

# MSEN660 Final Project

Akshay Rao

Dec 2018

## 1 Part A

- We use the intermediate layers to featurize images.
- We create six pairwise classifiers, each trained on featurized image training sets for each of the 5 intermediate layers.
- Here is the convolution layer used and the cross-validated error estimate for each of the six pairwise two-label classifiers.

---

```

Pairwise Classifier 1 - spheroidite vs network
Layer 1 10 Fold CV Score: 0.395
Layer 2 10 Fold CV Score: 0.39
Layer 3 10 Fold CV Score: 0.395
Layer 4 10 Fold CV Score: 0.4000000000000001
Layer 5 10 Fold CV Score: 0.9650000000000001
Pairwise Classifier 2 - spheroidite vs pearlite
Layer 1 10 Fold CV Score: 0.42000000000000004
Layer 2 10 Fold CV Score: 0.42000000000000004
Layer 3 10 Fold CV Score: 0.42000000000000004
Layer 4 10 Fold CV Score: 0.425
Layer 5 10 Fold CV Score: 0.9550000000000001
Pairwise Classifier 3 - spheroidite vs spheroidite+widmanstatten
Layer 1 10 Fold CV Score: 0.625
Layer 2 10 Fold CV Score: 0.625
Layer 3 10 Fold CV Score: 0.625
Layer 4 10 Fold CV Score: 0.625
Layer 5 10 Fold CV Score: 0.74375
Pairwise Classifier 4 - network vs pearlite
Layer 1 10 Fold CV Score: 0.43500000000000005
Layer 2 10 Fold CV Score: 0.4450000000000001
Layer 3 10 Fold CV Score: 0.43000000000000005
Layer 4 10 Fold CV Score: 0.44000000000000006
Layer 5 10 Fold CV Score: 0.975
Pairwise Classifier 5 - network vs spheroidite+widmanstatten
Layer 1 10 Fold CV Score: 0.625
Layer 2 10 Fold CV Score: 0.625
Layer 3 10 Fold CV Score: 0.625
Layer 4 10 Fold CV Score: 0.625
Layer 5 10 Fold CV Score: 0.9875
Pairwise Classifier 6 - pearlite vs spheroidite+widmanstatten
Layer 1 10 Fold CV Score: 0.625
Layer 2 10 Fold CV Score: 0.625
Layer 3 10 Fold CV Score: 0.625
Layer 4 10 Fold CV Score: 0.625
Layer 5 10 Fold CV Score: 0.8875

```

Figure 1: Cross Validated Error Estimates for Each Pairwise Classifier for Each of the 5 Layers

- We observe that the last layer provides the lowest error for all of the pairwise classifiers. In fact we see that the error for a given pair wise classifier decreases as we use intermediate layers from deeper in the VGG 16 neural network.
- This could be partially because as we go deeper into the network, the features are more refined. Parts of the image like lines, edges, points etc are more apparent.

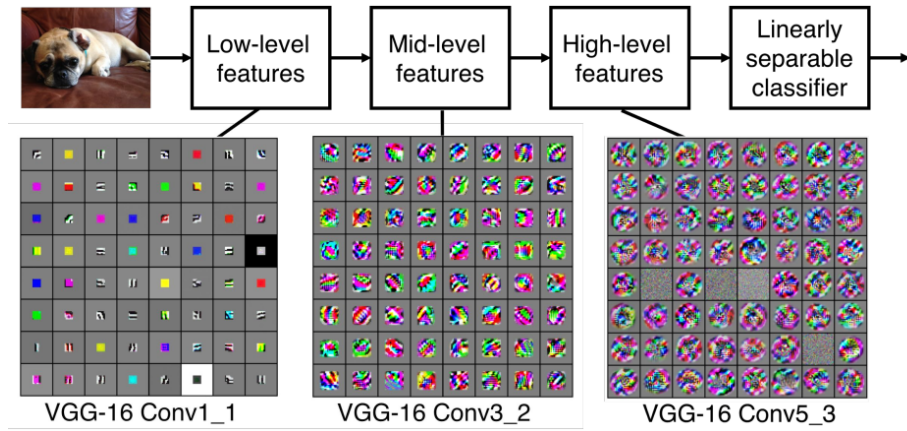


Figure 2: Illustration of increasing complexity of features for deeper layers-  
'From Stanford Course CS 231n'

