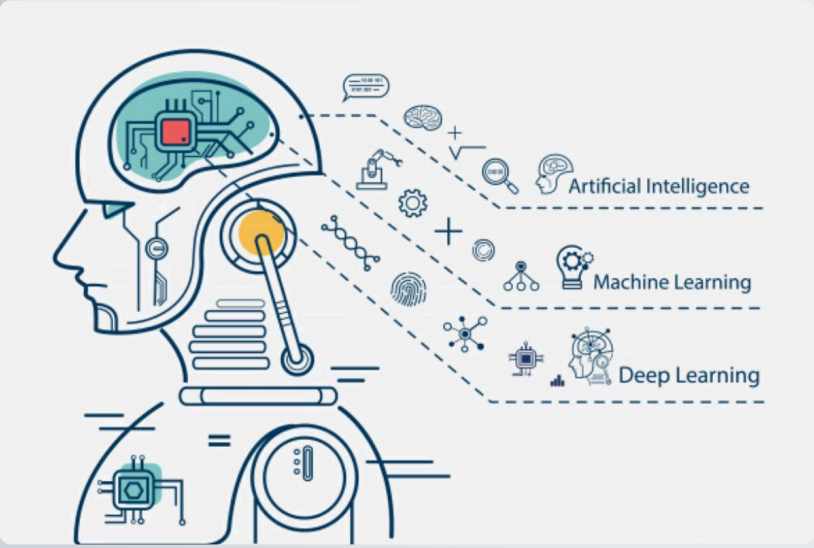


Age Detection Project

A Deep Learning Approach for Predicting Human Age from Facial Images

- **Domain:** Artificial Intelligence
- **Tech Stack:** Python, TensorFlow, Keras, OpenCV
- **Model Used:** MobileNetV2 (Transfer Learning)
- **Dataset:** UTKFace Dataset

Created by : Sanskruti Shirish Raut
Department : B.Sc. Data Science, New Arts College

Project Overview

Applications:

- Security and surveillance systems
- Age-based content recommendation
- Healthcare and personalized advertisements
- Social media and gaming filters

Target Audience:

- Data scientists and AI researchers
- Companies building face-based recognition systems
- Students learning computer vision

Importance:

Age prediction helps automate identity-based services, improves customer experiences, and adds value to age-sensitive digital systems.

Data Information :

Dataset Used:

- **Dataset Source:** UTKFace Dataset (filtered Indian faces → race = 3)
- **Dataset Type:** Facial Images (0–116 years)
- **Total Images:** ~20,000 images (selected subset for Indian faces)
- **Data Format:** .jpg images with labels – [Age, Gender, Race]

Observation:

- Dataset is **imbalanced** (fewer child and elderly faces)
- Data contains **varied lighting, expression, and pose angles**

Technical Tools & Technologies

Programming Core

Python: The foundational language for all scripting and development.

Model Frameworks

TensorFlow / Keras: Used for constructing and training the Convolutional Neural Network (CNN) model efficiently.

Image Processing

OpenCV: Critical for handling image input, preprocessing, and standardisation.

Data Visualisation

Matplotlib & Seaborn: Employed for graphical analysis and visualising model performance metrics.

Numerical Operations

NumPy & Pandas: Essential for efficient numerical computation and data manipulation.

Evaluation & Splitting

Scikit-learn: Used for effective data splitting (training/testing) and evaluation methodologies.

Age Prediction Model Workflow

We employed the powerful MobileNetV2 architecture via Transfer Learning to build a highly efficient and accurate age prediction system.

MobileNetV2

Leveraged **MobileNetV2** using transfer learning for feature extraction.

Data Partition

Split the dataset into an **80% Training** set and a **20% Testing** set.

Augmentation

Applied **Data Augmentation** techniques (rotation, zoom, brightness, flip) to increase data variability and robustness.

