Julia Nelson
CS-558 Homework 3
5/15/2021

## Source Code:

### *histclassifier.py*

```
import os
import numpy as np
import pandas as pd
from matplotlib import image
from matplotlib import pyplot as plt
from sklearn.neighbors import KNeighborsClassifier

rgb_index = {'r': 0, 'g': 1, 'b': 2}
colors = {'r': 'RED', 'g': 'GREEN', 'b': 'BLUE'}


def scan_images(imgdirs):
    """
    Scan images and parse the file names in a list of source directories
    :param imgdirs: A list of directories
    :return: A list of images with parsed information
    """
    imglist = []
    imglist_test = []
    imglist_train = []
    # imdirs = ['ImClass/']
    for imdir in imgdirs:
        for filename in os.listdir(imdir):
            label, role = filename[:-5].split('_')
            id = int(filename.split('.')[0][-1])
            if role == 'train':
                imglist_train.append({'label': label,
                            'filename': filename,
                            'dir': imdir,
                            'role': role,
                            'id': id})
            else:
                imglist_test.append({'label': label,
                            'filename': filename,
                            'dir': imdir,
                            'role': role,
                            'id': id})
            imglist.append({'label': label,
                            'filename': filename,
                            'dir': imdir,
                            'role': role,
                            'id': id})
    # print("Training images:\n{}\n\n".format(imglist_train))
    # print("Testing images:\n{}".format(imglist_test))
    return imglist

def compute_multihist(imglist, bins, savefig=False):
    """
    Computes color histogram for each image in the set
