

```

import os
import pandas as pd
import matplotlib.pyplot as plt

# Function to update the lists with new pairs
def update_similarity_lists(similarity, query, target, list_to_update,
reverse=False, store_pairs=10):
    list_to_update.append((similarity, query, target))
    list_to_update.sort(key=lambda x: x[0], reverse=reverse)
    if len(list_to_update) > store_pairs:
        list_to_update.pop()

def collecting(directory_path, store_pairs=10, save=False):
    # List to store all similarity scores
    similarity_scores = []

    # Initialize lists to store the top K lowest and highest
similarity pairs
    lowest_similarity_pairs = []
    highest_similarity_pairs = []

    # Loop through each file in the directory
    for filename in os.listdir(directory_path):
        if filename.endswith('.csv'):
            file_path = os.path.join(directory_path, filename)

            # Read the CSV file
            df = pd.read_csv(file_path, dtype={'Query': str, 'Target': str, 'Similarity': float})

            # Assume the similarity scores are in a column named
'similarity'
            similarity_scores.extend(df['Similarity'].tolist())

            # Iterate through each row in the DataFrame
            for index, row in df.iterrows():
                similarity = row['Similarity']
                query_image_name = row['Query']
                target_image_name = row['Target']

                # Update the lowest and highest similarity lists
                update_similarity_lists(similarity, query_image_name,
target_image_name, lowest_similarity_pairs, store_pairs=store_pairs)
                update_similarity_lists(similarity, query_image_name,
target_image_name, highest_similarity_pairs, reverse=True,
store_pairs=store_pairs)

    # Convert the lists to DataFrames
    lowest_df = pd.DataFrame(lowest_similarity_pairs,

```

```

columns=['Similarity', 'Query', 'Target'])
    highest_df = pd.DataFrame(highest_similarity_pairs,
columns=['Similarity', 'Query', 'Target'])

    # Ensure 'Query Image Name' and 'Target Image Name' are treated as
strings
    lowest_df['Query'] = lowest_df['Query'].astype(str)
    lowest_df['Target'] = lowest_df['Target'].astype(str)
    highest_df['Query'] = highest_df['Query'].astype(str)
    highest_df['Target'] = highest_df['Target'].astype(str)

    # Save to CSV files
    if save:
        # Define paths for the output CSV files
        lowest_similarity_csv = directory_path + 'min_max_pairs/' +
'lowest_similarity_pairs.csv'
        highest_similarity_csv = directory_path + 'min_max_pairs/' +
'highest_similarity_pairs.csv'

        lowest_df.to_csv(lowest_similarity_csv, index=False)
        highest_df.to_csv(highest_similarity_csv, index=False)

        print(f"Saved lowest similarity pairs to
{lowest_similarity_csv}")
        print(f"Saved highest similarity pairs to
{highest_similarity_csv}")

    return similarity_scores, lowest_similarity_pairs,
highest_similarity_pairs

def plotting(similarity_scores, bins=50, title=""):
    weights = [1 / len(similarity_scores)] * len(similarity_scores)

    # Plot the distribution of similarity scores
    plt.figure(figsize=(10, 6))
    plt.hist(similarity_scores, bins=bins, alpha=0.7, weights=weights)
    plt.title(title + " Distribution of Similarity")
    plt.xlabel('Similarity Score')
    plt.ylabel('Frequency')
    plt.xlim(-1, 1)
    plt.grid(axis='y', alpha=0.75)
    plt.show()

def compare_plotting(similarity_scores1, similarity_scores2, bins=50,
title="", label=['', '']):
    weights1 = [1 / len(similarity_scores1)] * len(similarity_scores1)
    weights2 = [1 / len(similarity_scores2)] * len(similarity_scores2)

    # Plot the distribution of similarity scores

```

```

plt.figure(figsize=(10, 6))
plt.hist([similarity_scores1, similarity_scores2], bins=bins,
alpha=0.7, weights=[weights1, weights2], label=label)
plt.title(title + " Distribution of Similarity")
plt.xlabel('Similarity Score')
plt.ylabel('Frequency')
plt.xlim(0, 1)
plt.grid(axis='y', alpha=0.75)
plt.legend()
plt.show()

