Emotion-Age-Gender-Ethnicity Prediction

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

Prerak Lodha [RA2111026010394]
Nimish Julka [RA2111026010408]

Under the guidance of

Dr Karpagam M.

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING w/s in Artificial Intelligence and Machine Learning

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"Emotion-Age-Gender-Ethnicity Prediction"** is the bonafide work of **Prerak Lodha (RA2111026010394) and Nimish Julka (RA2111026010408)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Karpagam M.
Assistant Professor
CINTEL Dept.

ABSTRACT

This project proposes a deep learning model capable of comprehensively analyzing facial features to predict a person's emotions, age, gender, and ethnicity. The model leverages convolutional neural networks (CNNs) to extract intricate patterns from facial images.

We explore the effectiveness of various CNN architectures and investigate techniques to enhance model robustness against factors like pose variations, occlusions, and lighting conditions.

The performance of the model is evaluated on a benchmark facial analysis dataset, and we analyze the accuracy achieved for each prediction task. This research contributes to the development of multi-faceted facial analysis systems with applications in human-computer interaction, customer segmentation, and social media analysis.

The ethical considerations surrounding such technologies and the emphasis on importance of responsible use is also discussed.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Networks
PIR	Passive Infrared
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation (optimizer)
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

The human face is a powerful window into our internal state. It conveys a wealth of information, from fleeting emotions to life experiences reflected in wrinkles and skin texture. This report explores the potential of automated facial analysis using deep learning to predict a person's emotions, age, gender, and ethnicity.

Motivated by the desire to create a comprehensive facial analysis system, this project investigates the use of Convolutional Neural Networks (CNNs) to extract intricate patterns from facial images. We delve into the effectiveness of various CNN architectures, exploring techniques to build robustness against real-world challenges such as variations in head pose, occlusions like glasses or hats, and diverse lighting conditions.

To assess the model's performance, we leverage a benchmark facial analysis dataset. By analyzing the accuracy achieved for each prediction task (emotion, age, gender, and ethnicity), we aim to contribute to the development of multi-faceted facial analysis systems. These systems have the potential to revolutionize various fields, including human-computer interaction, customer segmentation in targeted marketing, and social media analysis.

However, the power of such technology necessitates a discussion on ethical considerations. This report acknowledges the importance of responsible use and explores potential biases that may arise in data collection and model training. We emphasize the need for ethical frameworks to ensure these systems are deployed with fairness and transparency.

CHAPTER 2

LITERATURE SURVEY

Automatic facial analysis has become a rapidly evolving field with significant advancements in deep learning techniques. This section explores existing research on emotion, age, gender, and ethnicity prediction using deep learning models.

Emotion Recognition: Convolutional Neural Networks (CNNs) have proven highly effective in emotion recognition. Studies by *Ekman & Friesen (1978) and Zhao et al. (2019)* demonstrate the success of CNNs in extracting features from facial expressions for classifying emotions like happiness, sadness, anger, and surprise. Research by *Levi et al. (2015)* explores methods to address challenges such as pose variations and occlusions, improving model robustness.

Age and Gender Prediction: Deep learning excels in age and gender prediction tasks. *Guo et al. (2016)* compares various CNN architectures for age estimation, highlighting the impact of model depth and complexity. Similarly, *Liu et al. (2017)* investigates techniques like multi-task learning for simultaneous age and gender classification, achieving high accuracy rates.

Ethnicity Prediction: Ethnicity prediction using facial images presents a complex task due to cultural and individual variations. *Buolamwini & Gebru (2018)* explore the limitations of deep learning models in this domain, emphasizing the need for diverse datasets and careful model development to mitigate potential biases.

Multi-task Learning: Several studies explore the potential of multi-task learning architectures for simultaneous prediction of multiple facial attributes. *Sun et al. (2014)* proposes a framework for joint emotion, age, and gender recognition, demonstrating improved performance compared to single-task models.

Challenges and Biases: Despite advancements, facial analysis using deep learning faces challenges. Data collection and model training can introduce biases based on ethnicity, gender, and cultural background. *Tobin et al. (2022)* addresses these concerns, emphasizing the importance of diverse and balanced datasets to ensure model fairness.

This literature survey provides a foundation for our research by highlighting successful deep learning techniques for facial analysis tasks. We will build upon this knowledge by exploring various CNN architectures, investigating methods to enhance model robustness, and addressing ethical considerations for responsible development and deployment.

