# Facial Emotion Detection - Capstone Project

## MIT-PE: Applied Data Science Program


Author: Atif Saleem

Date: July 10, 2024

Course: Applied Data Science Program by MIT Professional Education

# Executive Summary

Facial emotion detection and/or recognition (aka FER) is a computer vision process aimed at detecting and classifying human emotional expressions. FER systems are currently used in a vast range of applications from areas such as education, healthcare, or public safety; therefore, detection and recognition accuracies with lower losses are very important.

## Key Takeaways

This project focused on developing a robust facial emotion detection system capable of classifying grayscale images of faces into four primary emotions: happy, sad, surprise, and neutral. The goal was to explore various deep learning models and identify the most effective approach, with high accuracy and low loss, for this task, ultimately enhancing human-computer interaction in diverse applications.

Key findings include:

- **Grayscale Focus:** Due to the nature of the dataset, models trained on grayscale images demonstrated superior performance compared to those trained on RGB images. Overall, CNN models performed better than pre-trained models with transfer learning architectures. We could observe higher prediction precision for **"Happy" (87%)** and **"Surprise" (82%)** classes in the classification report and confusion matrix of final chosen model *(Please, see Appendix-3 for classficiation report and confusion matrix of "**complex_cnn_model_1**")*.

- **Complex CNN as Optimal Solution:** After evaluating various deep learning models, including transfer learning architectures (VGG16, ResNet50V2, EfficientNetV2B2) and traditional CNNs, a custom-designed complex CNN model (**'complex_cnn_model_1'**) emerged as the most effective solution. This model achieved the highest accuracy and loest loss on the validation set while maintaining reasonable computational efficiency. (*Please, see Appendix-2 for models performance comparison and Appendix-3 for its architecure, compilation, and trainig code snippets, and for accuracy and loss plots.)*

- **Data Augmentation and Hyperparameter Tuning:** These techniques played a crucial role in achieving optimal model performance. The data was already split into about 75/25 to reduce baises in the dataset *(Please, see Appendix-1)*. Data augmentation increased the diversity of the training data, improving the model's generalization capabilities. Careful hyperparameter tuning, including adjusting learning rate, batch size, optimization algorithms, downsampling, and regularization techniques, further enhanced model accuracy.

## Key Next Steps

To fully leverage the potential of this facial emotion detection system and maximize its impact, stakeholders should consider the following next steps:

**Model Deployment:**

- **Cloud-Based API:** Deploy the model as a REST API on cloud platforms (e.g., Google Cloud, AWS) to provide scalable and accessible emotion detection capabilities to various applications.
- **Edge Devices:** Explore deploying the model on edge devices (e.g., smartphones, embedded systems) for real-time emotion detection in privacy-sensitive or low-latency scenarios.

**Model Optimization:**

- **Quantization:** Investigate model quantization techniques to reduce the model size and improve inference speed, enabling efficient deployment on resource-constrained devices.
- **Fine-tuning with a Larger Dataset:** Collect or utilize a larger, more diverse dataset to fine-tune the model, further enhancing its accuracy and generalization to real-world scenarios.

**Future Research and Development:**

- **Expanded Emotion Recognition:** Explore the possibility of recognizing a broader range of emotions beyond the basic four, enabling more nuanced human-computer interaction.
- **Bias Mitigation:** Analyze and address potential biases within the dataset to ensure fair and equitable emotion recognition across different demographics.
- **Real-world Robustness:** Investigate techniques to improve the model's performance in challenging real-world conditions, such as varying lighting, head poses, and occlusions.

By prioritizing these next steps, stakeholders can unlock the full potential of this facial emotion detection system, driving innovation and enhancing human-computer interaction across a wide range of applications.

# Problem and Solution Summary

In this chapter we'll look at the problem being solved and summary of the final solution proposed after evaluating the different deep learning models to solve the problem.

## Problem Being Solved

The core problem addressed in this project was the accurate and efficient detection of human emotions from grayscale facial images. The ability to understand and respond to human emotions is critical for enhancing human-computer interaction across various domains, including:

- **Customer Service:** Detecting customer emotions can enable personalized interactions, leading to improved satisfaction and loyalty.
- **Education:** Understanding student emotions can help tailor educational content and support, addressing frustration or confusion in real-time.
- **Healthcare:** Emotion detection can aid in the early diagnosis and intervention of mental health conditions by identifying subtle emotional cues.
- **Marketing:** Analyzing consumer emotions during product interactions can provide valuable insights into preferences and purchasing behavior.
- **Security:** Systems can identify potential security threats by recognizing aggression or deceit in facial expressions

However, the task of accurately classifying facial emotions from grayscale images presents several challenges, including *(Please, see Appendix-1)*:

- **Variations in Facial Features:** Diverse facial structures, expressions, and ages can make it difficult to identify consistent patterns.
- **Lighting Conditions:** Changes in lighting can significantly affect the appearance of facial features, impacting the accuracy of emotion detection.
- **Ambiguous Expressions:** Some facial expressions may be subtle or ambiguous, making it challenging to assign a definitive emotion label.
- **Partially Covered:** Some images were either partially covered by either the person or by the object.
- **Copied Images:** Some images seemed to be either screenshots or re-used copies as they had text written on those images. Text was also covering the images, making it hard for the models to detect effectively.

