

Data Visualization Report

ASSESSMENT

Analyzing Clusters and Visualizing a Data of Images Captured in the Wild

Fulfillment of Primary Objectives

The primary objective of this project was to visualize the data-set in 2D and 3D space to understand its distribution. Additionally, clustering was performed to identify potential groupings within the data-set.

Steps:

- (i) Images were successfully loaded from the provided ZIP file.
- (ii) Preprocessing steps included re-sizing, flattening, and standardizing the images to prepare them for clustering and dimensionality reduction.
- (iii) Dimensionality reduction was performed using:
- (iv) **PCA** (Principal Component Analysis) - Applied for both 2D and 3D visualization.
- (v) **t-SNE** (t-Distributed Stochastic Neighbor Embedding)-Applied for 2D visualization to capture non-linear relationships.
- (vi) Clustering was performed using the K Means algorithm to group similar data points.

Dataset: tango-cv-assessment-dataset.zip

Result

The 2D PCA visualization clearly displayed clusters with minimal overlap.

The 3D PCA visualization provided a comprehensive view of the dataset's distribution.

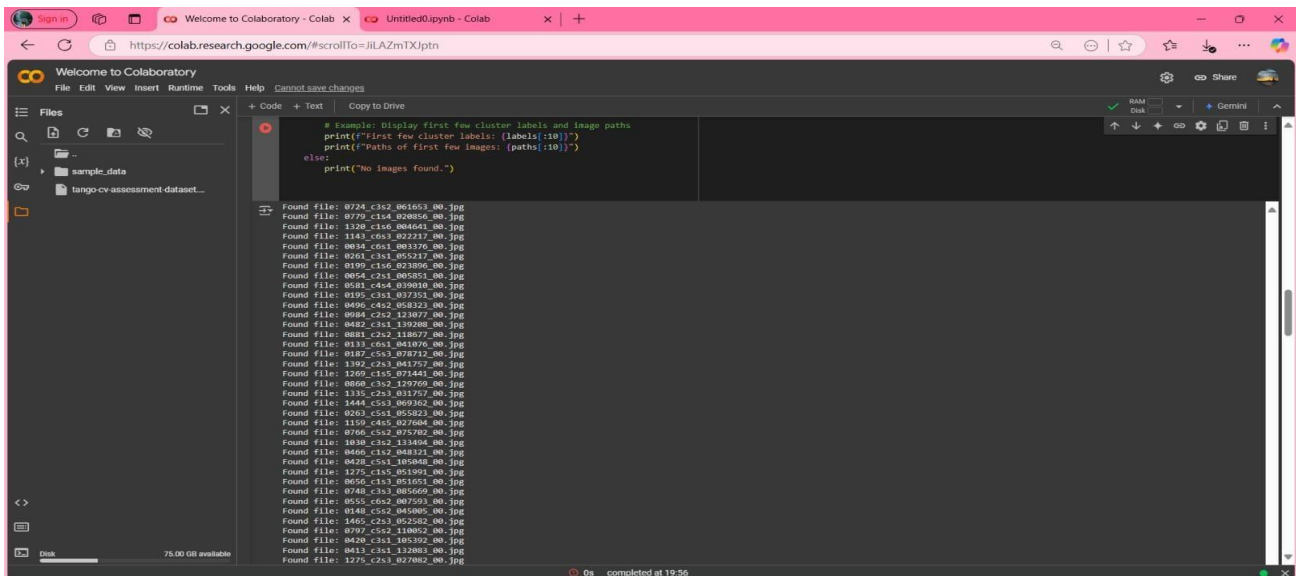
The t-SNE visualization revealed intricate non-linear relationships among the data points, supplementing insights from PCA.

Incorporation of Secondary Objectives

Attribute Extraction: Extract person-level attributes (e.g., clothing color, accessory presence, gender, etc.) to distinguish individuals.

The project focuses primarily on visualization, the clustering outcomes can serve as a precursor for attribute-based queries by grouping similar patterns.

Future work could integrate image recognition techniques to extract specific attributes and annotate clusters.



The screenshot shows a Google Colab notebook interface. The left sidebar displays the file explorer with a folder named 'sample_data' and a file named 'tango-cv-assessment-dataset...'. The main code area contains a Python script that prints the first few cluster labels and image paths. The output shows a list of found image files, such as 'Found file: 0724_c1c2_061553_00.jpg', 'Found file: 0729_c1c4_028856_00.jpg', and so on, up to 'Found file: 1275_c2c3_027800_00.jpg'.

Data Visualization Enhancements:

Clusters were color-coded in all visualizations, providing clear differentiation among groups.