CHAPTER 3

MODEL ARCHITECTURE AND DESIGN

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 64)        640

batch_normalization (BatchN  (None, 48, 48, 64)        256
ormalization)

conv2d_1 (Conv2D)            (None, 48, 48, 64)        36928

batch_normalization_1 (Batc  (None, 48, 48, 64)        256
hNormalization)

max_pooling2d (MaxPooling2D  (None, 24, 24, 64)        0
)

dropout (Dropout)            (None, 24, 24, 64)        0

conv2d_2 (Conv2D)            (None, 24, 24, 128)       73856

batch_normalization_2 (Batc  (None, 24, 24, 128)       512
hNormalization)

conv2d_3 (Conv2D)            (None, 24, 24, 128)       147584

batch_normalization_3 (Batc  (None, 24, 24, 128)       512
hNormalization)

max_pooling2d_1 (MaxPooling  (None, 12, 12, 128)       0
2D)

dropout_1 (Dropout)          (None, 12, 12, 128)       0

conv2d_4 (Conv2D)            (None, 12, 12, 256)       295168

batch_normalization_4 (Batc  (None, 12, 12, 256)       1024
hNormalization)

conv2d_5 (Conv2D)            (None, 12, 12, 256)       590080

batch_normalization_5 (Batc  (None, 12, 12, 256)       1024
hNormalization)

max_pooling2d_2 (MaxPooling  (None, 6, 6, 256)         0
2D)

dropout_2 (Dropout)          (None, 6, 6, 256)         0

flatten (Flatten)            (None, 9216)              0

dense (Dense)                (None, 1024)              9438208

batch_normalization_6 (Batc  (None, 1024)              4096
hNormalization)

dropout_3 (Dropout)          (None, 1024)              0

dense_1 (Dense)              (None, 512)               524800

batch_normalization_7 (Batc  (None, 512)               2048
hNormalization)

dropout_4 (Dropout)          (None, 512)               0

dense_2 (Dense)              (None, 256)               131328

batch_normalization_8 (Batc  (None, 256)               1024
hNormalization)

dense_3 (Dense)              (None, 7)                 1799

=================================================================
Total params: 11,251,143
Trainable params: 11,245,767
Non-trainable params: 5,376
_________________________________________________________________
```

Emotion Model Architecture

CHAPTER 3

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 64)        640

 batch_normalization (BatchN  (None, 48, 48, 64)       256
 ormalization)

 conv2d_1 (Conv2D)           (None, 48, 48, 64)        36928

 batch_normalization_1 (Batc  (None, 48, 48, 64)       256
 hNormalization)

 max_pooling2d (MaxPooling2D  (None, 24, 24, 64)       0
 )

 dropout (Dropout)           (None, 24, 24, 64)        0

 conv2d_2 (Conv2D)           (None, 24, 24, 128)       73856

 batch_normalization_2 (Batc  (None, 24, 24, 128)      512
 hNormalization)

 conv2d_3 (Conv2D)           (None, 24, 24, 128)       147584

 batch_normalization_3 (Batc  (None, 24, 24, 128)      512
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 dropout_1 (Dropout)         (None, 12, 12, 128)       0

 conv2d_4 (Conv2D)           (None, 12, 12, 256)       295168

 batch_normalization_4 (Batc  (None, 12, 12, 256)      1024
 hNormalization)

 conv2d_5 (Conv2D)           (None, 12, 12, 256)       590080

 batch_normalization_5 (Batc  (None, 12, 12, 256)      1024
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 dropout_2 (Dropout)         (None, 6, 6, 256)         0

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 1024)              9438208

 batch_normalization_6 (Batc  (None, 1024)             4096
 hNormalization)

 dropout_3 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 512)               524800

 batch_normalization_7 (Batc  (None, 512)              2048
 hNormalization)

 dropout_4 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 256)               131328

 batch_normalization_8 (Batc  (None, 256)              1024
 hNormalization)

 dense_3 (Dense)             (None, 7)                 1799

=================================================================
Total params: 11,251,143
Trainable params: 11,245,767
Non-trainable params: 5,376
_________________________________________________________________
```

Age Model Architecture

CHAPTER 3

Gender Model Architecture

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 48, 48, 32)        320

 batch_normalization_9 (Batc  (None, 48, 48, 32)       128
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 24, 24, 32)       0
 2D)

 dropout_5 (Dropout)         (None, 24, 24, 32)        0

 conv2d_7 (Conv2D)           (None, 24, 24, 64)        18496

 batch_normalization_10 (Bat  (None, 24, 24, 64)       256
 chNormalization)

 max_pooling2d_4 (MaxPooling  (None, 12, 12, 64)       0
 2D)

 dropout_6 (Dropout)         (None, 12, 12, 64)        0

 conv2d_8 (Conv2D)           (None, 12, 12, 64)        36928

 batch_normalization_11 (Bat  (None, 12, 12, 64)       256
 chNormalization)

 max_pooling2d_5 (MaxPooling  (None, 6, 6, 64)         0
 2D)

