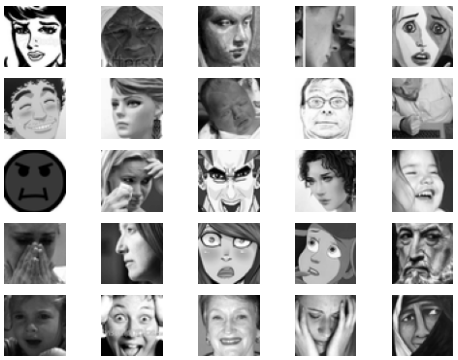
Given that the total number of images with a non-zero NF score is relatively small in comparison to the overall dataset size, it was deemed prudent to remove all images with a non-zero NF score from the dataset.

1.2.2 Eliminating Images Labeled as 'Unknown'

A similar curation process must also be applied to images labeled as 'Unknown.' The distribution of 'unknown' scores for these images is detailed below:

- number of images with score unknown=8 is 3.

(a) Images with NF=10



(b) Images with NF=2

Figure 1.2: Images with "Not Face" score of non-zero

- number of images with score unknown=7 is 3.
- number of images with score unknown=6 is 18.
- number of images with score unknown=5 is 55.
- number of images with score unknown=4 is 224.
- number of images with score unknown=3 is 751.
- number of images with score unknown=2 is 2526.
- number of images with score unknown=1 is 8220.

(a) Images with unknown=6

(b) Images with unknown=5

Figure 1.3: Images with "unknown" score of non-zero

In view of the substantial number of images with non-zero 'Unknown' scores, a decision was reached after careful consideration. It was determined that images with a 'Unknown' score of 5 or greater should be excluded from the dataset. This action aims to maintain the dataset's integrity and coherence for subsequent analysis and machine learning model development.

For a visual representation of the images with non-zero 'unknown' scores, please refer to Figure 1.3, which provides insight into the selection criteria employed for image removal.

1.3 Labeling

In the context of this project, it is imperative to classify images into four distinct labels: Anger, Neutral, Bored, and Focused. While the original dataset inherently includes the emotions Anger and Neutral, it lacks explicit representations for Bored and Focused emotions. This chapter elucidates the methodology employed to extract these target labels from the original emotion scores.

1.3.1 Anger

The process of labeling an image as 'Angry' based on the emotion scores is relatively straightforward. An image is labeled as 'Angry' if the score for anger surpasses the scores of all other emotions, signifying it as the predominant emotion. However, to account for rare cases where all emotion scores are equal, an additional criterion is established: the score for anger must be at least 2, ensuring a minimal threshold for labeling an image as 'Angry.'

1.3.2 Neutral

The labeling of an image as 'Neutral' hinges on the emotion scores, with 'Neutral' being assigned if the score for neutrality holds the highest value among all the emotions.

1.3.3 Focused

The criteria for labeling an image as 'Focused' were derived from a comprehensive analysis of facial expressions that convey focus. Notably, these expressions exhibit a lack of sadness, and the neutrality score should outweigh the scores of other emotions.

1.3.4 Bored

The criteria for labeling an image as 'Bored' are rooted in the observation that when individuals are bored, they typically do not exhibit happiness. Furthermore, boredom is unaccompanied by fear. In cases where a person may appear slightly angry, the criterion is that the level of anger should be lower than the expressed level of sadness in the facial expression.

1.4 Dataset Visualization

After meticulous data cleaning, removal of irrelevant data, and the extraction of the desired emotion labels, the dataset is now composed of the following:

- Number of samples labeled as 'Neutral': 3789
- Number of samples labeled as 'Angry': 3954
- Number of samples labeled as 'Focused': 4553
- Number of samples labeled as 'Bored': 3960

The bar-plot of the plot's per emotion is shown in Fig. 1.4

Table 1.3: Summary of criterion's for labeling images based on emotion's scores

Label	criteria
Anger	• Anger score > other emotion's score • Anger score > 2
Neutral	• Neutral score > other emotion's score • Neutral score > 6
Focused	• sadness score == 0 • anger score == 0 • neutral score > other emotion's score
Bored	• happiness score == 0 • fear score == 0 • sadness score $\neq$ 0 • neutral score $\neq$ 0 • anger score + 2 < sadness score

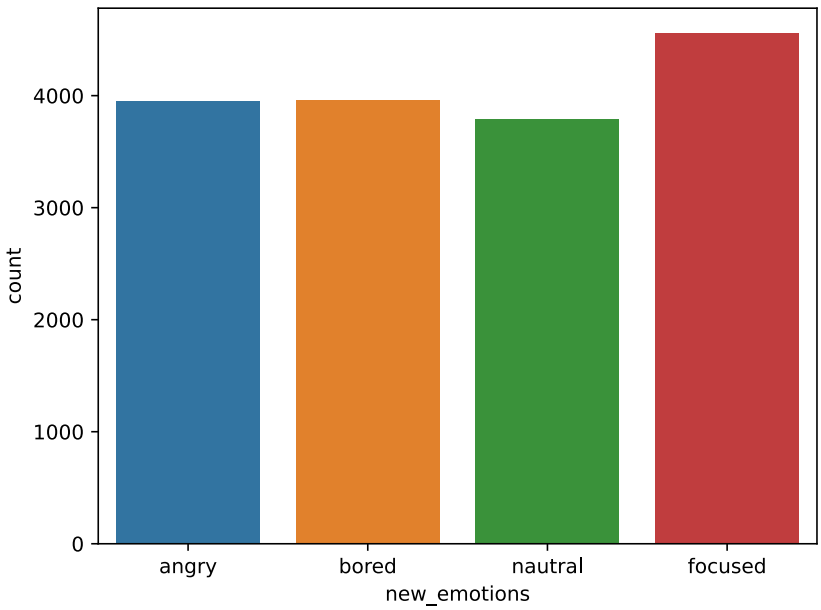
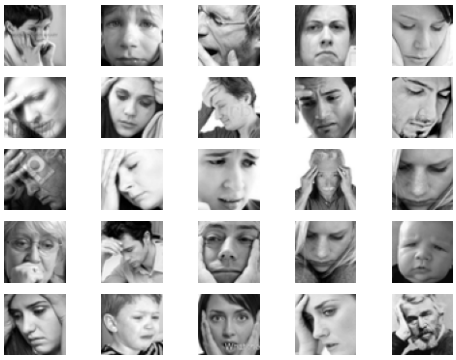


