CYSHIELD

Face Matching Across Ages System

Technical Report

Computer Vision Assignment

Task: CYCV001

Submitted to

Cyshield – Computer Vision Team (Hiring Process)

Submitted by

Ahmed Gamal Noureddine

Email: eng.ahmed.gamal.cu@gmail.com

Date: 3 September 2025

Contents

1. Objective

This system determines whether two input photos belong to the same individual at different ages. The pipeline detects and crops faces (using MTCNN), predicts each face's age (using custom ResNet-50 age regressor trained on UTKFace dataset from scratch), produces 512-dimension face embeddings (using FaceNet /InceptionResnetV1 pretrained on VGGFace2), and decides "MATCHED / NOT MATCHED" using cosine similarity with a certain threshold. The design balances accuracy, speed, and implementation simplicity while remaining extensible for future improvements.

2. System Architecture and End-to-End Flow

2.1 High-Level Modules

- Input Handler: accepts exactly two input images (JPG/PNG).
- Face Detection & Cropping: MTCNN finds the largest face per image and returns a crop for each face.
- Age Prediction: custom from scratch ResNet-50 regresses age (in years) from the cropped face (trained on UTKFace dataset).
- Face Embedding: InceptionResnetV1 (FaceNet family, pretrained on VGGFace2) produces a 512-dimension embedding vector.
- Similarity & Decision: L2-normalize embeddings then compute cosine similarity using dot product between the two vectors, compare the similarity score with a threshold to output MATCHED/NOT MATCHED.
- Reporting: prints age1, age2, cosine similarity score (X%), and the final decision.

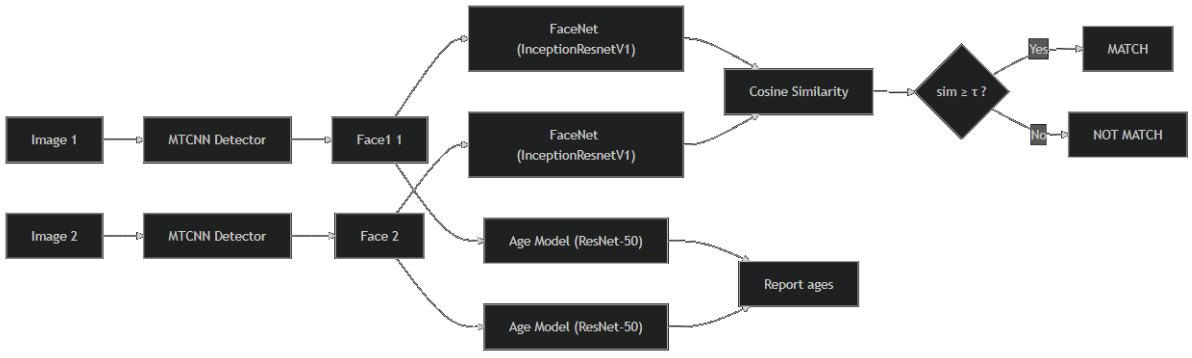

Figure 1. Full pipeline of the system

2.2 Data Flow (Input → Output)

1. Read two input images.

Figure 2. Two sample input for the same person in different ages

2. **Face Detection**: or each image, run MTCNN to get square cropped face (largest face in the image with a margin).

Figure 3. The two detected faces by MTCNN

3. **Age model**: process the 224×224 faces (with ImageNet mean/std) using custom from scratch ResNet-50 to get the age in years.
4. **Embedding model**: preprocess to 160×160 faces and outputs 512-dimension embedding vectors.
5. **Cosine similarity**: L2 normalize each embedding vector then dot product them to get similarity score (%).
6. **Decision**: if similarity score ≥ threshold output MATCHED else NOT MATCHED then report ages, similarity score,and decision.

```
First person is: 14.91 years old

Second person is: 19.78 years old

Similarity: 65.51%

Faces are MATCHED ... This is the same person
```

Figure 4. Matching system output

3. Dataset Selection & Preprocessing

3.1 Dataset: UTKFace

I used UTKFace dataset which has wide age coverage (children to elderly) and pruned to max of 60 years old to speed up training as I trained from scratch. It has variable pose, illumination, and background. But UTKFace dataset contains demographic imbalance.

3.2 Splits and Sampling

- Split: 80% training (18965 samples) and 20% validation (4742 samples).
- Sampling: Pruned the faces of ages higher than 60 years to speed up training and remove extreme outliers.