## Final Proposed Solution Design

The final proposed solution is a custom-designed complex Convolutional Neural Network (CNN) model, referred to as **'complex_cnn_model_1'**. Key points of this solution design include *(Please, see Appendix-3 for more details of this model and also see Appendix-2 for models comparison)*:

- **Grayscale Input:** The model is specifically designed to process grayscale images, leveraging the inherent characteristics of the dataset for optimal performance.

- **Deep Architecture:** The complex CNN architecture incorporates multiple convolutional layers, pooling layers, and activation functions to extract intricate features from facial images.
- **Regularization Techniques:** To prevent overfitting and enhance generalization, the model incorporates regularization techniques such as dropout regularization.
- **Data Augmentation:** A comprehensive data augmentation strategy is employed to artificially increase the diversity of the training data, making the model more robust to variations in facial features and lighting conditions.

## Validity and Impact of the Solution

This proposed solution is considered valid and likely to solve the problem for several reasons:

- **Superior Performance:** The complex CNN model (**'complex_cnn_model_1'**) demonstrated the highest accuracy while lowest loss on the validation set compared to other models evaluated, indicating its effectiveness in classifying facial emotions. Although **cnn_model_1** also performed really good but our chosen model had lowest loss and deeper architecure for further enhancesments and applications *(Please! see Appendix-2 )*.
- **Tailored Architecture:** The model's architecture is specifically designed for grayscale images, optimizing its ability to extract relevant features from the dataset *(Please see Appendix-3)*.
- **Robustness:** The incorporation of optimization and regularization techniques, and data augmentation enhances the model's ability to generalize to unseen data, improving its performance in real-world scenarios.

The implementation of this solution is expected to have a significant positive impact:

- **Enhanced Human-Computer Interaction:** By accurately detecting human emotions, applications can provide more personalized and empathetic experiences, leading to increased user satisfaction and engagement.
- **Improved Decision-Making:** In various fields, understanding human emotions can inform decision-making processes, leading to better outcomes in customer service, education, healthcare, and marketing.
- **Innovation in AI Applications:** This solution contributes to the advancement of artificial intelligence and its application in understanding and responding to human emotions, opening up new possibilities for technological innovation.

By addressing the challenges of facial emotion detection and providing a robust, accurate solution, this project paves the way for more intuitive and effective human-computer interaction across a wide range of applications.

# Implementation Recommendations

There are many aspects to implementation recommendations. For the sake of verbosity we are going to cover as much as I could think of at this stage.

## Key Recommendations

To successfully implement the proposed facial emotion detection solution, the following key recommendations should be considered:

1. **Prioritize Deployment Strategy:**
    - **Cloud API:** If broad accessibility and scalability are paramount, deploy the model as a REST API on a cloud platform like Google Cloud Platform or AWS. This allows easy integration into various applications through API calls.
    - **Edge Deployment:** For real-time, privacy-focused applications, prioritize deployment on edge devices like smartphones or embedded systems. This minimizes latency and reduces reliance on cloud connectivity.
2. **Optimize Model Performance:**
    - **Quantization:** Implement model quantization techniques to reduce the model size and improve inference speed, especially for deployment on resource-constrained edge devices.
    - **Continuous Fine-tuning:** Establish a process for continuous fine-tuning of the model using new and diverse data. This ensures the model remains accurate and adapts to evolving real-world scenarios.
3. **Establish Data Governance:**
    - **Data Collection:** Develop a robust data collection strategy to gather diverse and representative facial image data, addressing potential biases and ensuring ethical considerations.
    - **Data Annotation:** Implement a rigorous annotation process to ensure high-quality labels for training and validating the model.
4. **Develop User-Friendly Interfaces:**
    - **API Documentation:** Provide clear and comprehensive API documentation for developers to easily integrate the emotion detection functionality into their applications.
    - **Intuitive UI/UX:** For end-user applications, design intuitive and user-friendly interfaces that effectively communicate the detected emotions and provide actionable insights.

## Key Actionables for Stakeholders

- **Technical Teams:**
    - Prioritize model deployment based on the chosen strategy (cloud or edge).
    - Implement optimization techniques (quantization, fine-tuning) to enhance performance.
    - Develop robust data collection and annotation processes.

  - Create user-friendly interfaces (API documentation, UI/UX) for seamless integration and user experience.
- **Business Leaders:**
  - Secure necessary resources (budget, personnel) for implementation and ongoing maintenance.
  - Identify target applications and industries where the solution can provide the most value.
  - Develop a clear communication strategy to educate stakeholders and potential users about the benefits of the solution.
- **Data Ethics Committee:**
  - Establish guidelines and protocols to ensure ethical data collection, annotation, and usage.
  - Regularly review the model's performance to identify and mitigate potential biases.