```

Eigen Places

```

# Directory containing the CSV files
EP_neighbors_path =
"/users/eleves-a/2022/haiyang.jiang/Trans_Proj/repo/VPR_Proj/result/
EigenPlaces/neighbors/"

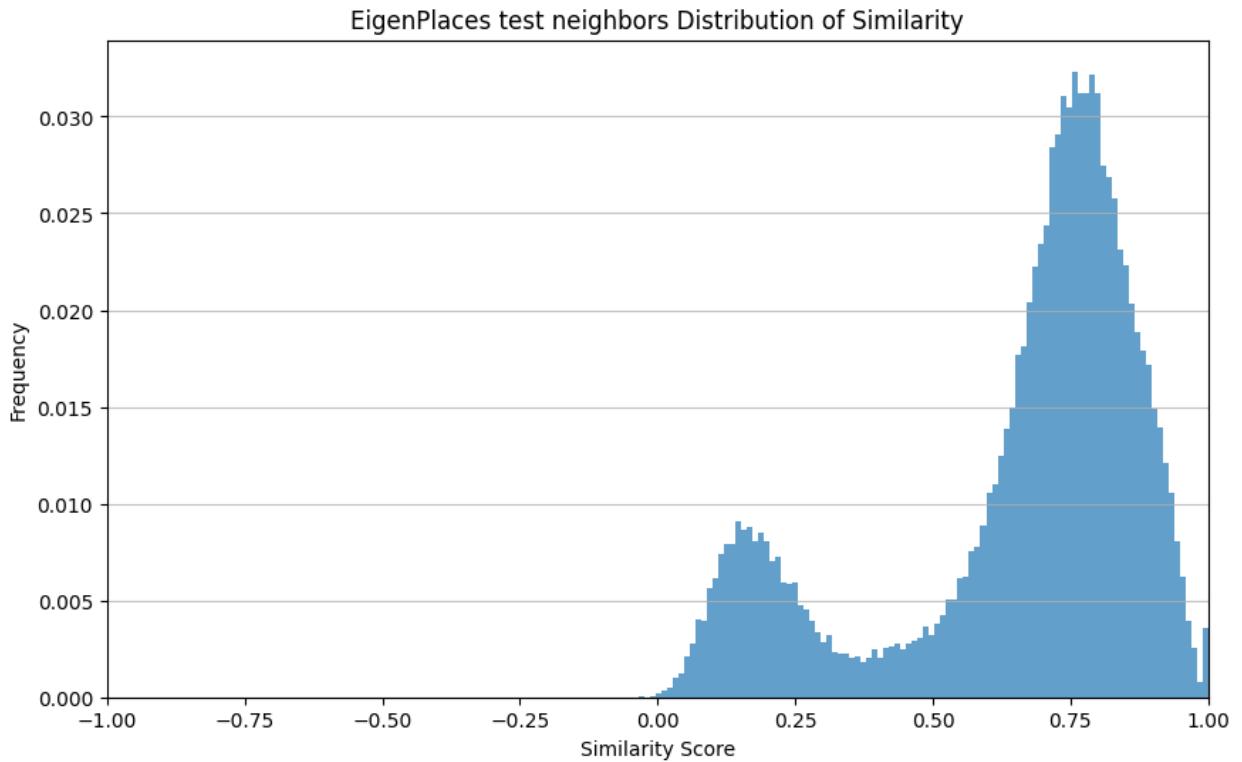
similarity_EP_neighbors, lowest_pairs_EP_neighbors,
highest_pairs_EP_neighbors = collecting(EP_neighbors_path)

print(len(similarity_EP_neighbors))

119140

plotting(similarity_EP_neighbors, bins=100, title="EigenPlaces test
neighbors")

