A

CASE STUDY REPORT

ON

“UTKFace Gender Estimator”

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Gunti Sai Pranitha **2453-22-733-148**

Under the guidance

Of

Dr. VAISHALI PURUDE

Assistant Professor,

Department of Computer Science and Engineering

AFFILIATED TO OSMANIA UNIVERSITY HYDERABAD

1. INTRODUCTION

1.1 Problem Statement

Gender detection from facial images is an essential task in computer vision with applications in security, surveillance, marketing, and human-computer interaction. The objective of this project is to build a deep learning model that accurately predicts the gender of a person based on their facial image using the UTKFace dataset. The model should learn and generalize from facial features to distinguish between male and female individuals efficiently.

1.2 Motivation

With the rapid advancement in artificial intelligence and machine learning, automated gender recognition systems have become increasingly valuable in various fields. Manual gender identification can be subjective and error-prone, especially in large-scale applications. This project is motivated by the need to develop an efficient, automated, and reliable method to classify gender from facial images, leveraging the UTKFace dataset and convolutional neural networks (CNNs). It also provides hands-on experience with multi-output models and facial feature learning.

2. PROPOSED MODEL

Algorithm:

The proposed model uses a **Convolutional Neural Network (CNN)** to detect the **gender** of a person from facial images in the **UTKFace dataset**. The key steps are:

2.1 Data Preprocessing:

- Extract facial images and **gender labels** from file names in the dataset.
- Convert images to **grayscale** and resize to a fixed dimension of **128x128**.
- Normalize pixel values and reshape the data for CNN input.

2.2 CNN Model Architecture:

- The model consists of **4 convolutional layers**, each followed by **max pooling**, to extract spatial features.
- A **Flatten layer** converts features into a 1D vector.
- A **Dense layer with ReLU activation** is used for learning.
- A final **Dense layer with Sigmoid activation** outputs the gender prediction (0 = Male, 1 = Female).

2.3 Training Setup:

- **Loss Function:** Binary Cross-Entropy
- **Optimizer:** Adam
- **Metric:** Accuracy

The training of the gender classification model was collaboratively carried out by three team members to observe performance improvement over different training durations:

- **Team Member 1** trained the model for **10 epochs** to observe early convergence behavior.
- **Team Member 2** extended training to **20 epochs** to analyze stability and improvements in accuracy.
- **Team Member 3** trained the model for **30 epochs**, achieving the **best validation accuracy and lowest loss**, indicating optimal performance.

This step-by-step training by different members allowed us to compare learning trends, avoid overfitting, and finalize the most effective training strategy.

2.4 Model Architecture Flow:

3. REQUIREMENTS

3.1 Software Requirements:

To implement and run the deep learning model efficiently, the following software tools and libraries were used:

Programming Language: Python 3.7 or higher

Development Environment: Jupyter Notebook / Visual Studio Code

Libraries:

- TensorFlow and Keras – for designing and training the CNN model
- NumPy and Pandas – for data handling and preprocessing
- PIL – for image reading and manipulation
- Matplotlib and Seaborn – for visualization of training metrics

3.2 Hardware Requirements:

The system specifications required to handle image processing and training tasks are as follows:

Processor: Intel i5 or higher

RAM: Minimum 8 GB (Recommended 16 GB for faster performance)

GPU (Optional): NVIDIA GPU with CUDA support for faster model training (if available)

3.3 Dataset Requirements:

The dataset used for this project is:

Dataset Name: UTKFace

Source: Kaggle - UTKFace Dataset

Description: The dataset contains over 20,000 facial images with labels for **Age, Gender, and Ethnicity**.

Usage in Project: For this case study, only the **Gender** label was extracted and used to train the binary classification model (Male/Female).