## Expected Benefits and Costs

**Benefits:**

- **Enhanced Customer Experiences:** Personalized interactions based on detected emotions can improve customer satisfaction, loyalty, and retention.
- **Improved Educational Outcomes:** Tailored educational content and support based on student emotions can lead to better learning outcomes and reduced frustration.
- **Early Mental Health Intervention:** Emotion detection can aid in identifying individuals at risk of mental health conditions, enabling timely intervention and support.
- **Data-Driven Marketing Insights:** Understanding consumer emotions can inform marketing strategies, leading to more effective campaigns and increased sales.

**Costs:**

- **Development and Deployment:** Costs associated with model development, deployment infrastructure (cloud or edge), and ongoing maintenance. **Rational Assumption:** $50,000 - $200,000 depending on the scale and complexity of deployment.
- **Data Collection and Annotation:** Costs associated with gathering and annotating a diverse and representative dataset. **Rational Assumption:** $10,000 - $50,000 depending on the dataset size and annotation requirements.
- **Training and Fine-tuning:** Computational costs for model training and ongoing fine-tuning. **Rational Assumption:** $500 - $5,000 per training cycle, depending on the model complexity and dataset size.

**Note:** These cost assumptions are highly variable and should be tailored to the specific needs and resources of the implementing organization at the time of planning.

## Key Risks and Challenges

Following are some of the key risks and challeneges.

- **Data Privacy:** Ensuring the ethical and responsible collection and usage of facial image data is crucial to maintain user trust and comply with privacy regulations.

- **Accuracy in Real-world Conditions:** The model's performance may be affected by challenging real-world conditions, such as varying lighting, head poses, color patterns, and occlusions. Ongoing fine-tuning and optimization are necessary to address these challenges.
- **Bias Mitigation:** Addressing potential biases in the dataset and the model's predictions is essential to ensure fair and equitable emotion recognition for all users.

## Further Analysis

To further refine the facial emotion detection solution and maximize its impact, the following areas warrant additional analysis and investigation:

1. **Real-World Performance Evaluation:** Conduct rigorous testing of the deployed model in real-world scenarios across diverse user demographics and environmental conditions. This will help identify specific challenges and inform targeted optimizations for improved robustness.
2. **Cross-Cultural Validation:** Assess the model's performance across different cultures and ethnicities. Facial expressions and their interpretations can vary across cultures, so validating the model's accuracy in diverse cultural contexts is essential.
3. **Integration with Multimodal Data:** Explore integrating facial emotion detection with other modalities, such as speech analysis and physiological signals (heart rate, skin conductance). This could lead to more comprehensive and accurate emotion detection.
4. **Explainability and Interpretability:** Investigate techniques to make the model's predictions more explainable and interpretable. This is crucial for building trust and understanding how the model arrives at its decisions, especially in sensitive applications like healthcare or law enforcement.
5. **Addressing Edge Cases:** Analyze the model's performance on edge cases, such as extreme expressions, occlusions, and low-quality images. Develop strategies to improve the model's robustness in handling these challenging scenarios. This will help especailly with the applications that require low-latency and real-time detection for example it can really help in autonomous vehicles where emotions and expressions can be detected in time for the vehicle to react human conditions and not just to the road conditions.
6. **Longitudinal Studies:** Conduct longitudinal studies to assess the long-term impact of the solution on user experiences, business outcomes, and societal implications. This will provide valuable insights for continuous improvement and responsible deployment.

By addressing these areas and continuously refining the solution, stakeholders can maximize the benefits of facial emotion detection, paving the way for more intuitive and impactful human-computer interaction across a wide range of applications.

