\*\*參考底下列的這些，如果有多調其他參數，要把前面其他實驗補上原本的數字\*\*

\*\* svm新版的一定要開gpu才能跑成功

### 待處理：(做好就~~橫線~~槓掉)

\*~~SVM模型有一個參數kernel='rbf'我忘記加到報告了，你看需不需要加~~

~~自製dataset 分別用cnn和svm各一個同影像大小的model去test~~

~~椒鹽雜訊 先去雜訊再test 或是 先去雜訊再train~~ (SVM的好了, CNN的我懶得弄，掰了個理由)

~~related work有點空虛~~

看google test set分對的照片跟分錯的

改SVM kernel看看

縮小圖片裡物體看看

CNN視覺化看看

影片

CNN 128\*~~128~~\*3 1層conv 各層大小參數及結果 \*\*如果最後有時間再弄\*\*

檢查有沒有錯字或語意不順XD

報告和code貼到github

1. 影像大小：
2. 模型：
3. 訓練資料：
4. 測試資料：
5. 模型訓練時間：
6. 模型存檔：
7. 結果：(訓練和測試資料都測)

--------------------------------------------------------

### SVM 128 蘋果

(應該用不到，有需要再補時間)

1. 影像大小：(128,128,3)
2. 模型：model = SVC(C=100, kernel='rbf')
3. 訓練資料：kaggle train 蘋果(新鮮/腐爛)
4. 測試資料：kaggle test 蘋果(新鮮/腐爛)
5. 模型訓練時間：???
6. 模型存檔：model\_svm\_only\_apples.pkl
7. 結果：

Train data

The model is 98.14126394052045% accurate

report:

\*\*\*\*需要再補\*\*\*\*

confusion matrix:

[[1628 65]

[ 10 2332]]

Test Data

The model is 91.86746987951807% accurate

report:

\*\*\*\*需要再補\*\*\*\*

confusion matrix:

[[342 53]

[ 28 573]]

--------------------------------------------------------

### SVM 128 崩RAM

影像大小：(128,128,3)

模型：model = SVC(C=100, kernel='rbf')

訓練資料：kaggle

結果：在model.fit爆掉RAM

--------------------------------------------------------

### SVM 64 C=100

1. 影像大小：(64,64,3)
2. 模型：

model = SVC(C=100, kernel='rbf')

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：46.40410494804382 seconds
4. 模型存檔：model\_svm\_64.pkl
5. 結果：

Train data

The model is 96.68837721309971% accurate

report:

precision recall f1-score support

freshapples 0.9780 0.9457 0.9616 1693

freshbanana 0.9968 0.9949 0.9959 1581

freshoranges 0.9576 0.9714 0.9644 1466

rottenapples 0.9134 0.9778 0.9445 2342

rottenbanana 0.9996 0.9996 0.9996 2224

rottenoranges 0.9734 0.8959 0.9331 1595

accuracy 0.9669 10901

macro avg 0.9698 0.9642 0.9665 10901

weighted avg 0.9679 0.9669 0.9668 10901

confusion matrix:

[[1601 1 5 77 0 9]

[ 1 1573 0 0 0 7]

[ 5 0 1424 31 0 6]

[ 18 2 16 2290 0 16]

[ 0 0 0 0 2223 1]

[ 12 2 42 109 1 1429]]

Test data

The model is 89.43661971830986% accurate

report:

precision recall f1-score support

freshapples 0.8952 0.8430 0.8683 395

freshbanana 0.9446 0.9396 0.9421 381

freshoranges 0.9125 0.8866 0.8993 388

rottenapples 0.8108 0.9268 0.8649 601

rottenbanana 0.9843 0.9434 0.9634 530

rottenoranges 0.8560 0.7965 0.8252 403

accuracy 0.8944 2698

macro avg 0.9005 0.8893 0.8939 2698

weighted avg 0.8975 0.8944 0.8947 2698

confusion matrix:

[[333 5 2 47 0 8]

[ 9 358 0 2 3 9]

[ 1 4 344 28 0 11]

[ 18 2 10 557 2 12]

[ 3 5 0 8 500 14]

[ 8 5 21 45 3 321]]

--------------------------------------------------------

### SVM 64 C=10

1. 影像大小：(64,64,3)
2. 模型：

model = SVC(C=10, kernel='rbf')

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：42.514564752578735 seconds
4. 模型存檔：model\_svm\_64\_C10.pkl
5. 結果：

Train data

The model is 87.16631501697091% accurate

report:

precision recall f1-score support

freshapples 0.8578 0.8086 0.8325 1693

freshbanana 0.9519 0.9507 0.9513 1581

freshoranges 0.8650 0.9093 0.8866 1466

rottenapples 0.7853 0.8873 0.8332 2342

rottenbanana 0.9515 0.9537 0.9526 2224

rottenoranges 0.8382 0.6884 0.7559 1595

accuracy 0.8717 10901

macro avg 0.8750 0.8663 0.8687 10901

weighted avg 0.8731 0.8717 0.8704 10901

confusion matrix:

[[1369 27 26 210 22 39]

[ 14 1503 1 8 12 43]

[ 30 6 1333 75 2 20]

[ 107 3 64 2078 15 75]

[ 11 15 0 42 2121 35]