    :param imglist: A list of images
    :param bins: Number of bins
    :param savefig: The histogram figure is saved in a file if savefig=True
    :return: Computed Data
    """
    bins_names = generate_bins_names(bins)
    # print(bins_names)
    df_header = {bin_name: [] for bin_name in bins_names}
    df_header['role'] = []
    df_header['label'] = []
    df_header['id'] = []
    # print(df_header)
    df_img = pd.DataFrame(df_header)
    for i, img in enumerate(imglist):
        img_data = image.imread(os.path.join(img['dir'], img['filename']))
        # print("{:02d} - Processing image ({} bins): {} ...".format(i, bins, img['filename']))
        # print("     Class:", img['label'])
        # print("     Role: ", img['role'])
        # print("     Shape: {}".format(img_data.shape))
        hists = np.empty(0)
        for color, index in rgb_index.items():
            hist, bins_ranges = np.histogram(img_data[:, :, index], bins=bins)
            hists = np.concatenate((hists, hist))
        df_img = df_img.append(pd.Series(hists, bins_names), ignore_index=True)
        df_img.loc[i, ['role', 'label', 'id']] = [img['role'], img['label'], img['id']]
    if savefig:
        save_histfig(df_img, bins)
    return df_img
```

```
def save_histfig(df_img, bins):
    """
    Save histogram figures to image files within 'plots/' directory
    :param df_img: Data of color histogram
    :param bins: Number of bins
    :return: None
    """
    bins_names_sep = generate_bins_names_sep(bins)
    bins_ranges = np.arange(1, bins + 1) * 255 / bins
    # print(bins_ranges, len(bins_ranges))
    for i, row in df_img.iterrows():
        for c, bins_names in bins_names_sep.items():
            plt.plot(
                bins_ranges,