# Appendix

## Appendix-1: Exploratory Data Analysis

**Images:** Some of the images taken from each class during the exploratory data analysis.

**Happy Class**

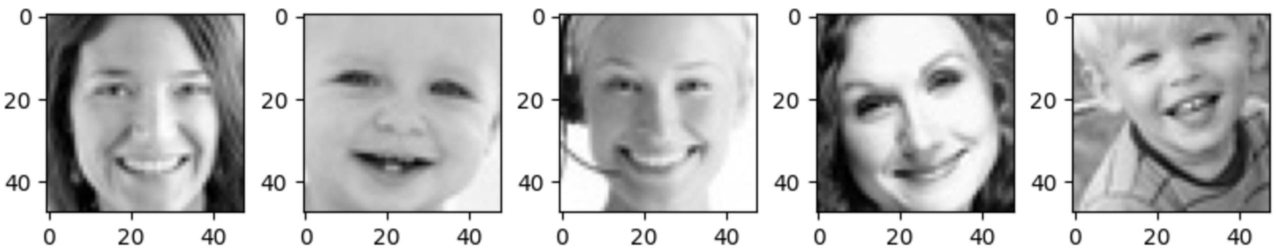

*Figure 1: Few Images from Happy Class*

**Sad Class**

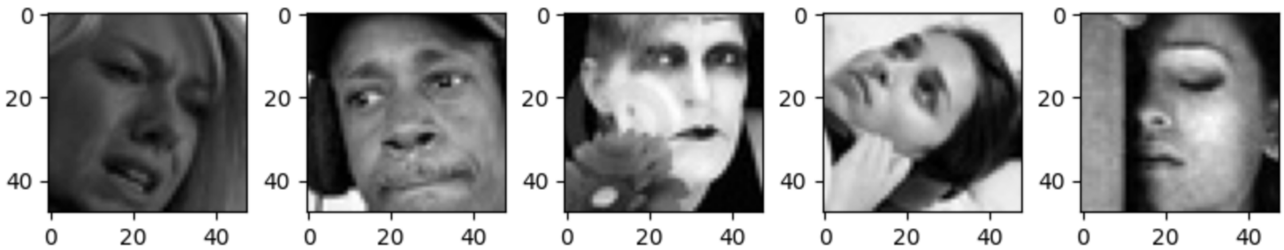

*Figure 2: : Few Images from Sad Class*

**Neutral Class**

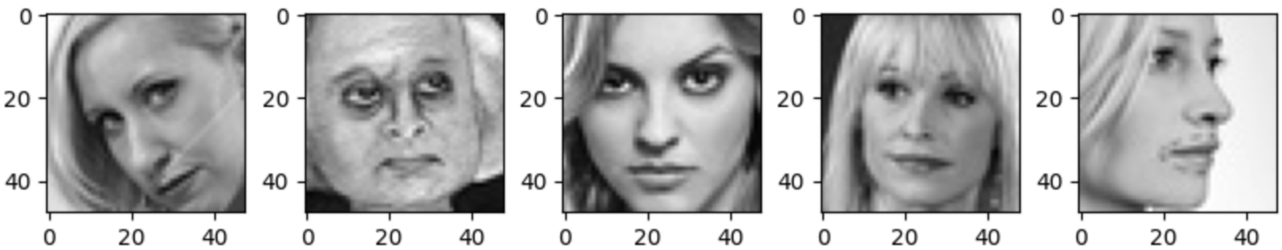

*Figure 3: : Few Images from Neutral Class*

## Surprise Class

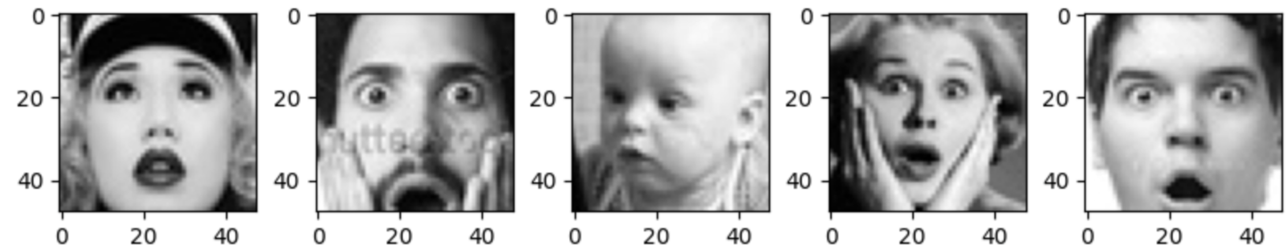

*Figure 4: : Few Images from Surprise Class*

## Data Split and Distribution:

The below table shows the data split of Training and Validation datasets.

| Data Set | Count | Precentage |
|---|---|---|
| **Train** | 15109 | 74.74% |
| **Validation** | 4977 | 24.62% |
| **Test** | 128 | 0.006% |

Table 1: Data Split

Below images show the distribution of each emotion class within Training and Validation datasets.