[ 65 25 117 233 57 1098]]

Test data

The model is 83.72868791697555% accurate

report:

precision recall f1-score support

freshapples 0.8095 0.7747 0.7917 395

freshbanana 0.8917 0.9291 0.9100 381

freshoranges 0.8495 0.8582 0.8538 388

rottenapples 0.7522 0.8436 0.7953 601

rottenbanana 0.9422 0.9226 0.9323 530

rottenoranges 0.7988 0.6700 0.7287 403

accuracy 0.8373 2698

macro avg 0.8407 0.8330 0.8353 2698

weighted avg 0.8386 0.8373 0.8364 2698

confusion matrix:

[[306 9 7 58 3 12]

[ 6 354 0 4 4 13]

[ 9 9 333 32 0 5]

[ 43 4 16 507 7 24]

[ 2 10 0 15 489 14]

[ 12 11 36 58 16 270]]

--------------------------------------------------------

### SVM 64 C=500

1. 影像大小：(64,64,3)
2. 模型：model = SVC(C=500, kernel='rbf')
3. 訓練資料：kaggle train all
4. 測試資料：kaggle test all
5. 模型訓練時間：64.0880835056305 seconds
6. 模型存檔：model\_svm\_64\_C500.pkl
7. 結果：

Train data

The model is 99.88074488579029% accurate

report:

precision recall f1-score support

freshapples 0.9988 0.9988 0.9988 1693

freshbanana 1.0000 1.0000 1.0000 1581

freshoranges 0.9993 0.9980 0.9986 1466

rottenapples 0.9962 0.9991 0.9977 2342

rottenbanana 1.0000 1.0000 1.0000 2224

rottenoranges 0.9994 0.9962 0.9978 1595

accuracy 0.9988 10901

macro avg 0.9989 0.9987 0.9988 10901

weighted avg 0.9988 0.9988 0.9988 10901

confusion matrix:

[[1691 0 0 2 0 0]

[ 0 1581 0 0 0 0]

[ 1 0 1463 2 0 0]

[ 1 0 0 2340 0 1]

[ 0 0 0 0 2224 0]

[ 0 0 1 5 0 1589]]

Test data

The model is 89.73313565604151% accurate

report:

precision recall f1-score support

freshapples 0.8529 0.8810 0.8667 395

freshbanana 0.9373 0.9423 0.9398 381

freshoranges 0.9079 0.8892 0.8984 388

rottenapples 0.8636 0.8952 0.8791 601

rottenbanana 0.9748 0.9472 0.9608 530

rottenoranges 0.8458 0.8164 0.8308 403

accuracy 0.8973 2698

macro avg 0.8970 0.8952 0.8959 2698

weighted avg 0.8980 0.8973 0.8975 2698

confusion matrix:

[[348 6 4 29 0 8]

[ 9 359 1 2 3 7]

[ 6 5 345 16 0 16]

[ 30 1 11 538 2 19]

[ 4 6 0 8 502 10]

[ 11 6 19 30 8 329]]

--------------------------------------------------------

### SVM 64 C=1000

1. 影像大小：64\*64\*3
2. 模型：model = SVC(C=1000, kernel='rbf')
3. 訓練資料：kaggle train all
4. 測試資料：kaggle test all
5. 模型訓練時間：65.36453485488892 seconds
6. 模型存檔：model\_svm\_64\_C1000.pkl
7. 結果：

Train data

The model is 99.99082652967618% accurate

report:

precision recall f1-score support

freshapples 1.0000 0.9994 0.9997 1693

freshbanana 1.0000 1.0000 1.0000 1581

freshoranges 1.0000 1.0000 1.0000 1466

rottenapples 0.9996 1.0000 0.9998 2342

rottenbanana 1.0000 1.0000 1.0000 2224

rottenoranges 1.0000 1.0000 1.0000 1595

accuracy 0.9999 10901

macro avg 0.9999 0.9999 0.9999 10901

weighted avg 0.9999 0.9999 0.9999 10901

confusion matrix:

[[1692 0 0 1 0 0]

[ 0 1581 0 0 0 0]

[ 0 0 1466 0 0 0]

[ 0 0 0 2342 0 0]

[ 0 0 0 0 2224 0]

[ 0 0 0 0 0 1595]]

Test data

The model is 89.54781319495922% accurate

report:

precision recall f1-score support

freshapples 0.8463 0.8785 0.8621 395

freshbanana 0.9375 0.9449 0.9412 381

freshoranges 0.9026 0.8840 0.8932 388

rottenapples 0.8615 0.8902 0.8756 601

rottenbanana 0.9729 0.9472 0.9598 530

rottenoranges 0.8501 0.8164 0.8329 403

accuracy 0.8955 2698

macro avg 0.8952 0.8935 0.8941 2698

weighted avg 0.8961 0.8955 0.8956 2698

confusion matrix:

[[347 6 5 29 0 8]

[ 9 360 1 2 3 6]

[ 7 5 343 17 0 16]

[ 32 1 13 535 2 18]

[ 4 6 0 8 502 10]

[ 11 6 18 30 9 329]]