                row[bins_names], c,
                label='{} Histogram'.format(colors[c]))
            plt.legend()
            plt.xlabel('Pixel values (0-255)')
            plt.ylabel('Number of pixels')
            try:
                os.mkdir('plots')
            except:
                pass
            plt.savefig('plots/bins_{:02d}_{}_{}_{}.jpg'.format(
                bins, row['label'], row['role'], int(row['id'])
            ))
        plt.clf()

def classify(df_img, bins, k):
    """
    Classify the data of color histogram for a given bins number and k nearest neighbor
    :param df_img: Data of color histogram
    :param bins: Number of bins
    :param k: Number of nearest neighbor
    :return: Classified data and test accuracy
    """
    bins_names = generate_bins_names(bins)
    X_train, Y_train, X_test, Y_test = split_train_test(df_img, bins_names)
    classifer = KNeighborsClassifier(algorithm='brute', n_neighbors=k)
    classifer.fit(X_train, Y_train['label'])
    pred_test = classifer.predict(X_test)
    acc_test = compute_accuracy(Y_test['label'], pred_test)
    return Y_test, pred_test, acc_test


def split_train_test(df_img, bins_names):
    """
    Split the data into training/testing data based on the filename
    :param df_img: Dataframe of images histograms
    :param bins_names: Set of bins names
    :return: Split data
    """
    # Split training data
    X_train = df_img[df_img['role'] == 'train']
    Y_train = X_train[['label', 'id']]
    X_train = X_train[bins_names]

    # Split testing data
    X_test = df_img[df_img['role'] == 'test']
    Y_test = X_test[['label', 'id']]
    X_test = X_test[bins_names]
    return X_train, Y_train, X_test, Y_test


def generate_bins_names(bins=8):
    """
