In [1]:

```
import os
from tqdm import tqdm

import pandas as pd
import numpy as np

import torch

import cv2
from PIL import Image
import warnings

import numpy as np
import pandas as pd

from PIL import Image

import torch
import torch.nn.functional as F

device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
print('device', device)
```

```
device cuda:0
```

image processing

In [3]:

```
from torchvision import transforms

train_transform = transforms.Compose([transforms.RandomResizedCrop(224),
                                      transforms.RandomHorizontalFlip(),
                                      transforms.ToTensor(),
                                      transforms.Normalize([0.485, 0.456, 0.406]
                                     ])


test_transform = transforms.Compose([transforms.Resize(256),
                                     transforms.CenterCrop(224),
                                     transforms.ToTensor(),
                                     transforms.Normalize(
                                         mean=[0.485, 0.456, 0.406],
                                         std=[0.229, 0.224, 0.225])
                                    ])
```

load test sets

In [6]:

```
dataset_dir = '/home/featurize/data'
test_path = os.path.join(dataset_dir, 'val')
from torchvision import datasets

test_dataset = datasets.ImageFolder(test_path, test_transform)
print('Number of test set images', len(test_dataset))
print('Number of categories', len(test_dataset.classes))
print('Name of each category', test_dataset.classes)
```

```
idx_to_labels = np.load('idx_to_labels.npy', allow_pickle=True).item()

classes = list(idx_to_labels.values())
print(classes)
```

```
Number of test set images 898
Number of categories 25
Name of each category ['Cherrytomatoes', 'Mangosteen', 'MomordicaCharantia', 'Nav
elOrange', 'Sandsugaroranges', 'apple', 'banana', 'carrot', 'cherries', 'cucumbe
r', 'durian', 'grape', 'hamimelon', 'kiwi', 'lemon', 'lichee', 'longan', 'mango',
'pear', 'pineapple', 'pitaya', 'pomegranate', 'strawberry', 'tomato', 'watermelo
n']
['CherryTomatoes', 'Mangosteen', 'MomordicaCharantia', 'NavelOrange', 'Sandsugaro
ranges', 'apple', 'banana', 'carrot', 'cherries', 'cucumber', 'durian', 'grape',
'hamimelon', 'kiwi', 'lemon', 'lichee', 'longan', 'mango', 'pear', 'pineapple',
'pitaya', 'pomegranate', 'strawberry', 'tomato', 'watermelon']
```

In [7]:

```
model = torch.load('/home/featurize/fruit25_pytorch.pth')
model = model.eval().to(device)
```

Form A - Test Set Image Path and Annotation

In [8]:

```
test_dataset.imgs[:10]
```

Out[8]:

```
[('/home/featurize/data/val/Cherrytomatoes/100.jpeg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/102.jpeg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/114.jpg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/12.jpg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/132.png', 0),
 ('/home/featurize/data/val/Cherrytomatoes/135.jpeg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/138.jpeg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/144.jpeg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/150.jpg', 0),
 ('/home/featurize/data/val/Cherrytomatoes/151.jpg', 0)]
```

In [9]:

```
img_paths = [each[0] for each in test_dataset.imgs]
```

In [10]:

```
df = pd.DataFrame()
df['image path'] = img_paths
df['Labelled category ID'] = test_dataset.targets
df['Name of labelling category'] = [idx_to_labels[ID] for ID in test_dataset.tar
```

In [11]:

```
df
```

Out[11]:

	image path	Labelled category ID	Name of labelling category
0	/home/featurize/data/val/Cherrytomatoes/100.jpeg	0	CherryTomatoes
1	/home/featurize/data/val/Cherrytomatoes/102.jpeg	0	CherryTomatoes
2	/home/featurize/data/val/Cherrytomatoes/114.jpg	0	CherryTomatoes
3	/home/featurize/data/val/Cherrytomatoes/12.jpg	0	CherryTomatoes
4	/home/featurize/data/val/Cherrytomatoes/132.png	0	CherryTomatoes
...	...	...	...
893	/home/featurize/data/val/watermelon/60.jpg	24	watermelon
894	/home/featurize/data/val/watermelon/69.jpg	24	watermelon
895	/home/featurize/data/val/watermelon/72.jpg	24	watermelon
896	/home/featurize/data/val/watermelon/87.jpg	24	watermelon
897	/home/featurize/data/val/watermelon/9.jpg	24	watermelon