--------------------------------------------------------

### SVM 64 C=1000 linear

1. 影像大小：64\*64\*3
2. 模型：model = SVC(C=1000, kernel='rbf')
3. 訓練資料：kaggle train all
4. 測試資料：kaggle test all
5. 模型訓練時間：65.36453485488892 seconds
6. 模型存檔：model\_svm\_64\_C1000.pkl
7. 結果：

Train data

The model is 100.0% accurate

report:

precision recall f1-score support

freshapples 1.0000 1.0000 1.0000 1693

freshbanana 1.0000 1.0000 1.0000 1581

freshoranges 1.0000 1.0000 1.0000 1466

rottenapples 1.0000 1.0000 1.0000 2342

rottenbanana 1.0000 1.0000 1.0000 2224

rottenoranges 1.0000 1.0000 1.0000 1595

accuracy 1.0000 10901

macro avg 1.0000 1.0000 1.0000 10901

weighted avg 1.0000 1.0000 1.0000 10901

confusion matrix:

[[1693 0 0 0 0 0]

[ 0 1581 0 0 0 0]

[ 0 0 1466 0 0 0]

[ 0 0 0 2342 0 0]

[ 0 0 0 0 2224 0]

[ 0 0 0 0 0 1595]]

Test data

The model is 70.38547071905114% accurate

report:

precision recall f1-score support

freshapples 0.5344 0.7089 0.6094 395

freshbanana 0.8274 0.8556 0.8413 381

freshoranges 0.7198 0.7216 0.7207 388

rottenapples 0.6824 0.6007 0.6389 601

rottenbanana 0.8621 0.7906 0.8248 530

rottenoranges 0.6197 0.5782 0.5982 403

accuracy 0.7039 2698

macro avg 0.7076 0.7093 0.7056 2698

weighted avg 0.7125 0.7039 0.7054 2698

confusion matrix:

[[280 11 25 50 19 10]

[ 19 326 3 3 9 21]

[ 34 9 280 33 2 30]

[125 1 43 361 13 58]

[ 39 23 1 24 419 24]

[ 27 24 37 58 24 233]]

--------------------------------------------------------

### CNN 3層conv ; size 128

1. 影像大小：128\*128\*3
2. 模型：

self.cnn\_layers = nn.Sequential(

nn.Conv2d(3, 64, 3, 1, 1),#output 64\*128\*128

nn.BatchNorm2d(64),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 64\*64\*64

nn.Conv2d(64, 128, 3, 1, 1),#output 128\*64\*64

nn.BatchNorm2d(128),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 128\*32\*32

nn.Conv2d(128, 256, 3, 1, 1),#output 256\*32\*32

nn.BatchNorm2d(256),

nn.ReLU(),

nn.MaxPool2d(4, 4, 0),#output 256\*8\*8

)

self.fc\_layers = nn.Sequential(

nn.Linear(256 \* 8 \* 8, 256),

nn.ReLU(),

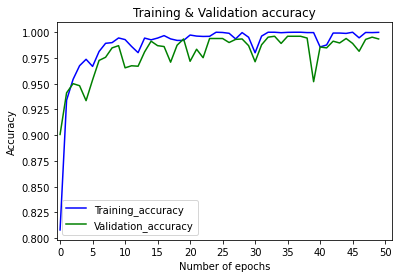
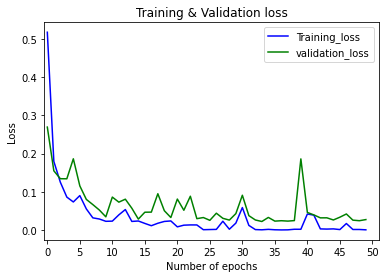
nn.Linear(256, 256),

nn.ReLU(),

nn.Linear(256, 6)

)

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：2896.7664663791656 seconds
4. 模型存檔：FreshnessDetector-1.pt
5. 結果：



Train data

The model is 99.85091743119267% accurate

report:

precision recall f1-score support

freshapples 0.9985 0.9993 0.9989 1350

freshbanana 0.9992 1.0000 0.9996 1287

freshoranges 0.9957 0.9991 0.9974 1155

rottenapples 0.9990 0.9979 0.9984 1914

rottenbanana 0.9994 0.9989 0.9991 1757

rottenoranges 0.9984 0.9960 0.9972 1257

accuracy 0.9985 8720

macro avg 0.9984 0.9985 0.9984 8720

weighted avg 0.9985 0.9985 0.9985 8720

confusion matrix:

[[1349 0 1 0 0 0]

[ 0 1287 0 0 0 0]

[ 0 1 1154 0 0 0]

[ 2 0 1 1910 0 1]

[ 0 0 0 1 1755 1]

[ 0 0 3 1 1 1252]]

Test data

The model is 99.36990363232023% accurate

report:

precision recall f1-score support

freshapples 0.9949 0.9949 0.9949 395

freshbanana 0.9948 0.9948 0.9948 381

freshoranges 0.9847 0.9974 0.9910 388

rottenapples 1.0000 0.9850 0.9925 601

rottenbanana 0.9981 1.0000 0.9991 530

rottenoranges 0.9852 0.9926 0.9889 403

accuracy 0.9937 2698

macro avg 0.9930 0.9941 0.9935 2698

weighted avg 0.9937 0.9937 0.9937 2698

confusion matrix:

[[393 2 0 0 0 0]

[ 0 379 0 0 0 2]

[ 0 0 387 0 0 1]

[ 2 0 3 592 1 3]

[ 0 0 0 0 530 0]

[ 0 0 3 0 0 400]]

--------------------------------------------------------

### CNN 3層conv ; size 64

1. 影像大小：64\*64\*3
2. 模型：

self.cnn\_layers = nn.Sequential(

nn.Conv2d(3, 64, 3, 1, 1),#output 64\*64\*64

nn.BatchNorm2d(64),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 64\*32\*32

nn.Conv2d(64, 128, 3, 1, 1),#output 128\*32\*32

nn.BatchNorm2d(128),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 128\*16\*16

nn.Conv2d(128, 256, 3, 1, 1),#output 256\*16\*16

nn.BatchNorm2d(256),

nn.ReLU(),

nn.MaxPool2d(4, 4, 0),#output 256\*4\*4

)

self.fc\_layers = nn.Sequential(

nn.Linear(256 \* 4 \* 4, 256),

nn.ReLU(),

nn.Linear(256, 256),

nn.ReLU(),

nn.Linear(256, 6)

)

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：3127.182713508606 seconds
4. 模型存檔：FreshnessDetector-64\_64\_3.pt
5. 結果：

Train data

The model is 97.46559633027523% accurate

report:

precision recall f1-score support

freshapples 1.0000 0.9807 0.9903 1348

freshbanana 0.9968 1.0000 0.9984 1265

freshoranges 0.9949 1.0000 0.9974 1169

rottenapples 0.8992 0.9984 0.9462 1886

rottenbanana 1.0000 0.9531 0.9760 1771

rottenoranges 1.0000 0.9149 0.9556 1281

accuracy 0.9747 8720

macro avg 0.9818 0.9745 0.9773 8720

weighted avg 0.9771 0.9747 0.9749 8720

confusion matrix:

[[1322 0 0 26 0 0]

[ 0 1265 0 0 0 0]

[ 0 0 1169 0 0 0]

[ 0 0 3 1883 0 0]

[ 0 4 0 79 1688 0]

[ 0 0 3 106 0 1172]]

Test data

The model is 96.10822831727205% accurate

report:

precision recall f1-score support

freshapples 1.0000 0.9519 0.9754 395

freshbanana 0.9948 0.9974 0.9961 381

freshoranges 0.9898 0.9974 0.9936 388

rottenapples 0.8584 0.9983 0.9231 601

rottenbanana 1.0000 0.9283 0.9628 530

rottenoranges 1.0000 0.8883 0.9409 403

accuracy 0.9611 2698

macro avg 0.9738 0.9603 0.9653 2698

weighted avg 0.9662 0.9611 0.9616 2698

confusion matrix:

[[376 1 1 17 0 0]

[ 0 380 0 1 0 0]

[ 0 0 387 1 0 0]

[ 0 0 1 600 0 0]

[ 0 1 0 37 492 0]

[ 0 0 2 43 0 358]]

--------------------------------------------------------

### CNN 3層conv ; size 64 v2

(重跑一遍)

1. 影像大小：64\*64\*3
2. 模型：

self.cnn\_layers = nn.Sequential(

nn.Conv2d(3, 64, 3, 1, 1),#output 64\*64\*64

nn.BatchNorm2d(64),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 64\*32\*32

nn.Conv2d(64, 128, 3, 1, 1),#output 128\*32\*32

nn.BatchNorm2d(128),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 128\*16\*16

nn.Conv2d(128, 256, 3, 1, 1),#output 256\*16\*16

nn.BatchNorm2d(256),

nn.ReLU(),

nn.MaxPool2d(4, 4, 0),#output 256\*4\*4

)

self.fc\_layers = nn.Sequential(

nn.Linear(256 \* 4 \* 4, 256),

nn.ReLU(),

nn.Linear(256, 256),

nn.ReLU(),

nn.Linear(256, 6)

)

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：3127.182713508606 seconds
4. 模型存檔：FreshnessDetector-64\_64\_3\_v2.pt
5. 結果：

(Train 待補)

Test data

The model is 98.1097108969607% accurate

report:

precision recall f1-score support

freshapples 0.9517 0.9975 0.9740 395

freshbanana 0.9896 1.0000 0.9948 381

freshoranges 0.9483 0.9923 0.9698 388

rottenapples 0.9982 0.9401 0.9683 601

rottenbanana 0.9981 0.9981 0.9981 530

rottenoranges 0.9899 0.9752 0.9825 403

accuracy 0.9811 2698

macro avg 0.9793 0.9839 0.9813 2698

weighted avg 0.9818 0.9811 0.9811 2698

confusion matrix:

[[394 1 0 0 0 0]

[ 0 381 0 0 0 0]

[ 0 0 385 0 0 3]

[ 20 2 12 565 1 1]

[ 0 1 0 0 529 0]

[ 0 0 9 1 0 393]]

--------------------------------------------------------

### CNN 3層conv ; size 64 v3

(再再跑一遍)