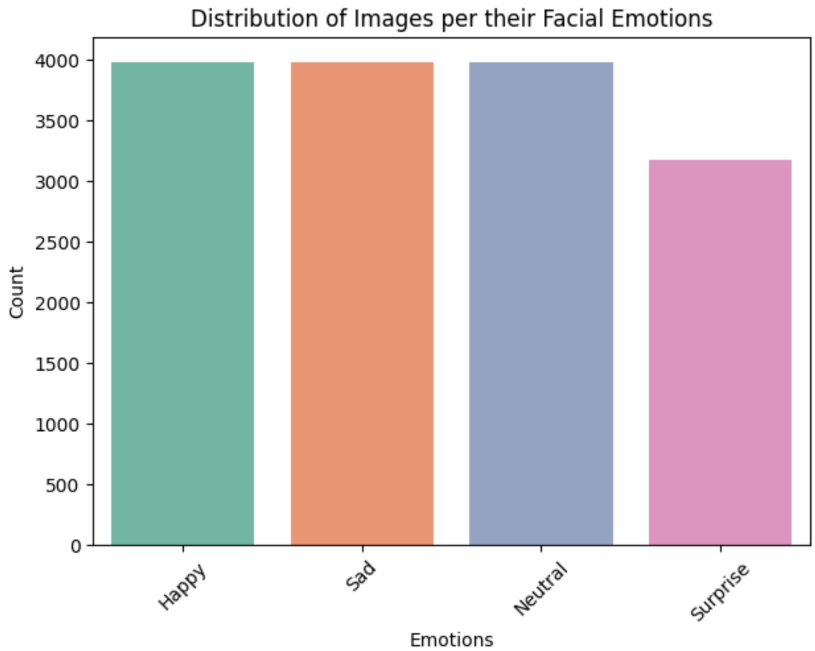


*Figure 5: Data/Images Distribution – Training Set*

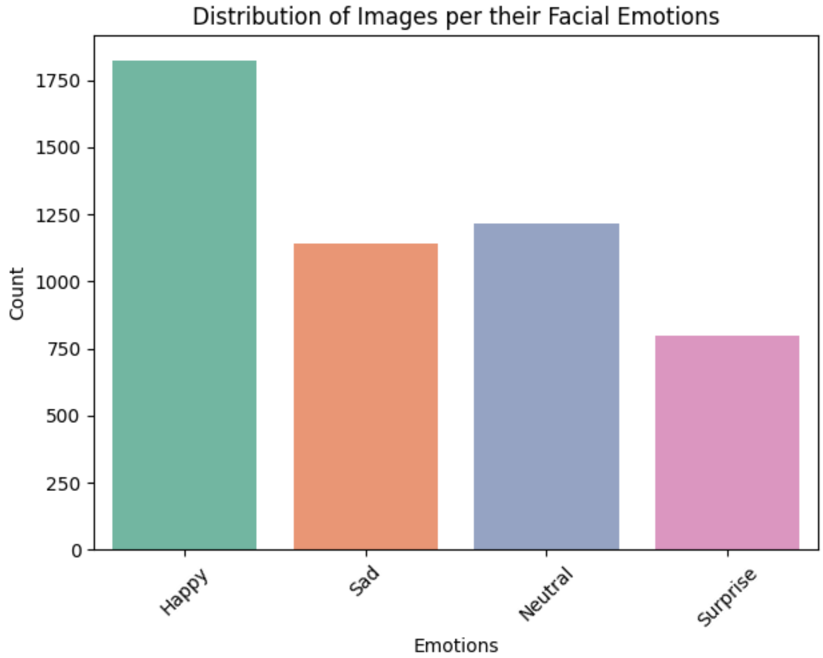


*Figure 6: Data/Images Distribution – Validation Set*

## Appendix-2: Models Perofrmance Comparison

| | Parameters | Train Loss | Train Accuracy | Val Loss | Val Accuracy | Test Loss | Test Accuracy |
|---|---|---|---|---|---|---|---|
| **cnn_model-1**: Baseline (Gray) | 684,036 | 0.77 | 0.68 | 0.62 | 0.76 | 0.62 | 0.76 |
| **cnn_model_2**: Baseline (RGB) | 2,765,316 | 0.49 | 0.80 | 0.67 | 0.75 | 0.58 | 0.76 |
| **vgg_model_1**: VGG16 (RGB) | 14,714,688 | 1.15 | 0.49 | 1.05 | 0.54 | 1.08 | 0.48 |
| **resnet_model-1**: ResNet50V2 (RGB) | 25,703,876 | 1.03 | 0.56 | 0.99 | 0.58 | 0.99 | 0.58 |
| **efficient_model_1**: EfficientNetV2B2 (RGB) | 10,253,090 | 1.38 | 0.26 | 1.36 | 0.37 | 1.36 | 0.37 |
| **complex_cnn_model_1**: Complex CNN (Gray - Chosen Model) | 1,839,876 | 0.69 | 0.72 | 0.60 | 0.76 | 0.55 | 0.80 |
| **complex_cnn_model_2**: Complex CNN (Gray) | 3,412,100 | 0.86 | 0.66 | 0.67 | 0.74 | 0.62 | 0.74 |

*Figure 7: Models Comparison – Complex CNN Model-1*

# Appendix-3: Final Proposed Model – Complex CNN Model-1

## Data Augmentation:

In this section, we are creating data loaders which we will use as inputs to the more Complicated Convolutional Neural Network. We will go ahead with color_mode = 'grayscale'.

```
# # Data augmentation
train_datagen = ImageDataGenerator(horizontal_flip = True,
                                   brightness_range = (0.,2.),
                                   rescale = 1./255,
                                   zoom_range = 0.2,
                                   rotation_range = 20,
                                   shear_range = 0.3)
test_datagen = ImageDataGenerator(rescale = 1./255) # No data augmentation for test and/or validation

# Create image generators -- Grayscale Images
print("\nGrayscale Images:")

train_generator_gray = train_datagen.flow_from_directory(
    train_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=32,
    color_mode='grayscale',
    class_mode='categorical',
    shuffle=True
)

validation_generator_gray = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=32,
    color_mode='grayscale',
    class_mode='categorical',
    shuffle=False
)

test_generator_gray = test_datagen.flow_from_directory(
    test_dir,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=32,
    color_mode='grayscale',
    class_mode='categorical',
    shuffle=False
)

# Define the number of classes (emotions)
num_classes_gray = len(train_generator_gray.class_indices)
print("\nNumber of training classes for Grayscale Images:", num_classes)
print("Training Class Indices for Grayscale Images:", train_generator_gray.class_indices)
```

```
Grayscale Images:
Found 15109 images belonging to 4 classes.
Found 4977 images belonging to 4 classes.
Found 128 images belonging to 4 classes.

Number of training classes for Grayscale Images: 4
Training Class Indices for Grayscale Images: {'happy': 0, 'neutral': 1, 'sad': 2, 'surprise': 3}
```