```



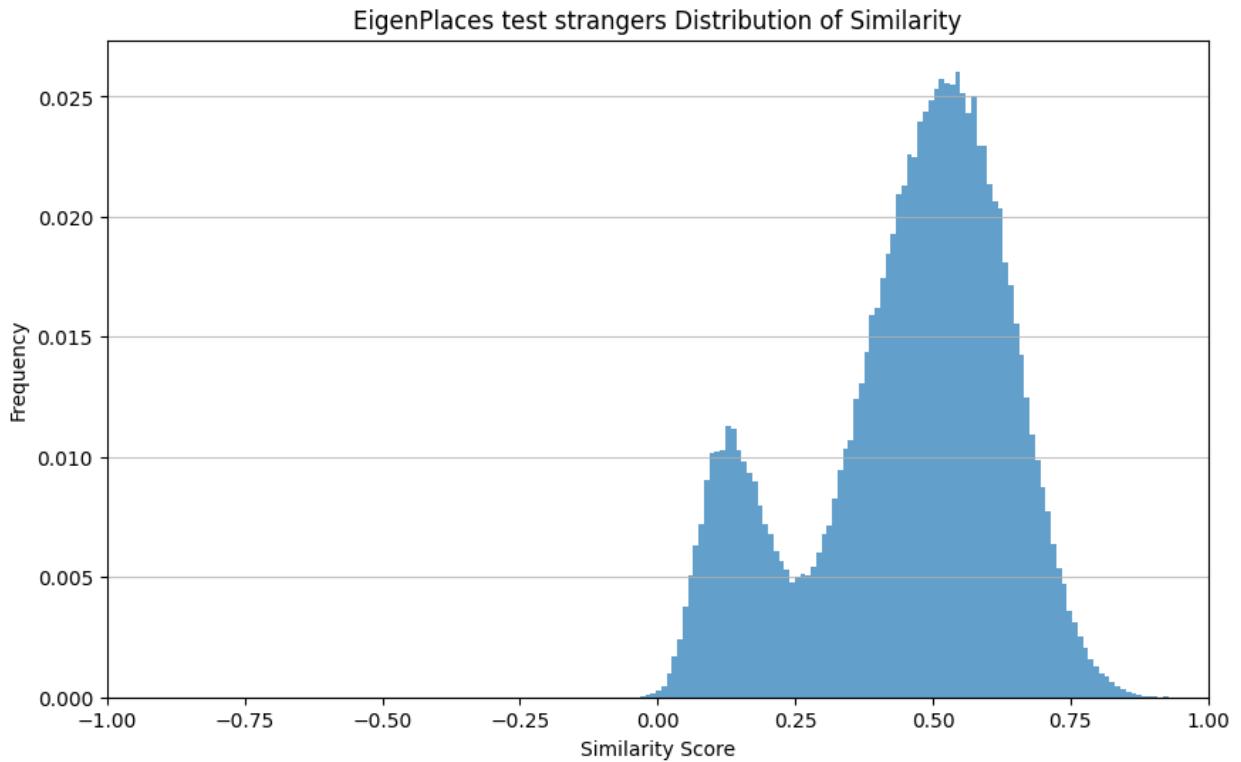
```
# Directory containing the CSV files
EP_strangers_path =
"/users/elevens-a/2022/haiyang.jiang/Trans_Proj/repo/VPR_Proj/result/
EigenPlaces/strangers/"

similarity_EP_strangers, lowest_pairs_EP_strangers,
highest_pairs_EP_strangers = collecting(EP_strangers_path)

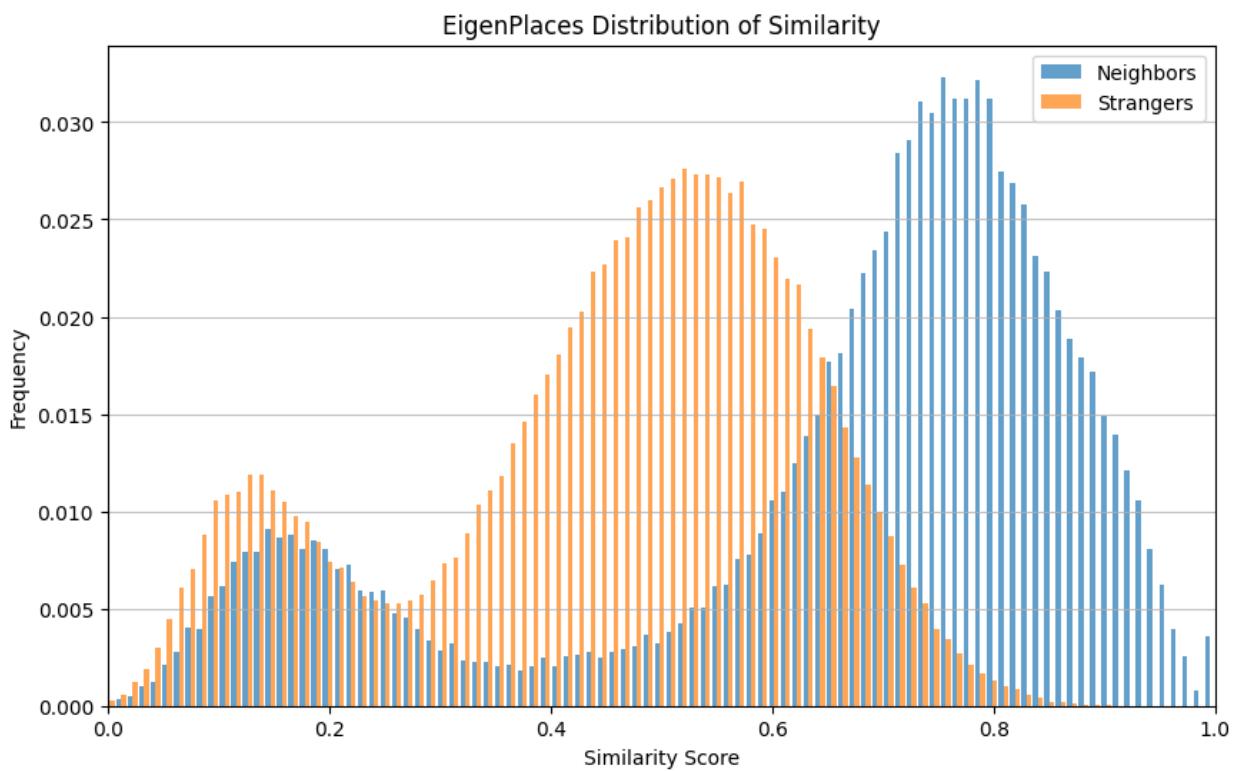
print(len(similarity_EP_strangers))

216509

plotting(similarity_EP_strangers, bins=100, title="EigenPlaces test
strangers")
```



```
compare_plotting(similarity_EP_neighbors, similarity_EP_strangers,
bins=100, title="EigenPlaces", label=['Neighbors', 'Strangers'])
```