898 rows × 3 columns

Table B - Image classification prediction results for each image in the test set, and confidence levels for each category

In [82]:

```
n = 3
```

In [25]:

```
df_pred = pd.DataFrame()
for idx, row in tqdm(df.iterrows()):
    img_path = row['image path']
    img_pil = Image.open(img_path).convert('RGB')
    input_img = test_transform(img_pil).unsqueeze(0).to(device)
    pred_logits = model(input_img)
    pred_softmax = F.softmax(pred_logits, dim=1)

    pred_dict = {}

    top_n = torch.topk(pred_softmax, n)
    pred_ids = top_n[1].cpu().detach().numpy().squeeze()


    for i in range(1, n+1):
        pred_dict['top-{}-anticipateID'.format(i)] = pred_ids[i-1]
        pred_dict['top-{}-Forecast name'.format(i)] = idx_to_labels[pred_ids[i-1
    pred_dict['top-nThe prediction is correct.'] = row['Labelled category ID'] i

    for idx, each in enumerate(classes):
        pred_dict['{}-predictive confidence'.format(each)] = pred_softmax[0][idx

    df_pred = df_pred._append(pred_dict, ignore_index=True)
```

```
898it [00:14, 62.65it/s]
```

In [26]:

```
df_pred
```

Out[26]:

	top-1-anticipateID	top-1-Forecast name	top-2-anticipateID	top-2-Forecast name	top-3-anticipateID	
0	23	tomato	0	CherryTomatoes	4	Sa
1	0	CherryTomatoes	23	tomato	21	
2	0	CherryTomatoes	23	tomato	22	
3	0	CherryTomatoes	23	tomato	22	
4	23	tomato	0	CherryTomatoes	22	
...	...	...	...	...	...	
893	24	watermelon	23	tomato	21	
894	23	tomato	0	CherryTomatoes	24	
895	24	watermelon	17	mango	21	
896	9	cucumber	24	watermelon	2	Mon
897	23	tomato	0	CherryTomatoes	24	

898 rows × 32 columns

Splice the two AB forms

In [27]:

```
df = pd.concat([df, df_pred], axis=1)
```

In [28]:

```
df
```

Out[28]:

	image path	Labelled category ID	Name of labelling category	ar
0	/home/featurize/data/val/Cherrytomatoes/100.jpeg	0	CherryTomatoes	
1	/home/featurize/data/val/Cherrytomatoes/102.jpeg	0	CherryTomatoes	
2	/home/featurize/data/val/Cherrytomatoes/114.jpg	0	CherryTomatoes	
3	/home/featurize/data/val/Cherrytomatoes/12.jpg	0	CherryTomatoes	
4	/home/featurize/data/val/Cherrytomatoes/132.png	0	CherryTomatoes	
...	...	...	...	
893	/home/featurize/data/val/watermelon/60.jpg	24	watermelon	
894	/home/featurize/data/val/watermelon/69.jpg	24	watermelon	
895	/home/featurize/data/val/watermelon/72.jpg	24	watermelon	
896	/home/featurize/data/val/watermelon/87.jpg	24	watermelon	
897	/home/featurize/data/val/watermelon/9.jpg	24	watermelon	