Figure 1.4: Bar-plot of images per desired emotion

(a) Samples of images labeled as 'Angry'



(b) Samples of images labeled as 'Bored'

(c) Samples of images labeled as 'Neutral'

(d) Samples of images labeled as 'Focused'

Figure 1.5: Samples of the cleaned dataset

Chapter 2

Training Models

In Section 1, we created a dataset consisting of 14,831 samples. The data split ratios were chosen as 70%, 15%, and 15% for the training, validation, and test sets, respectively. This resulted in 10,382 samples for training, 2,225 samples for validation, and 2,224 samples for testing.

To enhance the diversity of our training data and improve the robustness of the model, we applied two augmentation techniques during training:

- **Horizontal Flip:** Images were horizontally flipped with a probability of 0.5. This augmentation helps the model learn features invariant to left-right orientation changes.
- **Random Rotation:** Images were randomly rotated between -10 and 10 degrees. This augmentation introduces variations in object orientations, aiding the model in becoming more rotation-invariant.

2.1 CNN architecture

Implementation of the models is conducted in the PyTorch library[3]. 12 distinct CNN-based models are introduced to proficiently categorize images into 4 classes (Angry, Focused, Bored, Neutral). The architectural share common hyperparameters, including learning rate, training epochs, batch size, optimizer, and loss function, meticulously detailed in Table 2.1.

Table 2.1: Hyper Parameters used in Training

Parameter	Values
Learning Rate	0.001
Epochs	100
Batch Size	64
Optimizer	Adam
Loss Function	Cross-Entropy
Weight Decay	0.0001

The overall architecture of the models, illustrated in Fig. 2.2, comprises three fundamental components:

1. **ResBlocks:** These form the initial segment of the models and were configured with depths of 4 and 8 layers. Convolutional layers within the ResBlocks utilized two configurations of 16 and 32 channels.

   For the ResBlock configuration, kernel sizes of (3×3), (5×5), and (7×7) were employed. Corresponding padding values of 1, 2, and 3 were used to ensure the output dimensions matched the input dimensions. Convolutional blocks within the ResBlocks were set to a striding of 1.

   - **Kernel Sizes and Padding:**
     - (3×3): Padding was set to 1.

- (5×5): Padding was set to 2.
- (7×7): Padding was set to 3.

2. **Average Pooling and Flattening:** Following the ResBlocks, the output underwent Average Pooling to achieve a final size of (20×20).

3. **Fully Connected Layers:** The final part of the model consisted of fully connected layers. The number of these layers was dynamically correlated with the number of ResBlocks, always maintaining a ratio of one-third of the ResBlocks, resulting in a range of 1 to 3 layers.

