

Q3

```
In [7]: import pandas as pd
import numpy as np
import cv2
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

df = pd.read_csv(r"C:\Users\ABC\Desktop\BAI\BAI-S7\CV Lab\Lab 08\Task 03\train

def load_and_preprocess_images(file_paths):
    images = []
    for file_path in file_paths:
        img = cv2.imread(r"C:\Users\ABC\Desktop\BAI\BAI-S7\CV Lab\Lab 08\Task 0
        img = cv2.resize(img, (224, 224))
        images.append(img)
    return np.array(images)

X = load_and_preprocess_images(df['ID'].values)
y = df['Class'].values

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

X_train, X_val, y_train, y_val = train_test_split(X, y_encoded, test_size=0.2,

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='linear'))

model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_val, y

val_loss, val_mae = model.evaluate(X_val, y_val)
print(f"Validation Mean Absolute Error: {val_mae}")

model.save("age_estimation_model.h5")
```

```

Epoch 1/10
498/498 [=====] - 1199s 2s/step - loss: 10342.1006 -
mae: 6.4624 - val_loss: 0.8018 - val_mae: 0.7515
Epoch 2/10
498/498 [=====] - 1485s 3s/step - loss: 1.1094 - ma
e: 0.7983 - val_loss: 0.8762 - val_mae: 0.8198
Epoch 3/10
498/498 [=====] - 1405s 3s/step - loss: 0.8141 - ma
e: 0.8279 - val_loss: 0.8206 - val_mae: 0.8414
Epoch 4/10
498/498 [=====] - 1071s 2s/step - loss: 0.7899 - ma
e: 0.8250 - val_loss: 0.8079 - val_mae: 0.8379
Epoch 5/10
498/498 [=====] - 1092s 2s/step - loss: 0.7743 - ma
e: 0.8131 - val_loss: 0.8161 - val_mae: 0.8352
Epoch 6/10
498/498 [=====] - 1071s 2s/step - loss: 0.7655 - ma
e: 0.8050 - val_loss: 0.8127 - val_mae: 0.8338
Epoch 7/10
498/498 [=====] - 1071s 2s/step - loss: 0.7532 - ma
e: 0.7959 - val_loss: 0.8220 - val_mae: 0.8364
Epoch 8/10
498/498 [=====] - 1050s 2s/step - loss: 0.7467 - ma
e: 0.7914 - val_loss: 0.8214 - val_mae: 0.8358
Epoch 9/10
498/498 [=====] - 1127s 2s/step - loss: 0.7328 - ma
e: 0.7809 - val_loss: 0.8354 - val_mae: 0.8341
Epoch 10/10
498/498 [=====] - 1078s 2s/step - loss: 0.7215 - ma
e: 0.7714 - val_loss: 0.8334 - val_mae: 0.8371
125/125 [=====] - 66s 525ms/step - loss: 0.8334 - ma
e: 0.8371
Validation Mean Absolute Error: 0.8370787501335144

```

```

C:\Users\ABC\anaconda3\Lib\site-packages\keras\src\engine\training.py:3079: U
serWarning: You are saving your model as an HDF5 file via `model.save()`. Thi
s file format is considered legacy. We recommend using instead the native Ker
as format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(

```



```

In [40]: import cv2
import numpy as np
from keras.models import load_model
from sklearn.preprocessing import LabelEncoder

face_cascade_path = cv2.data.harcascades + 'haarcascade_frontalface_alt.xml'
face_cascade = cv2.CascadeClassifier(face_cascade_path)

def classify_age(video_path, model_path, label_encoder):
    model = load_model(model_path)

    cap = cv2.VideoCapture(video_path)

    if not hasattr(label_encoder, 'classes_') or len(label_encoder.classes_) == 0:
        raise ValueError("LabelEncoder is not fitted. Fit the LabelEncoder before")

    while True:
        ret, frame = cap.read()
        if not ret:
            break

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

        for (x, y, w, h) in faces:
            face = frame[y:y + h, x:x + w]
            face = cv2.resize(face, (224, 224))

            face = face / 255.0
            face = np.expand_dims(face, axis=0)

            predicted_age = model.predict(face).squeeze()

            # Need to define threshold since the classification was done according to age
            young_age_threshold = 25
            old_age_threshold = 60

            if predicted_age < young_age_threshold:
                age_label = "YOUNG"
            elif predicted_age >= old_age_threshold:
                age_label = "OLD"
            else:
                age_label = "MIDDLE"

            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(frame, f"Age: {age_label}", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0))

        cv2.imshow('Age Estimation', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

model_path = "age_estimation_model.h5"

```

```
label_encoder = LabelEncoder()
label_encoder.fit(y_train)

video_path = r"C:\Users\ABC\Videos\Captures\Baby's Day Out full movie - YouTube
classify_age(video_path, model_path, label_encoder)
```

```
1/1 [=====] - 0s 151ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 50ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 49ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
```

```
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 44ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 46ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
```



```
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 45ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 44ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
```

```
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 46ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 81ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 43ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 41ms/step
```

```
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
```