MixVPR

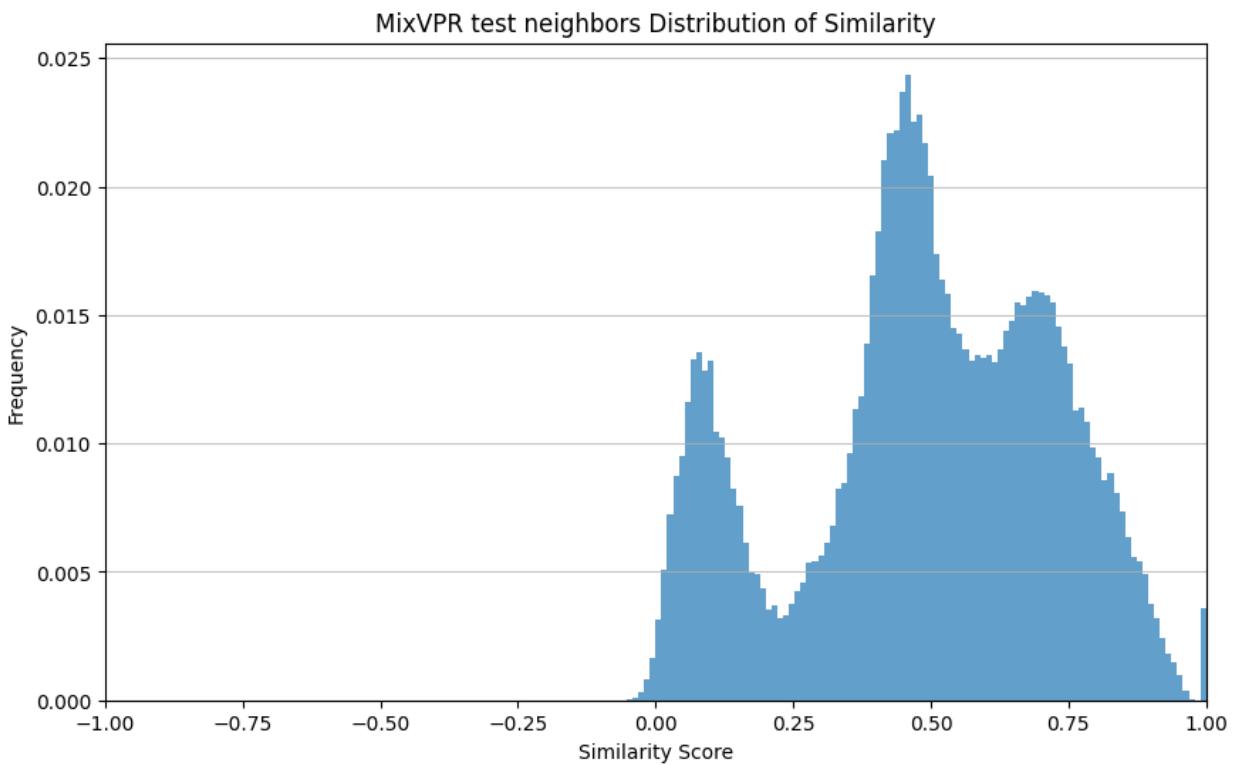
```
# Directory containing the CSV files
MVPR_neighbors_path =
"/users/elevens-a/2022/haiyang.jiang/Trans_Proj/repo/VPR_Proj/result/
MixVPR/neighbors/"

similarity_MVPR_neighbors, lowest_pairs_MVPR_neighbors,
highest_pairs_MVPR_neighbors = collecting(MVPR_neighbors_path)

print(len(similarity_MVPR_neighbors))

119144

plotting(similarity_MVPR_neighbors, bins=100, title="MixVPR test
neighbors")
```