1. 影像大小：64\*64\*3
2. 模型：

self.cnn\_layers = nn.Sequential(

nn.Conv2d(3, 64, 3, 1, 1),#output 64\*64\*64

nn.BatchNorm2d(64),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 64\*32\*32

nn.Conv2d(64, 128, 3, 1, 1),#output 128\*32\*32

nn.BatchNorm2d(128),

nn.ReLU(),

nn.MaxPool2d(2, 2, 0),#output 128\*16\*16

nn.Conv2d(128, 256, 3, 1, 1),#output 256\*16\*16

nn.BatchNorm2d(256),

nn.ReLU(),

nn.MaxPool2d(4, 4, 0),#output 256\*4\*4

)

self.fc\_layers = nn.Sequential(

nn.Linear(256 \* 4 \* 4, 256),

nn.ReLU(),

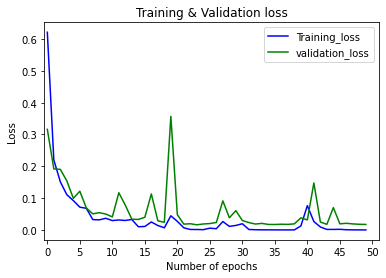
nn.Linear(256, 256),

nn.ReLU(),

nn.Linear(256, 6)

)

1. 訓練資料：kaggle train all
2. 測試資料：kaggle test all
3. 模型訓練時間：2111.1394336223602 seconds
4. 模型存檔：FreshnessDetector-64\_64\_3\_v3.pt
5. 結果：



Train data

The model is 100.0% accurate

report:

precision recall f1-score support

freshapples 1.0000 1.0000 1.0000 1373

freshbanana 1.0000 1.0000 1.0000 1257

freshoranges 1.0000 1.0000 1.0000 1159

rottenapples 1.0000 1.0000 1.0000 1880

rottenbanana 1.0000 1.0000 1.0000 1778

rottenoranges 1.0000 1.0000 1.0000 1273

accuracy 1.0000 8720

macro avg 1.0000 1.0000 1.0000 8720

weighted avg 1.0000 1.0000 1.0000 8720

confusion matrix:

[[1373 0 0 0 0 0]

[ 0 1257 0 0 0 0]

[ 0 0 1159 0 0 0]

[ 0 0 0 1880 0 0]

[ 0 0 0 0 1778 0]

[ 0 0 0 0 0 1273]]

Test data

The model is 99.51816160118607% accurate

report:

precision recall f1-score support

freshapples 0.9949 0.9975 0.9962 395

freshbanana 0.9974 0.9974 0.9974 381

freshoranges 0.9948 0.9871 0.9909 388

rottenapples 0.9983 0.9933 0.9958 601

rottenbanana 0.9962 1.0000 0.9981 530

rottenoranges 0.9877 0.9950 0.9913 403

accuracy 0.9952 2698

macro avg 0.9949 0.9951 0.9950 2698

weighted avg 0.9952 0.9952 0.9952 2698

confusion matrix:

[[394 1 0 0 0 0]

[ 0 380 0 0 1 0]

[ 0 0 383 0 0 5]

[ 2 0 1 597 1 0]

[ 0 0 0 0 530 0]

[ 0 0 1 1 0 401]]

--------------------------------------------------------

### 自製dataset CNN 64

1. 影像大小：64\*64\*3
2. 使用模型：FreshnessDetector-64\_64\_3\_v3.pt
3. 測試資料：google test randomly selected (number of fresh apples/bananas/orange = 11/15/9; rotten = 8/10/10)
4. 結果：

Test Google data

The model is 85.48387096774194% accurate

report:

precision recall f1-score support

freshapples 1.0000 0.7273 0.8421 11

freshbanana 0.9286 0.8667 0.8966 15

freshoranges 0.8889 0.8889 0.8889 9

rottenapples 0.6250 0.7143 0.6667 7

rottenbanana 0.8333 1.0000 0.9091 10

rottenoranges 0.8182 0.9000 0.8571 10

accuracy 0.8548 62

macro avg 0.8490 0.8495 0.8434 62

weighted avg 0.8680 0.8548 0.8555 62

confusion matrix:

[[ 8 0 0 3 0 0]

[ 0 13 0 0 1 1]

[ 0 0 8 0 0 1]

[ 0 1 0 5 1 0]

[ 0 0 0 0 10 0]

[ 0 0 1 0 0 9]]

Wrong Prediction:

Actual: 0

Prediction: tensor([[ 5.1963, -4.3327, -4.4005, 5.7218, 0.8252, -5.4761]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 0

Prediction: tensor([[ 6.5216e-03, -5.6723e+00, -2.7116e+00, 6.7267e+00, 2.4903e+00,

-1.9817e+00]], device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 0

Prediction: tensor([[ 3.6262, 3.3291, -6.4075, 4.8143, 4.3398, -13.7616]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 1

Prediction: tensor([[ -9.5483, 0.9548, -12.0914, -9.2488, -1.0836, 14.9709]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 1

Prediction: tensor([[-23.6222, 5.6954, 6.7946, -5.8096, 10.4715, 0.6019]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 2

Prediction: tensor([[-13.4634, -1.5082, 9.6249, -15.6473, -7.3164, 12.5432]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 3

Prediction: tensor([[-17.7315, -1.1014, -11.0031, 9.0969, 17.2697, -1.5451]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 3

