**We used DeepFace in combination with manual verification. Open-source Python library for facial analysis. It supports race/ethnicity classification into categories like: *white, black, asian, indian, middle eastern, latino*. It Uses pre-trained models (e.g., VGG-Face, Facenet, ArcFace).**

**Google Colab Code (Full Pipeline)**

**# STEP 1: Install DeepFace**

!pip install deepface –quiet

**# STEP 2: Import Required Libraries**

import os

import pandas as pd

from deepface import DeepFace

from google.colab import files

from PIL import Image

from IPython.display import display

**# STEP 3: Upload Images**

print("Upload your images (JPG/PNG)...")

uploaded = files.upload()

image\_paths = list(uploaded.keys())

**# STEP 4: Helper Function to Convert Age**

def age\_group(age):

try:

age = float(age)

if age < 40:

return "Young"

elif 40 <= age < 60:

return "Middle-aged"

else:

return "Older"

except:

return "Unknown"

**# STEP 5: Analyze Images**

results = []

for img\_path in image\_paths:

try:

img = Image.open(img\_path)

display(img)

# Run DeepFace analysis (all 4 attributes)

analysis = DeepFace.analyze(img\_path=img\_path, actions=['age', 'gender', 'emotion', 'race'], enforce\_detection=False)[0]

result = {

"Image": img\_path,

"Predicted\_Age": analysis["age"],

"Age\_Group": age\_group(analysis["age"]),

"Gender": analysis["gender"],

"Dominant\_Emotion": analysis["dominant\_emotion"],

"Dominant\_Race": analysis["dominant\_race"]

}

# Add detailed race scores

for race, score in analysis["race"].items():

result[f"Race\_{race}"] = score

# Add detailed emotion scores

for emotion, score in analysis["emotion"].items():

result[f"Emotion\_{emotion}"] = score

results.append(result)

except Exception as e:

print(f"Error processing {img\_path}: {e}")

results.append({

"Image": img\_path,

"Error": str(e)

})

**# STEP 6: Create DataFrame & Save CSV**

df = pd.DataFrame(results)

csv\_filename = "deepface\_analysis.csv"

df.to\_csv(csv\_filename, index=False)

print(f"\n✅ Results saved as: {csv\_filename}")

**# STEP 7: Download CSV**

files.download(csv\_filename)