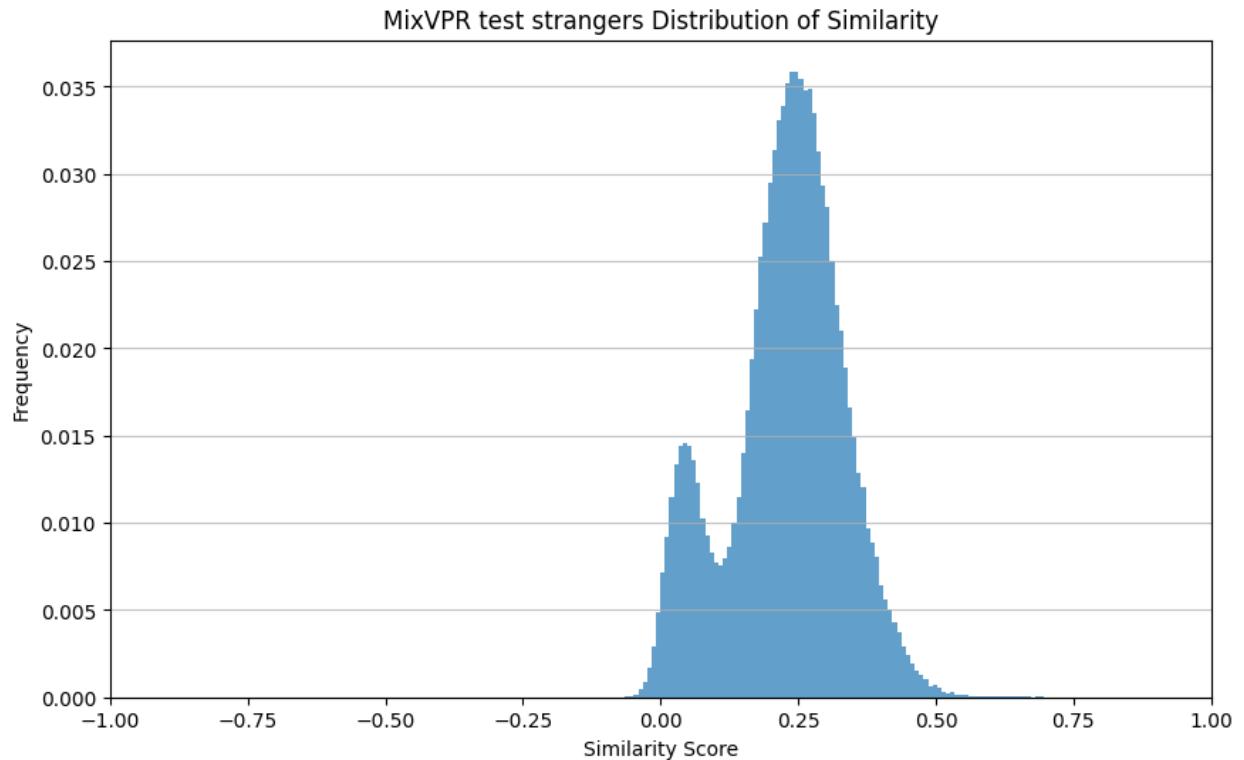
```
# Directory containing the CSV files
MVPR_strangers_path =
"/users/elevens-a/2022/haiyang.jiang/Trans_Proj/repo/VPR_Proj/result/
MixVPR/strangers/"

similarity_MVPR_strangers, lowest_pairs_MVPR_strangers,
highest_pairs_MVPR_strangers = collecting(MVPR_strangers_path)

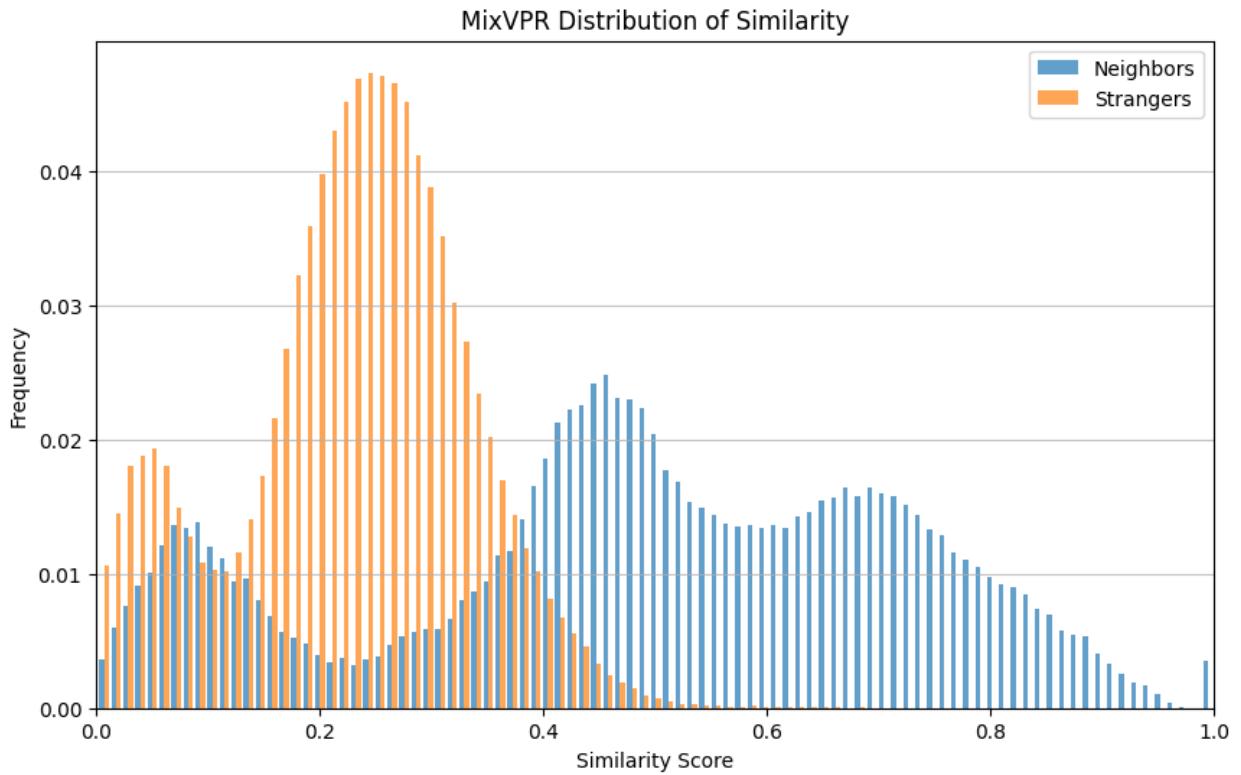
print(len(similarity_MVPR_strangers))
```

```
211530
```

```
plotting(similarity_MVPR_strangers, bins=100, title="MixVPR test  
strangers")
```

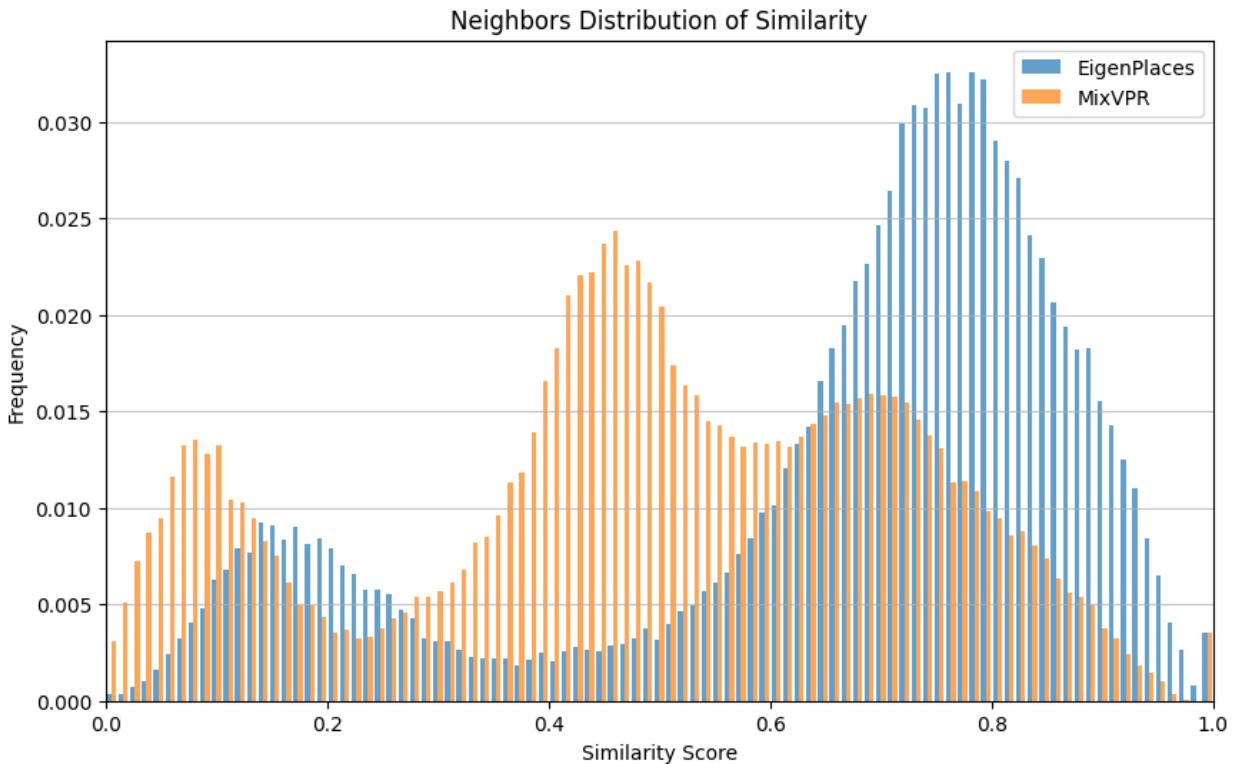


```
compare_plotting(similarity_MVPR_neighbors, similarity_MVPR_strangers,  
bins=100, title="MixVPR", label=['Neighbors', 'Strangers'])
```