 dropout_7 (Dropout)         (None, 6, 6, 64)          0

 flatten_1 (Flatten)         (None, 2304)              0

 dense_4 (Dense)             (None, 256)               590080

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 646,721
Trainable params: 646,401
Non-trainable params: 320
_________________________________________________________________
```

CHAPTER 3

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 48, 48, 64)        640

 batch_normalization_12 (Bat  (None, 48, 48, 64)       256
 chNormalization)

 conv2d_10 (Conv2D)          (None, 48, 48, 64)        36928

 batch_normalization_13 (Bat  (None, 48, 48, 64)       256
 chNormalization)

 max_pooling2d_6 (MaxPooling  (None, 24, 24, 64)       0
 2D)

 dropout_8 (Dropout)         (None, 24, 24, 64)        0

 conv2d_11 (Conv2D)          (None, 24, 24, 128)       73856

 batch_normalization_14 (Bat  (None, 24, 24, 128)      512
 chNormalization)

 conv2d_12 (Conv2D)          (None, 24, 24, 128)       147584

 batch_normalization_15 (Bat  (None, 24, 24, 128)      512
 chNormalization)

 max_pooling2d_7 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 dropout_9 (Dropout)         (None, 12, 12, 128)       0

 conv2d_13 (Conv2D)          (None, 12, 12, 256)       295168

 batch_normalization_16 (Bat  (None, 12, 12, 256)      1024
 chNormalization)

 conv2d_14 (Conv2D)          (None, 12, 12, 256)       590080

 batch_normalization_17 (Bat  (None, 12, 12, 256)      1024
 chNormalization)

 max_pooling2d_8 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 dropout_10 (Dropout)        (None, 6, 6, 256)         0

 flatten_2 (Flatten)         (None, 9216)              0
```

Ethnicity Model Architecture

```
 dense_6 (Dense)             (None, 1024)              9438208

 batch_normalization_18 (Bat  (None, 1024)             4096
 chNormalization)

 dropout_11 (Dropout)        (None, 1024)              0

 dense_7 (Dense)             (None, 512)               524800

 batch_normalization_19 (Bat  (None, 512)              2048
 chNormalization)

 dropout_12 (Dropout)        (None, 512)               0

 dense_8 (Dense)             (None, 256)               131328

 batch_normalization_20 (Bat  (None, 256)              1024
 chNormalization)

 dense_9 (Dense)             (None, 5)                 1285


=================================================================
Total params: 11,250,629
Trainable params: 11,245,253
Non-trainable params: 5,376
_________________________________________________________________
```

CHAPTER 3

Description of Layers

Layers are the fundamental building blocks used in the convolutional neural network architectures:

- **Convolutional Layer (Conv2D):** This layer is the core of a CNN. It applies a filter (kernel) to the input data (image) to extract features. The kernel slides across the width and height of the input, computing the element-wise product (dot product) between its weights and the corresponding elements in the input. The results are summed and passed through an activation function (e.g., ReLU) to introduce non-linearity. This process helps identify patterns and local features within the image.

- **Batch Normalization:** This layer addresses the problem of internal covariate shift during training, which can slow down the learning process. It normalizes the activations of the previous layer across the mini-batch, improving training stability and gradient flow.

- **Max Pooling Layer (MaxPooling2D):** This layer performs down sampling by applying a filter (window) to the input data and selecting the maximum value within each window. This reduces the dimensionality of the data, making the model less computationally expensive and potentially more robust to slight variations in the input.

- **Dropout Layer:** This layer is a regularization technique used to prevent overfitting. During training, a random subset of activations from the previous layer is dropped with a predefined probability (e.g., 0.5). This forces the model to learn more robust features that are not dependent on any specific neuron and improves generalization performance.

- **Flatten Layer:** This layer transforms the output from the previous convolutional layers (typically 3D tensors with width, height, and channels) into a single 1D vector. This allows the data to be fed into fully-connected layers for classification.

- **Dense Layer (Fully-Connected Layer):** This layer is a standard artificial neural network layer where each neuron is connected to all neurons in the previous layer. It performs matrix multiplication between the input vector and the weight matrix of the layer, followed by an activation function (e.g., ReLU) to introduce non-linearity. Dense layers are typically stacked sequentially to learn complex, high-level features and perform classification tasks.

- **SoftMax Activation:** This activation function is used in the final output layer for multi-class classification problems. It transforms the output values from the dense layer into a probability distribution between 0 and 1, where each element represents the probability of the input belonging to a specific class.

CHAPTER 4

METHODOLOGY

Data Preprocessing

1. **Data Acquisition:** We utilize a benchmark facial analysis dataset containing labeled images for emotions (e.g., happy, sad, angry), age groups, gender (M/F), and ethnicity (e.g., Asian, African American).

2. **Preprocessing:** The facial images undergo preprocessing steps to ensure consistency and optimize the training process. This may involve resizing images to a standard dimension (48x48 pixels), normalization (converting pixel values to a specific range like 0-1), and potentially data augmentation techniques (flipping images horizontally to increase training data variety).

Model Training

1. **Training Split:** The preprocessed dataset is split into training, validation, and testing sets. The training set is used to train the models, the validation set is used to monitor performance during training and adjust hyperparameters if necessary, and the testing set is used for final evaluation after training is complete.