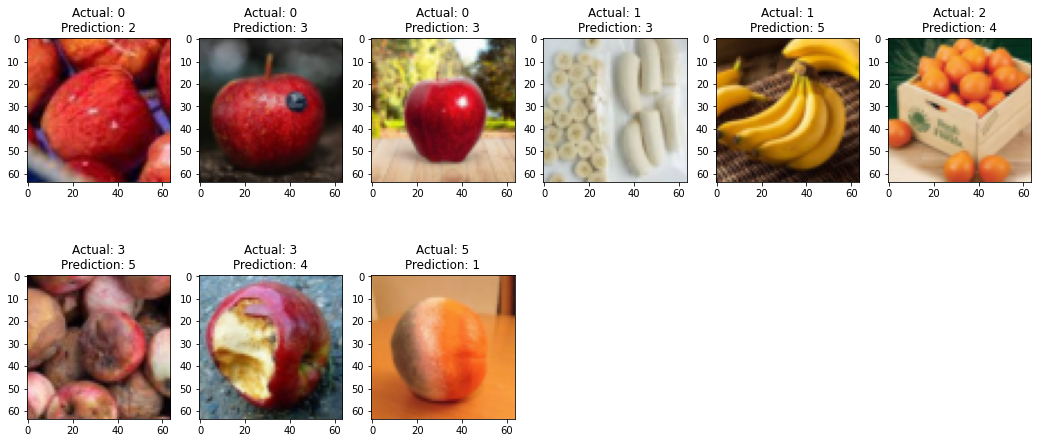
Prediction: tensor([[ 0.5761, 7.3652, -8.8401, 2.5531, 5.4042, -13.6454]],

device='cuda:0', grad\_fn=<AddmmBackward>)

Actual: 5

Prediction: tensor([[-8.5907, -5.6297, 10.9613, -4.3626, -9.3298, 8.0646]],

device='cuda:0', grad\_fn=<AddmmBackward>)



--------------------------------------------------------

### 自製dataset SVM 64 C=500

1. 影像大小：64\*64\*3
2. 使用模型：model\_svm\_64\_C500.pkl
3. 測試資料：google test randomly selected (number of fresh apples/bananas/orange = 11/15/9; rotten = 8/10/10)
4. 結果：

Test Google data

The model is 46.774193548387096% accurate

report:

precision recall f1-score support

freshapples 0.5000 0.3636 0.4211 11

freshbanana 0.8000 0.5333 0.6400 15

freshoranges 0.5000 0.5556 0.5263 9

rottenapples 0.3333 0.4286 0.3750 7

rottenbanana 0.4118 0.7000 0.5185 10

rottenoranges 0.2500 0.2000 0.2222 10

accuracy 0.4677 62

macro avg 0.4658 0.4635 0.4505 62

weighted avg 0.4992 0.4677 0.4678 62

confusion matrix:

[[4 0 3 2 2 0]

[1 8 1 0 3 2]

[1 0 5 0 2 1]

[0 0 0 3 2 2]

[0 1 0 1 7 1]

[2 1 1 3 1 2]]

--------------------------------------------------------

### 自製dataset SVM 64 C=100

1. 影像大小：64\*64\*3
2. 使用模型：model\_svm\_64.pkl
3. 測試資料：google test randomly selected (number of fresh apples/bananas/orange = 11/15/9; rotten = 8/10/10)
4. 結果：

Test Google data

The model is 48.38709677419355% accurate

report:

precision recall f1-score support

freshapples 0.6667 0.3636 0.4706 11

freshbanana 0.7778 0.4667 0.5833 15

freshoranges 0.5000 0.6667 0.5714 9

rottenapples 0.3000 0.4286 0.3529 7

rottenbanana 0.4667 0.7000 0.5600 10

rottenoranges 0.3000 0.3000 0.3000 10

accuracy 0.4839 62

macro avg 0.5019 0.4876 0.4730 62

weighted avg 0.5366 0.4839 0.4861 62

confusion matrix:

[[4 0 3 2 2 0]

[1 7 1 1 2 3]

[0 0 6 0 2 1]

[0 0 0 3 2 2]

[0 1 0 1 7 1]

[1 1 2 3 0 3]]

--------------------------------------------------------

### 自製dataset SVM 64 C=10

1. 影像大小：64\*64\*3
2. 使用模型：model\_svm\_64\_C10.pkl
3. 測試資料：google test randomly selected (number of fresh apples/bananas/orange = 11/15/9; rotten = 8/10/10)
4. 結果：

Test Google data

The model is 56.451612903225815% accurate

report:

precision recall f1-score support

freshapples 0.8571 0.5455 0.6667 11

freshbanana 0.8000 0.5333 0.6400 15

freshoranges 0.5556 0.5556 0.5556 9

rottenapples 0.4286 0.4286 0.4286 7

rottenbanana 0.5000 0.8000 0.6154 10

rottenoranges 0.3846 0.5000 0.4348 10

accuracy 0.5645 62

macro avg 0.5876 0.5605 0.5568 62

weighted avg 0.6173 0.5645 0.5715 62

confusion matrix:

[[6 0 3 0 1 1]

[1 8 0 0 3 3]