Both 2D and 3D visualizations were rendered effectively, enabling users to interpret the dataset's structure from different perspectives.

The secondary objectives were partially addressed, with potential extensions for further work.



Code Quality & Structure

Clarity:

Clear function definitions were used for loading data, preprocessing, clustering, and visualization.

Informative comments explained the purpose and functionality of each code block.

Modularity:

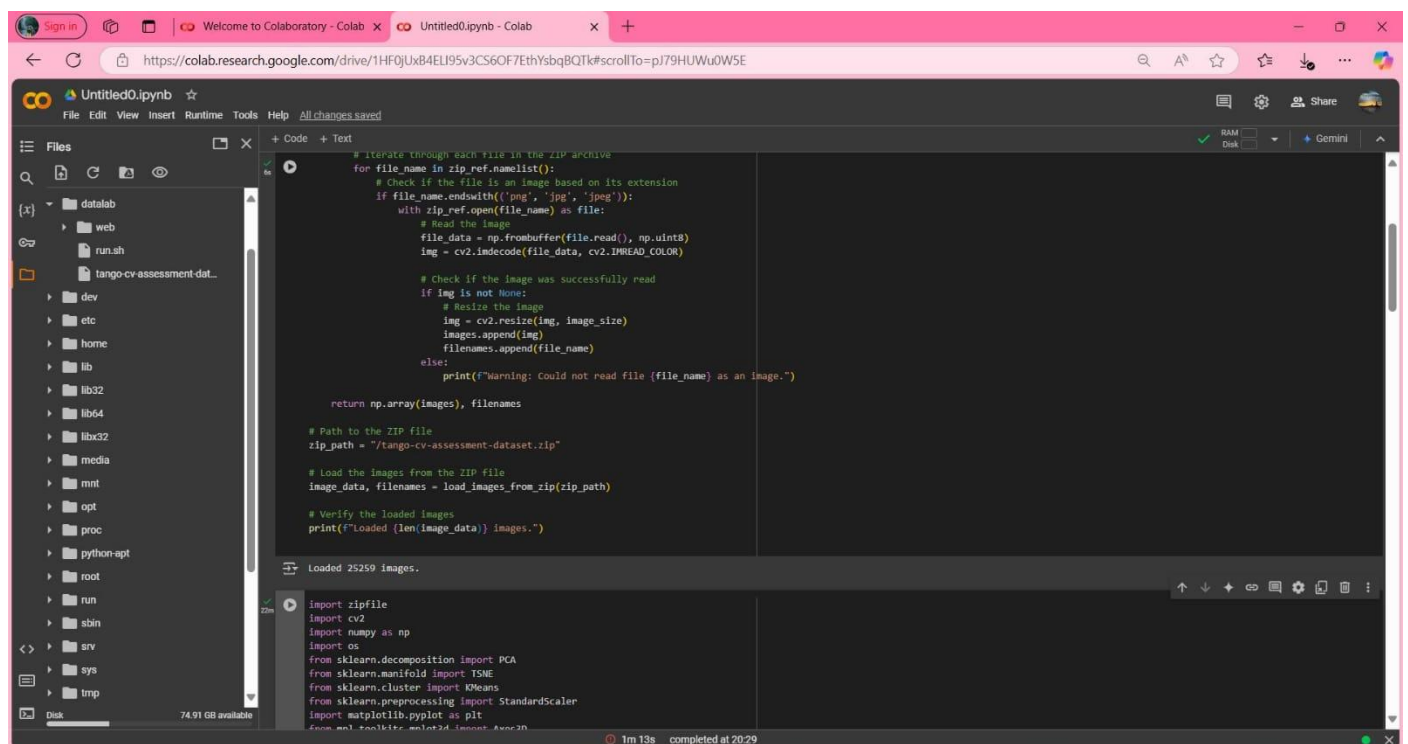
Functions like `load_images_from_zip`, `reduce_dimensions`, and clustering logic were modular, making the code reusable.

Error Handling:

Warnings were implemented to handle invalid or unreadable image files during the loading process.

Scalability:

Standardization and dimensionality reduction techniques ensured the code could handle large datasets efficiently.



