

Source Code-

```
import tensorflow as tf
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cv2
# Define constants
img_height = 128
img_width = 128
batch_size = 32
epochs = 10
# Load the "UTKFace" dataset
df = pd.read_csv('UTKFace.csv')
```

```
SITRC,Nashik
```

```
df['age'] = df['age'].apply(lambda x: min(x, 100)) # limit age to 100
df = df.sample(frac=1).reset_index(drop=True) # shuffle the dataset
df['image_path'] = 'UTKFace/' + df['image_path']
df_train = df[:int(len(df)*0.8)] # 80% for training
df_val = df[int(len(df)*0.8):int(len(df)*0.9)] # 10% for validation
df_test = df[int(len(df)*0.9):] # 10% for testing
# Define data generators for training, validation, and testing sets
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_dataframe(
    dataframe=df_train,
    x_col='image_path',
    y_col=['male', 'age'],
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='raw')
val_generator = val_datagen.flow_from_dataframe(
    dataframe=df_val,
    x_col='image_path',
    y_col=['male', 'age'],
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='raw')
test_generator = test_datagen.flow_from_dataframe(
```

```

dataframe=df_test,

SITRC,Nashik
x_col='image_path',
y_col=['male', 'age'],
target_size=(img_height, img_width),
batch_size=batch_size,
class_mode='raw')
# Define the neural network model
model = Sequential([
Conv2D(32, (3,3), activation='relu', input_shape=(img_height, img_width, 3)),
MaxPooling2D((2,2)),
Conv2D(64, (3,3), activation='relu'),
MaxPooling2D((2,2)),
Conv2D(128, (3,3), activation='relu'),
MaxPooling2D((2,2)),
Conv2D(128, (3,3), activation='relu'),
MaxPooling2D((2,2)),
Flatten(),
Dropout(0.5),
Dense(512, activation='relu'),
Dense(2)
])
# Compile the model
model.compile(optimizer='adam',
loss={'dense_1': 'binary_crossentropy', 'dense_2': 'mse'},
metrics={'dense_1': 'accuracy', 'dense_2': 'mae'})
# Train the model
history = model.fit(train_generator,

SITRC,Nashik

epochs=epochs,
validation_data=val_generator)
# Evaluate the model on the test set
loss, accuracy, mae = model.evaluate(test_generator)
print("Test accuracy:", accuracy)
print("Test MAE:", mae)
# Predict the gender and age of a sample image
img = cv2.imread('sample_image.jpg')
img

```