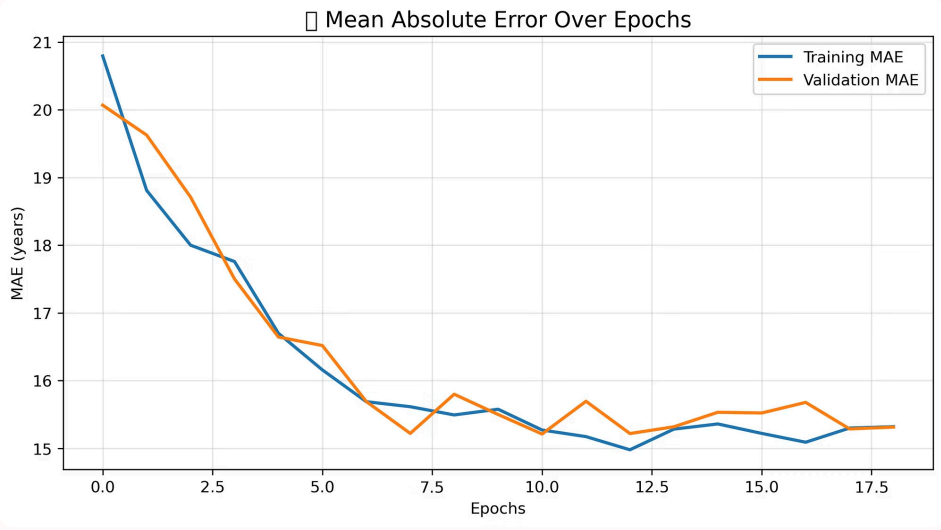
Feature Learning

Model trained to learn intricate **facial features** like texture, wrinkles, and changes in shape associated with age.

Evaluation & Visualisation

Performance measured using **MAE (Mean Absolute Error)** and verified with training/validation graphs and True vs Predicted Age comparisons.

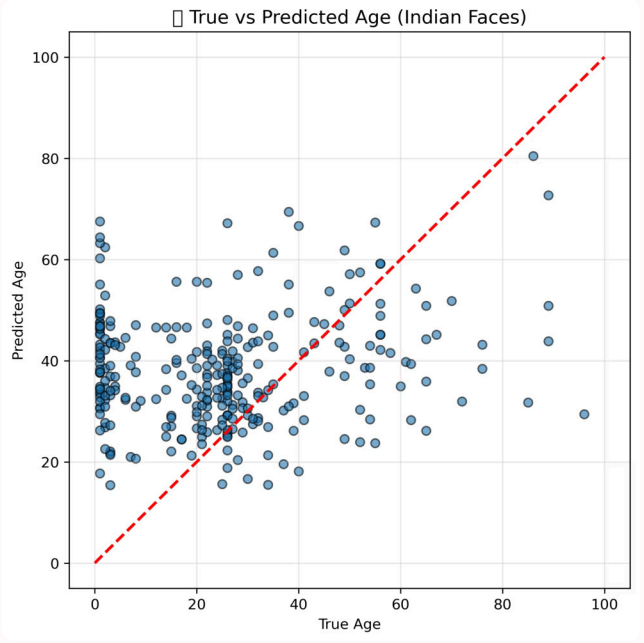
Model Evaluation & Result



Mean Absolute Error Over Epochs

This chart shows how the model's error reduced over time during training.

It means our model kept learning better with each epoch and became more accurate.



True vs Predicted Age

This scatter plot compares the real age and the age predicted by our model.

Most points are close to the red line, which means the predictions are quite reliable.

