kernel3d32a7bd5e

Python notebook using data from Fruits 360 · 24 views · 15h ago · Edit tags

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

#import os
#for dirname, _, filenames in os.walk('/kaggle/input'):
#    for filename in filenames:
#        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

In [2]:

```
import os
print(os.listdir("../input/fruits/fruits-360_dataset/fruits-360/Training"))
```

```
['Lemon', 'Pear Kaiser', 'Onion Red', 'Ginger Root', 'Apple Red 3',
'Limes', 'Pomegranate', 'Grape White 3', 'Pepper Green', 'Pineappl
e', 'Apple Golden 3', 'Nectarine Flat', 'Mangostan', 'Potato Whit
e', 'Cocos', 'Onion White', 'Clementine', 'Tomato Yellow', 'Chestnu
t', 'Walnut', 'Tomato Cherry Red', 'Physalis', 'Pepper Red', 'Peppe
r Yellow', 'Apple Red Delicious', 'Apple Crimson Snow', 'Grapefruit
White', 'Pomelo Sweetie', 'Blueberry', 'Cauliflower', 'Cherry 1',
'Pear', 'Banana Red', 'Pear Monster', 'Melon Piel de Sapo', 'Pear F
orelle', 'Grape White 4', 'Tomato 2', 'Cantaloupe 2', 'Physalis wit
h Husk', 'Pear Williams', 'Apple Golden 1', 'Rambutan', 'Mango Re
d', 'Grape Pink', 'Apple Golden 2', 'Potato Red Washed', 'Tamarill
o', 'Cherry Wax Red', 'Huckleberry', 'Salak', 'Passion Fruit', 'Haz
elnut', 'Raspberry', 'Orange', 'Lemon Meyer', 'Kohlrabi', 'Tomato M
aroon', 'Tomato 3', 'Onion Red Peeled', 'Kiwi', 'Cactus fruit', 'Pi
tahaya Red', 'Lychee', 'Tomato 1', 'Maracuja', 'Quince', 'Potato Re
d', 'Pear Red', 'Apple Red 1', 'Avocado ripe', 'Potato Sweet', 'App
le Red Yellow 1', 'Banana Lady Finger', 'Peach 2', 'Peach Flat', 'D
ates', 'Nut Pecan', 'Avocado', 'Mulberry', 'Pear Abate', 'Banana',
'Grape Blue', 'Mandarine', 'Mango', 'Peach', 'Strawberry Wedge', 'K
umquats', 'Plum 2', 'Papaya', 'Cherry Wax Yellow', 'Pepino', 'Grape
White 2', 'Kaki', 'Apple Red 2', 'Pineapple Mini', 'Nectarine', 'To
mato 4', 'Cantaloupe 1', 'Eggplant', 'Nut Forest', 'Beetroot', 'App
le Granny Smith', 'Cherry Wax Black', 'Apple Red Yellow 2', 'Caramb
ula', 'Apple Braeburn', 'Granadilla', 'Plum 3', 'Grapefruit Pink',
'Apricot', 'Strawberry', 'Grape White', 'Apple Pink Lady', 'Redcurr
ant', 'Plum', 'Cherry Rainier', 'Cherry 2', 'Tangelo', 'Guava']
```

In [3]:
```
import numpy as np
import cv2
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
from matplotlib.offsetbox import OffsetImage, AnnotationBbox
import glob
import os
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import svm
from sklearn.cluster import KMeans
```

In [4]:
```
fruit_images = []
labels = []
for fruit_dir_path in glob.glob("../input/fruits/fruits-360_dataset/fruits-360/Training/*"):
    fruit_label = fruit_dir_path.split("/")[-1]
    for image_path in glob.glob(os.path.join(fruit_dir_path, "*.jpg")):
        image = cv2.imread(image_path, cv2.IMREAD_COLOR)

        image = cv2.resize(image, (45, 45))
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        fruit_images.append(image)
        labels.append(fruit_label)
fruit_images = np.array(fruit_images)
labels = np.array(labels)
```

In [5]:
```
print(labels)
```

```
['Lemon' 'Lemon' 'Lemon' ... 'Guava' 'Guava' 'Guava']
```

In [6]:
```
label_to_id_dict = {v:i for i,v in enumerate(np.unique(labels))}
id_to_label_dict = {v: k for k, v in label_to_id_dict.items()}
```