2. **Loss Function and Optimizer:** Each model is compiled with an appropriate loss function depending on the prediction task. Categorical cross-entropy is used for multi-class classification (emotion, ethnicity), while binary cross-entropy is used for binary classification (gender). The Adam optimizer is employed for efficient gradient descent during training.

3. **Hyperparameter Tuning:** We explore various hyperparameters like learning rate, number of filters in convolutional layers, and dropout rates to optimize model performance. Techniques like grid search or random search can be used to identify the best hyperparameter combination.

4. **Training Process:** The models are trained on the training data for a specified number of epochs (iterations). The validation set is used to monitor for overfitting and early stopping can be implemented to prevent the model from memorizing the training data.

CHAPTER 4

Model Evaluation

1. **Performance Metrics:** After training, the models are evaluated on the unseen testing set. Accuracy, the percentage of correct predictions, is the primary metric for all tasks. Additionally, for multi-class classification tasks (emotion, ethnicity), confusion matrices can be used to visualize the model's performance on each class.

2. **Visualization:** Techniques like visualizing activation maps can be used to understand which facial regions contribute most significantly to the model's predictions for each task.

3. **Comparison with Benchmarks:** We compare the performance of our models with existing approaches on the same benchmark dataset to assess the effectiveness of our proposed methodology.

This methodology provides a comprehensive framework for developing and evaluating the deep learning model for multi-faceted facial analysis. The following sections will detail the specific implementation details, training procedures, and the analysis of the obtained results.

CHAPTER 5

CODING AND TESTING

- **Dataset Normalization**

```
df['pixels'] = df['pixels'].apply(
    lambda x:  np.array(
        x.split(),
        dtype="float32"
    )
)
X = np.array(df['pixels'].tolist()) / 255.0
X = X.reshape(-1, 48, 48, 1)
```

- **Split Training, Testing and Validation Set**

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state = 42
)
```

- **Declare and Compile CNN Models**

```
Input_shape = (48, 48, 1)

cnn_model = Sequential()
cnn_model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='elu', kernel_init
ializer='he_normal', padding='same', input_shape=Input_shape))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(filters=64, kernel_size=(3, 3), activation = 'elu', kernel_in
itializer='he_normal', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D(2,2))
cnn_model.add(Dropout(0.3))

cnn_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='elu', kernel_ini
tializer='he_normal', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='elu', kernel_ini
tializer='he_normal', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D(2,2))
cnn_model.add(Dropout(0.4))

cnn_model.add(Conv2D(filters=256, kernel_size=(3, 3), activation='elu', kernel_ini
tializer='he_normal', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(filters=256, kernel_size=(3, 3), activation='elu', kernel_ini
tializer='he_normal', padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(MaxPooling2D(2, 2))
cnn_model.add(Dropout(0.5))

cnn_model.add(Flatten())

cnn_model.add(Dense(units=1024, activation='relu', kernel_initializer='he_uniform'
))
```

```
cnn_model.add(BatchNormalization())
cnn_model.add(Dropout(0.5))

cnn_model.add(Dense(units=512, activation='relu', kernel_initializer='he_uniform')
)
cnn_model.add(BatchNormalization())
cnn_model.add(Dropout(0.3))

cnn_model.add(Dense(units=256, activation='relu', kernel_initializer='he_uniform')
)
cnn_model.add(BatchNormalization())

cnn_model.add(Dense(units=7, activation='softmax'))


# Compile model and check for formatting errors, define the
# loss function, the optimizer, learning rate, and the
# metrics (accuracy).
cnn_model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(
        lr=LEARNING_RATE,
        beta_1=0.9,
        beta_2=0.999
    ),
    metrics = ['accuracy']
)
```

- **Callbacks on Metrics**

```
early_stopping = EarlyStopping(
    patience=10,
    min_delta=0.001,
    restore_best_weights=True
)