Image Format: JPEG files with filenames in the format age_gender_race_date.jpg (e.g., 25_1_0_20170116.jpg)

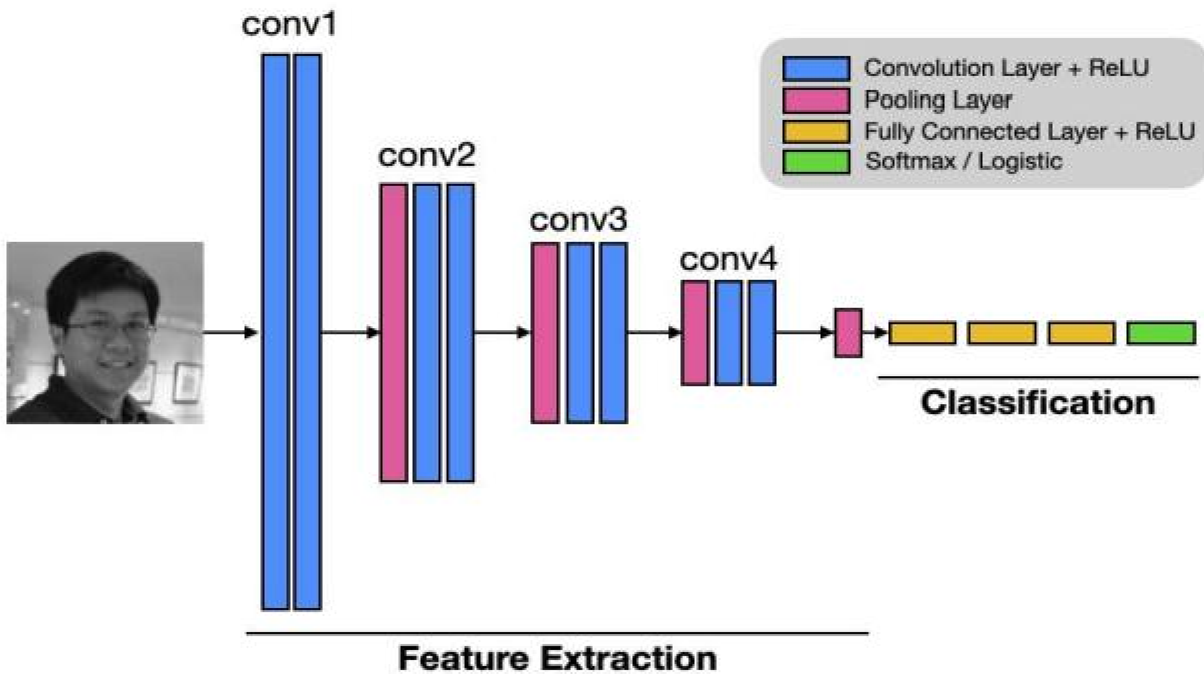
4. IMPLEMENTATION

4.1 Data Collection and Preparation:

- The **UTKFace dataset** was downloaded from **Kaggle**, containing facial images labeled with age, gender, and ethnicity.
- Only the **gender label** (0: male, 1: female) was extracted from image filenames.
- Images were **preprocessed** by:

1. Converting to **grayscale** to reduce complexity.
2. Resizing to a fixed size (e.g., 48x48 pixels)
3. Normalizing pixel values to the range [0,1]
4. Splitting the dataset into training and testing sets (e.g., 80% train, 20% test)

4.2 Model Architecture:



A convolutional Neural Network (CNN) was built using Keras:

- **Input Layer:** Accepts preprocessed grayscale images.
- **Convolutional Layers:** Extract spatial feature using filters
- **Pooling Layers:** Reduce dimensionality using MaxPooling
- **Flatten Layer:** Converts 2D features into 1D vector
- **Dense Layer**: Fully connected layers for learning complex patterns.
- **Output Layer:** Single neuron with sigmoid activation for binary classification(male/female)

4.3 Model Training:

- The model was compiled with:

Loss function: Binary Crossentropy

Optimizer: Adam

Metrics: Accuracy

- The model was trained using the training set:

One member with 10 epochs, another for 20 epochs and final training was completed for 30 epochs.

- The performance was monitored using training and validation accuracy and loss.

4.4 Evaluation and Prediction:

- After training, the model was evaluated on the test set.
- Metrics such as **accuracy, confusion matrix, and loss curve** were analyzed.
- Finally, the model was used to **predict gender** on new unseen images.