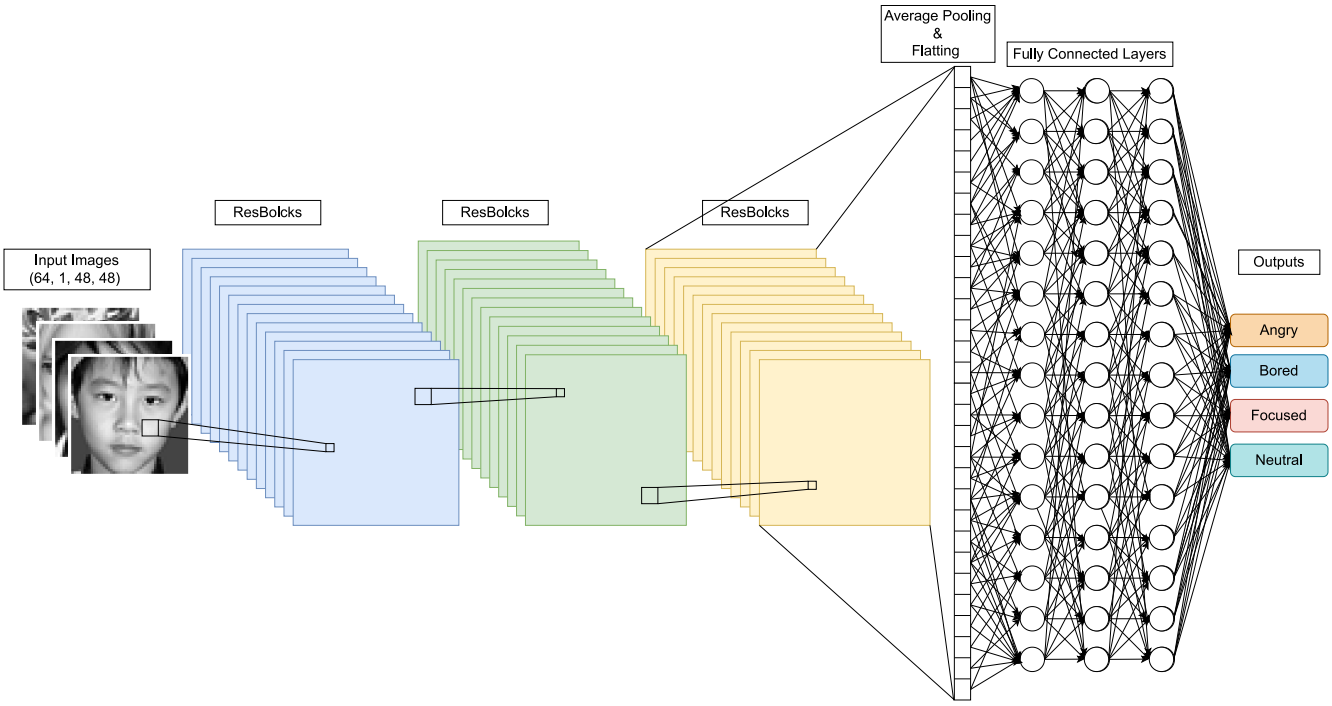


Figure 2.1: Schematic of Model Architecture

2.1.1 Resblock

Residual Networks, commonly known as ResNets, have become a cornerstone in deep learning architectures, particularly for image recognition tasks. At the heart of ResNets are residual blocks, a fundamental building block designed to address the challenge of training very deep neural networks[4]. A residual block introduces a shortcut or a "skip connection" that allows the network to learn residual functions. Mathematically, given an input x, the output of a residual block can be represented as $F(x) + x$, where $F(x)$ is the learned residual. This architecture facilitates the flow of information through the network, mitigating the vanishing gradient problem and enabling the training of exceedingly deep networks.

2.1.2 Convolution Layers

Each convolution layer systematically applies convolution operations to input data, sequentially transmitting results to subsequent layers.

2.1.3 Activation Functions

Rectified Linear Unit (ReLU) is a widely used activation function in neural networks. It is defined as $f(x) = \max(0, x)$, keeping positive values unchanged and setting negative values to zero. ReLU introduces non-linearity to

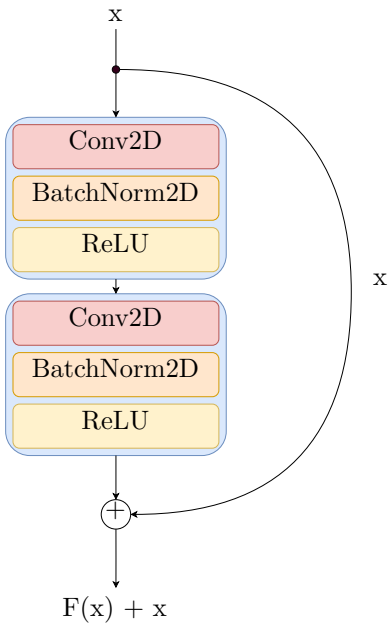


Figure 2.2: Resblock

neural networks, aiding in learning complex patterns. Its simplicity and ability to address the vanishing gradient problem make it a popular choice.

2.1.4 BatchNorm2D

Batch Normalization (BatchNorm) is a widely used technique in deep learning to improve the training stability and convergence of neural networks. It operates by normalizing the input of a layer across a mini-batch of data. For a given mini-batch of size m, the BatchNorm operation for a specific feature x can be expressed as follows:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2.1}$$

where $\hat{x}$ is the normalized input, μ is the mean, σ^2 is the variance, and ϵ is a small constant to avoid division by zero.

2.1.5 Pooling Layers

The inclusion of pooling layers is indispensable, serving to reduce the dimensions of feature maps. This reduction minimizes trainable parameters, expediting computations without compromising crucial features. Additionally, pooling layers contribute significantly to averting overfitting risks in expansive networks, bolstering overall robustness and generalization in facial expression detection models.

2.1.6 Flattening Layer

Following convolutional and pooling operations, a Flatten layer is introduced in the model architecture. This critical layer transforms entire filter maps into a flattened representation, providing a comprehensive set of features to the subsequent classifier.

2.1.7 Fully-Connected Layers

In a fully-connected layer, each neuron or node is connected to every neuron in the previous layer, forming a dense matrix of weights. Given an input vector x and weight matrix W, the output y of a fully-connected layer can be expressed as:

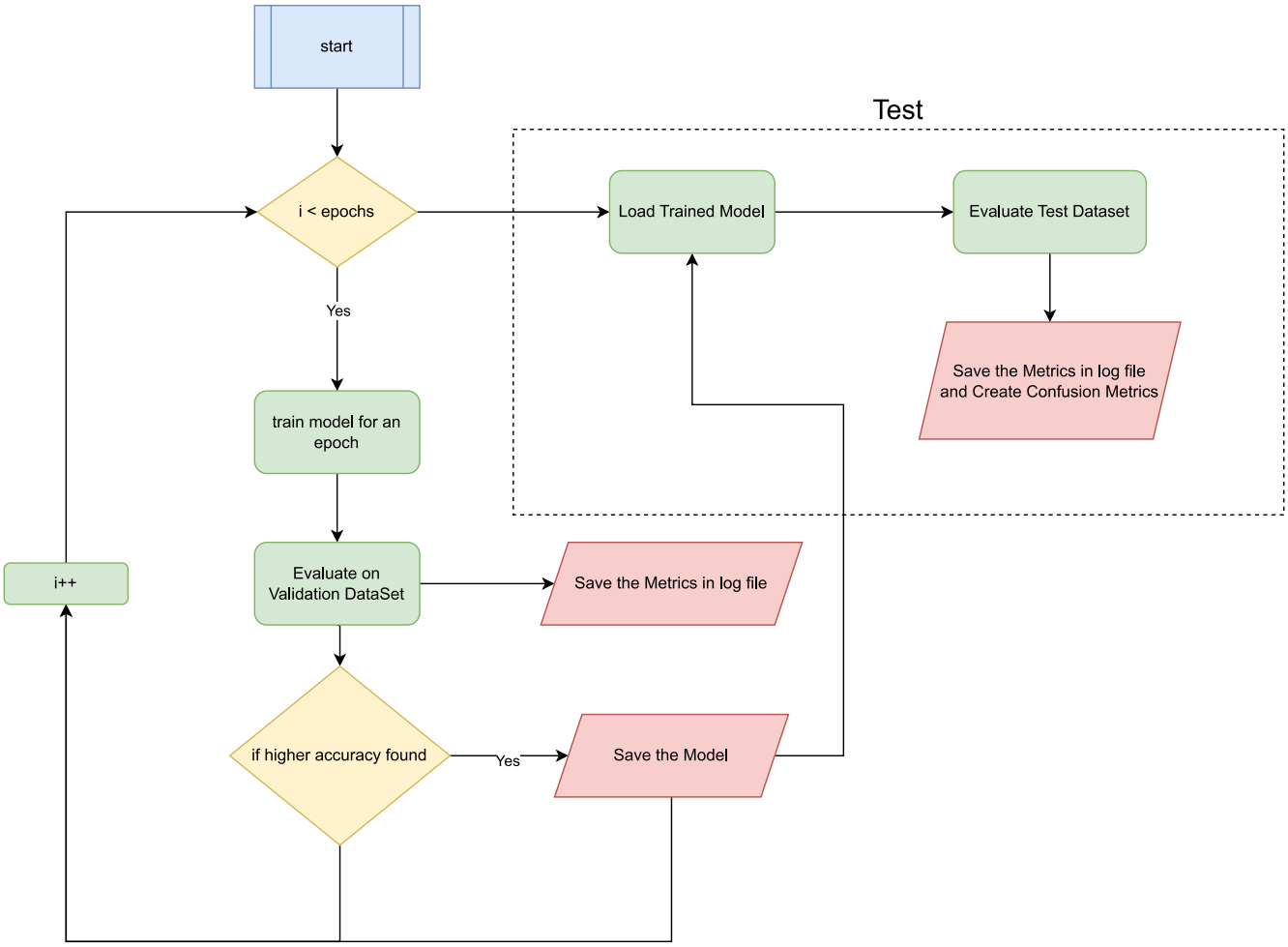


Figure 2.3: Flow-chart of training and evaluation of the models

$$y = \sigma(Wx + b) \tag{2.2}$$

where W is the weight matrix, b is the bias vector, and σ is the activation function. The matrix multiplication Wx captures the weighted sum of inputs, and the bias term b is added.

2.2 Training Process

After creating a Training Data Loader, Validation DataLoader and Test Data Loader, a Learner class was created with couples the data loader and model. the learner handles traning of the model, logging the metrics, and loss and saves and load the trained models specified. The Learner fallacy is taken from Fast.ai[5]. each model was trained by train dataset an epoch, followed by evaluted the model with validation dataset and if a new accuracy was found the model would be saved. An overall flow-chart of the training and evaluating process is shown in Fig.2.3.

2.3 Metrics

The performance of the model is evaluated using following metrics:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{2.3}$$

Table 2.2: Summary of Model Architecture and the represented Accuracy

Model Name	K Size	Block	Channels	FCs	Accuracy	Recall	Precision	F1 Score
B16_N4_FC1_K3_AP20	3	4	16	1	0.582472	0.581015	0.599142	0.585643
B32_N4_FC1_K3_AP20	3	4	32	1	0.59236	0.595532	0.622576	0.596638
B16_N8_FC2_K3_AP20	3	8	16	2	0.604494	0.600417	0.620104	0.59736
B32_N8_FC2_K3_AP20	3	8	32	2	0.622022	0.618432	0.650313	0.622679
B16_N4_FC1_K5_AP20	5	4	16	1	0.596404	0.598403	0.612813	0.595999
B32_N4_FC1_K5_AP20	5	4	32	1	0.59236	0.595048	0.629317	0.590056
B16_N8_FC2_K5_AP20	5	8	16	2	**0.623371**	0.621407	0.654631	0.611707
B32_N8_FC2_K5_AP20	5	8	32	2	0.612135	0.615304	0.652023	0.613317
B16_N4_FC1_K7_AP20	7	4	16	1	0.58382	0.580199	0.619654	0.579667
B32_N4_FC1_K7_AP20	7	4	32	1	0.577978	0.582606	0.592769	0.584087
B16_N8_FC2_K7_AP20	7	8	16	2	0.617978	0.611103	0.627087	0.609395
B32_N8_FC2_K7_AP20	7	8	32	2	0.607191	0.610399	0.651253	0.608782

Accuracy measures the overall correctness of the model's predictions.