*Figure 8: Data Augmentation – Complex CNN Model-1*

## Model Architecture:

- Let's change input shape to (48, 48, 1)

```
# Define our first custome model with 5 Convolutional Blocks
complex_cnn_model_1 = Sequential([
  # Block 1
  Conv2D(32, (3, 3), activation='relu', input_shape= (IMG_SIZE, IMG_SIZE, 1), padding= 'same'),
  MaxPooling2D((2, 2)),
  # Block 2 (Let's add BatchNormalizatio, LeakyRelu, and GaussianNoise onwards)
  Conv2D(64, (3, 3), activation='relu', padding= 'same'),
  BatchNormalization(),
  LeakyReLU(alpha=0.1),
  MaxPooling2D((2, 2)),
  GaussianNoise(0.1),
  # Block 3
  Conv2D(128, (3, 3), activation='relu', padding= 'same'),
  BatchNormalization(),
  LeakyReLU(alpha=0.1),
  MaxPooling2D((2, 2)),
  Dropout(0.2), # Add dropout for regularization
  # Block 4
  Conv2D(256, (3, 3), activation='relu', padding= 'same'),
  BatchNormalization(),
  LeakyReLU(alpha=0.1),
  MaxPooling2D((2, 2)),
  GaussianNoise(0.1),
  # Block 5
  Conv2D(512, (3, 3), activation='relu', padding= 'same'),
  BatchNormalization(),
  LeakyReLU(alpha=0.1),
  MaxPooling2D((2, 2)),
  Dropout(0.2),
  # Flatten for dense layers
  Flatten(),
  # Dense layers for classification
  Dense(256, activation='relu'),
  BatchNormalization(),
  Dropout(0.2),
  # Add one more dense layer before output layer
  Dense(512, activation='relu'),
  BatchNormalization(),
  Dropout(0.2),
  Dense(num_classes_gray, activation='softmax')  # Output layer with softmax for multi-class classification
])

# Model summary
complex_cnn_model_1.summary()
```

```
=================================================================
Total params: 1839876 (7.02 MB)
Trainable params: 1836420 (7.01 MB)
Non-trainable params: 3456 (13.50 KB)
_________________________________________________________________
```

*Figure 9: Model Architecture – Complex CNN Model-1*

## Model Compilation and Training:

```
# Compile the model with an optimizer, loss function, and metrics
complex_cnn_model_1.compile(optimizer=Adam(learning_rate = 0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
# Train the model with training and validation data
history_complex_cnn1 = complex_cnn_model_1.fit (train_generator_gray,
    epochs=30,
    validation_data=validation_generator_gray,
    verbose=1,
    callbacks=callbacks_list
)
```