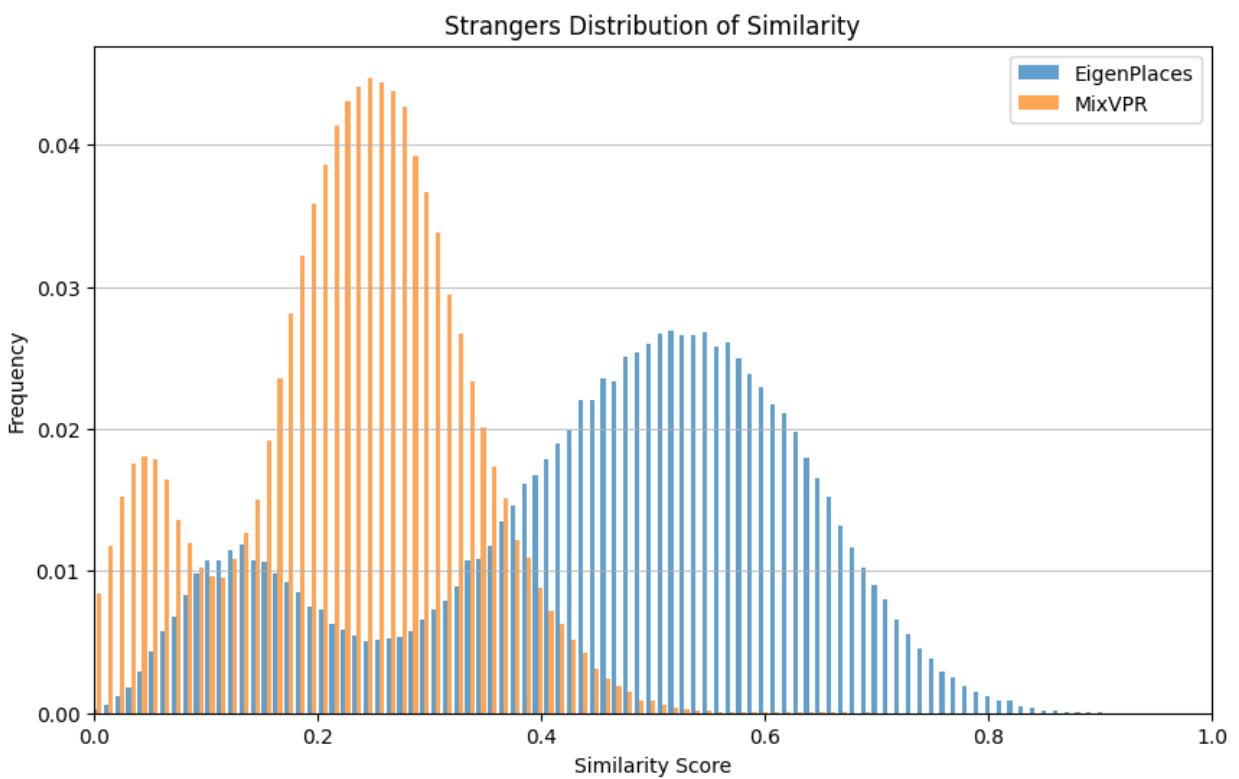


Cross Comparison

```
compare_plotting(similarity_EP_neighbors, similarity_MVPR_neighbors,  
bins=100, title="Neighbors", label=['EigenPlaces', 'MixVPR'])
```



```
compare_plotting(similarity_EP_strangers, similarity_MVPR_strangers,
bins=100, title='Strangers', label=['EigenPlaces', 'MixVPR'])
```