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{2.4}$$

Precision represents the accuracy of the positive predictions made by the model.

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{2.5}$$

Recall, also known as sensitivity, gauges the model's ability to capture all positive instances.

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

The F1 score is the harmonic mean of precision and recall, providing a balanced assessment of the model's performance.

In the above equations, T_P represents True Positive, T_N represents True Negative, F_P represents False Positive, and F_N represents False Negative.

2.4 Evaluation

2.4.1 Hyper-paremeter's effect

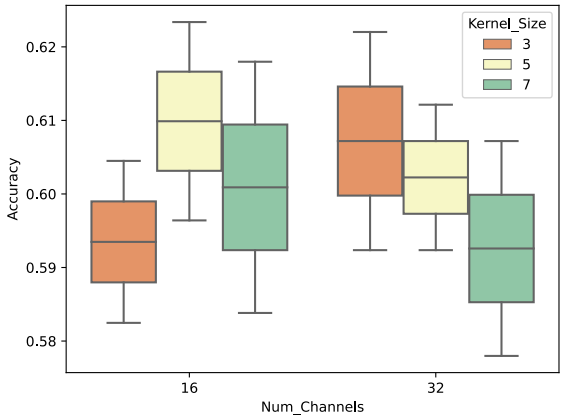
As illustrated in Fig. 2.4c and Fig. 2.4d, accuracy trends come to the forefront, highlighting the superiority of models kernel size of(5×5), over alternative sizes of (3×3) and (7×7).

Examining Fig. 2.4e and Fig. 2.4f, a positive correlation emerges between an increased number of residual blocks and heightened accuracy. Notably, models with 8 ResBlocks outperform those with 4 ResBlocks.

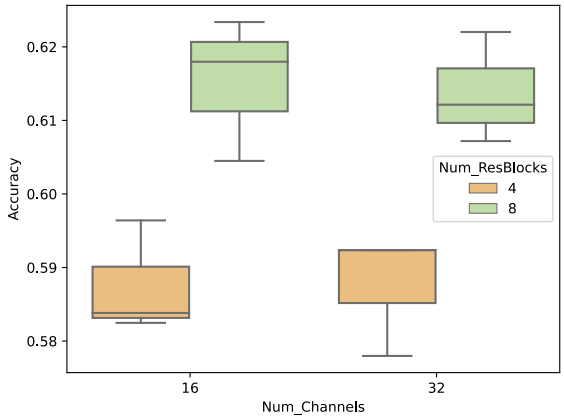
Concerning the number of channels within each ResBlock, no significant correlation is observed with accuracy. In Fig. 2.4c, the relationship is not straightforward, as models with a kernel size of (5×5) and 16 channels achieve higher accuracy compared to those with 32 channels. However, this trend does not hold for kernel size (3×3).

Remarkably, our model configuration B16_N8_FC2_K5_AP20 stands out as the pinnacle of performance, boasting an accuracy of 0.623371. The complete model architectures and respective accuracy values for each model are presented in Table 2.2.

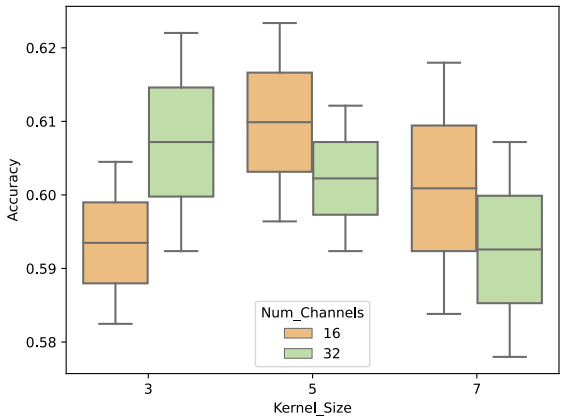
2.4.2 Confusion Matrix Analysis



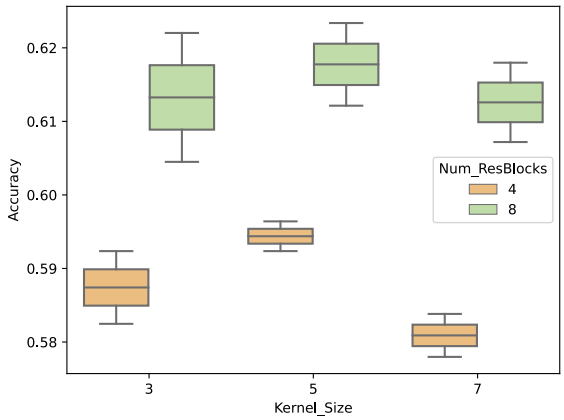
(a) Accuracy scores compared to ResBlock number of Channels and Kernel Sizes



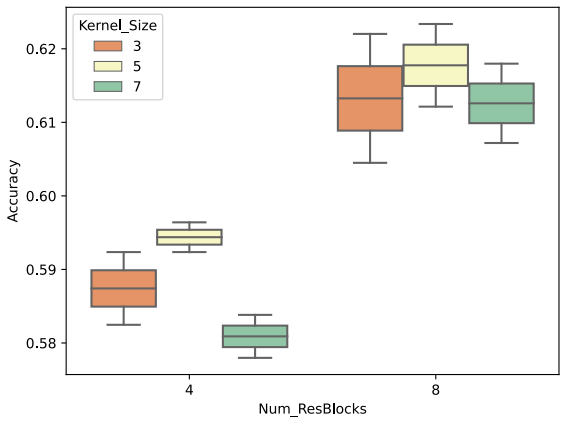
(b) Accuracy scores compared to ResBlock number of channels and number of ResBlocks