    Generates a set of bins names for the dataset
    :param bins: Number of bins
    :return: Set of bins names for r, g, b
    """
    return ['{}_{:02d}'.format(color, bin_index)
            for color in rgb_index
            for bin_index in range(bins)]


def generate_bins_names_sep(bins=8):
    """
    Generates a set of separate bins names for the dataset
    :param bins: Number of bins
    :return: Set of bins names
    """
    r_bins = ['{}_{:02d}'.format('r', index) for index in range(bins)]
    g_bins = ['{}_{:02d}'.format('g', index) for index in range(bins)]
    b_bins = ['{}_{:02d}'.format('b', index) for index in range(bins)]
    return {'r': r_bins, 'g': g_bins, 'b': b_bins}


def compute_accuracy(actual, predicted):
    """
    Computes the test accuracy given a test sample with actual and predicted labels
    :param actual: Actual labels
    :param predicted: Predicted Labels
    :return: Test Accuracy (float)
    """
    matches = 0
    for i, act in enumerate(actual):
        if act == predicted[i]:
            matches += 1
    return matches / len(actual)
```

***main.py***

```python
import histclassifier


if __name__ == '__main__':
    bins_set = [4, 8, 16, 32]
    for bins in bins_set:
        imglist = histclassifier.scan_images(['ImClass/', ])
        df_img = histclassifier.compute_multihist(imglist, bins, True)
        for k in [1, 3]:
            actual, predicted, acc = histclassifier.classify(df_img, bins, k)
            print("\n\n================================")
            print("Classification using {} bins:".format(bins))
            print("--------------------------------")
            actual = actual.reset_index()
            for index, row in actual.iterrows():
                # print(index, row)