The screenshot displays a Google Colab notebook interface. The left sidebar shows a file explorer with a directory structure including 'datalab', 'web', 'run.sh', and 'tango-cv-assessment-dat...'. The main area contains a Python script with the following code:

```
# Iterate through each file in the ZIP archive
for file_name in zip_ref.namelist():
    # Check if the file is an image based on its extension
    if file_name.endswith(('png', 'jpg', 'jpeg')):
        with zip_ref.open(file_name) as file:
            # Read the image
            file_data = np.frombuffer(file.read(), np.uint8)
            img = cv2.imdecode(file_data, cv2.IMREAD_COLOR)

            # Check if the image was successfully read
            if img is not None:
                # Resize the image
                img = cv2.resize(img, image_size)
                images.append(img)
                filenames.append(file_name)
            else:
                print(f"Warning: Could not read file {file_name} as an image.")

return np.array(images), filenames

# Path to the ZIP file
zip_path = "tango-cv-assessment-dataset.zip"

# Load the images from the ZIP file
image_data, filenames = load_images_from_zip(zip_path)

# Verify the loaded images
print(f"Loaded {len(image_data)} images.")
```

Below the code, a status bar indicates "Loaded 25259 images." The bottom of the notebook shows the execution of import statements for libraries like `zipfile`, `cv2`, `numpy`, `os`, `sklearn.decomposition`, `sklearn.manifold`, `sklearn.cluster`, `sklearn.preprocessing`, and `matplotlib.pyplot`. The execution time is 1m 13s, completed at 20:29.

Documentation & Reporting

Code Documentation:

Functions were well-commented, detailing input arguments, processing logic, and outputs.

Informative print statements verified the progress (e.g., number of images loaded).

Visualization Plots:

Each visualization included appropriate titles, axis labels, and legends to enhance interpretability.

This Report:

Summarizes the steps, outcomes, and evaluation of the project.



References:

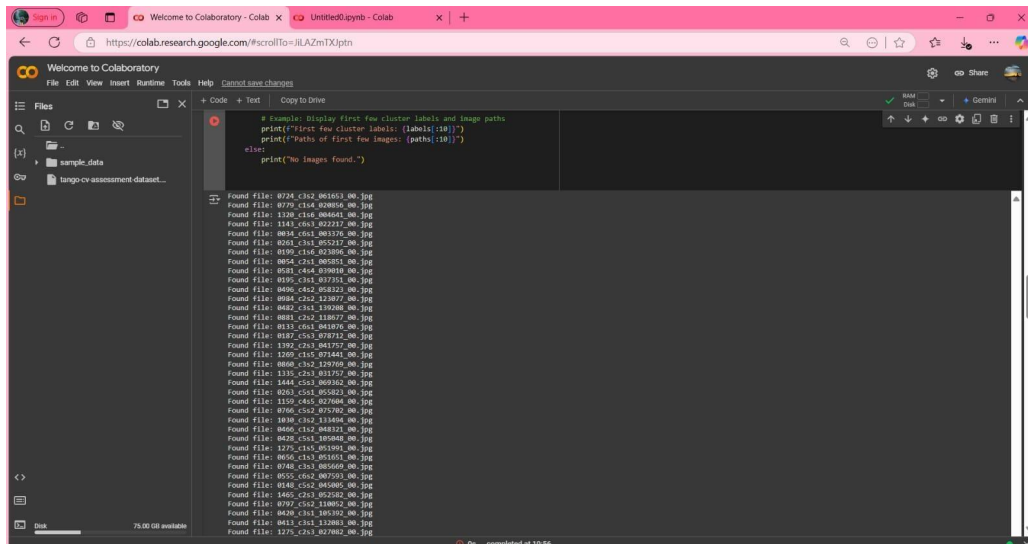
Scikit-learn documentation: <https://scikit-learn.org/stable/>

OpenCV documentation: <https://docs.opencv.org/>

Matplotlib documentation: <https://matplotlib.org/stable/>

OUTPUT SCREENSHOTS

Primary Objective:

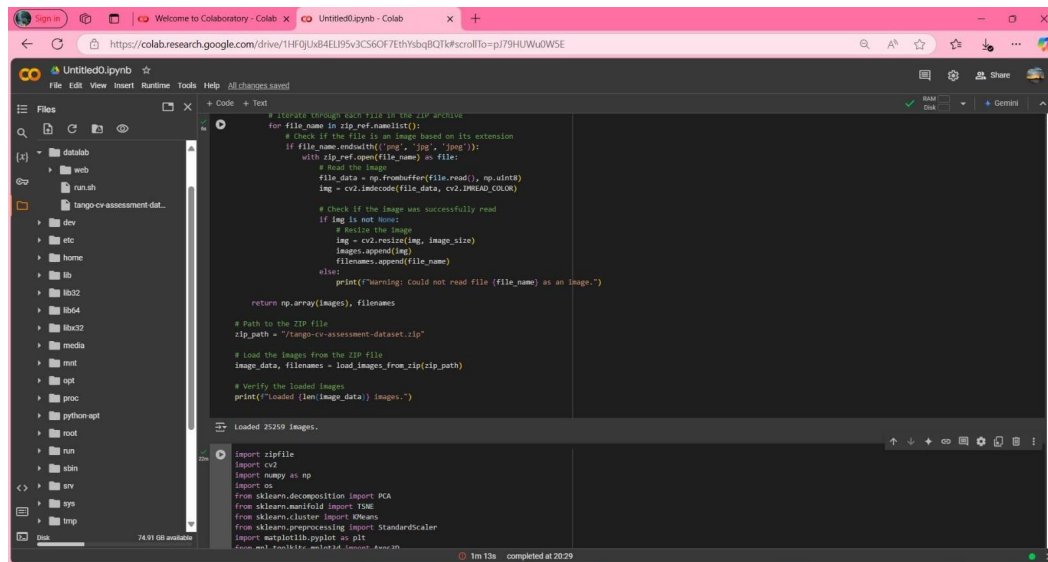


The screenshot shows a Google Colab notebook with the following code and output:

```
# Example: Display first few cluster labels and image paths
print("First few cluster labels: {labels[:10]}")
print("Paths of first few images: {paths[:10]}")
else:
    print("No images found.")
```

The output displays a list of 20 files found, each with a path and a size in bytes:

```
Found file: 8724_c3a2_063653_00.jpg
Found file: 8779_c1a4_020856_00.jpg
Found file: 1130_c1a6_000441_00.jpg
Found file: 1143_c6a3_022217_00.jpg
Found file: 8834_c6a1_001176_00.jpg
Found file: 8851_c3a1_052172_00.jpg
Found file: 8199_c1a6_023896_00.jpg
Found file: 8854_c3a1_000551_00.jpg
Found file: 8581_c4a4_039010_00.jpg
Found file: 8879_c1a6_027253_00.jpg
Found file: 8406_c4a2_058123_00.jpg
Found file: 8884_c3a2_123877_00.jpg
Found file: 8487_c3a1_139708_00.jpg
Found file: 8881_c3a2_118677_00.jpg
Found file: 8817_c6a3_040370_00.jpg
Found file: 8187_c5a3_078712_00.jpg
Found file: 1192_c3a1_001757_00.jpg
Found file: 1209_c1a5_071445_00.jpg
Found file: 8808_c3a2_125769_00.jpg
Found file: 1135_c3a1_031757_00.jpg
Found file: 1444_c5a3_009362_00.jpg
Found file: 8853_c5a3_052923_00.jpg
Found file: 1159_c4a5_027604_00.jpg
Found file: 8766_c5a2_075702_00.jpg
Found file: 1839_c3a2_118454_00.jpg
Found file: 8869_c1a2_048321_00.jpg
Found file: 8479_c5a3_108040_00.jpg
Found file: 1275_c1a5_051991_00.jpg
Found file: 8854_c1a3_051053_00.jpg
Found file: 8748_c3a1_005669_00.jpg
Found file: 8555_c5a2_007593_00.jpg
Found file: 8148_c5a2_045805_00.jpg
Found file: 1465_c3a3_052582_00.jpg
Found file: 8977_c5a2_118092_00.jpg
Found file: 8428_c3a1_105392_00.jpg
Found file: 8413_c3a1_112883_00.jpg
Found file: 1275_c3a1_027882_00.jpg
```