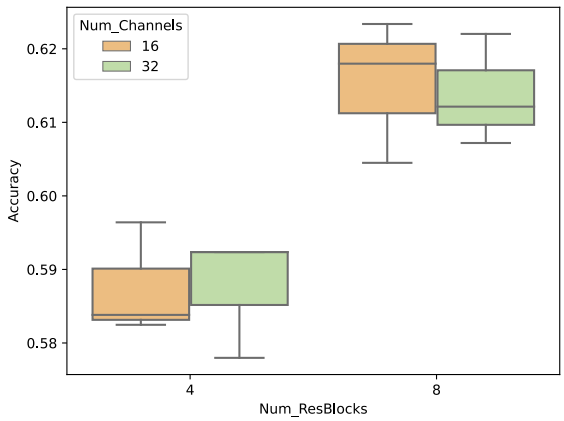
(c) Accuracy scores compared to Kernel Sizes and ResBlock number of Channels



(d) Accuracy scores compared to Kernel Sizes and number of ResBlocks

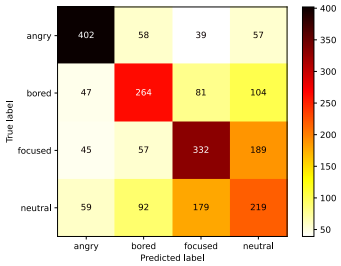


(e) Accuracy scores compared to Kernel Sizes and number of ResBlocks

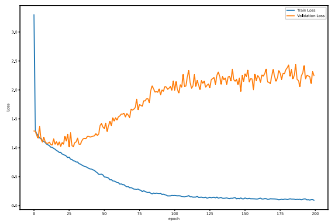


(f) Accuracy scores compared to ResBlock's number of channels and number of ResBlocks

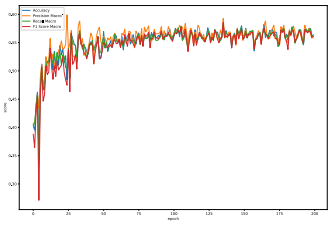
Figure 2.4: Overall Caption for All Figures



(a) Confusion Matrix on Test

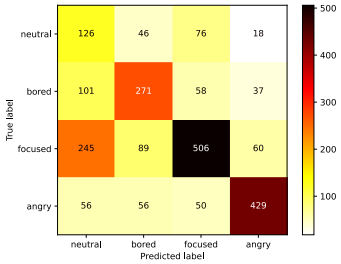


(b) Loss

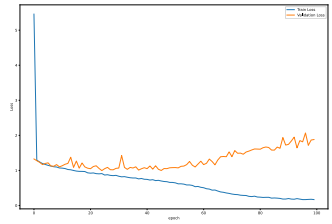


(c) Metrics Macro

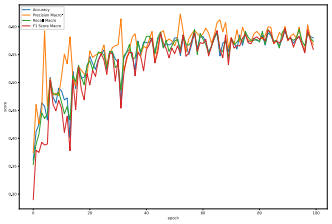
Figure 2.5: Caption derived from the upper directory: B16_N4_FC1_K3_AP20



(a) Confusion Matrix on Test

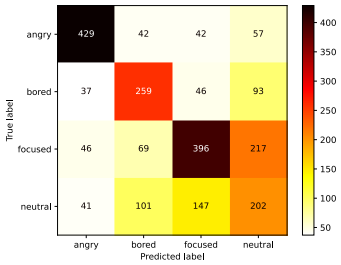


(b) Loss

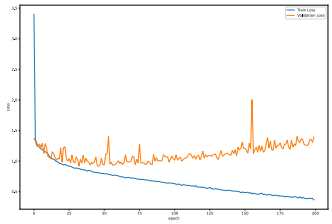


(c) Metrics Macro

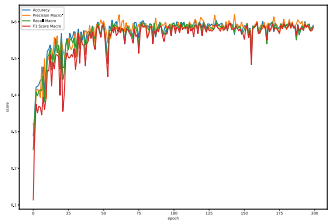
Figure 2.6: Caption derived from the upper directory: B32_N4_FC1_K3_AP20



(a) Confusion Matrix on Test

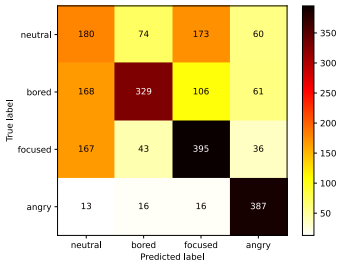


(b) Loss

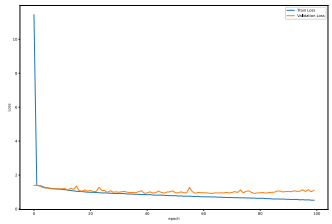


(c) Metrics Macro

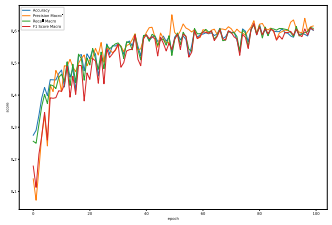
Figure 2.7: Caption derived from the upper directory: B16_N8_FC2_K3_AP20