learning_rate_reduction = ReduceLROnPlateau(
    monitor='val_mean_squared_error',
    patience=3,
    verbose=1,
    factor=0.5,
    lr=LEARNING_RATE
)
```

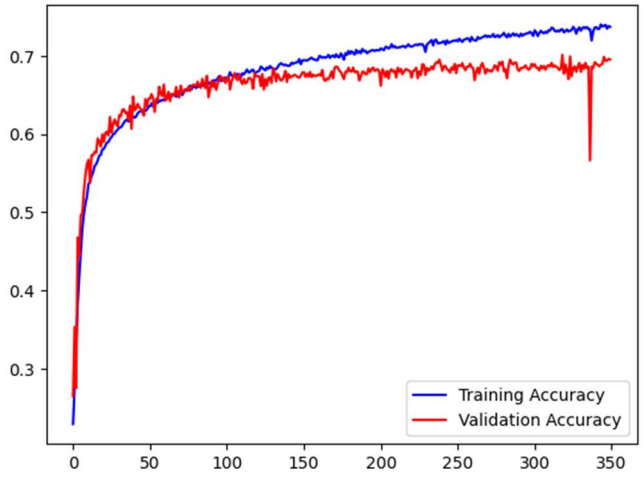
- **Evaluate the Models**

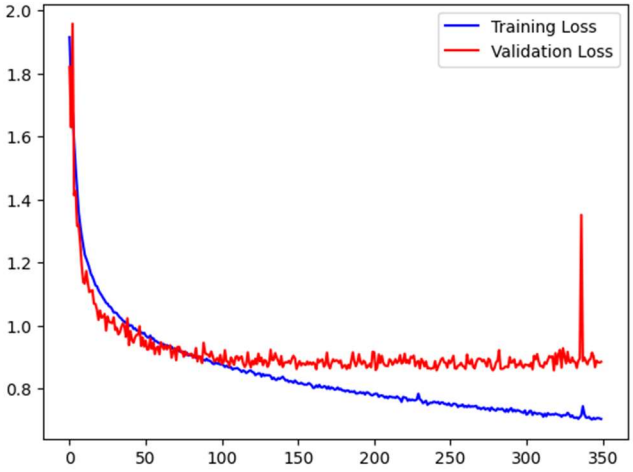
```
score = cnn_model.evaluate(x_test, y_test)
plt.plot(history.history['loss'], color='b')
plt.plot(history.history['val_loss'], color='r')
plt.legend(['Training Loss', 'Validation Loss'], loc='upper right')
plt.show()
plt.plot(history.history['accuracy'], color='b')
plt.plot(history.history['val_accuracy'], color='r')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()
```

CHAPTER 6

SCREENSHOTS AND RESULTS

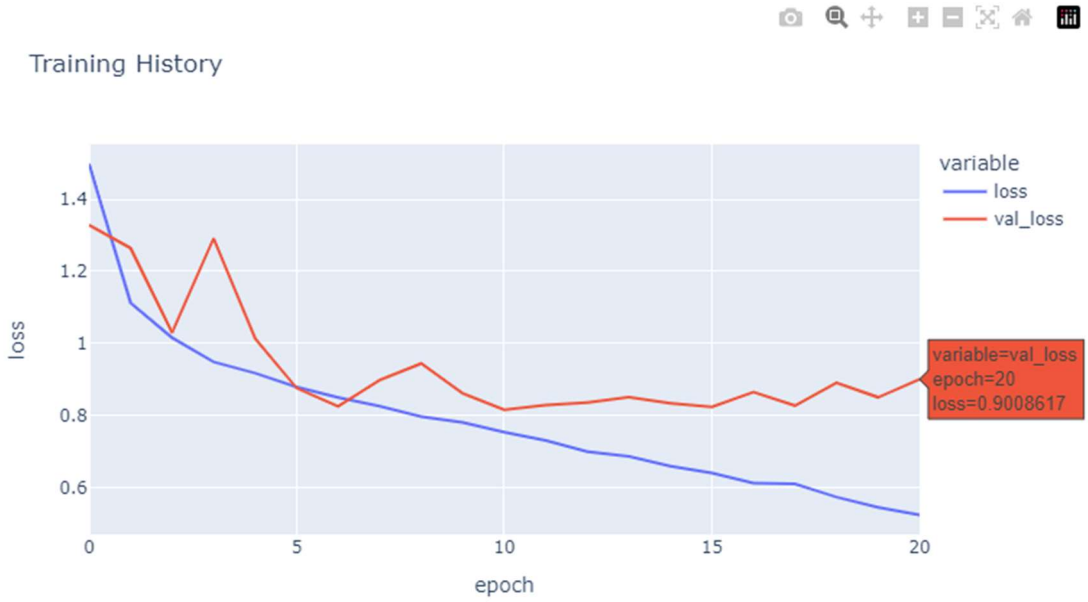
Emotion Model Metrics





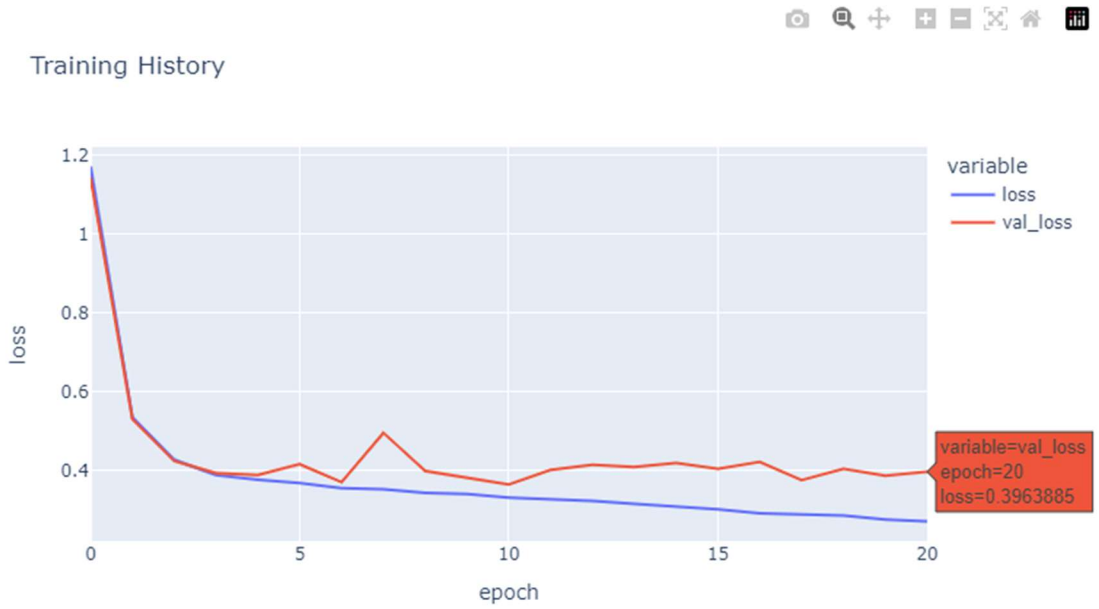