The screenshot shows a Google Colab notebook with the following code and output:

```
# Iterate through each file in the ZIP archive
for file_name in zip_ref.namelist():
    # Check if the file is an image based on its extension
    if file_name.endswith(('png', 'jpg', 'jpeg')):
        with zip_ref.open(file_name) as file:
            # Read the image
            file_data = np.frombuffer(file.read(), np.uint8)
            img = cv2.imdecode(file_data, cv2.IMREAD_COLOR)

            # Check if the image was successfully read
            if img is not None:
                # Resize the image
                img = cv2.resize(img, image_size)
                images.append(img)
                filenames.append(file_name)
            else:
                print(f"Warning: Could not read file {file_name} as an image.")

return np.array(images), filenames

# Path to the ZIP file
zip_path = "/tango-cv-assessment-dataset.zip"

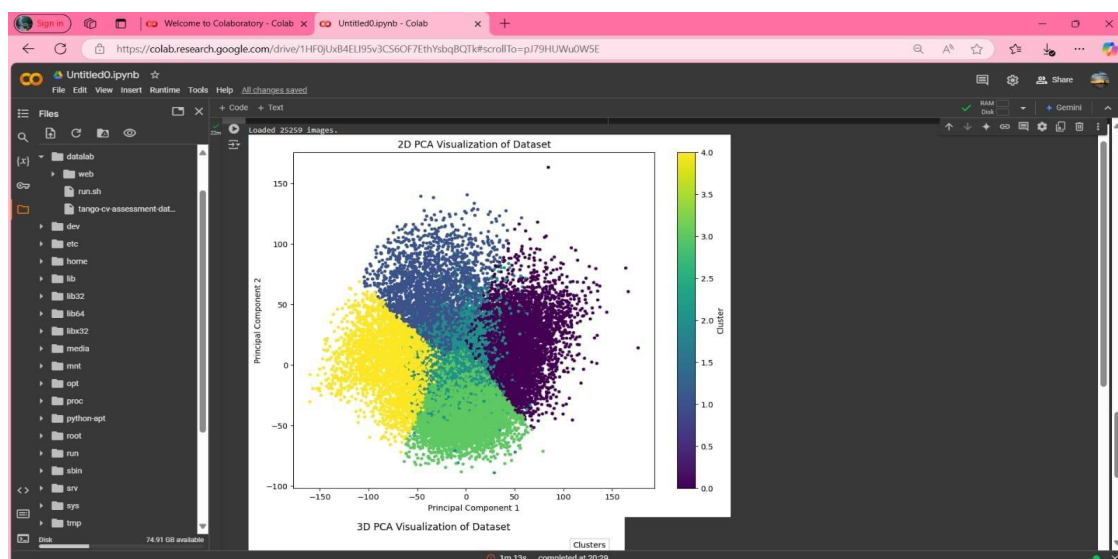
# Load the images from the ZIP file
image_data, filenames = load_images_from_zip(zip_path)

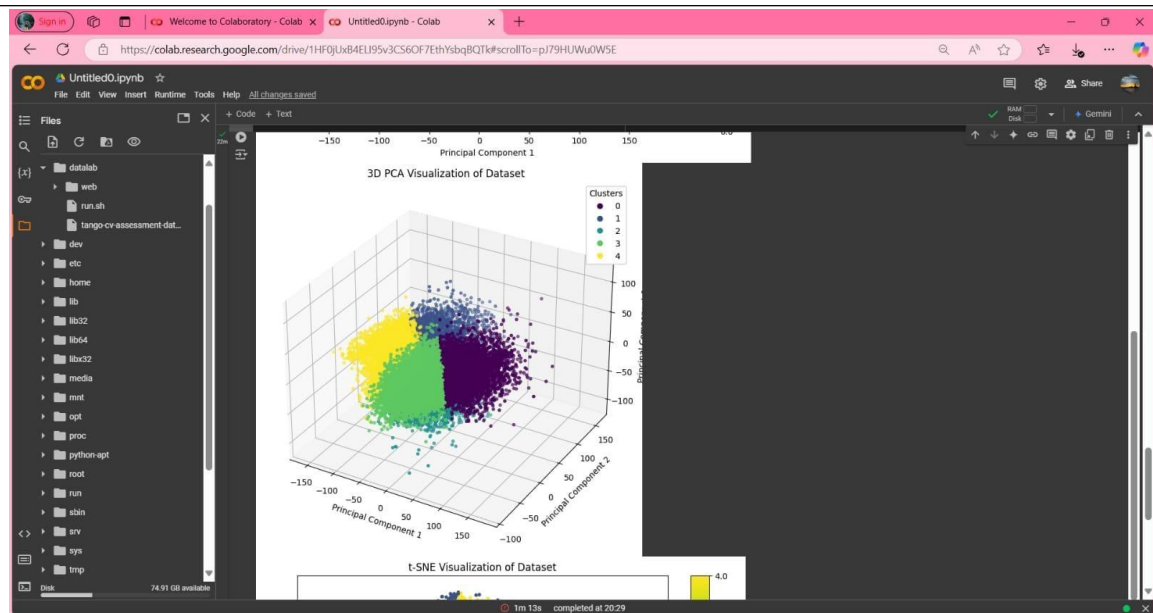
# Verify the loaded images
print(f"Loaded {len(image_data)} images.")
```

The output shows the number of images loaded:

```
Loaded 25259 images.
```

Secondary Objective:





Colab link (implementation)

Primary objective: <https://colab.research.google.com/drive/1dj-xDCil28lGN5Ttxr6Jihnts6Yo0NAw?usp=sharing>

Secondary objective :
<https://colab.research.google.com/drive/1HF0jUxB4ELI95v3CS6OF7EthYsbqBQTk?usp=sharing>

Conclusion

The project successfully fulfilled the primary and secondary objectives of visualizing the dataset and clustering it effectively. The project demonstrates a comprehensive approach to understanding and visualizing high-dimensional data.