

```
[160] import pandas as pd
import numpy as np
from PIL import Image
import os
import matplotlib.pyplot as plt
```

[/content/Final](#) (ctrl + click)

```
df=pd.read_excel("/content/Final Dataset (1).xlsx")
df
```



	Name	Gender	Waist Size	Height	Chest Size
0	RP	Female	34	178.0	32
1	CS	Female	30	167.0	32
2	VS	Female	36	176.0	34
3	SP	Female	30	167.0	28
4	SR	Female	28	5.4	32
...	...	...	...	...	...
69	NM	Female	36	167.0	36
70	HS	Male	36	158.0	44
71	SS	Male	34	182.0	38
72	SD	Female	26	186.0	30
73	TR	Female	30	151.0	32



```
df1=pd.read_excel("/content/Image dataset csv.xlsx")
df1
```

	State	Image-Male	
0	Andhra Pradesh	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
1	Arunachal Pradesh	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
2	Assam	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
3	Bihar	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
4	Chhattisgarh	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
5	Goa	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
6	Gujarat	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
7	Haryana	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
8	Himachal Pradesh	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
9	Jharkhand	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
10	Karnataka	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
11	Kerela	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
12	Madhya Pradesh	<a href="https://svkmmumbai-my.sharepoint.com/:u/g/per...">https://svkmmumbai-my.sharepoint.com/:u/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:u/g/per...">https://svkmmumbai-my.sharepoint.com/:u/g/per...</a>
13	Maharashtra	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>
14	Manipur	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>	<a href="https://svkmmumbai-my.sharepoint.com/:i/g/per...">https://svkmmumbai-my.sharepoint.com/:i/g/per...</a>

0s completed at 11:10 PM



```
image_path = '/content/nauvari-art-silk-saree-207x300.webp' # Replace with the actual path  
img = Image.open(image_path)  
img.show()  
plt.imshow(img)  
plt.show()
```





```
import pandas as pd
from PIL import Image

# Load the Excel file into a Pandas DataFrame
excel_file_path = '/content/Image dataset csv.xlsx' # Replace with the actual path to
df = pd.read_excel(excel_file_path)

# Assuming you have 'male_image_name' and 'female_image_name' columns in your Excel fi

female_image_paths = df['Image-Female'].tolist()

male_image_paths = df['Image-Male'].tolist()
```

[ ] Start coding or [generate](#) with AI.

```
[66] # Resize function (similar to the previous example)
def resize_images(image_paths, target_size=(100, 100)):
    for img_path in image_paths:
        try:
            img = Image.open(img_path)
            resized_img = img.resize(target_size)
            resized_img.save(img_path)
            print(f"Resized and saved: {img_path}")
        except Exception as e:
            print(f"Error processing {img_path}: {e}")
```

```
def resize_image(img_path, target_size=(100, 50)):
    try:
        # Open the image
        img = Image.open(img_path)

        # Resize the image
        resized_img = img.resize(target_size)

        # Display the resized image using matplotlib
        plt.imshow(resized_img)
        plt.axis('off') # Turn off axis labels
        plt.show()
    except Exception as e:
        print(f"Error processing {img_path}: {e}")

# Example usage
image_path = '/content/nauvari-art-silk-saree-207x300.webp' # Replace with the actual path
resize_image(image_path, target_size=(300, 300))
```





```
3] pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in /usr/local/lib/pyt
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: ml-dtypes~=0.2.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dis
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/py
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python3.10/dist
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local/lib/pyth
Requirement already satisfied: keras<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-pack
```

```

] from PIL import Image
import numpy as np

def load_and_preprocess_images_from_dataframe(dataframe, row_index, column_name, target_size):
    # Use iloc to get the image path from the specified row and column
    img_path = df1.iloc[0:20]["Image-Male"]

    try:
        # Open the image, resize, and convert to NumPy array
        img = Image.open(img_path).resize(target_size)
        img_array = np.array(img) / 255.0 # Normalize pixel values to [0, 1]
    except Exception as e:
        print(f"Error processing {img_path}: {e}")
        return None

    return img_array

# Example usage
row_index = 0 # Replace with the desired row index
column_name = 'Image-Male' # Replace with the desired column name
X_male = load_and_preprocess_images_from_dataframe(df1, row_index, column_name, target_size)

# Similarly, you can use iloc for the female images
column_name = 'Image-Female'
X_female = load_and_preprocess_images_from_dataframe(df1, row_index, column_name, target_size)

```

```

import tensorflow as tf
from tensorflow.keras import layers, models

# Define the CNN model
def create_simple_cnn(input_shape, num_classes):
    model = models.Sequential()

    # Convolutional layers
    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))

    # Flatten layer
    model.add(layers.Flatten())

    # Fully connected layers
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model

# Define input shape and number of classes
input_shape = (28, 28, 1) # Adjust the input shape based on your dataset
num_classes = 10 # Change this based on the number of classes in your dataset

```

```

# Create the CNN model
model = create_simple_cnn(input_shape, num_classes)

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Print model summary
model.summary()

```

```

] # Assuming you have X_train and y_train as your training data
model.fit(X_train, y_train, epochs=10, validation_split=0.2)

```