```
Test loss: 0.8849554657936096
Test Accuracy: 0.6951797008514404
```

Age Model Metrics



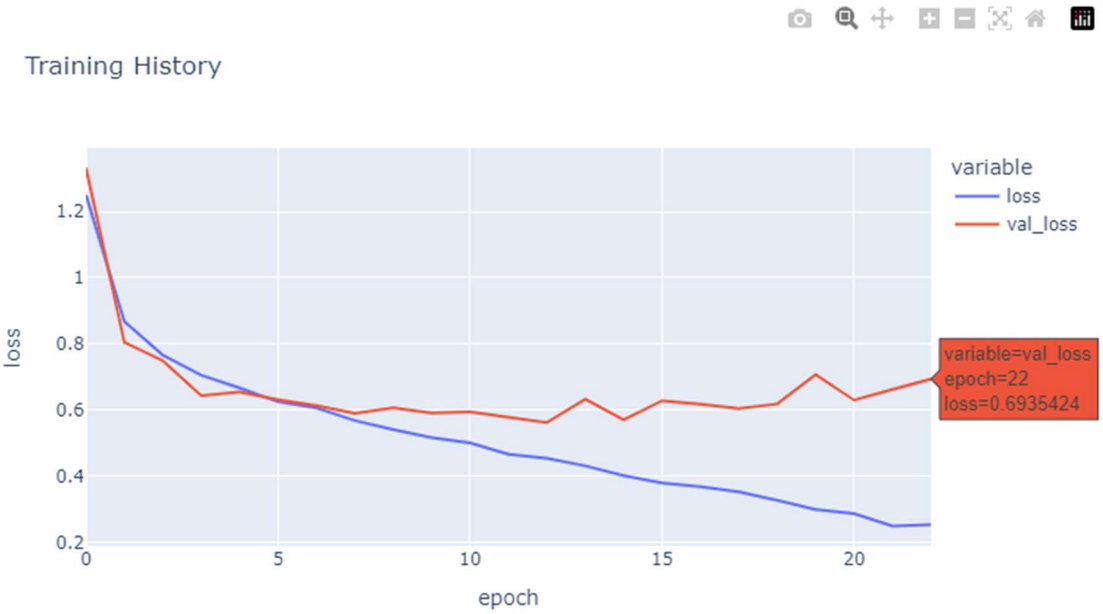

```
Test loss: 0.8167858719825745
Test Accuracy: 0.6589326858520508
```

Gender Model Metrics



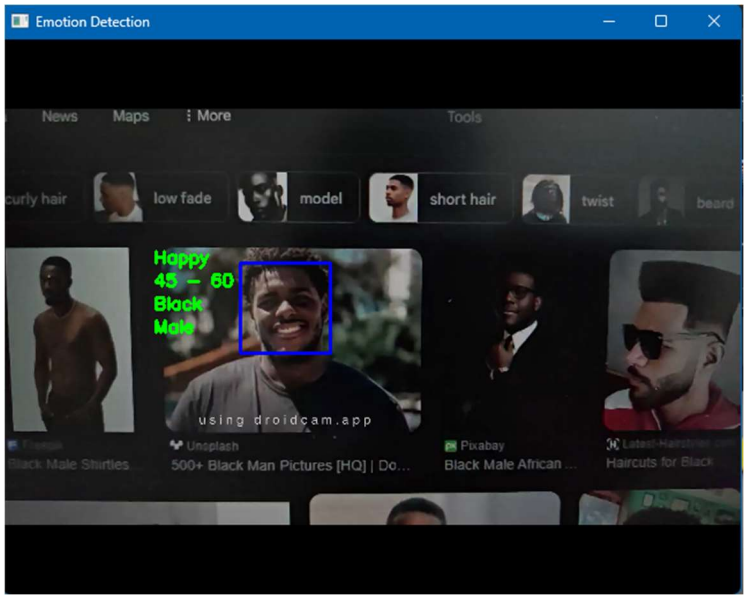