3.3 Preprocessing and Augmentation

- Training Data Augmentation:
  - Random Resized Cropping of size = 224 and scale = 0.9~1.0
  - Random Horizontal Flip with probability = 0.5
  - Color Jitter with brightness = 0.2, contrast = 0.2
  - Random Grayscale with probability = 0.1
  - Gaussian Blur with kernel size = 3 and sigma = (0.1, 1.5)
  - Normalization with ImageNet mean/std.
- Validation Data Preprocessing:
  - Normalization with ImageNet mean/std.

4. Age Prediction Model Architecture (From Scratch ResNet-50)

4.1 ResNet-50 Architecture

- Input: 3×224×224 cropped face.
- Stem: 7×7 conv → batch norm → ReLU → maxpool.
- Residual stages: {3, 4, 6, 3} bottleneck blocks (1×1 → 3×3 → 1×1) with identity/projection shortcuts.
- Head: Global Average Pool → Linear(2048 → 1) → scalar age.
- Stabilization: zero-init the final BN scale in bottleneck branch to ease optimization. This reduce the effect of residual block at first of training and they found this better.
- Output: scalar age in years (from 1 to 60 years).
- This model has 23,510,081 parameters (backbone + linear head).
- This model has also 4.1 billion FLOPs!

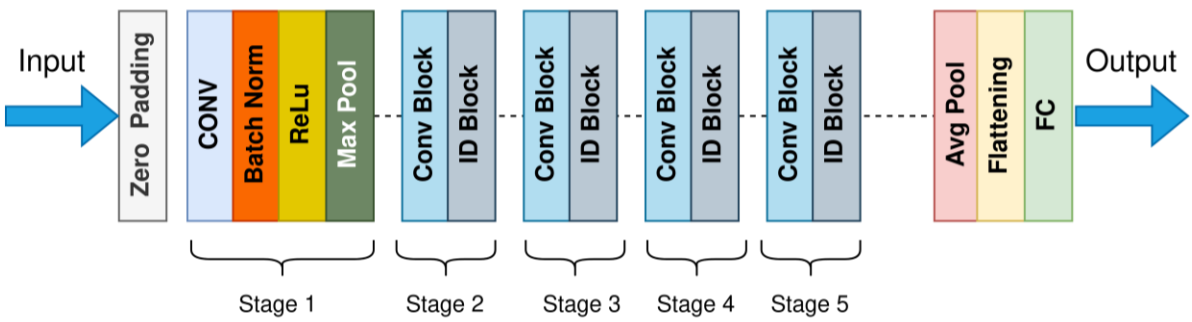

Figure 5. ResNet-50 model architecture

4.2 Loss Function

- First, I used mean absolute error (MAE) loss because it expresses the age difference well (MAE = actual age error)
- But I found that the training MAE saturated at about 6 years after 50 epochs and the validation MAE was higher than 9 years.
- So, I then used Smooth L1 (Huber) Loss with beta = 6 for quadratically penalizing age errors less than or equal 6 years to enhance and reduce the training MAE.
- The Smooth L1 (Huber) with beta = 6 performed better and ended with better age error of about 4 years.

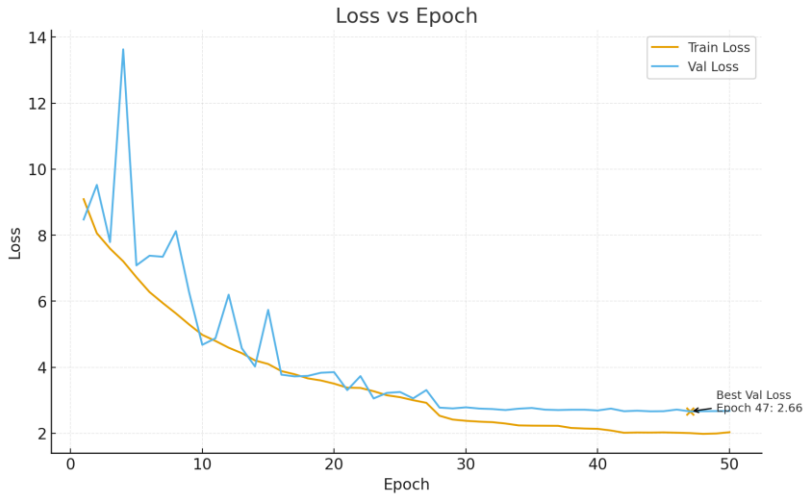

Figure 6. Smooth L1 (Huber) Loss

4.3 Training Setup

- Optimizer: SGD with learning rate = 0.005; momentum = 0.9, and weight decay = 0.0001 for regularization.
- Learning Rate Schedule: Reduce LR On Plateau with factor = 0.1 and patience = 3 epochs.
- Batch Size: 64 samples/batch
- Number of Epochs: 50 Epochs.
- Stability: gradient clipping (5.0) to avoid spikes.