Examples

```
import matplotlib.pyplot as plt
from PIL import Image
import os

data_dir_path = "/Data/Trans_Proj/data/street_view_images_raw/"

def plot_image_pairs(image_pairs, title=''):
    # Set up the figure size and subplots
    fig, axes = plt.subplots(len(image_pairs), 2, figsize=(10, 3 * len(image_pairs)))

    # If only one pair, axes dimensions are reduced
    if len(image_pairs) == 1:
        axes = [axes]

    for i, (similarity, query_filename, target_filename) in enumerate(image_pairs):
        # Construct the full path to the images
        query_image_path = os.path.join(data_dir_path, query_filename + '.jpg')
        target_image_path = os.path.join(data_dir_path, target_filename + '.jpg')

        # Load the images
        query_image = Image.open(query_image_path)
        target_image = Image.open(target_image_path)

        # Plot query image
        axes[i][0].imshow(query_image)
        axes[i][0].axis('off') # Hide axes ticks

        # Plot target image
        axes[i][1].imshow(target_image)
        axes[i][1].axis('off')

        # Set title with similarity score
        axes[i][0].set_title(f'Query: {query_filename}')
        axes[i][1].set_title(f'Target: {target_filename} with Similarity: {similarity}')

    plt.suptitle(title, fontsize=16, fontweight='bold')
    plt.tight_layout(rect=[0, 0, 1, 0.95]) # Adjust the layout to make room for the big title
    plt.show()
```

EigenPlaces

Neighbor Positive Examples

```
plot_image_pairs(highest_pairs_EP_neighbors, title='EigenPlaces  
Neighbors Highest Pairs')
```

EigenPlaces Neighbors Highest Pairs

Query: 609561270181965

Target: 1748778565312532 with Similarity: 1.000000238418579



Query: 2954285588124617

Target: 399536178548774 with Similarity: 1.000000238418579



Query: 309100640760054

Target: 395343575474046 with Similarity: 1.000000238418579



Query: 899758967340446

Target: 1322617848178398 with Similarity: 1.000000238418579



Query: 1322617848178398

Target: 899758967340446 with Similarity: 1.000000238418579



Neighbor Negative Examples

```
plot_image_pairs(lowest_pairs_EP_neighbors, title='EigenPlaces  
Neighbors Lowest Pairs')
```

EigenPlaces Neighbors Lowest Pairs

Query: 293401535719720



Target: 487430195935464 with Similarity: -0.0340641215443611



Query: 487430195935464



Target: 293401535719720 with Similarity: -0.0340641215443611



Query: 105663068593368



Target: 201394418710336 with Similarity: -0.0107577145099639



Query: 201394418710336



Target: 105663068593368 with Similarity: -0.0107577145099639



Query: 145564747881430



Target: 1084084408786448 with Similarity: -0.0083721987903118



Stranger Positive Examples

```
plot_image_pairs(highest_pairs_EP_strangers, title='EigenPlaces  
Strangers Highest Pairs')
```

EigenPlaces Strangers Highest Pairs

Query: 1928284654039842



Target: 101613699132731 with Similarity: 0.9374951124191284



Query: 272283278396009



Target: 109342625038079 with Similarity: 0.920319437980652



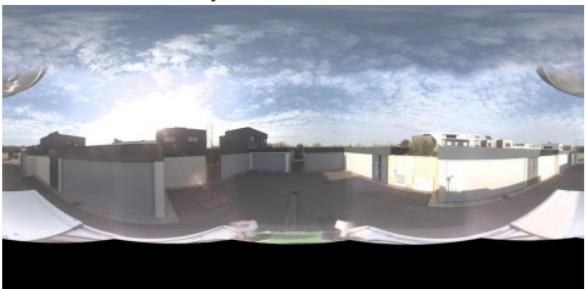
Query: 1928284654039842



Target: 3686422294815581 with Similarity: 0.9188045263290404



Query: 227993586147282



Target: 1228611517660836 with Similarity: 0.9184324741363524



Query: 1116889862183486



Target: 113824371155056 with Similarity: 0.9139269590377808



Stranger Negative Examples

```
plot_image_pairs(lowest_pairs_EP_strangers, title='EigenPlaces  
Strangers Lowest Pairs')
```

EigenPlaces Strangers Lowest Pairs

Query: 204868471763380



Target: 762363814439257 with Similarity: -0.031948335468769



Query: 876612829865940



Target: 283332893498027 with Similarity: -0.0316493883728981



Query: 1253944315062416



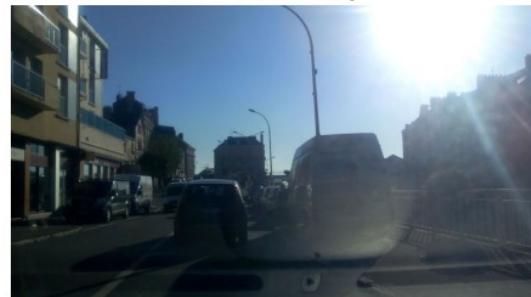
Target: 159131120229826 with Similarity: -0.0303909629583358



Query: 473473797101148



Target: 265212648658442 with Similarity: -0.0278368312865495



Query: 329246372328414



Target: 1084084408786448 with Similarity: -0.0259264633059501



MixVPR

Neigbhor Positive Examples

```
plot_image_pairs(highest_pairs_MVPR_neighbors, title='MixVPR Neighbors  
Highest Pairs')
```

MixVPR Neighbors Highest Pairs

Query: 251295937032632

Target: 936099213655292 with Similarity: 1.000000238418579



Query: 315554103349801

Target: 3154350254887281 with Similarity: 1.000000238418579



Query: 1779285525591503

Target: 4189073451204206 with Similarity: 1.000000238418579



Query: 4450942208353830

Target: 3195268940772588 with Similarity: 1.000000238418579



Query: 432670024878334

Target: 4328511490601335 with Similarity: 1.000000238418579



Neighbor Negative Examples

```
plot_image_pairs(lowest_pairs_MVPR_neighbors, title='MixVPR Neighbors  
Lowest Pairs')
```