CODE:

```
input_shape = (128, 128, 1)
inputs = Input(shape=input_shape)

x = Conv2D(32, (3, 3), activation='relu')(inputs)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation='relu')(x)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(128, (3, 3), activation='relu')(x)
x = MaxPooling2D((2, 2))(x)
x = Conv2D(256, (3, 3), activation='relu')(x)
x = MaxPooling2D((2, 2))(x)
x = Flatten()(x)

# Two branches: Gender (classification) and Age (regression)
fc_gender = Dense(256, activation='relu')(x)
fc_gender = Dropout(0.3)(fc_gender)
out_gender = Dense(1, activation='sigmoid', name='gender_out')(fc_gender)

fc_age = Dense(256, activation='relu')(x)
fc_age = Dropout(0.3)(fc_age)
out_age = Dense(1, activation='relu', name='age_out')(fc_age)

model = Model(inputs=inputs, outputs=[out_gender, out_age])

model.compile(optimizer='adam',
              loss=['binary_crossentropy', 'mae'],
              metrics={'gender_out': 'accuracy', 'age_out': 'mae'})

model.summary()
```

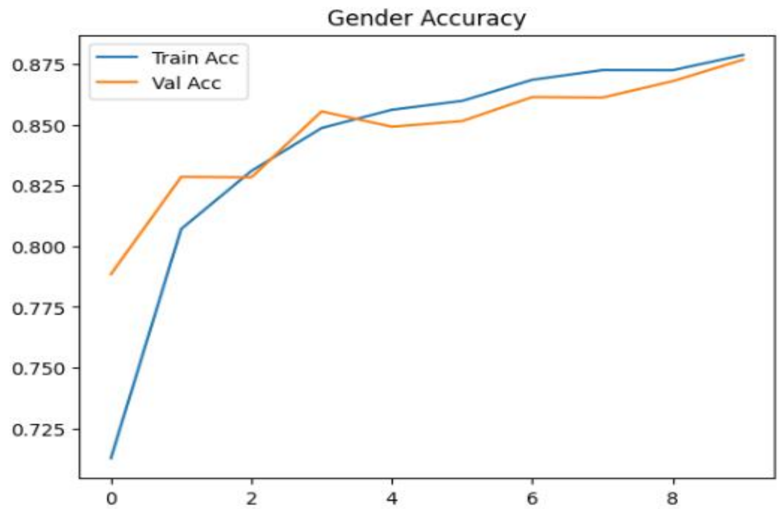
Convolutional Neural Network Code.

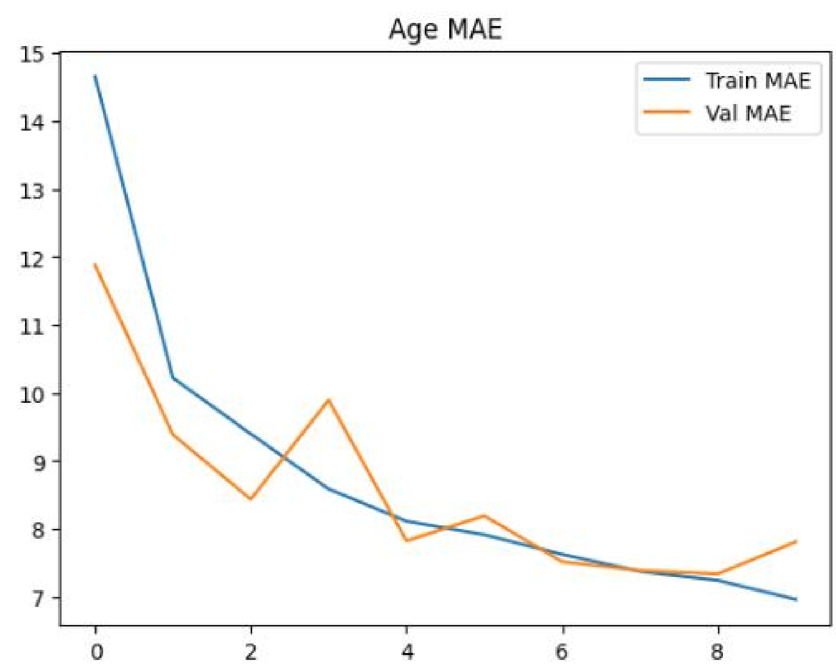
5. OUTPUT SCREENSHOTS

```
sample_idx = 1000
img = X[sample_idx]
true_age = y_age[sample_idx]
true_gender = gender_dict[y_gender[sample_idx]]

pred_gender, pred_age = model.predict(img.reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred_gender[0][0])]
pred_age = round(pred_age[0][0])

print(f"Original → Gender: {true_gender}, Age: {true_age}")
print(f"Predicted → Gender: {pred_gender}, Age: {pred_age}")
plt.imshow(img.reshape(128, 128), cmap='gray')
plt.axis('off')
plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step
Original → Gender: Male, Age: 43
Predicted → Gender: Male, Age: 50
```

```python
plt.plot(history.history['gender_out_accuracy'], label='Train Acc')
plt.plot(history.history['val_gender_out_accuracy'], label='Val Acc')
plt.title('Gender Accuracy')
plt.legend()
plt.show()

# MAE Plot for Age
plt.plot(history.history['age_out_mae'], label='Train MAE')
plt.plot(history.history['val_age_out_mae'], label='Val MAE')
plt.title('Age MAE')
plt.legend()
plt.show()
```