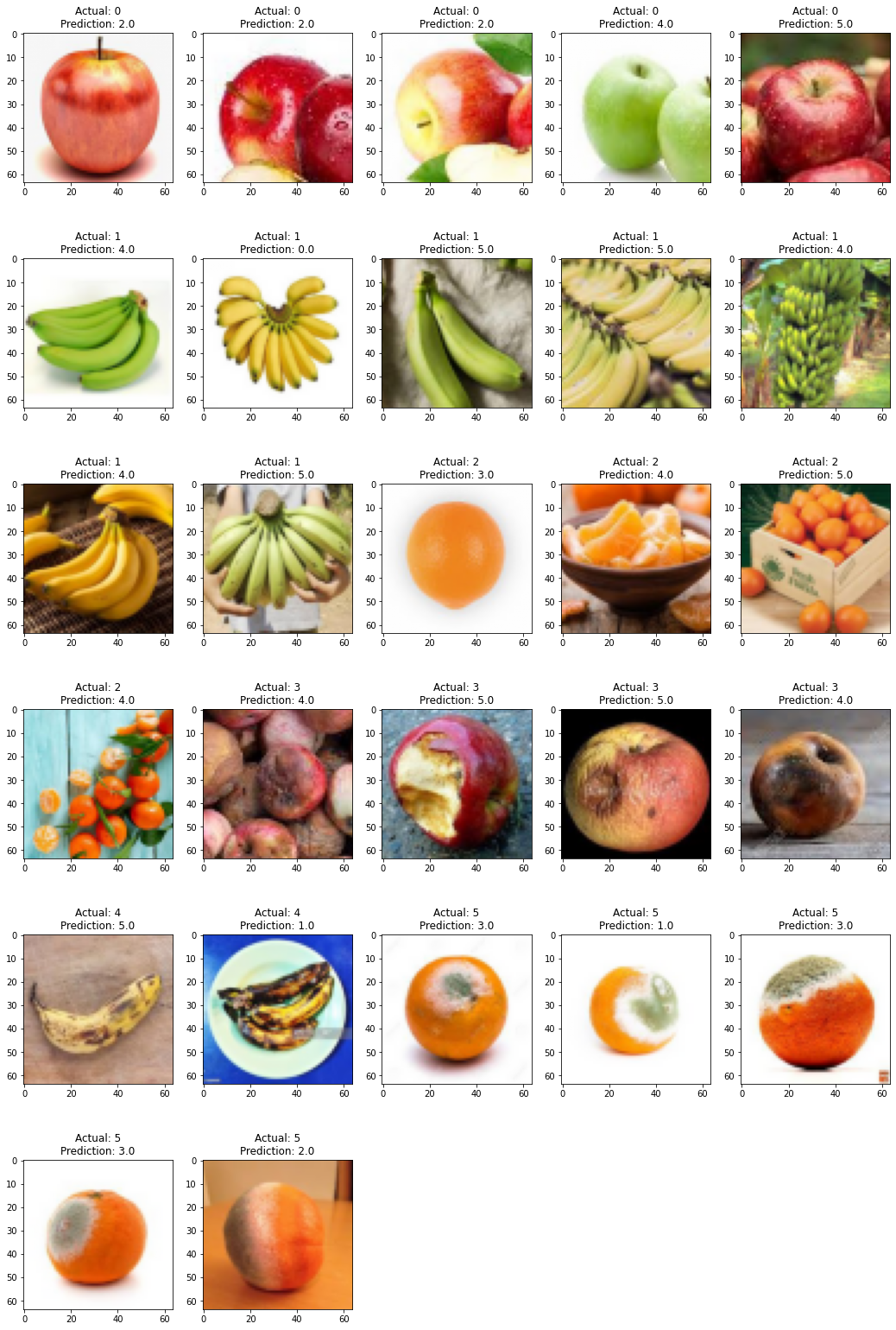
[0 0 5 1 2 1]

[0 0 0 3 2 2]

[0 1 0 0 8 1]

[0 1 1 3 0 5]]

Wrong Prediction:



--------------------------------------------------------

### 自製dataset SVM 64 C=1000

1. 影像大小：64\*64\*3
2. 使用模型：model\_svm\_64\_C1000.pkl
3. 測試資料：google test randomly selected (number of fresh apples/bananas/orange = 11/15/9; rotten = 8/10/10)
4. 結果：

Test Google data

The model is 48.38709677419355% accurate

report:

precision recall f1-score support

freshapples 0.5556 0.4545 0.5000 11

freshbanana 0.8000 0.5333 0.6400 15

freshoranges 0.5556 0.5556 0.5556 9

rottenapples 0.3333 0.4286 0.3750 7

rottenbanana 0.4118 0.7000 0.5185 10

rottenoranges 0.2500 0.2000 0.2222 10

accuracy 0.4839 62

macro avg 0.4844 0.4787 0.4685 62

weighted avg 0.5171 0.4839 0.4860 62

confusion matrix:

[[5 0 2 2 2 0]

[1 8 1 0 3 2]

[1 0 5 0 2 1]

[0 0 0 3 2 2]

[0 1 0 1 7 1]

[2 1 1 3 1 2]]