(a) Confusion Matrix on Test

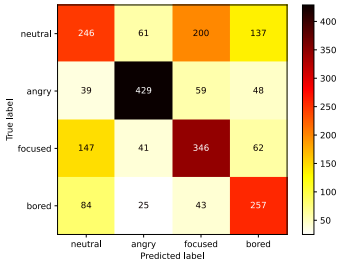


(b) Loss

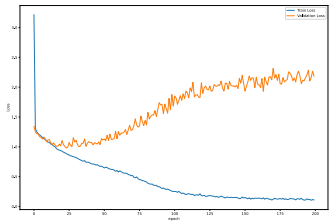


(c) Metrics Macro

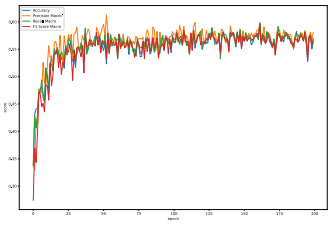
Figure 2.8: Caption derived from the upper directory: B32_N8_FC2_K3_AP20



(a) Confusion Matrix on Test

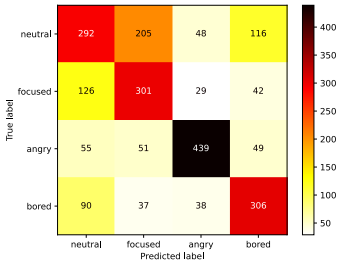


(b) Loss

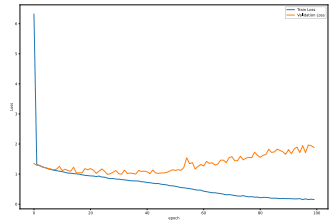


(c) Metrics Macro

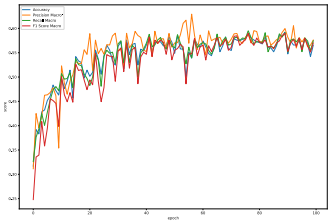
Figure 2.9: Caption derived from the upper directory: B16_N4_FC1_K5_AP20



(a) Confusion Matrix on Test

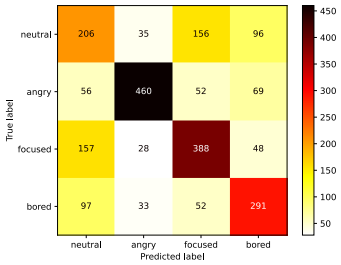


(b) Loss

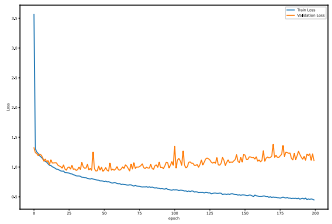


(c) Metrics Macro

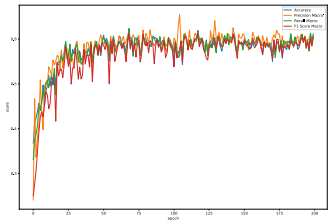
Figure 2.10: Caption derived from the upper directory: B32_N4_FC1_K5_AP20



(a) Confusion Matrix on Test

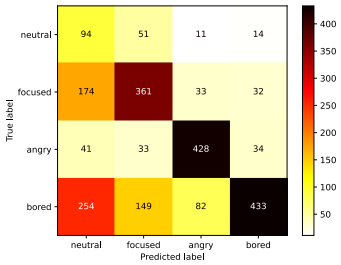


(b) Loss

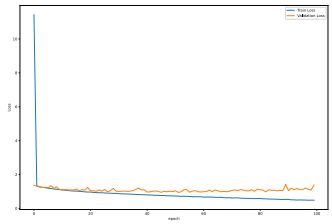


(c) Metrics Macro

Figure 2.11: Caption derived from the upper directory: B16_N8_FC2_K5_AP20



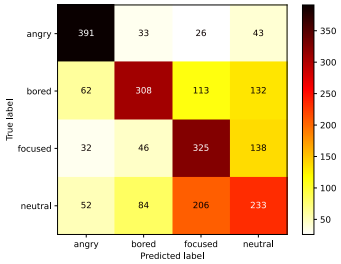
(a) Confusion Matrix on Test



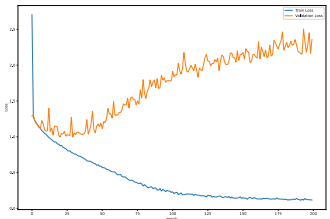
(b) Loss

(c) Metrics Macro

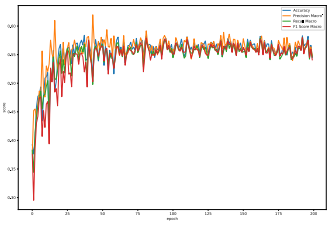
Figure 2.12: Caption derived from the upper directory: B32_N8_FC2_K5_AP20



(a) Confusion Matrix on Test

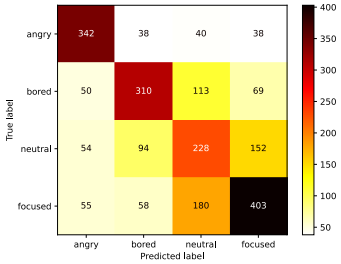


(b) Loss

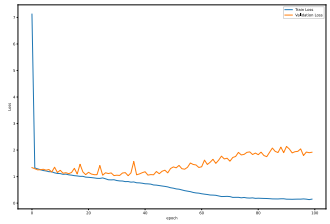


(c) Metrics Macro

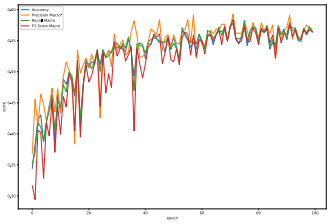
Figure 2.13: Caption derived from the upper directory: B16_N4_FC1_K7_AP20