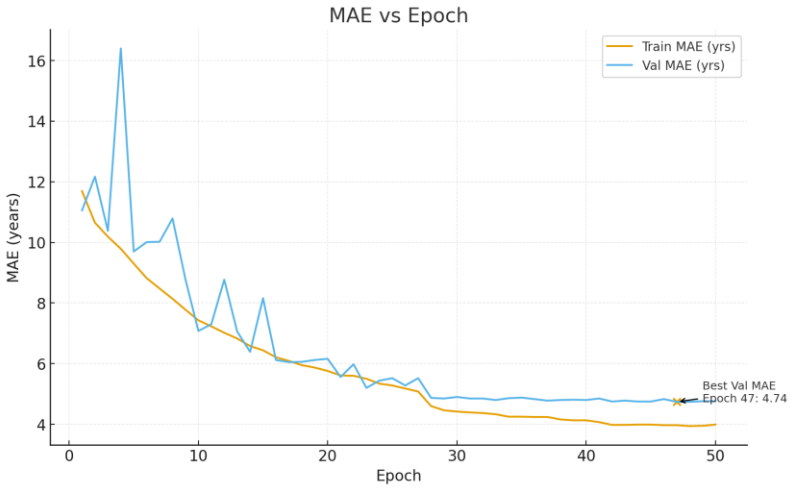

Figure 7. Age mean absolute error in years

5. Face Detection and Embedding Models

- **MTCNN Face Detector parameters:** *margin=40, select_largest=True, keep_all=False*
  This ensures that the model returns only one face per image with a 40 pixels margin to make the cropped face fit the age prediction model well (not tight crops).

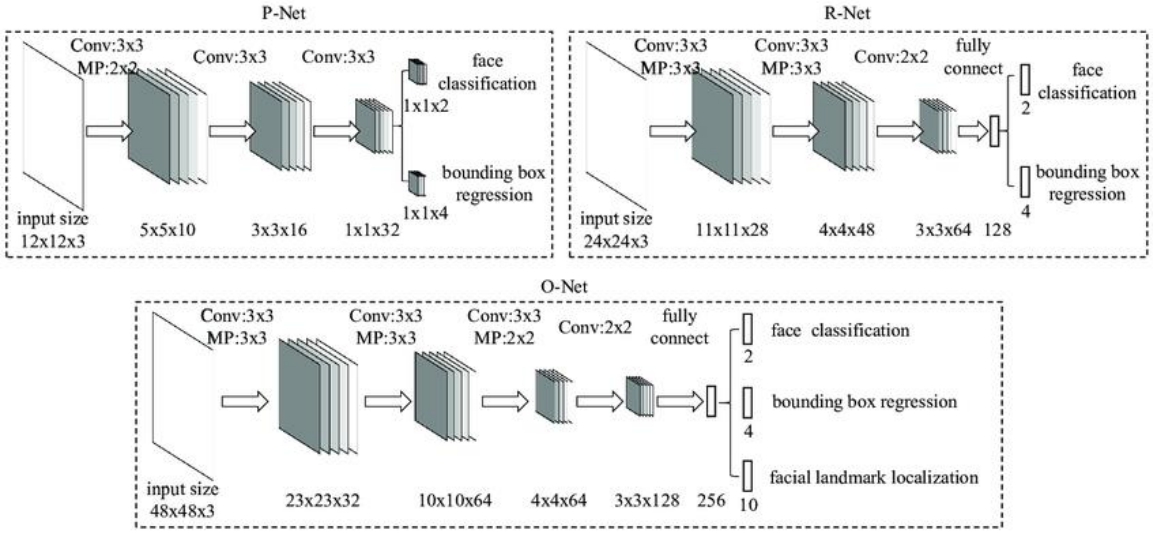

Figure 8. MTCNN model architecture

- **Face Embedding** (FaceNet/InceptionResnetV1): input face of dimensions 160×160 normalized to [−1,1]. It generates 512-dimension embeddings vector per face.