*Figure 10: Compilation and Training Code – Complex CNN Model-1*

**Model Evaluation:**

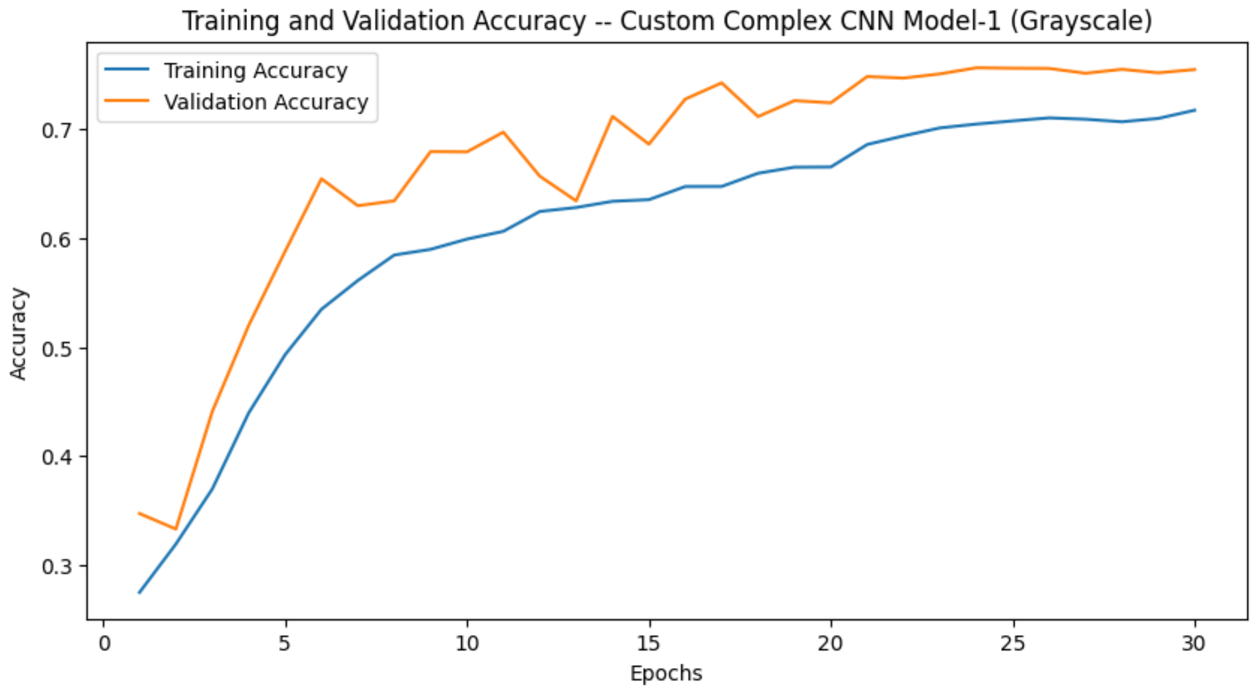


*Figure 11: Training vs Validation Accuracy – Complex CNN Model-1*

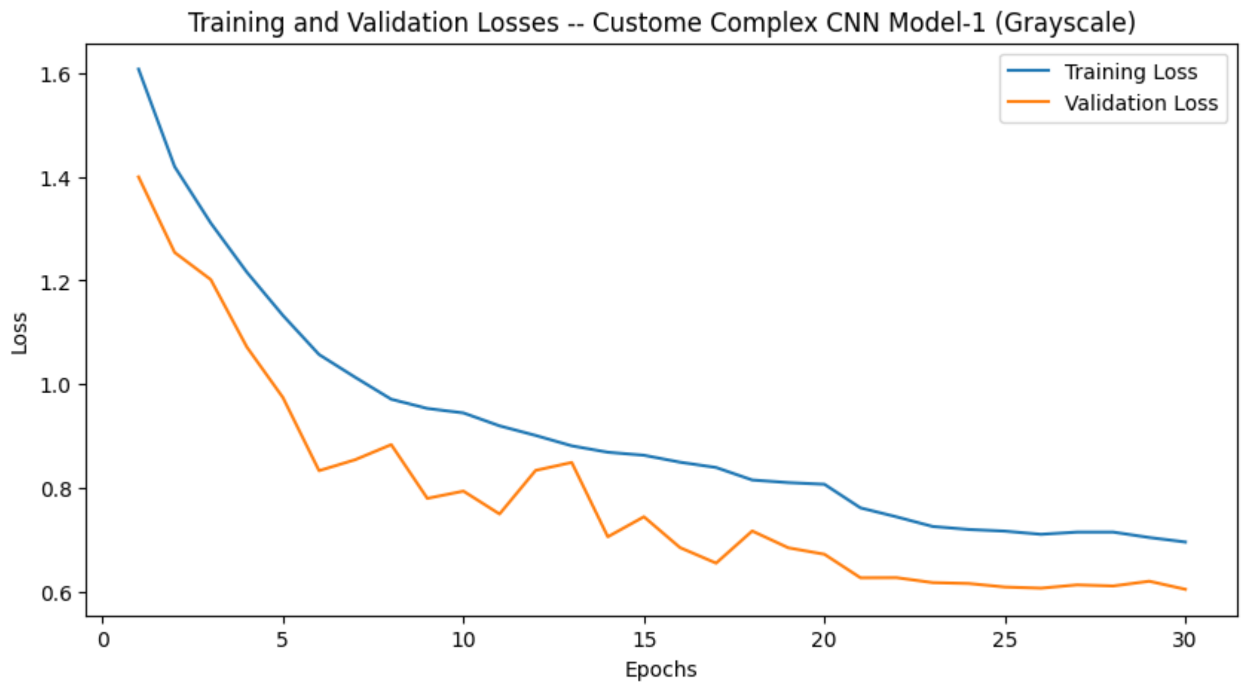


*Figure 12: Training vs Validation Loss – Complex CNN Model-1*

Classification Report for Complex CNN Model-1 (Grayscale):

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Happy | 0.87 | 0.82 | 0.85 | 1825 |
| Sad | 0.62 | 0.72 | 0.67 | 1216 |
| Neutral | 0.68 | 0.61 | 0.65 | 1139 |
| Surprise | 0.82 | 0.86 | 0.84 | 797 |
| | | | | |
| accuracy | | | 0.75 | 4977 |
| macro avg | 0.75 | 0.75 | 0.75 | 4977 |
| weighted avg | 0.76 | 0.75 | 0.76 | 4977 |