(a) Confusion Matrix on Test

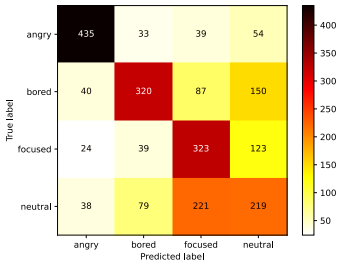


(b) Loss

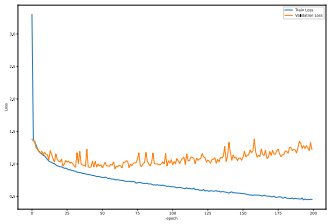


(c) Metrics Macro

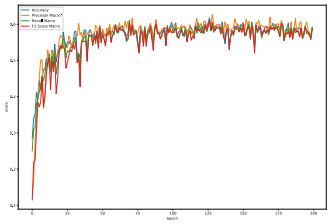
Figure 2.14: Caption derived from the upper directory: B32_N4_FC1_K7_AP20



(a) Confusion Matrix on Test

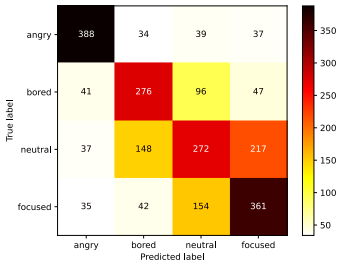


(b) Loss

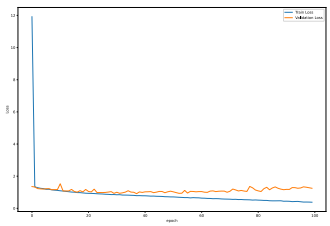


(c) Metrics Macro

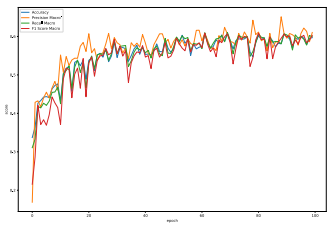
Figure 2.15: Caption derived from the upper directory: B16_N8_FC2_K7_AP20



(a) Confusion Matrix on Test



(b) Loss



(c) Metrics Macro

Figure 2.16: Caption derived from the upper directory: B32_N8_FC2_K7_AP20

The Confusion Matrix for all models reveals that they generally perform well in predicting "Angry" images, achieving high accuracy. This is attributed to the distinct features of angry faces, such as large eyes, open mouths, and an aggressive facial expression.

Furthermore, the models demonstrate proficiency in correctly classifying "Bored" images, as the bored facial expression, characterized by downcast eyes and closed, straight mouths, is easily distinguishable.

However, challenges arise in distinguishing between "Focused" and "Neutral" expressions. This is reasonable, considering that both expressions involve eyes staring at the camera and closed lips. Even at a human level, differentiation between focused and neutral faces proves challenging.

Examining the Loss plots, a notable trend emerges: as the model's layer count decreases, there is an increased risk of overfitting. With each training epoch, the validation loss consistently rises. Conversely, as the model size increases, the validation loss remains stable and fluctuates less compared to models with fewer layers.

2.5 Conclusions and Forward Look:

Addressing bias is imperative. Our dataset exhibits bias toward gender, age, and race, emphasizing the need for a balanced dataset for enhanced real-time application performance.

Furthermore, exploring advanced models like ResNets, R-CNNs, Fast RCNNs, Faster R-CNN, and YOLO could further enhance overall model performance.

Additionally, better labeling of the dataset and increasing the quality of the dataset might be significantly helpful.

Finally, the key notes of the model comparison are as follows:

- Models with (5×5) kernel sizes outperform (3×3) and (7×7).
- Positive correlation observed between ResBlocks and accuracy; 8 ResBlocks outperform 4.
- No significant correlation found between channel count in ResBlocks and accuracy.
- Model `B16_N8_FC2_K5_AP20` excels with an accuracy of 0.623371.
- Confusion Matrix shows strong performance in predicting "Angry" and "Bored" expressions.
- Difficulty distinguishing between "Focused" and "Neutral" expressions.
- Loss plots indicate increased overfitting risk with fewer layers; larger models show more stability.

Acknowledgements

This project was undertaken as part of the requirements for COMP 6721: Applied Artificial Inteligence, lectured by Professor Rene Witte at Concordia University, Montreal, Canada. The report was meticulously prepared using the LaTeX typesetting language on the Overleaf platform. Additionally, schematic diagrams were crafted using draw.io.Compute Canada Alliance, Cedar Computer located in Simon Fraser University, Vancouver, Canada, was used to perform High Performance Computings and training the models.

Bibliography

[1] W. C. Y. B. Dumitru, Ian Goodfellow, "Challenges in representation learning: Facial expression recognition challenge," 2013.

[2] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proceedings of the 18th ACM international conference on multimodal interaction*, pp. 279–283, 2016.

[3] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[5] J. Howard *et al.*, "fastai." https://github.com/fastai/fastai, 2018.