6. ADVANTAGES/ DISADVANTAGES

Advantages:

1. High Accuracy: CNN models can extract spatial features effectively, resulting in good accuracy in gender classification.

2. Automated Feature Extraction: No need for manual feature engineering; CNN’s learn relevant features directly from images.

3. Scalability: The model can be scaled to detect other facial attributes like age or emotion with slight modifications.

4. Real-World Applications: Useful in fields like surveillance, targeted advertising, and identity verification systems.

Disadvantages:

1. Dataset Bias: If the dataset has imbalanced gender or ethnicity distribution, the model may show biased predictions.

2. Overfitting Risk: If not properly validated, the model may memorize training data instead of generalizing.

3. High Computational Requirements: Training CNNs requires significant computational power, especially for large datasets.

4. Privacy Concerns: Using facial data may raise ethical and privacy issues in real-world applications.

7. USE OF PROJECT

The main purpose of this project is to develop a deep learning model that can automatically **predict the gender (male or female)** from facial images using the UTKFace dataset. This type of system has wide applications in:

Smart Surveillance: Helps in identifying and classifying individuals in security systems.

Personalized Applications: Useful in tailoring user experiences in apps based on gender.

Marketing & Analytics: Enables demographic analysis for targeted advertising.

Authentication Systems: Enhances accuracy in face-based login or verification methods.

By training a **Convolutional Neural Network (CNN)**, the project demonstrates how machine learning can be applied to facial recognition tasks, contributing to the growing field of **computer vision and AI in biometrics**.

8. CONCLUSION

In this project, we successfully implemented a Convolutional Neural Network (CNN) to classify gender using facial images from the UTKFace dataset. The model was trained and evaluated to distinguish between male and female faces, achieving satisfactory accuracy. This work highlights the effectiveness of deep learning techniques in solving real-world classification problems such as gender detection. Overall, the project provides a foundation for further development in facial analysis systems and shows potential for applications in security, personalization, and demographic analytics.

9. REFERENCES

- Sandeep Kumar, Sukhwinder Singh & Jagdish Kumar, “A Study on Face Recognition Techniques with Age and Gender Classification ” In IEEE International Conference on Computing, Communication and Automation (ICCCA), pp. 1001 - 1006, May 2017

https://scholar.google.com/scholar?as_q=A+Study+on+Face+Recognition+Techniques+with+Age+and+Gender+Classification&as_occt=title&hl=en&as_sdt=0%2C31

- Shan, C., Gong, S., & McOwan, P. W. (2008). Fusing gait and face cues for human gender recognition. Neurocomputing, 71 (10-12), 1931 - 1938.

https://scholar.google.com/scholar?as_q=Fusing+gait+and+face+cues+for+human+gender+recognition&as_occt=title&hl=en&as_sdt=0%2C31

- Human Gender Classification using Machine Learning - Vaishnavi Y. https://www.researchgate.net/publication/342115800_Human_Gender_Classification_using_Machine_Learning

- Human Gender Identification Employing Convolution Neural Networks for Veiled Face Images 30 ` Human Gender Identification Employing Convolution Neural Networks for Veiled Face Image

https://www.researchgate.net/publication/387501470_Human_Gender_Identification_Employing_Convolution_Neural_Networks_for_Veiled_Face_Images_30_Human_Gender_Identification_Employing_Convolution_Neural_Networks_for_Veiled_Face_Image