*Figure 13: Classification Report – Complex CNN Model-1*

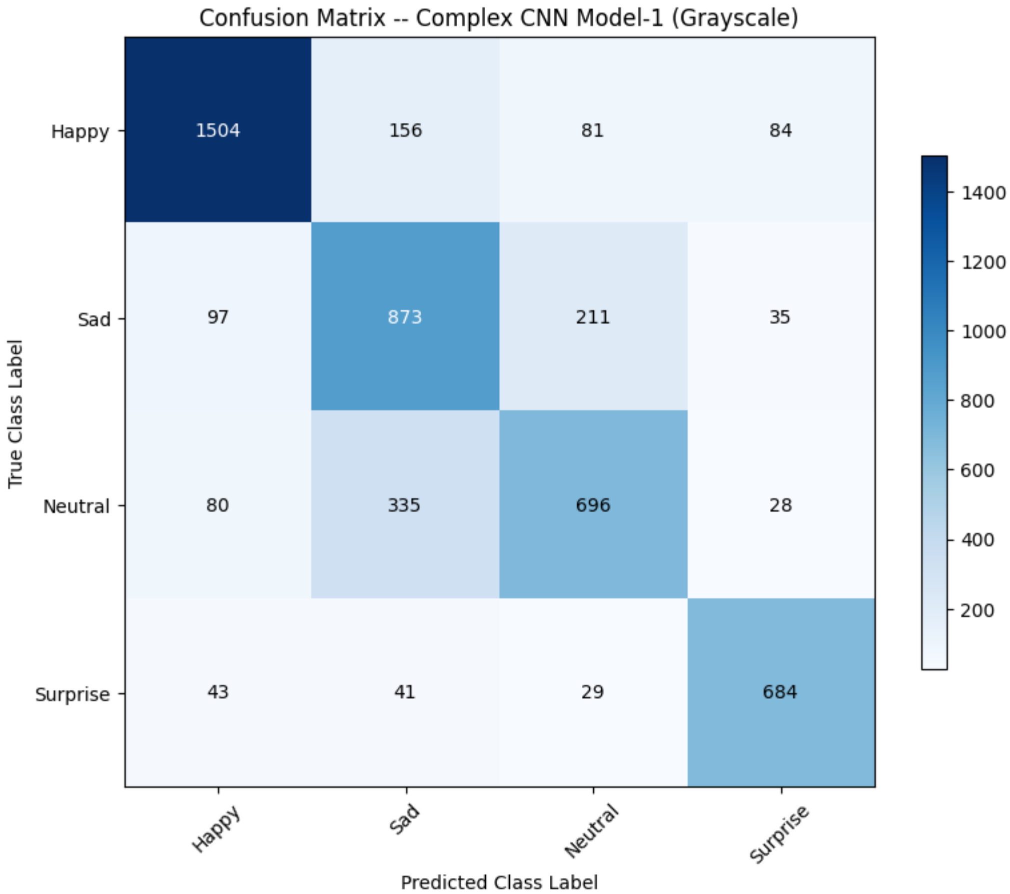


*Figure 14: Confusion Matrix – Complex CNN Model-1*

# References


- "Facial Expression Recognition Using Convolutional Neural Networks: State of the Art" by Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger (2017).
- "Facial Emotion Recognition: A Survey" by Aleix Martinez and Shichuan Du (2000).
- "Deep Facial Expression Recognition: A Survey" by Shan Li and Weihong Deng (2020).
- "Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning" by Hongwei Ng, Viet Dung Nguyen, Vassilios Vonikakis, Stefan Winkler
- "Implementing a Real-time Emotion Detection System using Convolutional Neural Network" by K Nalini, G. Lakshmi Indira, S. Mahathi Varma, P Vinith
- "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016).
- "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman (2014).
- "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" by Mingxing Tan, Quoc V. Le (2019).
- "Multi-task Learning for Multi-modal Emotion Recognition and Sentiment Analysis" by Soujanya Poria, M. Shad Akhtar, Deepanway Ghosal, Dushyant Singh Chauhan (2019).
- "Unmasking the Face: A Guide to Recognizing Emotions From Facial Clues" by Paul Ekman and Wallace V. Friesen (2003).
- "Emotional Expressions Reconsidered: Challenges to Inferring Emotion From Human Facial Movements" by Barrett, Lisa Feldman et al. (2019).
- "Automatic Analysis of Facial Actions: A Survey" by by Michel Valstar, Bihan Jiang, Brais Martinez, and Maja Pantic (2017).
- "Meta-Analysis of the First Facial Expression Recognition Challenge" by Valstar, Michel F.; Mehu, Marc; Jiang, Bo; Pantic, Maja; Scherer, Klaus R.
- "Emotion Methods for Facial Expression Recognition with Applications in Challenging Situations" by Anil Audumbar Pise, Mejdal A. Alqahtani, Priti Verma, Purushothama K, Dimitrios A. Karras, Prathibha S, and Awal Halifa
- "Human-Computer Interaction Using Emotion Recognition from Facial Expression" by Abdat, F. et al. (2011).
- Facial Emotion Recognition by European Data Protection Supervisor, issue 1, 2021.
- "Python tutorial. Technical Report CS-R9128, Centrum voor Wiskunde en Informatica (CWI), Amsterdam" *by* Van Rossum, G. (1991).
- "TensorFlow: Large-scale machine learning on heterogeneous systems" by Abadi, M., Agarwal, A., Barham, P., ... & Zheng, X. (2015).
- Keras. *GitHub repository,* https://github.com/fchollet/keras by Chollet, F. (2015).
- "The OpenCV Library. *Dr. Dobb's Journal of Software Tools*" by Bradski, G. (2000).
- "Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830" by Pedregosa, F., Varoquaux, G., Gramfort, A., ... & Duchesnay, E. et al. (2011).
- "Array programming with NumPy. *Nature, 585*, 357-362 (2020)" by Harris, C. R., Millman, K. J., van der Walt, S. J., ... & Oliphant, T. E. et al. (2020).
- "Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95" by Hunter, J. D. (2007).
- "Seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021" by Waskom, M. (2021).