898 rows × 35 columns

Export complete table

In [29]:

```
df.to_csv('Test sets prediction results.csv', index=False)
```

In [30]:

```
idx_to_labels = np.load('idx_to_labels.npy', allow_pickle=True).item()
classes = list(idx_to_labels.values())
print(classes)
```

```
['CherryTomatoes', 'Mangosteen', 'MomordicaCharantia', 'NavelOrange', 'Sandsugaro
ranges', 'apple', 'banana', 'carrot', 'cherries', 'cucumber', 'durian', 'grape',
'hamimelon', 'kiwi', 'lemon', 'lichee', 'longan', 'mango', 'pear', 'pineapple',
'pitaya', 'pomegranate', 'strawberry', 'tomato', 'watermelon']
```

In [31]:

```
sum(df['Name of labelling category'] == df['top-1-Forecast name']) / len(df)
```

Out[31]:

```
0.6269487750556793
```

In [32]:

```
from sklearn.metrics import classification_report
```

In [33]:

```
print(classification_report(df['Name of labelling category'], df['top-1-Forecast
                    precision    recall  f1-score   support

    CherryTomatoes       0.42      0.71      0.52        38
        Mangosteen       0.78      0.80      0.79        35
MomordicaCharantia       0.55      0.80      0.65        35
       NavelOrange       0.51      0.51      0.51        37
  Sandsugaroranges       0.53      0.83      0.64        35
             apple       0.84      0.74      0.79        35
            banana       0.67      0.11      0.19        36
            carrot       0.73      0.53      0.61        36
          cherries       0.68      0.59      0.63        32
          cucumber       0.54      0.63      0.58        35
            durian       0.94      0.47      0.63        36
             grape       0.86      0.84      0.85        38
         hamimelon       0.65      0.59      0.62        37
              kiwi       0.80      0.43      0.56        37
             lemon       0.51      0.83      0.63        29
            lichee       0.78      0.47      0.59        38
            longan       0.77      0.71      0.74        38
             mango       0.45      0.61      0.52        33
              pear       0.40      0.68      0.51        37
         pineapple       0.74      0.70      0.72        37
            pitaya       0.79      0.94      0.86        36
       pomegranate       0.53      0.73      0.61        37
        strawberry       0.82      0.47      0.60        38
            tomato       0.63      0.47      0.54        36
        watermelon       0.79      0.51      0.62        37

          accuracy                           0.63       898
         macro avg       0.67      0.63      0.62       898
      weighted avg       0.67      0.63      0.62       898
```

In [34]:

```
report = classification_report(df['Name of labelling category'], df['top-1-Forec
del report['accuracy']
df_report = pd.DataFrame(report).transpose()
```

In [35]:

```
df_report
```

Out[35]:

	precision	recall	f1-score	support
CherryTomatoes	0.415385	0.710526	0.524272	38.0
Mangosteen	0.777778	0.800000	0.788732	35.0
MomordicaCharantia	0.549020	0.800000	0.651163	35.0
NavelOrange	0.513514	0.513514	0.513514	37.0
Sandsugaroranges	0.527273	0.828571	0.644444	35.0
apple	0.838710	0.742857	0.787879	35.0
banana	0.666667	0.111111	0.190476	36.0
carrot	0.730769	0.527778	0.612903	36.0
cherries	0.678571	0.593750	0.633333	32.0
cucumber	0.536585	0.628571	0.578947	35.0
durian	0.944444	0.472222	0.629630	36.0
grape	0.864865	0.842105	0.853333	38.0
hamimelon	0.647059	0.594595	0.619718	37.0
kiwi	0.800000	0.432432	0.561404	37.0
lemon	0.510638	0.827586	0.631579	29.0
lichee	0.782609	0.473684	0.590164	38.0
longan	0.771429	0.710526	0.739726	38.0
mango	0.454545	0.606061	0.519481	33.0
pear	0.403226	0.675676	0.505051	37.0
pineapple	0.742857	0.702703	0.722222	37.0
pitaya	0.790698	0.944444	0.860759	36.0
pomegranate	0.529412	0.729730	0.613636	37.0
strawberry	0.818182	0.473684	0.600000	38.0
tomato	0.629630	0.472222	0.539683	36.0
watermelon	0.791667	0.513514	0.622951	37.0
macro avg	0.668621	0.629115	0.621400	898.0
weighted avg	0.671058	0.626949	0.621458	898.0

