```
import numpy as np
import cv2
import matplotlib.pyplot as plt
from skimage import io
from skimage.color import rgb2gray
from PIL import Image as Image
from skimage.transform import resize
from sklearn.metrics import mean_squared_error
from skimage.measure import compare_ssim
from skimage.filters import sobel
from skimage.filters.rank import median
from skimage.morphology import disk
from skimage.measure.fit import ransac
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.metrics import r2_score
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.mixture import GaussianMixture
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor
from sklearn.svm import OneClassSVM
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.feature_selection import RFE
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score
from sklearn.metrics import cohen_kappa_score
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import log_loss
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import explained_variance_score
from sklearn.metrics import median_absolute_error
from sklearn.metrics import max_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import mean_poisson_deviance
from sklearn.metrics import mean_gamma_deviance
from sklearn.metrics import mean_tweedie_deviance
from sklearn.metrics import d2_tweedie_score
from sklearn.metrics import d2_pinball_score
from sklearn.metrics import d2_absolute_error
from sklearn.metrics import multilabel_confusion_matrix
from sklearn.metrics import hamming_loss
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import zero_one_loss
from sklearn.metrics import hinge_loss
from sklearn.metrics import brier_score_loss
from sklearn.metrics import calibration_curve
from sklearn.metrics import learning_curve
from sklearn.metrics import validation_curve
from sklearn.metrics import permutation_importance
from sklearn.metrics import partial_dependence
from sklearn.metrics import plot_partial_dependence
from sklearn.metrics import plot_permutation_importance
from sklearn.metrics import plot_learning_curve
from sklearn.metrics import plot_validation_curve
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import plot_roc_curve
from sklearn.metrics import plot_precision_recall_curve
from sklearn.metrics import plot_average_precision_score
from sklearn.metrics import plot_cohen_kappa_score
from sklearn.metrics import plot_matthews_corrcoef
from sklearn.metrics import plot_log_loss
from sklearn.metrics import plot_mean_absolute_error
from sklearn.metrics import plot_mean_squared_error
from sklearn.metrics import plot_r2_score
from sklearn.metrics import plot_explained_variance_score
from sklearn.metrics import plot_median_absolute_error
from sklearn.metrics import plot_max_error
from sklearn.metrics import plot_mean_absolute_percentage_error
from sklearn.metrics import plot_mean_squared_log_error
from sklearn.metrics import plot_mean_poisson_deviance
from sklearn.metrics import plot_mean_gamma_deviance
from sklearn.metrics import plot_mean_tweedie_deviance
from sklearn.metrics import plot_d2_tweedie_score
from sklearn.metrics import plot_d2_pinball_score
from sklearn.metrics import plot_d2_absolute_error
from sklearn.metrics import plot_multilabel_confusion_matrix
from sklearn.metrics import plot_hamming_loss
from sklearn.metrics import plot_jaccard_similarity_score
from sklearn.metrics import plot_zero_one_loss
from sklearn.metrics import plot_hinge_loss
from sklearn.metrics import plot_brier_score_loss
from sklearn.metrics import plot_calibration_curve
from sklearn.metrics import plot_learning_curve
from sklearn.metrics import plot_validation_curve
from sklearn.metrics import plot_permutation_importance
from sklearn.metrics import plot_partial_dependence
from sklearn.metrics import plot_permutation_importance
from sklearn.metrics import plot_learning_curve
from sklearn.metrics import plot_validation_curve
```

Question 1

Write a program to implement a region segmentation algorithm using the Sobel operator, a morphological filtering, and a connected 'Walking' type of an image. Assume that the segmentation is successful, and the segmentation results in two connected regions. (2 points)

```
img = io.imread('Problem1.jpg')
io.imshow(img)
io.show()
```

```
[[[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [255 255 255]
  [255 255 255]
  [255 255 255]]

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [255 255 255]
  [255 255 255]
  [255 255 255]]

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [255 255 255]
  [255 255 255]
  [255 255 255]]

 ...
```

```
img = io.imread('Problem1.jpg')
io.imshow(rgb2gray(img))
io.show()
```

```
img = io.imread('Problem1.jpg')
io.imshow(sobel(rgb2gray(img)))
io.show()
```

Question 2

Write a program to obtain the spatial and contrast cues using Kurt suggestions of an image instead of pixels. (2 points)

```
img = io.imread('Problem2.jpg')
io.imshow(rgb2gray(img))
io.show()
print(rgb2gray(img))
```

```
array([[0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75],
       [0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75],
       [0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75],
       ...,
       [0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75],
       [0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75],
       [0.75, 0.75, 0.75, ...,
        0.75, 0.75, 0.75]])
```

```
img = io.imread('Problem2.jpg')
io.imshow(rgb2gray(img))
io.show()
print(rgb2gray(img).shape)
print(rgb2gray(img).size)
print(rgb2gray(img).dtype)
print(rgb2gray(img).ndim)
print(rgb2gray(img).max())
print(rgb2gray(img).min()) # This is the minimum intensity of the image
print(rgb2gray(img).mean()) # This is the average intensity of the image
print(rgb2gray(img).std()) # This is the standard deviation of the image
print(rgb2gray(img).var()) # This is the variance of the image
print(rgb2gray(img)[0,0]) # This is the intensity of the pixel at (0,0)
print(rgb2gray(img)[0,0:5]) # This is the intensity of the pixels at (0,0) to (0,4)
print(rgb2gray(img)[0:5,0]) # This is the intensity of the pixels at (0,0) to (4,0)
print(rgb2gray(img)[0:5,0:5]) # This is the intensity of the pixels at (0,0) to (4,4)
print(rgb2gray(img)[0:5,0:5].shape)
print(rgb2gray(img)[0:5,0:5].size)
print(rgb2gray(img)[0:5,0:5].dtype)
print(rgb2gray(img)[0:5,0:5].ndim)
print(rgb2gray(img)[0:5,0:5].max())
print(rgb2gray(img)[0:5,0:5].min)
print(rgb2gray(img)[0:5,0:5].mean()
print(rgb2gray(img)[0:5,0:5].std()
print(rgb2gray(img)[0:5,0:5].var()
print(rgb2gray(img)[0:5,0:5][0,0])
print(rgb2gray(img)[0:5,0:5][0,0:5])
print(rgb2gray(img)[0:5,0:5][0:5,0])
print(rgb2gray(img)[0:5,0:5][0:5,0:5])
```

```
(1024, 1024)
1048576
float64
2
0.9882352941176471
0.0
0.7500000000000000
0.0000000000000000
0.7500000000000000
```