                print("Test image {} of class <{}> has been assigned to class <{}>".format(
                    int(row['id']), row['label'], predicted[index]
                ))
            print("~~~~~~~~~~~~~~~~~~~~~~")
            print("Test Accuracy(bins={}, k={}): {:.2f}%".format(
                bins, k, 100 * acc))
```

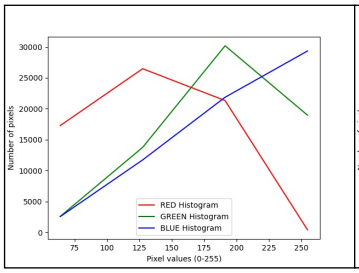
**Output:**

*output.txt*

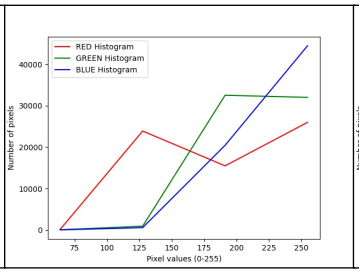
```
================================
Classification using 4 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <insidecity>
Test image 4 of class <forest> has been assigned to class <forest>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <insidecity>
Test image 4 of class <insidecity> has been assigned to class <coast>
Test image 2 of class <forest> has been assigned to class <forest>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=4, k=1): 83.33%


================================
Classification using 4 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <coast>
Test image 4 of class <forest> has been assigned to class <insidecity>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <insidecity>
Test image 4 of class <insidecity> has been assigned to class <coast>
Test image 2 of class <forest> has been assigned to class <insidecity>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=4, k=3): 58.33%


================================
Classification using 8 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <coast>
Test image 4 of class <forest> has been assigned to class <forest>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <insidecity>
Test image 4 of class <insidecity> has been assigned to class <insidecity>
Test image 2 of class <forest> has been assigned to class <forest>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=8, k=1): 83.33%