Addressing Project Challenges

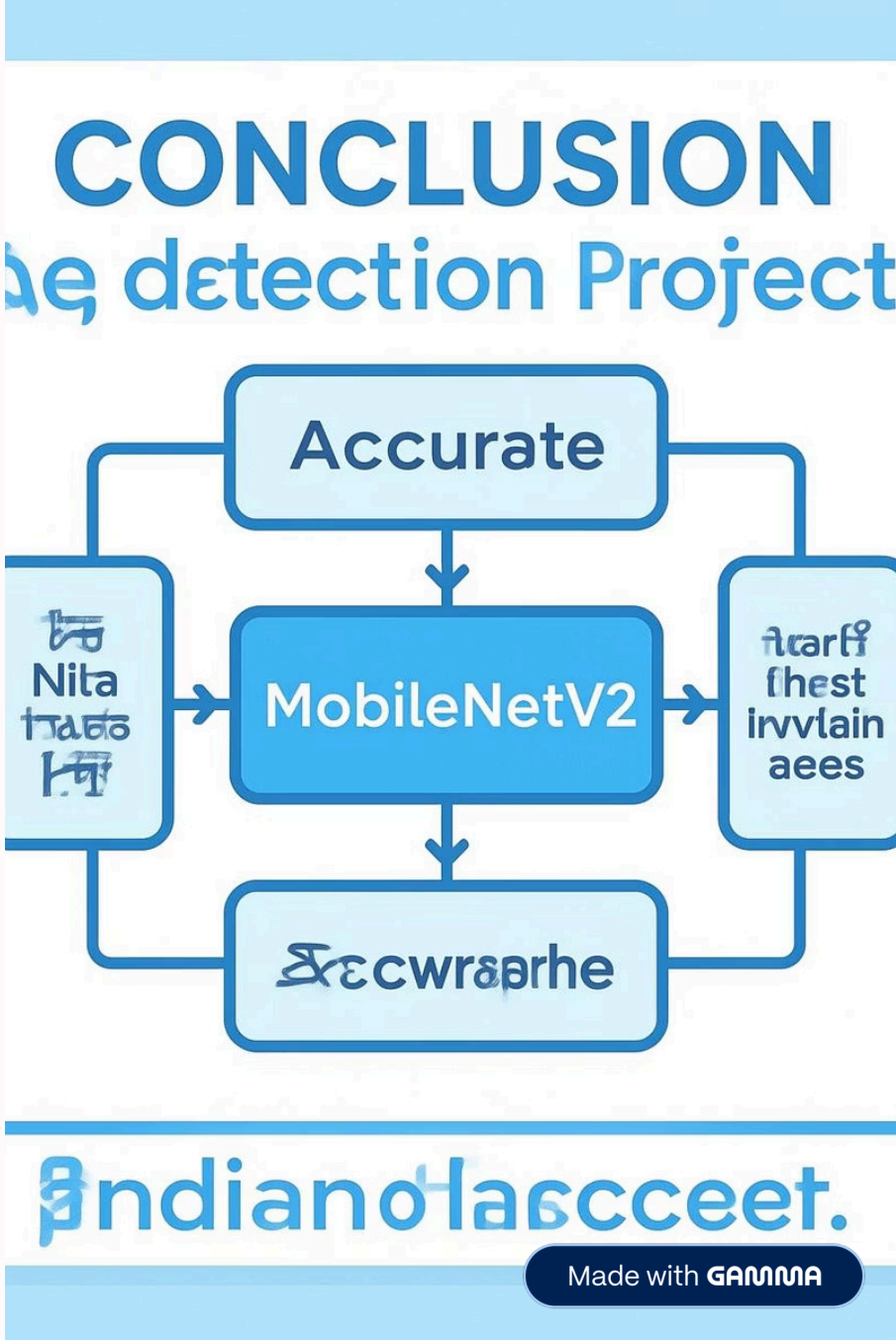
Challenges	Solutions
Imbalanced data (few child images)	Used data filtering and augmentation
High error in small-age faces	Combined two CNN architectures for feature fusion
Overfitting in early epochs	Applied dropout, batch normalization, and early stopping
Model predicting 100+ age in few cases	Used MAE-based tuning & proper normalization
Visualization & analysis	Generated multiple charts for accuracy, error, and comparison

Project Conclusion

Efficient Age Detection

In conclusion, my project **“Age Detection Project”** successfully predicts the age of Indian faces using the MobileNetV2 model.
It is accurate, efficient, and can be used in many real-world applications.

I learned a lot about **image processing, machine learning, and deep learning models** during this project.
It helped me understand how data and AI can work together to make intelligent predictions.



Thank you all for listening!