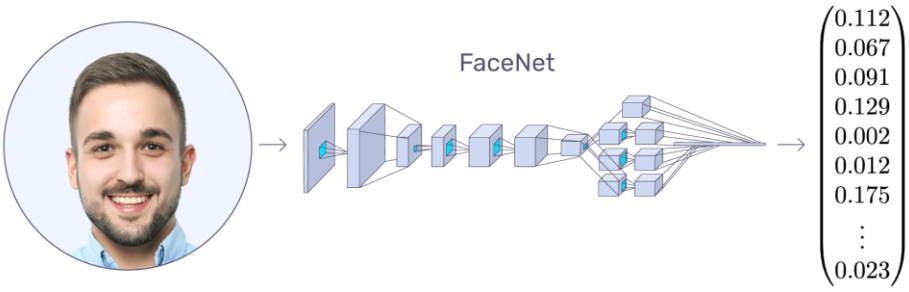

Figure 9. FaceNet embedding model

- **Similarity & Decision**: cosine similarity using dot product between the two L2-normalized embeddings. Faces are MATCHED if similarity score ≥ threshold.
- **Threshold**: used threshold of 0.50 tuned on test pairs. It balances the need for large threshold for small ages (children are confusing in matching) and the need for small threshold for accurately matching high age differences.
- We can also tune the threshold based on the application for which we use face matching system. It will depend on our interest in precision or recall, if we are interested in high precision we may increase the threshold, but for high recall we need to decrease the threshold.

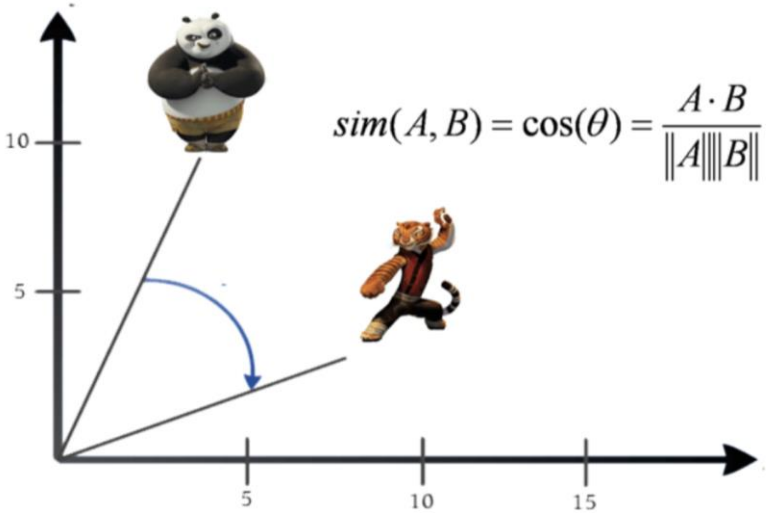

Figure 10. Cosine Similarity

6. Evaluation Metrics of Age Regressor

- Test set MAE (Mean Absolute Error in years): 4.76 years
- Calibration Plot: true ages vs mean predicted vs true.