MixVPR Neighbors Lowest Pairs

Query: 942976256500627



Target: 469632520991739 with Similarity: -0.0524557083845138



Query: 515998056096211



Target: 1214857095648492 with Similarity: -0.0452919229865074



Query: 1214857095648492



Target: 515998056096211 with Similarity: -0.0452919229865074



Query: 369109074542367



Target: 785259122384369 with Similarity: -0.0435949712991714



Query: 785259122384369



Target: 369109074542367 with Similarity: -0.0435949712991714



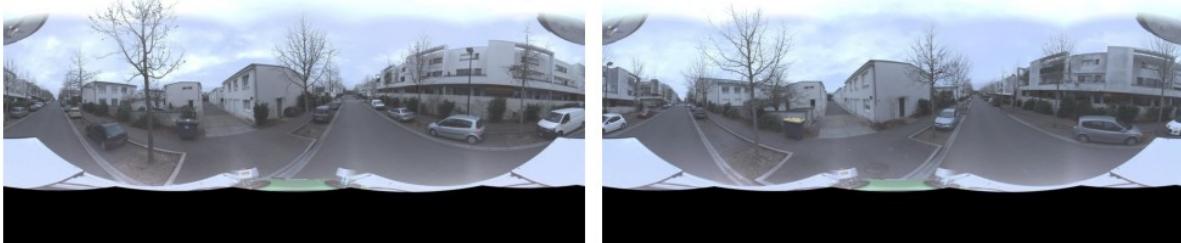
Stranger Positive Examples

```
plot_image_pairs(highest_pairs_MVPR_strangers, title='MixVPR Strangers  
Highest Pairs')
```

MixVPR Strangers Highest Pairs

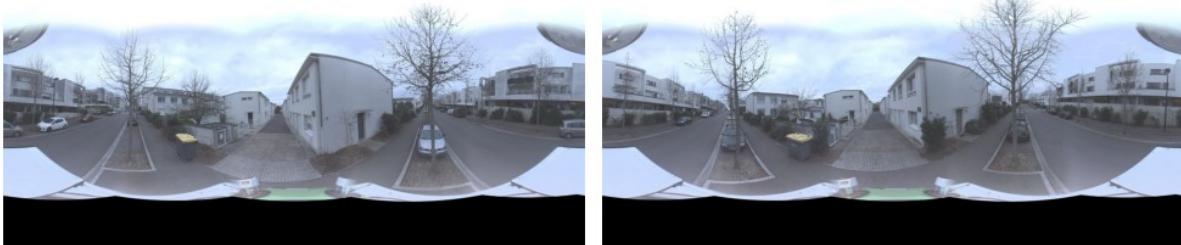
Query: 1116889862183486

Target: 113824371155056 with Similarity: 0.7368820905685425



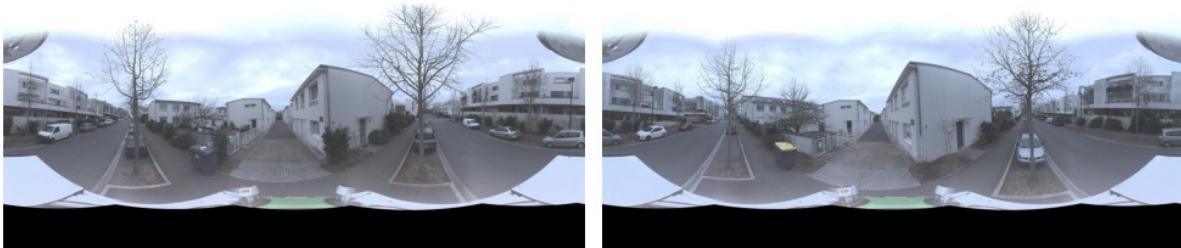
Query: 452749816292578

Target: 1363412787451632 with Similarity: 0.7310847043991089



Query: 1958446357668502

Target: 452749816292578 with Similarity: 0.7295624613761902



Query: 653869139383232

Target: 241320218079073 with Similarity: 0.7272611856460571



Query: 899319920718572

Target: 1101503527343644 with Similarity: 0.723192572593689



Stranger Negative Examples

```
plot_image_pairs(lowest_pairs_MVPR_strangers, title='MixVPR Strangers  
Lowest Pairs')
```

MixVPR Strangers Lowest Pairs

Query: 646214656640852



Target: 134216935300721 with Similarity: -0.0730977207422256



Query: 2190184974468035



Target: 165162768850606 with Similarity: -0.065721146762371



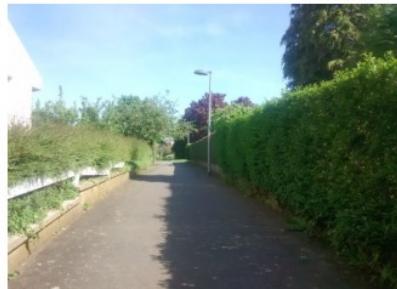
Query: 369823395144448



Target: 452291242456965 with Similarity: -0.0632998645305633



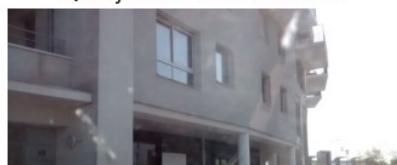
Query: 134216935300721



Target: 253634220266379 with Similarity: -0.0601971969008445



Query: 3034324073468170



Target: 108563788203194 with Similarity: -0.0601963400840759

