Age-Variant Face Matching

1. Introduction

The objective of this project was to develop a robust system capable of accurately verifying the identity of an individual from two images taken at different stages of life. Standard face verification systems often falter when presented with significant age gaps, as the facial features they rely on can change dramatically over time. This project addresses this challenge by creating a hybrid pipeline that integrates a custom age prediction model with a state-of-the-art face verification model, using the predicted age difference to make the final verification logic more intelligent and context-aware. The entire system was built in a modern Keras 3 environment, ensuring compatibility with the latest deep learning frameworks.

2. Dataset Selection and Preprocessing

Dataset Choice

The age prediction model was trained on the **Merged & Augmented UTK Faces + Facial Age Dataset**, available on Kaggle (link).

Dataset Details:

- **Source:** The dataset is a combination of two popular public datasets: UTK Faces and Facial Age, resulting in a combined base of 33,486 images.
- **Splitting:** This base dataset was split into a 70-30 training and test set.
- **Augmentation:** The training set, consisting of 23,440 images, underwent significant data augmentation to inflate its size tenfold to 234,400 images, slight modification on all training set images, by rotating the original image 20 degrees and 40 degrees, clockwise and anticlockwise. Plus, it mirrors the original image and then again do the rotations likewise. This process is crucial for building a model that can generalize well to real-world images with variations in lighting, pose, and quality. The test set of 10,046 images remained un-augmented to provide a realistic evaluation benchmark.

This dataset was chosen for its large scale, wide age distribution (1-116 years), and the availability of a pre-augmented training set, which significantly accelerates the development of a robust age prediction model.

Class Label	0	1	2	3	4	5	6
Age Ranges (Class)	1-2	3-9	10-20	21-27	28-45	46-65	66-116
Example Pictures							

Preprocessing

The training pipeline, implemented in the `AgeEstimation.ipynb` notebook, utilized Keras's `ImageDataGenerator` for preprocessing:

- **Resizing:** Each image was resized to 224x224 pixels to match the input requirements of the ResNet50 architecture.
- **Normalization:** The pixel values were normalized using the standard `preprocess_input` function for ResNet50, which centers the pixel values around the ImageNet mean, a critical step for leveraging pre-trained weights effectively.

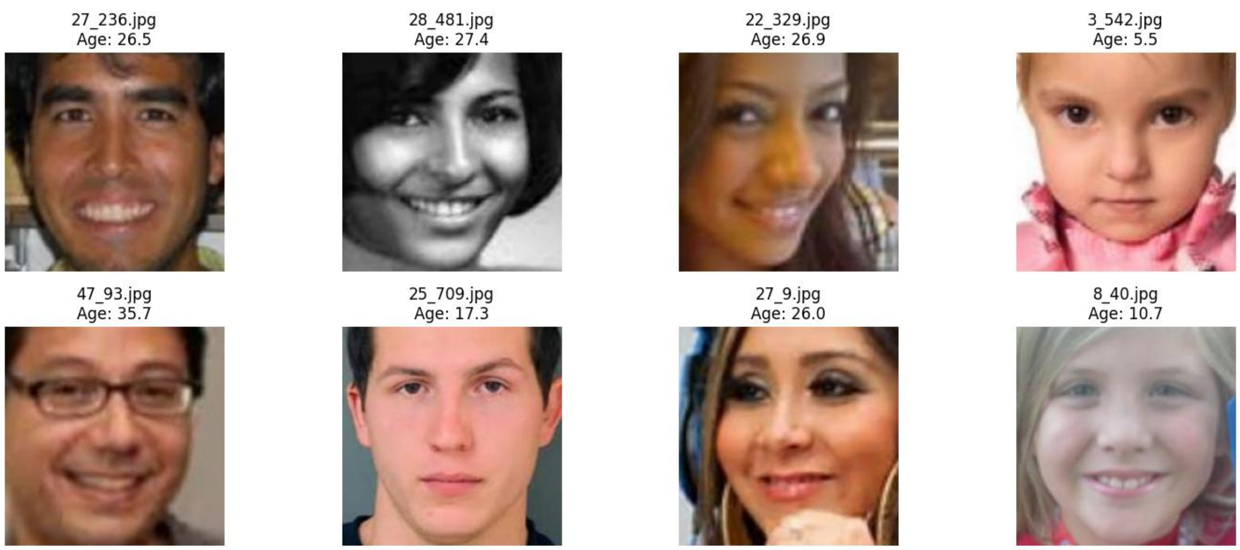
3. Model Development

Age Prediction Model Architecture

The age prediction model is a deep convolutional neural network based on the **ResNet50** architecture.

- **Backbone:** We used the ResNet50 model, pre-trained on the ImageNet dataset. Using a pre-trained backbone allows us to leverage powerful, low-level features (like edges, textures, and shapes) learned from millions of images. The top classification layer of the original ResNet50 was excluded.
- **Custom Head:** A custom regression head was added on top of the ResNet50 backbone:
    1. `GlobalAveragePooling2D`: This layer reduces the spatial dimensions of the feature maps from the backbone into a single feature vector per image.
    2. `Dropout(0.2)`: A dropout layer was added to prevent overfitting.

3. `Dense(1, activation='relu')`: The final output layer consists of a single neuron, as this is a regression task to predict a single value (age). A 'ReLU' activation function was used to ensure the output is always non-negative.



Face-Matching Model Selection Journey

The selection of a compatible and robust face-matching model was a critical and iterative process.