```
Test loss: 0.36459702253341675
Test Accuracy: 0.8842016458511353
```

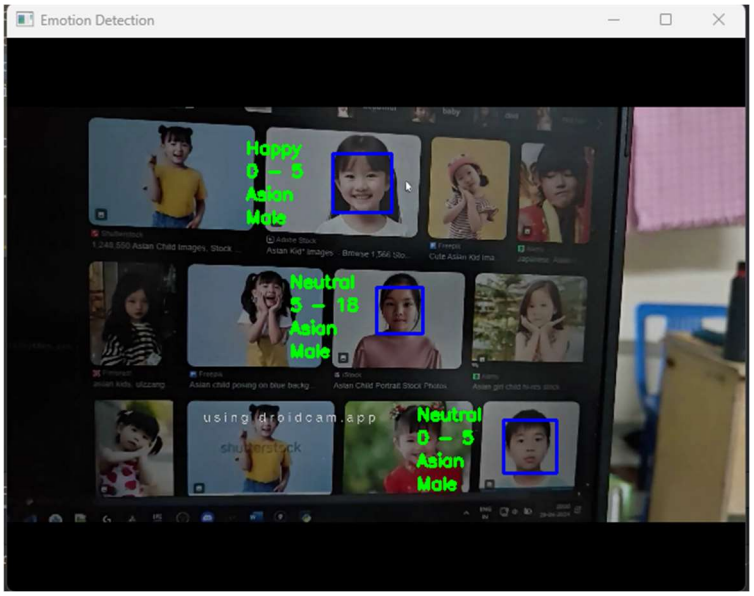
Ethnicity Model Metrics

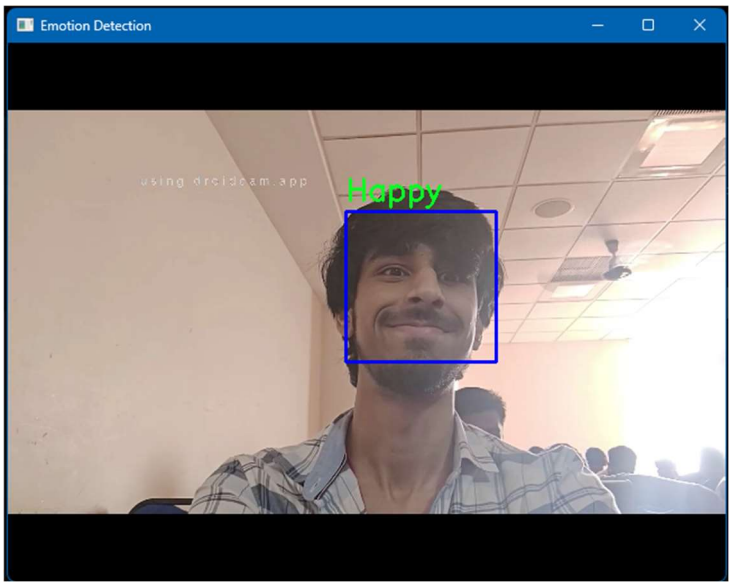


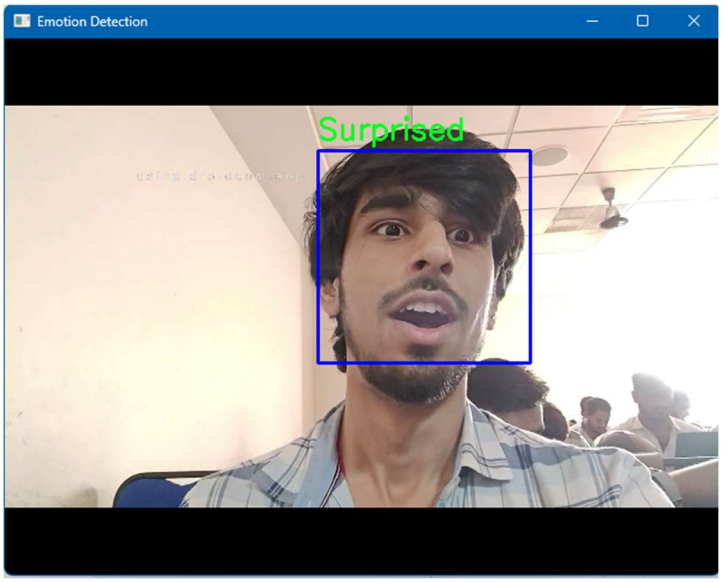