In [36]:

```
accuracy_list = []
for fruit in tqdm(classes):
    df_temp = df[df['Name of labelling category']==fruit]
    accuracy = sum(df_temp['Name of labelling category'] == df_temp['top-1-Forec
    accuracy_list.append(accuracy)
```

```
100%|██████████| 25/25 [00:00<00:00, 1104.71it/s]
```

In [37]:

```
# Calculate macro-averaged and weighted-averaged accuracies
acc_macro = np.mean(accuracy_list)
acc_weighted = sum(accuracy_list * df_report.iloc[:-2]['support'] / len(df))
```

```
accuracy_list.append(acc_macro)
accuracy_list.append(acc_weighted)

df_report['accuracy'] = accuracy_list
```

In [38]:
```
df_report
```

Out[38]:

	precision	recall	f1-score	support	accuracy
CherryTomatoes	0.415385	0.710526	0.524272	38.0	0.710526
Mangosteen	0.777778	0.800000	0.788732	35.0	0.800000
MomordicaCharantia	0.549020	0.800000	0.651163	35.0	0.800000
NavelOrange	0.513514	0.513514	0.513514	37.0	0.513514
Sandsugaroranges	0.527273	0.828571	0.644444	35.0	0.828571
apple	0.838710	0.742857	0.787879	35.0	0.742857
banana	0.666667	0.111111	0.190476	36.0	0.111111
carrot	0.730769	0.527778	0.612903	36.0	0.527778
cherries	0.678571	0.593750	0.633333	32.0	0.593750
cucumber	0.536585	0.628571	0.578947	35.0	0.628571
durian	0.944444	0.472222	0.629630	36.0	0.472222
grape	0.864865	0.842105	0.853333	38.0	0.842105
hamimelon	0.647059	0.594595	0.619718	37.0	0.594595
kiwi	0.800000	0.432432	0.561404	37.0	0.432432
lemon	0.510638	0.827586	0.631579	29.0	0.827586
lichee	0.782609	0.473684	0.590164	38.0	0.473684
longan	0.771429	0.710526	0.739726	38.0	0.710526
mango	0.454545	0.606061	0.519481	33.0	0.606061
pear	0.403226	0.675676	0.505051	37.0	0.675676
pineapple	0.742857	0.702703	0.722222	37.0	0.702703
pitaya	0.790698	0.944444	0.860759	36.0	0.944444
pomegranate	0.529412	0.729730	0.613636	37.0	0.729730
strawberry	0.818182	0.473684	0.600000	38.0	0.473684
tomato	0.629630	0.472222	0.539683	36.0	0.472222
watermelon	0.791667	0.513514	0.622951	37.0	0.513514
macro avg	0.668621	0.629115	0.621400	898.0	0.629115
weighted avg	0.671058	0.626949	0.621458	898.0	0.626949

In [39]:
```
df_report.to_csv('Indicators for assessing accuracy by category.csv', index_labe
In [44]: model = torch.load('/home/featurize/fruit25_pytorch.pth')
         model = model.eval().to(device)
```

```
In [45]: from torchvision.models.feature_extraction import create_feature_extractor
```

```
In [46]: model_trunc = create_feature_extractor(model, return_nodes={'avgpool': 'semantic
```

```
In [47]: img_path = '/home/featurize/data/val/kiwi/167.png'
         img_pil = Image.open(img_path)
         input_img = test_transform(img_pil)
         input_img = input_img.unsqueeze(0).to(device)
         pred_logits = model_trunc(input_img)
```

```
In [48]: pred_logits['semantic_feature'].squeeze().detach().cpu().numpy().shape
```

```
Out[48]: (512,)
```

```
In [49]: df = pd.read_csv('/home/featurize/Test sets prediction results.csv')
```

```
In [50]: df.head()
```