In [7]:
```
id_to_label_dict
```

Out[7]:
```
{0: 'Apple Braeburn',
 1: 'Apple Crimson Snow'
```

```
1: 'Apple Crimson Snow',
2: 'Apple Golden 1',
3: 'Apple Golden 2',
4: 'Apple Golden 3',
5: 'Apple Granny Smith',
6: 'Apple Pink Lady',
7: 'Apple Red 1',
8: 'Apple Red 2',
9: 'Apple Red 3',
10: 'Apple Red Delicious',
11: 'Apple Red Yellow 1',
12: 'Apple Red Yellow 2',
13: 'Apricot',
14: 'Avocado',
15: 'Avocado ripe',
16: 'Banana',
17: 'Banana Lady Finger',
18: 'Banana Red',
19: 'Beetroot',
20: 'Blueberry',
21: 'Cactus fruit',
22: 'Cantaloupe 1',
23: 'Cantaloupe 2',
24: 'Carambula',
25: 'Cauliflower',
26: 'Cherry 1',
27: 'Cherry 2',
28: 'Cherry Rainier',
29: 'Cherry Wax Black',
30: 'Cherry Wax Red',
31: 'Cherry Wax Yellow',
32: 'Chestnut',
33: 'Clementine',
34: 'Cocos',
35: 'Dates',
36: 'Eggplant',
37: 'Ginger Root',
38: 'Granadilla',
39: 'Grape Blue',
40: 'Grape Pink',
41: 'Grape White',
42: 'Grape White 2',
43: 'Grape White 3',
44: 'Grape White 4',
45: 'Grapefruit Pink',
46: 'Grapefruit White',
47: 'Guava',
48: 'Hazelnut',
49: 'Huckleberry',
50: 'Kaki',
51: 'Kiwi',
52: 'Kohlrabi',
53: 'Kumquats',
54: 'Lemon',
55: 'Lemon Meyer'
```

```
55: 'Lemon Meyer',
56: 'Limes',
57: 'Lychee',
58: 'Mandarine',
59: 'Mango',
60: 'Mango Red',
61: 'Mangostan',
62: 'Maracuja',
63: 'Melon Piel de Sapo',
64: 'Mulberry',
65: 'Nectarine',
66: 'Nectarine Flat',
67: 'Nut Forest',
68: 'Nut Pecan',
69: 'Onion Red',
70: 'Onion Red Peeled',
71: 'Onion White',
72: 'Orange',
73: 'Papaya',
74: 'Passion Fruit',
75: 'Peach',
76: 'Peach 2',
77: 'Peach Flat',
78: 'Pear',
79: 'Pear Abate',
80: 'Pear Forelle',
81: 'Pear Kaiser',
82: 'Pear Monster',
83: 'Pear Red',
84: 'Pear Williams',
85: 'Pepino',
86: 'Pepper Green',
87: 'Pepper Red',
88: 'Pepper Yellow',
89: 'Physalis',
90: 'Physalis with Husk',
91: 'Pineapple',
92: 'Pineapple Mini',
93: 'Pitahaya Red',
94: 'Plum',
95: 'Plum 2',
96: 'Plum 3',
97: 'Pomegranate',
98: 'Pomelo Sweetie',
99: 'Potato Red',
100: 'Potato Red Washed',
101: 'Potato Sweet',
102: 'Potato White',
103: 'Quince',
104: 'Rambutan',
105: 'Raspberry',
106: 'Redcurrant',
107: 'Salak',
108: 'Strawberry',
109: 'Strawberry Wedge'
```

```
110: 'Tamarillo',
111: 'Tangelo',
112: 'Tomato 1',
113: 'Tomato 2',
114: 'Tomato 3',
115: 'Tomato 4',
116: 'Tomato Cherry Red',
117: 'Tomato Maroon',
118: 'Tomato Yellow',
119: 'Walnut'}
```

In [8]:

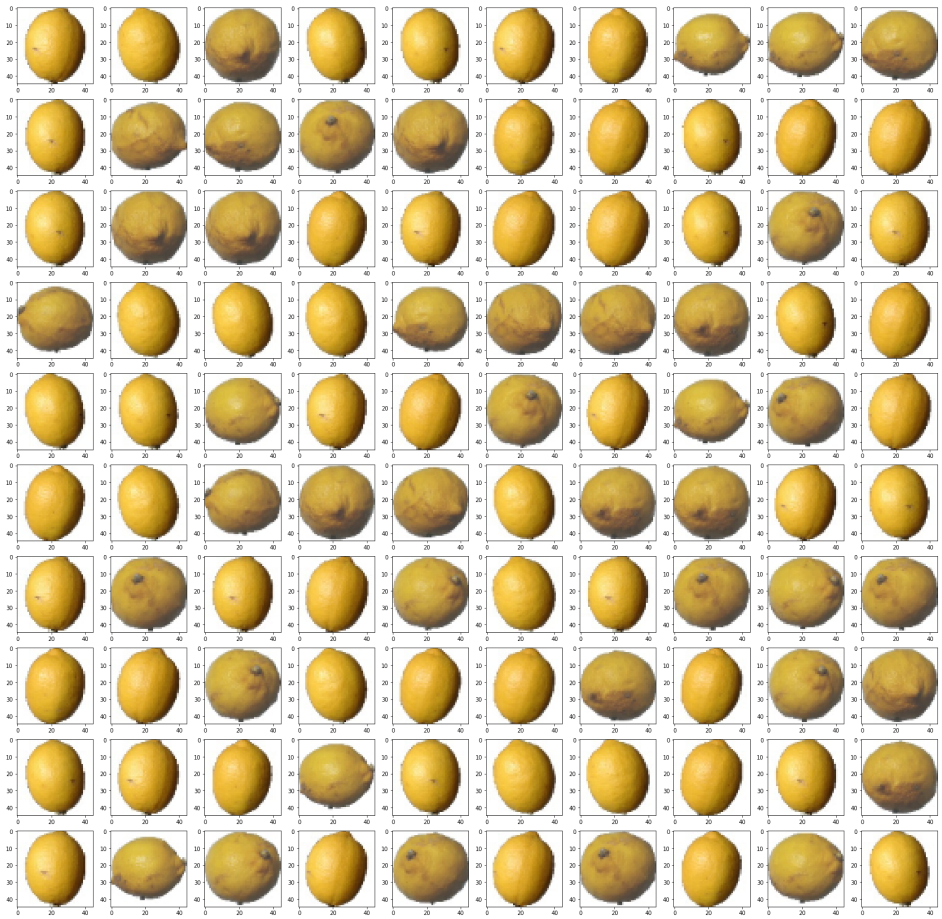
```
label_ids = np.array([label_to_id_dict[x] for x in labels])
```

In [9]:

```
def plot_image_grid(images, rows, columns):
    figure = plt.figure(figsize=(columns * 3, rows * 3))
    for i in range(columns * rows):
        figure.add_subplot(rows, columns, i + 1)
        plt.imshow(images[i])
    plt.show()
```

In [10]:

```
plot_image_grid(fruit_images[0:100], 10, 10)
```



In [11]:
```
scaler = StandardScaler()
```

In [12]:
```
images_scaled = scaler.fit_transform([i.flatten() for i in fruit_image
s])
```

In [13]:
```
pca = PCA(n_components=50)
pca_result = pca.fit_transform(images_scaled)
```

In [14]:
```
tsne = TSNE(n_components=2, perplexity=40.0)
tsne_result = tsne.fit_transform(pca_result)
tsne_result_scaled = StandardScaler().fit_transform(tsne_result)
```

In [15]:
```
tsnedf = pd.DataFrame()
tsnedf['x'] = list(tsne_result_scaled[:,0])
tsnedf['y'] = list(tsne_result_scaled[:,1])
tsnedf['label'] = labels
tsnedf.head()
```

Out[15]:

	x	y	label
0	-2.039382	-0.595430	Lemon
1	-1.989299	-0.766368	Lemon
2	0.640839	1.558324	Lemon
3	-2.002293	-0.719883	Lemon
4	-2.008383	-0.699009	Lemon