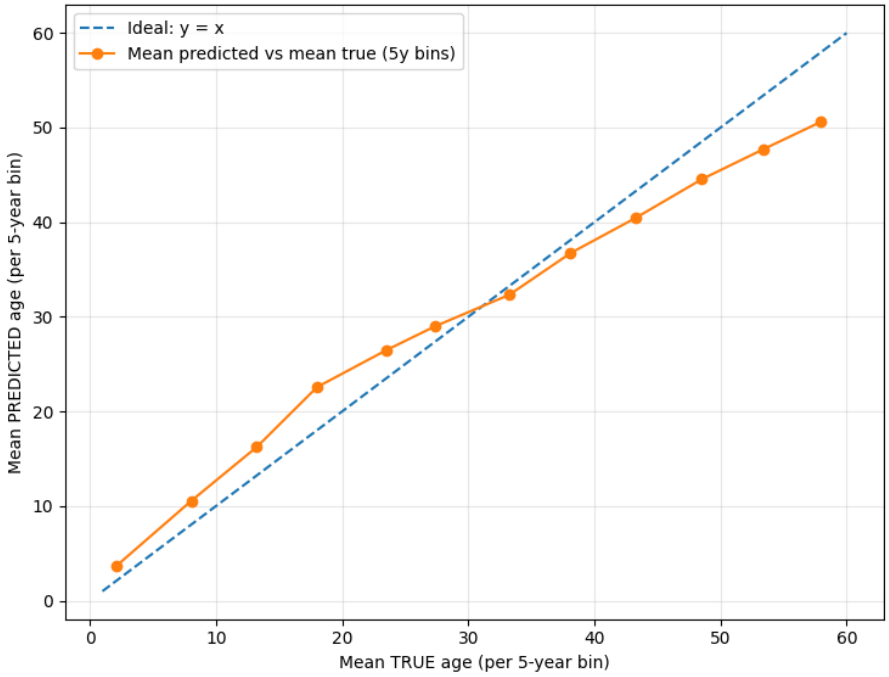

Figure 11. Mean predicted vs mean true values

- MAE by Age Group: 0–10, 11–20, 21–30, 31–40, 41–50, 51–60

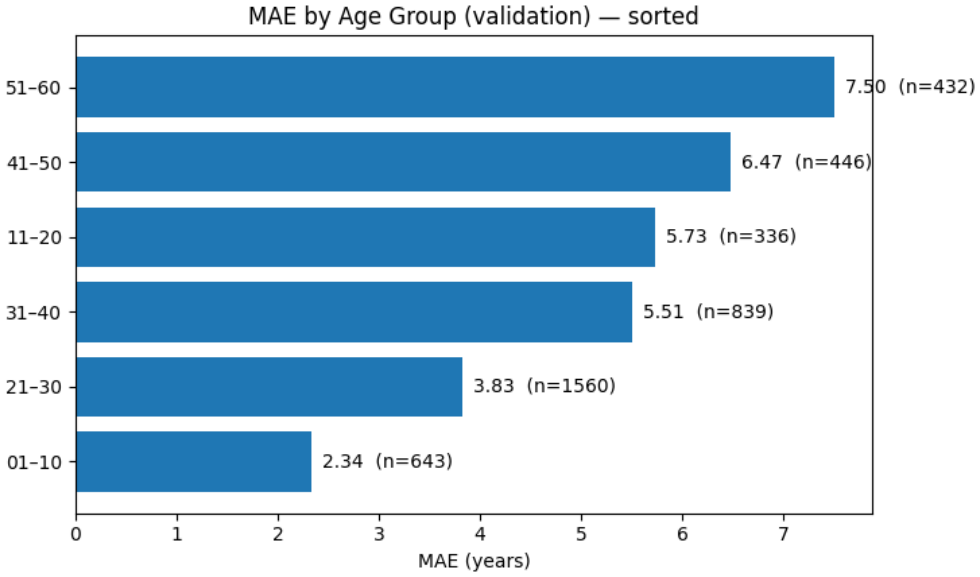

Figure 12. Mean absolute error for six age groups

7. Strengths and Limitations

7.1 Strengths

- Robust in face matching of the same person with moderate age differences, especially for clear, frontal faces and good lighting.
- Age predictor performs well on children and younger adults with strong MAE for ages 1–10 years and generally decent performance for under 40 years ages.
- Strong pretrained MTCNN face detector and FaceNet embedding model.
- Fast, modular pipeline; easy to deploy and extend.

7.2 Limitations

- Face matching would not perform well on images of lookalike people like brothers and twins.
- Age regression errors are larger for ages bigger than 40 years, as in these ages people faces do not change a lot, it is even hard for humans to predict their ages well.
- Age predictor is biased to this dataset faces. It cannot generalize well anywhere and MAE may drop on other demographics, sensors, and image styles.
- Global threshold may not generalize across domains, it would be better to calibrate it using domain test data.

8. Trade-offs and Future Improvement

8.1 Age Prediction

- **Regression:**
  - Predict simple continuous age and directly optimizes MAE (in years)
  - Straightforward, interpretable age estimate
- **Classification:**
  - Predict probabilities over age bins so it may be better for extreme outlier ages.
  - We may design clusters/bins of ages (say 0-15, 15-30, 30-45, 45-60 and so on) then apply classification on these clusters then regression on each cluster alone.

8.2 Face Embedding

- When developing system for specific domain data, we may consider using single multitasking model for both face embedding and age prediction by finetuning face embedding model for age prediction or take face representation vector from bottleneck layer in the age predictor.
- This may reduce costs by running inference on single model for dual tasks.

8.3 Similarity Score

- **Cosine Similarity:** Standard, simple, stable with L2-normlized embedding vectors.
- **Euclidean Distance:** We may consider calibrating Euclidean Distance measurement.