Out[50]:

	image path	Labelled category ID	Name of labelling category	antic
0	/home/featurize/data/val/Cherrytomatoes/100.jpeg	0	CherryTomatoes	
1	/home/featurize/data/val/Cherrytomatoes/102.jpeg	0	CherryTomatoes	
2	/home/featurize/data/val/Cherrytomatoes/114.jpg	0	CherryTomatoes	
3	/home/featurize/data/val/Cherrytomatoes/12.jpg	0	CherryTomatoes	
4	/home/featurize/data/val/Cherrytomatoes/132.png	0	CherryTomatoes	

5 rows × 35 columns

Calculate the semantic features of each image in the test set

```
In [51]: encoding_array = []
         img_path_list = []

         for img_path in tqdm(df['image path']):
             img_path_list.append(img_path)
             img_pil = Image.open(img_path).convert('RGB')
             input_img = test_transform(img_pil).unsqueeze(0).to(device)
             feature = model_trunc(input_img)['semantic_feature'].squeeze().detach().cpu(
             encoding_array.append(feature)
         encoding_array = np.array(encoding_array)
```

```
100%|██████████| 898/898 [00:10<00:00, 87.61it/s]
```

```
In [52]: encoding_array.shape
```

```
Out[52]: (898, 512)
```

```
In [53]: np.save('Test set semantic features.npy', encoding_array)
```

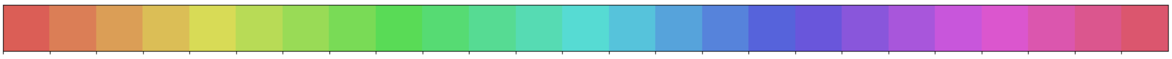
```
In [54]: import seaborn as sns
         marker_list = ['.', ',', 'o', 'v', '^', '<', '>', '1', '2', '3', '4', '8', 's',
```

```
In [55]: class_list = np.unique(df['Name of labelling category'])
```

```
In [56]: class_list
```

```
Out[56]: array(['CherryTomatoes', 'Mangosteen', 'MomordicaCharantia',
                'NavelOrange', 'Sandsugaroranges', 'apple', 'banana', 'carrot',
                'cherries', 'cucumber', 'durian', 'grape', 'hamimelon', 'kiwi',
                'lemon', 'lichee', 'longan', 'mango', 'pear', 'pineapple',
                'pitaya', 'pomegranate', 'strawberry', 'tomato', 'watermelon'],
               dtype=object)
```

```
In [57]: n_class = len(class_list) # Number of test set label categories
         palette = sns.hls_palette(n_class)
         sns.palplot(palette)
```



```
In [58]: import random
         random.seed(1234)
         random.shuffle(marker_list)
         random.shuffle(palette)
```

```
In [59]: from sklearn.manifold import TSNE

         tsne = TSNE(n_components=2, n_iter=20000)
         X_tsne_2d = tsne.fit_transform(encoding_array)
```

```
In [60]: X_tsne_2d.shape
```

```
Out[60]: (898, 2)
```

```
In [ ]:
```

```
In [61]: show_feature = 'Name of labelling category'
```