In [16]:
```
nb_classes = len(np.unique(label_ids))
sns.set_style('white')
#120 for 120 fruits, so 120 different colors
cmap = plt.cm.get_cmap("Spectral", 120)

plt.figure(figsize=(20,20))
for i, label_id in enumerate(np.unique(label_ids)):

    #plot matching labels to tsne results so labels are accurate
    plt.scatter(tsne_result_scaled[np.where(label_ids == label_id), 0
],
                tsne_result_scaled[np.where(label_ids == label_id), 1
],
                marker = '.',
                c = cmap(i),
                linewidth = '5',
                alpha=0.8,
```

```
                label = id_to_label_dict[label_id])
plt.title('T-SNE Plot (PCA 50 Components)', fontsize = 15)
plt.xlabel('X', fontsize = 15)
plt.ylabel('Y', fontsize = 15)
plt.legend(loc = 'center left', bbox_to_anchor = (1, 0.5), ncol = 2)
plt.show()
```

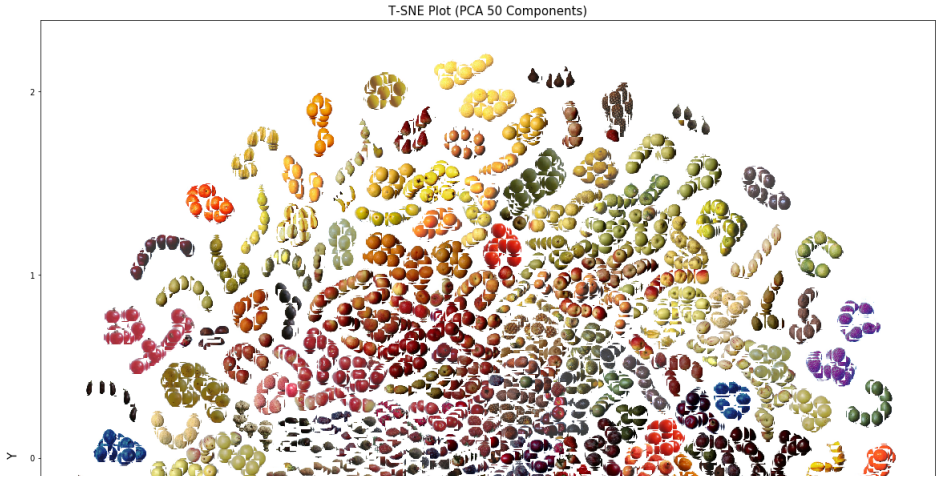
```
---------------------------------------------------------------------
--------
NameError                                 Traceback (most recent ca
ll last)
<ipython-input-16-c36c12a1d7c5> in <module>
      1 nb_classes = len(np.unique(label_ids))
----> 2 sns.set_style('white')
      3 #120 for 120 fruits, so 120 different colors
      4 cmap = plt.cm.get_cmap("Spectral", 120)
      5

NameError: name 'sns' is not defined
```

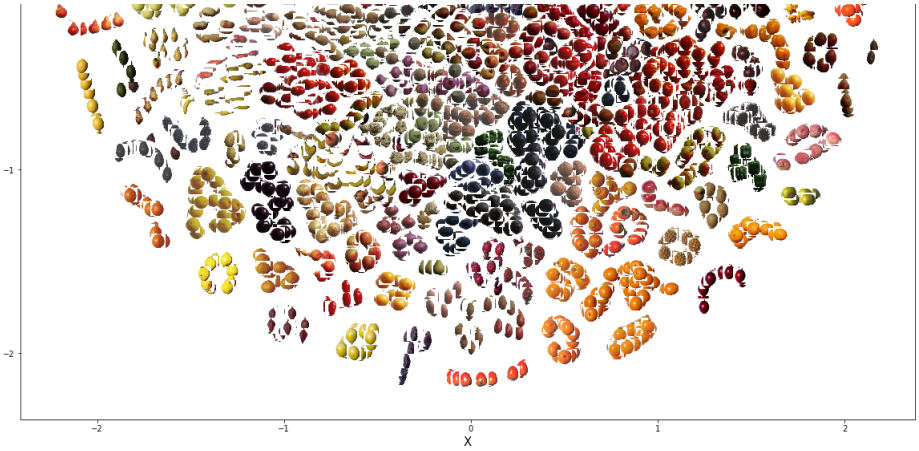
In [17]:

```
import seaborn as sns
```

In [18]:

```
fig, ax = plt.subplots(figsize=(20,20))
for df, i in zip(tsnedf.iterrows(), fruit_images):
    x = df[1]['x']
    y = df[1]['y']
    img = OffsetImage(i, zoom = .4)
    ab = AnnotationBbox(img, (x,y), xycoords = 'data', frameon = False
)
    ax.add_artist(ab)
ax.update_datalim(tsnedf[['x', 'y']].values)
ax.autoscale()
plt.title('T-SNE Plot (PCA 50 Components)', fontsize = 15)
plt.xlabel('X', fontsize = 15)
plt.ylabel('Y', fontsize = 15)
plt.show()
```