================================
Classification using 8 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <coast>
Test image 4 of class <forest> has been assigned to class <insidecity>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <insidecity> has been assigned to class <coast>
Test image 2 of class <forest> has been assigned to class <coast>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=8, k=3): 50.00%


================================
Classification using 16 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <forest> has been assigned to class <forest>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <insidecity> has been assigned to class <forest>
Test image 2 of class <forest> has been assigned to class <forest>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=16, k=1): 66.67%


================================
Classification using 16 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <forest> has been assigned to class <coast>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <insidecity> has been assigned to class <insidecity>
Test image 2 of class <forest> has been assigned to class <coast>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=16, k=3): 58.33%


================================
Classification using 32 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <forest> has been assigned to class <forest>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <insidecity> has been assigned to class <insidecity>
Test image 2 of class <forest> has been assigned to class <forest>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=32, k=1): 75.00%


================================
Classification using 32 bins:
--------------------------------
Test image 1 of class <forest> has been assigned to class <forest>
Test image 4 of class <coast> has been assigned to class <coast>
Test image 3 of class <forest> has been assigned to class <forest>
Test image 2 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <forest> has been assigned to class <coast>
Test image 3 of class <coast> has been assigned to class <coast>
Test image 1 of class <coast> has been assigned to class <coast>
Test image 2 of class <coast> has been assigned to class <coast>
Test image 3 of class <insidecity> has been assigned to class <forest>
Test image 4 of class <insidecity> has been assigned to class <insidecity>
Test image 2 of class <forest> has been assigned to class <coast>
Test image 1 of class <insidecity> has been assigned to class <forest>
~~~~~~~~~~~~~~~~~~~~~~
Test Accuracy(bins=32, k=3): 58.33%
```

## **Output Plots:**

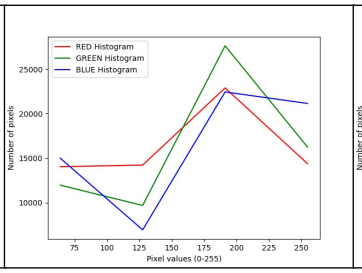
bins_04_**coast**_test_1



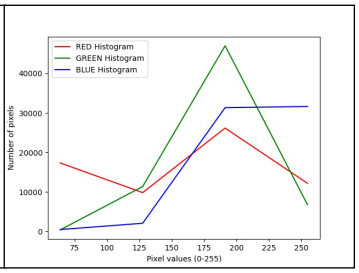

test_2



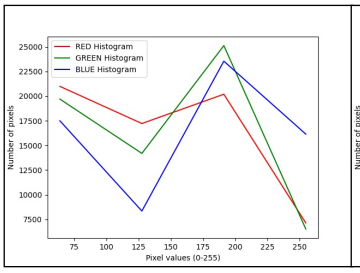

test_3



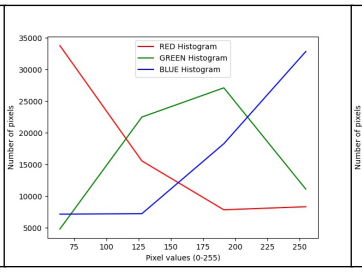

test_4



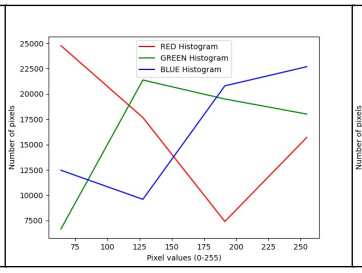

bins_04_**coast**_train_1



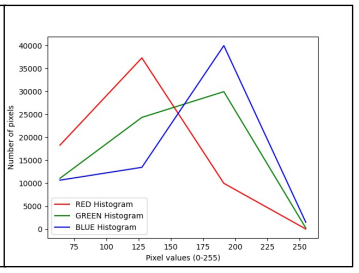

train_2



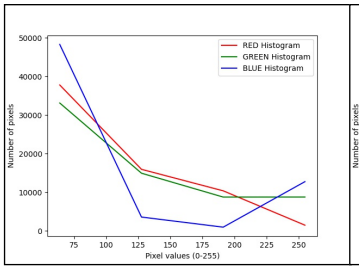

train_3



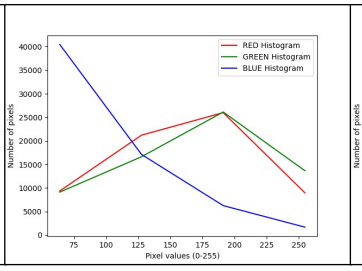

train_4



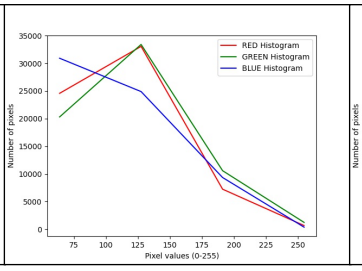

bins_04_**forest**_test_1



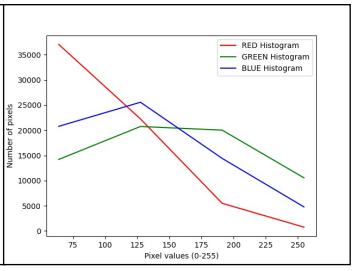

test_2



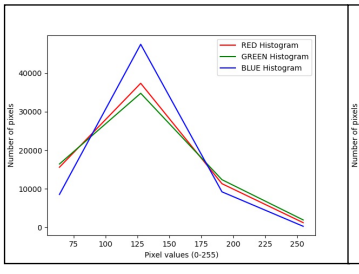

test_3



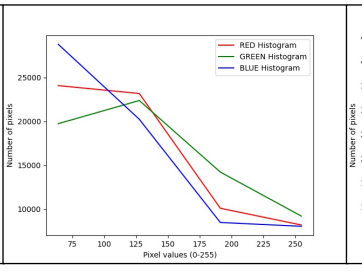

test_4



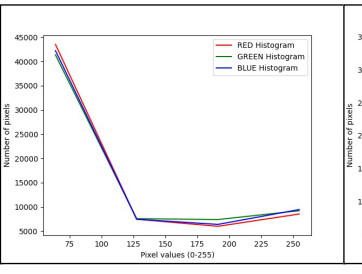

bins_04_**forest**_train_1



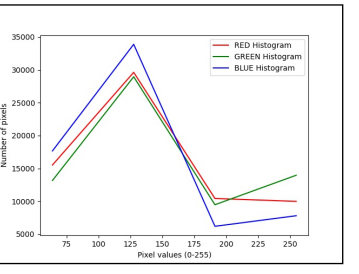

train_2



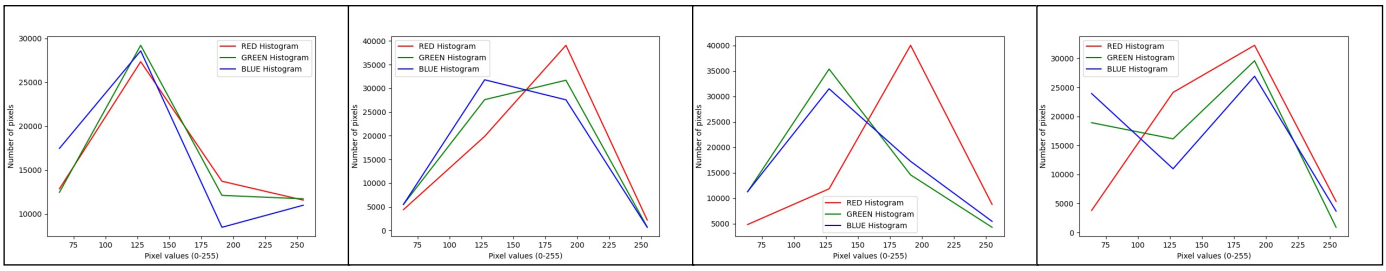

train_3



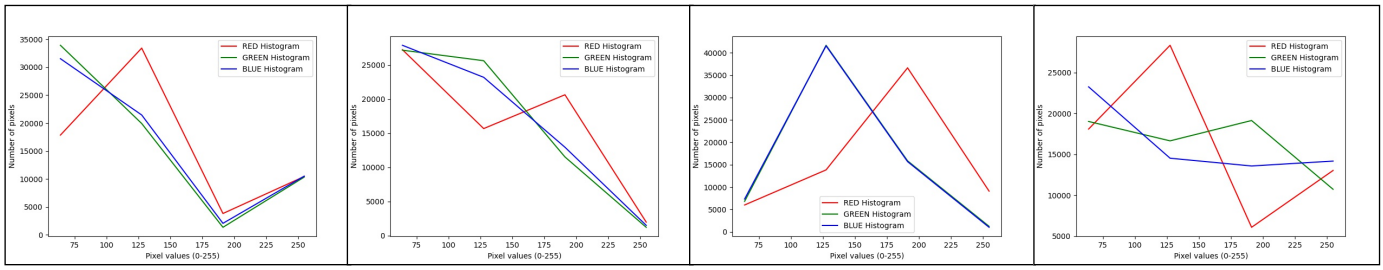

train_4