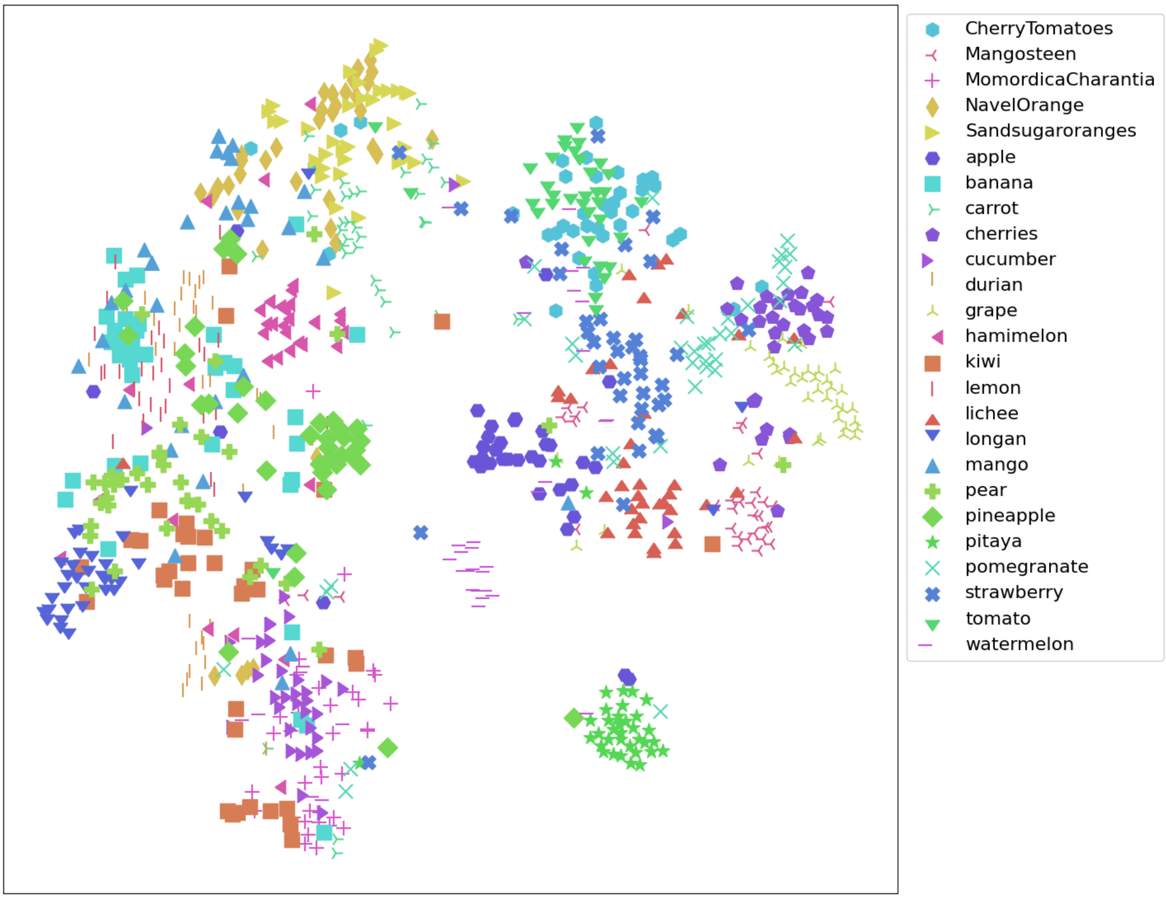
```
In [63]: import matplotlib
         import matplotlib.pyplot as plt
```

```
In [64]: plt.figure(figsize=(14, 14))
         for idx, fruit in enumerate(class_list):
             # Getting colours and point types
             color = palette[idx]
             marker = marker_list[idx%len(marker_list)]

             # Find the index numbers of all images labelled with the category as the cur
```

```
    indices = np.where(df[show_feature]==fruit)
    plt.scatter(X_tsne_2d[indices, 0], X_tsne_2d[indices, 1], color=color, marke

plt.legend(fontsize=16, markerscale=1, bbox_to_anchor=(1, 1))
plt.xticks([])
plt.yticks([])
plt.savefig('Two-dimensional downscaling visualisation of semantic features t-SN
plt.show()
```


CherryTomatoes
Mangosteen
MomordicaCharantia
NavelOrange
Sandsugaroranges
apple
banana
carrot
cherries
cucumber
durian
grape
hamimelon
kiwi
lemon
lichee
longan
mango
pear
pineapple
pitaya
pomegranate
strawberry
tomato
watermelon