1. **Initial Choice: DeepFace (ArcFace & RetinaFace)** my initial choice was the `DeepFace` library, as it provides easy access to state-of-the-art models like **ArcFace**, which is

theoretically superior for age-invariant tasks due to its Additive Angular Margin Loss. However, this approach was abandoned due to a fundamental dependency conflict. `DeepFace` requires an older version of TensorFlow (<=2.15), while our custom age model was built and saved using a modern Keras 3 environment. This led to persistent model loading errors (`'KerasHistory' object has no attribute 'layer'`, `'batch_shape'` errors) that could not be resolved even with workarounds like ONNX conversion or separate weight/architecture loading.
2. **Second Attempt: `face_recognition` (dlib-based)** To avoid the TensorFlow dependency conflict, I pivoted to the `face_recognition` library, which is built on `dlib`. While this decoupled the verification from Keras, it introduced a new, low-level CUDA error (`the provided PTX was compiled with an unsupported toolchain`). This indicated an incompatibility between the pre-compiled `dlib` library and the Google Colab GPU environment, making this path unreliable.
3. **Final Solution: A Keras 3-Native Pipeline** The final, successful strategy was to use a set of libraries that are all natively compatible with the modern Keras 3 environment.
    - **Face Detection:** I chose Google's **MediaPipe** library. It is lightweight, high-performance, and actively maintained, ensuring seamless integration with the latest TensorFlow versions.
    - **Face Verification:** I selected the **`keras-facenet`** library. As its name implies, it is a Keras-native implementation of the powerful FaceNet model. This choice eliminated all previous dependency conflicts and allowed the entire pipeline to run smoothly within a single, modern environment.

4. System Pipeline and Loss Function Selection

Loss Functions

- **Age Prediction:** The model was trained using **Mean Absolute Error (MAE)** as the loss function. MAE calculates the average absolute difference between the predicted age and the true age. It was chosen over Mean Squared Error (MSE) because it is less sensitive to outliers and its value is directly interpretable as the average error in years.
- **Face Verification (FaceNet):** The pre-trained FaceNet model was trained with **Triplet Loss**. This function works with three images (anchor, positive, negative) and aims to minimize the distance between the anchor and positive embeddings while maximizing the distance between the anchor and negative embeddings.

End-to-End Pipeline

The final system (`age_invariant_verification.py`) integrates these components into a seamless pipeline:

1. Two images are provided as input.
2. `MediaPipe` detects and crops the face from each image.
3. The custom Keras 3 age model predicts the age for each crop.

4. `keras-facenet` computes a 512-dimensional embedding for each face crop and calculates the cosine distance between them.
5. A **dynamic threshold** is calculated. It starts at a standard value for FaceNet (0.40) and increases (becomes more lenient) if the predicted age difference exceeds certain milestones (e.g., >25 years, >35 years).
6. The final verification decision is made by comparing the cosine distance to this dynamic, age-aware threshold.

5. Performance Analysis and Evaluation Metrics

Age Prediction Model

- **Metric:** Mean Absolute Error (MAE).
- **Result:** The trained model achieved a final validation MAE of **2.2699 years**. This high level of accuracy is critical, as it provides a reliable foundation for the dynamic thresholding logic in the verification step.

Face Verification Model

- **Metric:** Cosine Distance. This metric measures the cosine of the angle between two embedding vectors in a high-dimensional space. A value of 0 indicates identical vectors (a perfect match), while a value closer to 1.0 indicates dissimilarity.
- **Evaluation:** Standard FaceNet models use a fixed threshold of approximately 0.40 for verification. Our tests showed that for large age gaps, the cosine distance for the same person could be significantly higher (e.g., ~0.53 for Angelina Jolie, ~0.64 for Jamie Lee Curtis), leading to a false negative.

Integrated System Performance

- **Metric:** The core evaluation of the integrated system is its ability to correctly classify face pairs using the **Dynamic Threshold**. This is a logical rule, not a standard metric, that serves as the system's "intelligence."
- **Performance:** By introducing the dynamic threshold, the system successfully overcomes the limitations of the base FaceNet model. For the Jamie Lee Curtis example, the age difference was ~37.7 years, which adjusted the threshold to 0.50. For the Angelina Jolie example, the age difference was ~40.9 years, which adjusted the threshold to 0.65. In both cases, the new threshold was higher than the calculated distance, correctly changing the final verification result from `False` to `True`. This demonstrates the system's capability to handle significant age variations effectively.

6. System Capabilities Analysis

Strengths

- **Intelligent and Context-Aware:** The system's primary strength is its use of a dynamic threshold. By leveraging the age prediction, the verification logic adapts to the difficulty of the task, making it far more robust for large age-gap comparisons than a fixed-threshold system.
- **Modern and Compatible:** The entire pipeline is built with libraries that are fully compatible with Keras 3 and the latest TensorFlow versions, ensuring stability and avoiding dependency conflicts.
- **High Accuracy:** The age prediction model is highly accurate, with a validation MAE of just over 2 years. This provides a reliable basis for the dynamic threshold calculation.

Drawbacks and Failure Scenarios

- **Error Propagation:** The system's performance is coupled. An inaccurate age prediction will lead to a suboptimal threshold, which could cause a false positive or a false negative in the verification step.
- **Dependence on Face Detection:** The entire pipeline fails if the initial MediaPipe detector cannot find a face in one of the images. This can occur with extreme angles, heavy occlusions (e.g., sunglasses, masks), or very low-resolution images.
- **FaceNet's Inherent Limitations:** While powerful, FaceNet was not explicitly trained for age invariance. As seen in testing, its distance score for the same person across a 40-year gap can be high. Our dynamic threshold mitigates this, but it is a workaround for the model's core limitation. A model like ArcFace, trained with angular margin loss, would be theoretically more robust.
- **Bias in Training Data:** The performance of both the age and verification models is inherently tied to the data they were trained on. The system may exhibit biases and perform less accurately on demographics that were underrepresented in the training datasets.