--------------------------------------------------------

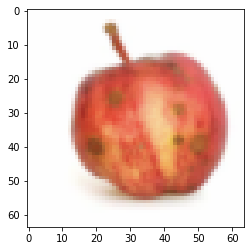
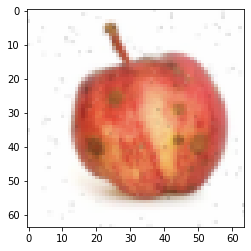
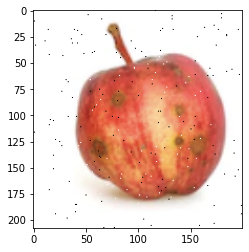
### 去掉椒鹽

用kaggle資料集

下面依序是

原圖 直接resize

去掉椒鹽 再resize



1. 用model\_svm\_64\_C500.pkl測試

把test data裡面有椒鹽的圖片(306張)用median filter過濾

\*\*報告可以比較accuracy和混淆矩陣

\*\*\*應該不能直接說多對一個，要用兩個的predict結果，再比較

有去椒鹽

Test data

The model is 94.44444444444444% accurate

report:

precision recall f1-score support

freshapples 0.8980 0.9778 0.9362 45

freshbanana 1.0000 0.9474 0.9730 38

freshoranges 1.0000 0.9333 0.9655 45

rottenapples 0.8767 0.9552 0.9143 67

rottenbanana 1.0000 0.9841 0.9920 63

rottenoranges 0.9318 0.8542 0.8913 48

accuracy 0.9444 306

macro avg 0.9511 0.9420 0.9454 306

weighted avg 0.9473 0.9444 0.9447 306

confusion matrix:

[[44 0 0 1 0 0]

[ 1 36 0 0 0 1]

[ 0 0 42 2 0 1]

[ 2 0 0 64 0 1]

[ 0 0 0 1 62 0]

[ 2 0 0 5 0 41]]

沒有去椒鹽

Test data

The model is 94.11764705882352% accurate

report:

precision recall f1-score support

freshapples 0.9167 0.9778 0.9462 45

freshbanana 1.0000 0.9474 0.9730 38

freshoranges 1.0000 0.9333 0.9655 45

rottenapples 0.8750 0.9403 0.9065 67

rottenbanana 1.0000 0.9841 0.9920 63

rottenoranges 0.8913 0.8542 0.8723 48

accuracy 0.9412 306

macro avg 0.9472 0.9395 0.9426 306

weighted avg 0.9433 0.9412 0.9415 306

confusion matrix:

[[44 0 0 1 0 0]

[ 1 36 0 0 0 1]

[ 0 0 42 2 0 1]

[ 1 0 0 63 0 3]

[ 0 0 0 1 62 0]

[ 2 0 0 5 0 41]]

actual: 3

no sp: 3.0

original predict: 5.0

actual: 3

no sp: 0.0

original predict: 5.0

2. 用C=500重新訓練SVM，train data和test data有椒鹽的先去掉椒鹽

1. 影像大小：64\*64\*3
2. 模型：model = SVC(C=500, kernel='rbf')
3. 訓練資料：kaggle train all 有椒鹽的去掉椒鹽

freshapples salt and pepper: 187

freshbanana salt and pepper: 180

freshoranges salt and pepper: 161

rottenapples salt and pepper: 260

rottenbanana salt and pepper: 243

rottenoranges salt and pepper: 174

1. 測試資料：kaggle test all 有椒鹽的去掉椒鹽
2. 模型訓練時間：62.12141513824463 seconds
3. 模型存檔：model\_svm\_64\_sp.pkl
4. 結果：

Train data

The model is 99.87157141546648% accurate

report:

precision recall f1-score support

freshapples 0.9988 0.9988 0.9988 1693

freshbanana 1.0000 1.0000 1.0000 1581

freshoranges 0.9993 0.9980 0.9986 1466

rottenapples 0.9957 0.9991 0.9974 2342

rottenbanana 1.0000 1.0000 1.0000 2224

rottenoranges 0.9994 0.9956 0.9975 1595

accuracy 0.9987 10901

macro avg 0.9989 0.9986 0.9987 10901

weighted avg 0.9987 0.9987 0.9987 10901

confusion matrix:

[[1691 0 0 2 0 0]

[ 0 1581 0 0 0 0]

[ 1 0 1463 2 0 0]

[ 1 0 0 2340 0 1]

[ 0 0 0 0 2224 0]

[ 0 0 1 6 0 1588]]

Test data

The model is 89.73313565604151% accurate

report:

precision recall f1-score support

freshapples 0.8526 0.8785 0.8653 395

freshbanana 0.9373 0.9423 0.9398 381

freshoranges 0.9079 0.8892 0.8984 388

rottenapples 0.8652 0.8968 0.8807 601

rottenbanana 0.9729 0.9472 0.9598 530

rottenoranges 0.8458 0.8164 0.8308 403

accuracy 0.8973 2698

macro avg 0.8969 0.8950 0.8958 2698

weighted avg 0.8979 0.8973 0.8974 2698

confusion matrix:

[[347 6 4 29 0 9]

[ 9 359 1 2 3 7]

[ 6 5 345 15 0 17]

[ 30 1 12 539 2 17]

[ 4 6 0 8 502 10]

[ 11 6 18 30 9 329]]

--------------------------------------------------------

### resize後的照片