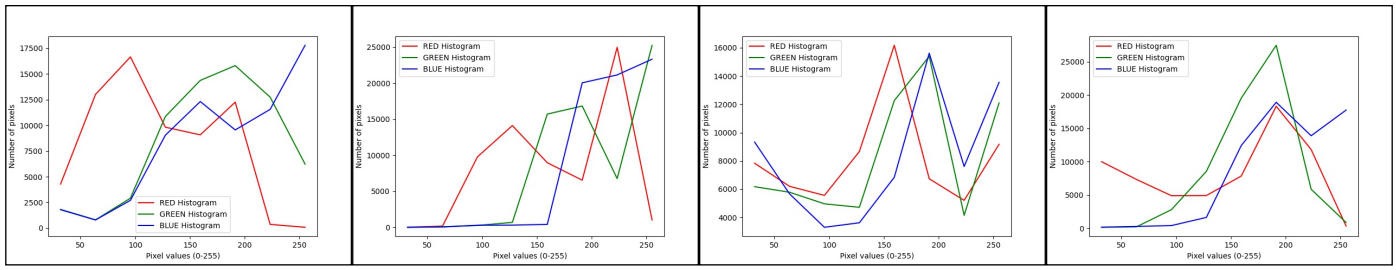
bins_04_**insidecity**_test_1 test_2 test_3 test_4



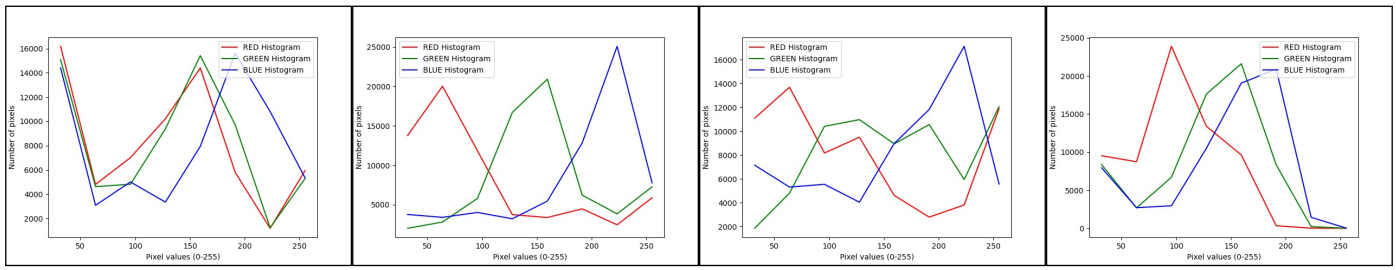

bins_04_**insidecity**_train_1 train_2 train_3 train_4



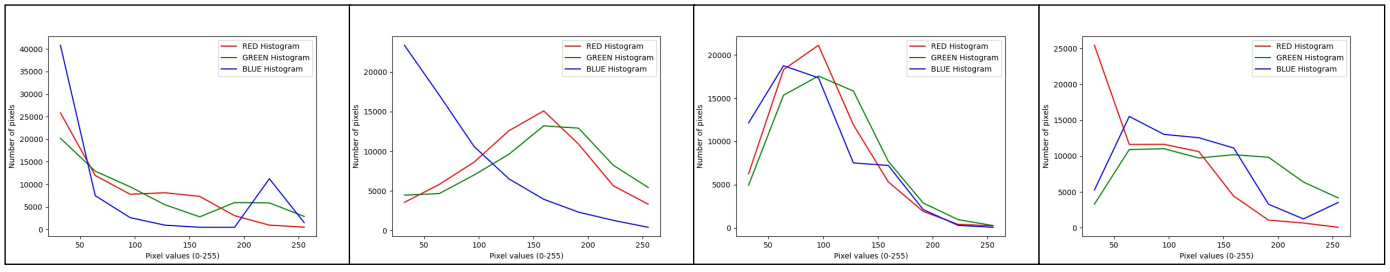

bins_08_**coast**_test_1 test_2 test_3 test_4



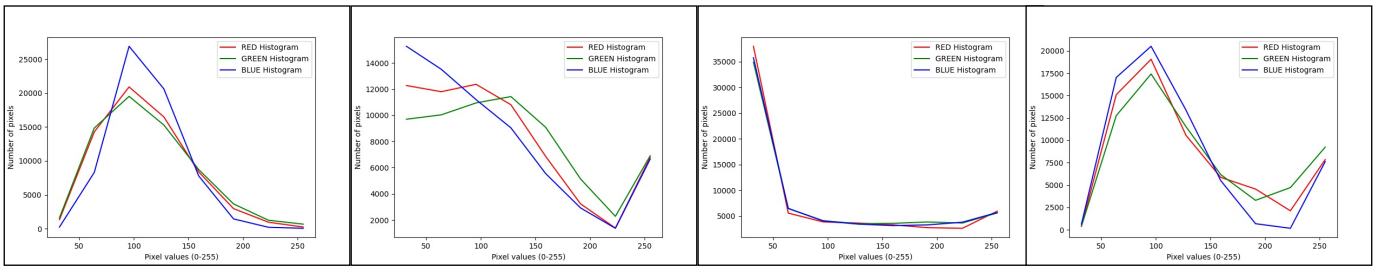

bins_08_**coast**_train_1 train_2 train_3 train_4



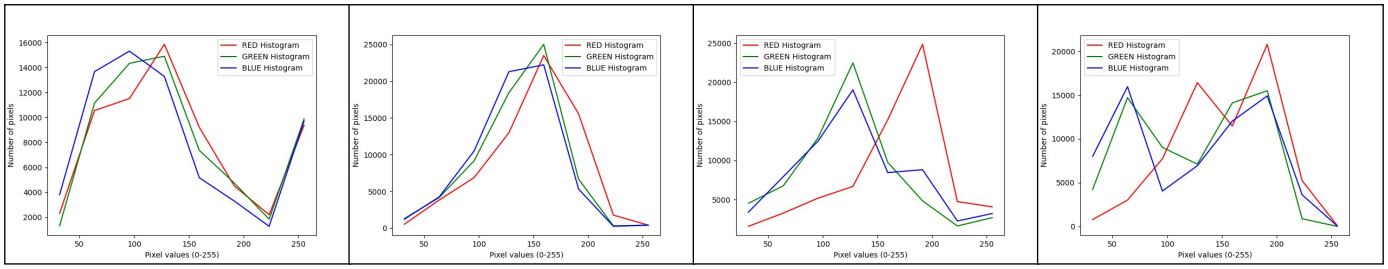

bins_08_**forest**_test_1 test_2 test_3 test_4



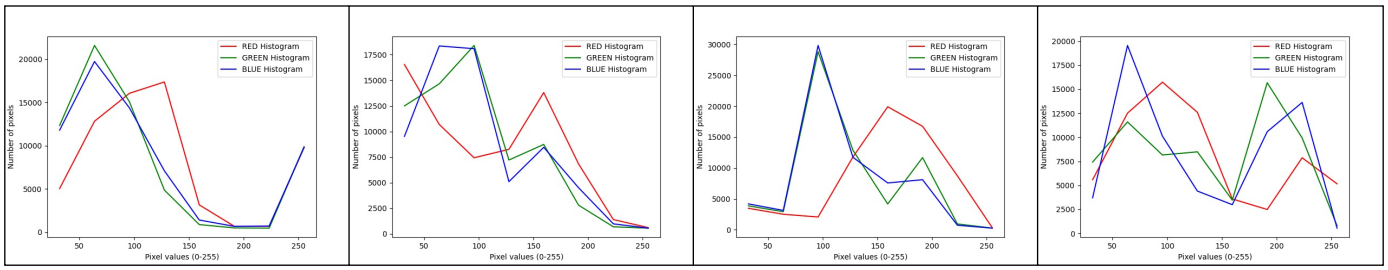
bins_08_**forest**_train_1 train_2 train_3 train_4



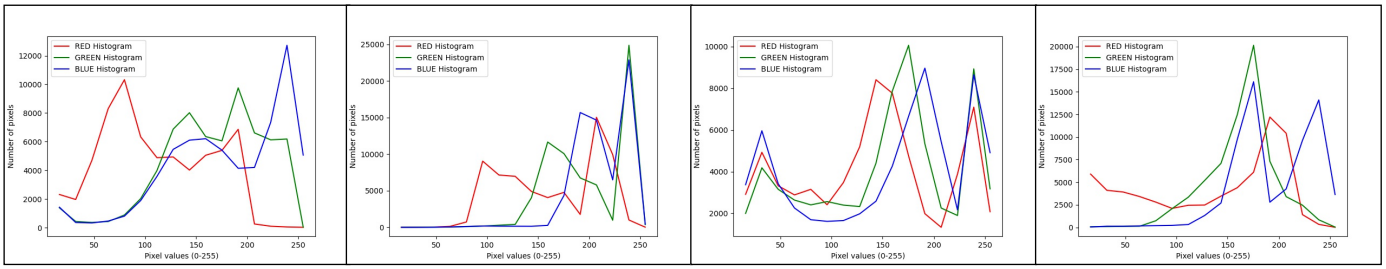

bins_08_**insidecity**_test_1 test_2 test_3 test_4



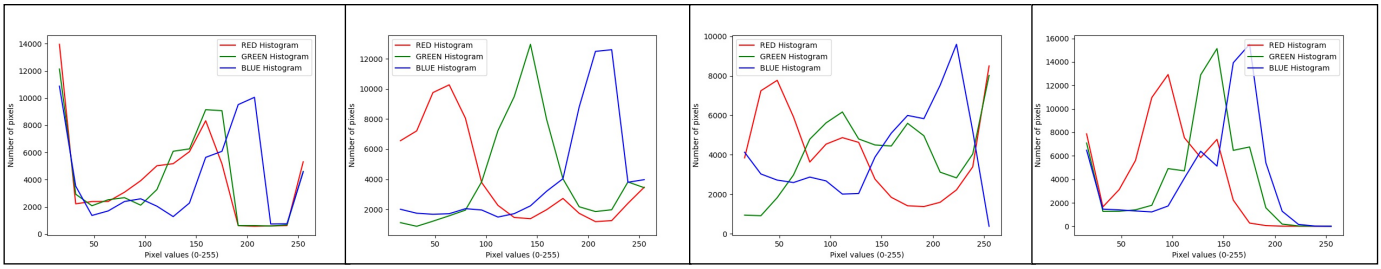

bins_08_**insidecity**_train_1 train_2 train_3 train_4



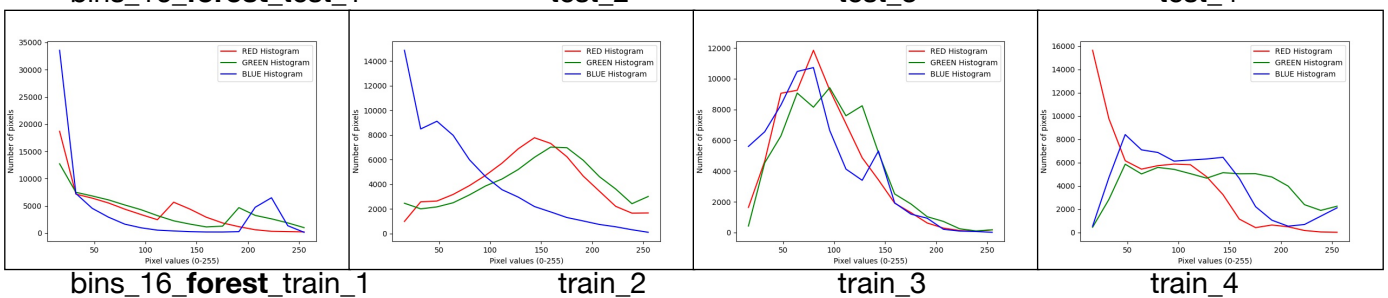

bins_16_**coast**_test_1 test_2 test_3 test_4



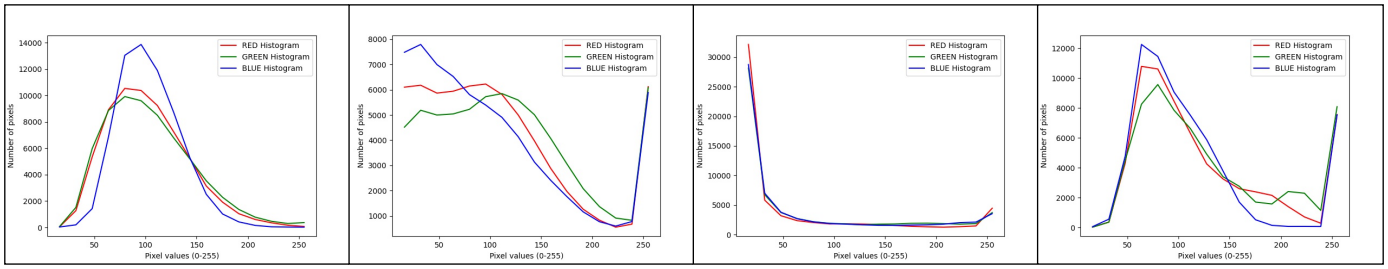

bins_16_**coast**_train_1 train_2 train_3 train_4



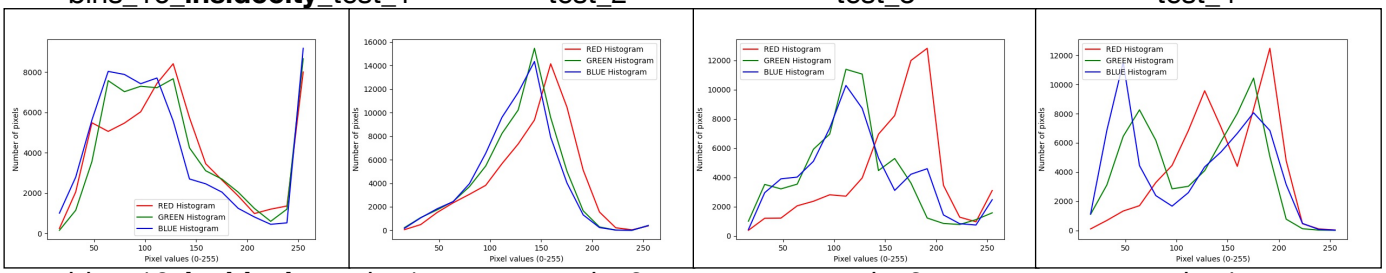
bins_16_**forest**_test_1 test_2 test_3 test_4



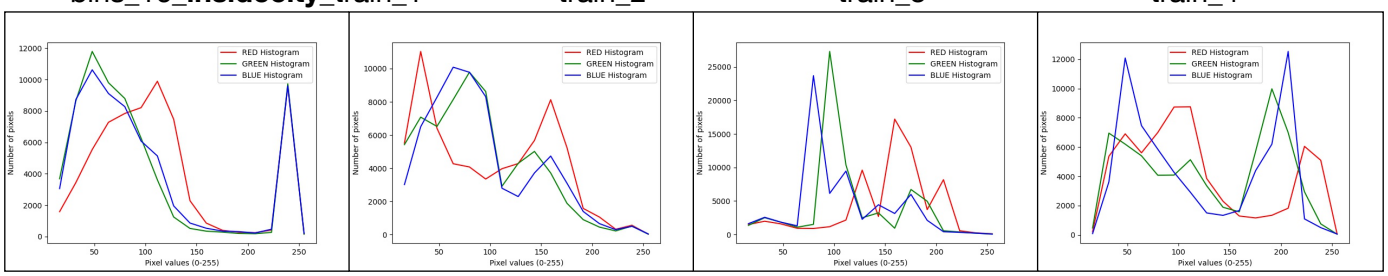

bins_16_**forest**_train_1 train_2 train_3 train_4



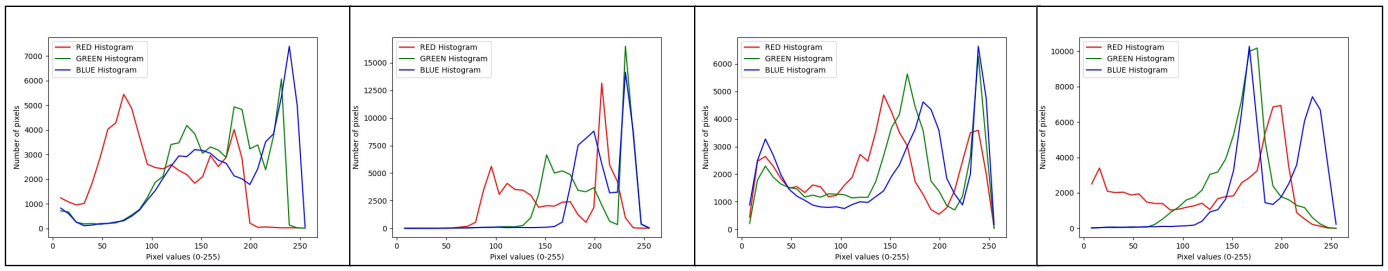

bins_16_**insidecity**_test_1 test_2 test_3 test_4



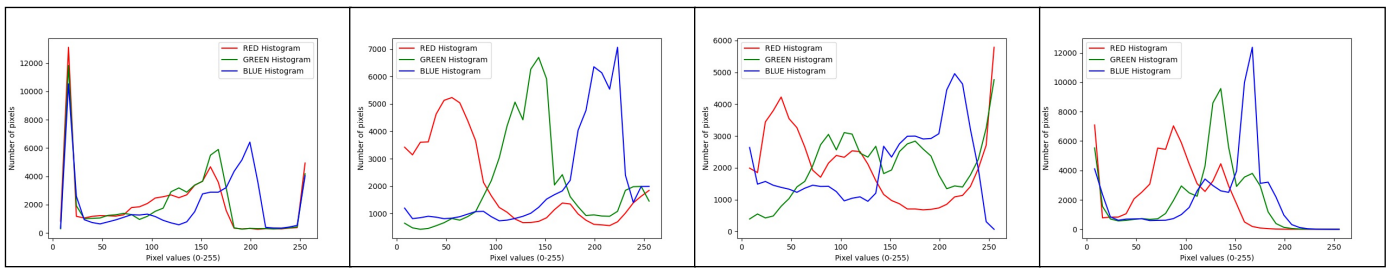

bins_16_**insidecity**_train_1 train_2 train_3 train_4



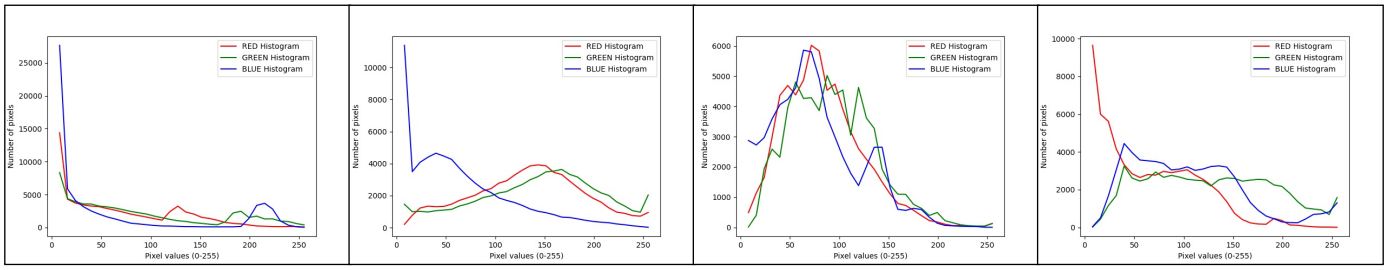

bins_32_**coast**_test_1 test_2 test_3 test_4



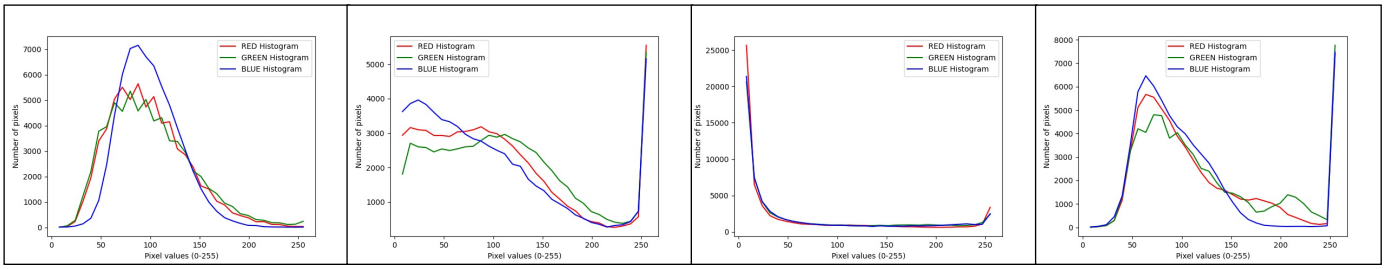

bins_32_**coast**_train_1 train_2 train_3 train_4



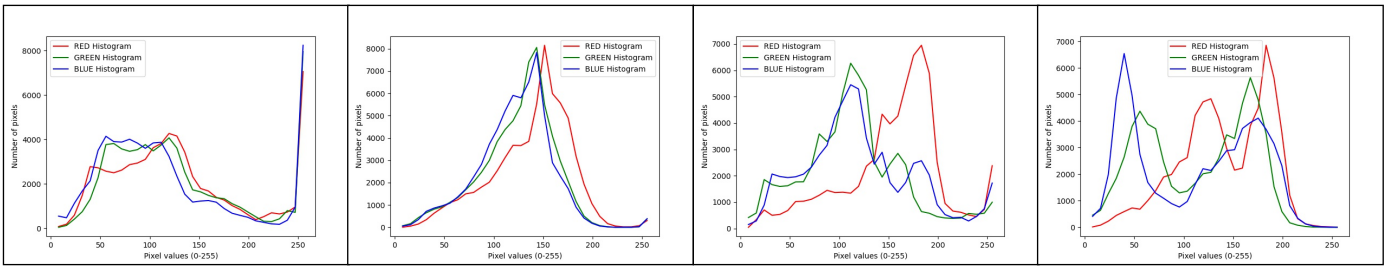
bins_32_**forest**_test_1 test_2 test_3 test_4




bins_32_**forest**_train_1 train_2 train_3 train_4




bins_32_**insidecity**_test_1 test_2 test_3 test_4




bins_32_**insidecity**_train_1 train_2 train_3 train_4