In [19]:
```
X_train, X_test, y_train, y_test = train_test_split(pca_result, label_
ids, test_size=0.25, random_state=42)
```

In [20]:
```
forest = RandomForestClassifier(n_estimators=10)
forest = forest.fit(X_train, y_train)
```

In [21]:
```
test_predictions = forest.predict(X_test)
```

In [22]:
```
precision = accuracy_score(test_predictions, y_test) * 100
print("Accuracy with RandomForest: {0:.6f}".format(precision))
```

```
Accuracy with RandomForest: 99.629752
```

In [23]:
```
svm_clf = svm.SVC()
svm_clf = svm_clf.fit(X_train, y_train)
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/svm/base.py:193: Fut
ureWarning: The default value of gamma will change from 'auto' to
'scale' in version 0.22 to account better for unscaled features. Se
t gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

In [24]:
```
test_predictions = svm_clf.predict(X_test)
```

In [25]:
```
precision = accuracy_score(test_predictions, y_test) * 100
print("Accuracy with SVM: {0:.6f}".format(precision))
```

```
Accuracy with SVM: 93.196694
```

In [26]:
```
validation_fruit_images = []
validation_labels = []
for fruit_dir_path in glob.glob("../input/fruits/fruits-360_dataset/fruits-360/Test/*"):
    fruit_label = fruit_dir_path.split("/")[-1]
    for image_path in glob.glob(os.path.join(fruit_dir_path, "*.jpg")):
        image = cv2.imread(image_path, cv2.IMREAD_COLOR)

        image = cv2.resize(image, (45, 45))
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        validation_fruit_images.append(image)
        validation_labels.append(fruit_label)
validation_fruit_images = np.array(validation_fruit_images)
validation_labels = np.array(validation_labels)
```

In [27]:
```
print(validation_labels)
```

```
['Lemon' 'Lemon' 'Lemon' ... 'Guava' 'Guava' 'Guava']
```

In [28]:
```
validation_label_ids = np.array([label_to_id_dict[x] for x in validation_labels])
```

In [29]:
```
validation_images_scaled = scaler.transform([i.flatten() for i in validation_fruit_images])
```

In [30]:
```
validation_pca_result = pca.transform(validation_images_scaled)
```

In [31]:
```
test_predictions = forest.predict(validation_pca_result)
```

In [32]:
```
precision = accuracy_score(test_predictions, validation_label_ids) * 100
print("Validation Accuracy with Random Forest: {0:.6f}".format(precision))
```

```
Validation Accuracy with Random Forest: 84.152846
```

In [33]:
```
test_predictions = svm_clf.predict(validation_pca_result)
```

In [34]:
```
precision = accuracy_score(test_predictions, validation_label_ids) * 1
00
print("Validation Accuracy with SVM: {0:.6f}".format(precision))
```


Data

Data Sources

- Fruits 360
  - fruits-360_dataset
    - fruits-360
      - LICENSE
      - papers
        - fruit_recognition_deep_learning.pdf
      - readme.md
      - Test
        - Apple Braeburn
          - 321_100.jpg
          - 322_100.jpg
          - 323_100.jpg
          - 324_100.jpg
          - 325_100.jpg
          - 326_100.jpg
          - 327_100.jpg
          - 32_100.jpg
          - 33_100.jpg
          - 34_100.jpg
          - 154 more
        - Apple Crimson Snow
          - 100_100.jpg
          - 101_100.jpg
          - 102_100.jpg
          - 103_100.jpg
          - 104_100.jpg

Fruits 360

A dataset with 82213 images of 120 fruits and vegetables

Last Updated: 2 months ago (Version 2)

About this Dataset

Fruits 360 dataset: A dataset of images containing fruits and vegetables

Version: 2019.09.21.0

Content

The following fruits and are included: Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambula, Cauliflower, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Eggplant, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Nut (Forest, Pecan), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Williams), Pepper (Red, Green, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow), Walnut.

Comments (0)

Click here to comment...