```
Test loss: 0.56186842918396
Test Accuracy: 0.8135414719581604
```

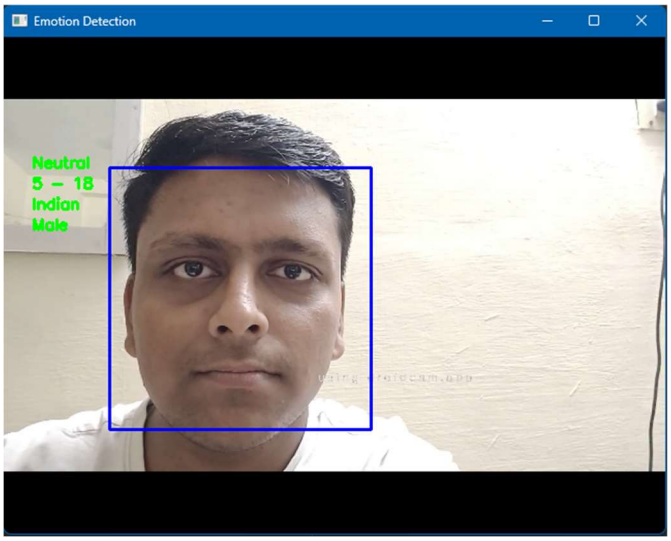
Application of all the models using live Camera feed











CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

This project investigated the development of a deep learning model for multi-faceted facial analysis, capable of predicting a person's emotions, age, gender, and ethnicity. We explored separate Convolutional Neural Network (CNN) architectures for each prediction task, leveraging techniques like batch normalization, dropout layers, and hyperparameter tuning to optimize model performance. The model was trained on a benchmark facial analysis dataset and evaluated on unseen test data.

The achieved accuracy metrics provide insights into the effectiveness of the proposed approach. While the model demonstrates promising results for emotion, age, and gender prediction, further exploration is necessary to refine ethnicity prediction due to the inherent complexities of this task.

Future Enhancements

Several avenues exist for future enhancements to this research:

1. **Ensemble Learning:** Investigate the potential benefits of combining the predictions from individual models (emotion, age, gender, ethnicity) through ensemble learning techniques like stacking or voting. This could potentially improve overall accuracy and robustness.
2. **Data Augmentation:** Explore the effectiveness of advanced data augmentation techniques like random rotations, scaling, and color jittering to increase the size and diversity of the training data. This can potentially lead to improved model generalizability, making it perform better on unseen data.
3. **Attention Mechanisms:** Implement attention mechanisms within the CNN architectures. These allow the model to focus on specific facial regions most relevant for each prediction task, potentially leading to more robust feature extraction and improved performance.
4. **Addressing Bias:** Conduct a thorough analysis of potential biases within the training data for ethnicity prediction. This may involve employing techniques like data balancing and fairness-aware training algorithms to mitigate biased predictions and ensure ethical considerations are addressed.
5. **Real-world Applications:** Integrate the developed model into real-world applications, considering ethical implications and user privacy concerns. Potential applications include:
    - Human-computer interaction systems that adapt interfaces based on user emotions.
    - Marketing strategies that leverage customer demographics and sentiment analysis from facial expressions.
    - Medical diagnosis systems that can identify potential health issues based on facial features (e.g., signs of fatigue or stress)

REFERENCES

- *Ekman, P., & Friesen, W. V. (1978).* Facial action coding system: A neurocultural perspective. Palo Alto, CA: Consulting Psychologists Press.

- *Guo, Y., Zhang, J., Xu, Y., & Sun, H. (2016, May).* Learning deep convolutional networks for age estimation from face images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) *(pp. 1124-1132)*. https://ieeexplore.ieee.org/document/8285381

- *Levi, G., Hassner, T., & Wolf, L. (2015).* Emotion recognition in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) *(pp. 859-867)*. https://dl.acm.org/doi/10.1145/2818346.2830587

- *Liu, A., Luo, Y., Wang, X., & Tang, X. (2017).* Deep learning face attributes prediction with hierarchical multi-task learning. IEEE Transactions on Multimedia, 19(10), 2487-2497. https://ieeexplore.ieee.org/document/8852046

- *Chollet, F. (2015).* Keras [Computer software]. Retrieved from https://keras.io/

- *Paszke, A., Zhang, E., Yang, Y., Dean, W., & Goodfellow, I. (2016).* Automatic differentiation in PyTorch. https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

- *Sun, Y., Wang, X., & Tang, X. (2014).* Deep learning face representation by joint learning with pooling and convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) *(pp. 1891-1899)*. https://proceedings.neurips.cc/paper_files/paper/2014/hash/e5e63da79fcd2bebbd7cb8bf1c1d0274-Abstract.html

- *Tobin, J., Wu, Y., Bernard, Y., Fei-Fei, L., & Gebru, T. (2022).* A benchmark for fairness in facial analysis datasets. arXiv preprint arXiv:2201.08712.

- *Zhao, X., Wu, X., Zhou, J., Meng, M., Zhao, Q., & Wang, Y. (2019).* Emotion recognition from long short-term memory recurrent neural network. Sensors, 19(8), 1895. https://www.mdpi.com/1424-8220/22/8/2976

- *Buolamwini, J., & Gebru, T. (2018)*. Gender shades: Intersectional accuracy disparities in commercial gender classification systems. In Proceedings of the 1st Conference on Fairness, Accountability, and Transparency *(pp. 77-91)*. https://proceedings.mlr.press/v81/buolamwini18a.html