## 2 Part B

- Here we have the separate test error rates on the unused micrographs of each of the four categories, for the pairwise two-label classifiers and the multilabel one-vs-one voting classifier.
- For the pairwise classifiers we use the test micrographs with one of the two labels used to train the classifier.
- For the multilabel classifier, we use the test micrographs with one of the four labels in the training data.

```

Pair wise Classifier spheroidite vs network for spheroidite :
Score 0.9927007299270073
Pair wise Classifier spheroidite vs pearlite for spheroidite :
Score 1.0
Pair wise Classifier spheroidite vs spheroidite+widmanstatten for spheroidite :
Score 0.9671532846715328
Pair wise Classifiers Average score for spheroidite :
0.98661800486618
Pair wise Classifier spheroidite vs network for network :
Score 0.9017857142857143
Pair wise Classifier network vs pearlite for network :
Score 0.9107142857142857
Pair wise Classifier network vs spheroidite+widmanstatten for network :
Score 0.9285714285714286
Pair wise Classifiers Average score for network :
0.9136904761904763
Pair wise Classifier spheroidite vs pearlite for pearlite :
Score 1.0
Pair wise Classifier network vs pearlite for pearlite :
Score 1.0
Pair wise Classifier pearlite vs spheroidite+widmanstatten for pearlite :
Score 1.0
Pair wise Classifiers Average score for pearlite :
1.0
Pair wise Classifier spheroidite vs spheroidite+widmanstatten for spheroidite+widmanstatten :
Score 0.5714285714285714
Pair wise Classifier network vs spheroidite+widmanstatten for spheroidite+widmanstatten :
Score 1.0
Pair wise Classifier pearlite vs spheroidite+widmanstatten for spheroidite+widmanstatten :
Score 0.8095238095238095
Pair wise Classifiers Average score for spheroidite+widmanstatten :
0.7936507936507936

```

Figure 3: Test Error Rates for the Pairwise Two Label Classifiers

```

For Micrograph Type spheroidite
OnevsOne has Test score: 0.9679144385026738
For Micrograph Type network
OnevsOne has Test score: 0.9481132075471698
For Micrograph Type pearlite
OnevsOne has Test score: 1.0
For Micrograph Type spheroidite+widmanstatten
OnevsOne has Test score: 0.8765432098765432

```

Figure 4: Test Error Rates for the Multilabel OnevsOne Classifier

### 3 Part C

- For the mixed pearlite + spheroidite and pearlite + widmanstatten micrographs, which were not used in training, we apply the trained one-vs-one multilabel voting classifier.

Micrograph Type: pearlite+spheroidite  
Micrograph Image 0  
One vs One Predicted: spheroidite  
Micrograph Image 1  
One vs One Predicted: spheroidite  
Micrograph Image 2  
One vs One Predicted: spheroidite  
Micrograph Image 3  
One vs One Predicted: pearlite  
Micrograph Image 4  
One vs One Predicted: spheroidite  
Micrograph Image 5  
One vs One Predicted: spheroidite  
Micrograph Image 6  
One vs One Predicted: spheroidite  
Micrograph Image 7  
One vs One Predicted: spheroidite  
Micrograph Image 8  
One vs One Predicted: spheroidite  
Micrograph Image 9  
One vs One Predicted: pearlite  
Micrograph Image 10  
One vs One Predicted: spheroidite  
Micrograph Image 11  
One vs One Predicted: spheroidite  
Micrograph Image 12  
One vs One Predicted: spheroidite  
Micrograph Image 13  
One vs One Predicted: spheroidite  
Micrograph Image 14  
One vs One Predicted: spheroidite  
Micrograph Image 15  
One vs One Predicted: spheroidite  
Micrograph Image 16  
One vs One Predicted: spheroidite

Figure 5: Snippet of Predictions for mixed pearlite + spheroidite

```

Micrograph Type: pearlite+widmanstatten
Micrograph Image 0
One vs One Predicted: pearlite
Micrograph Image 1
One vs One Predicted: pearlite
Micrograph Image 2
One vs One Predicted: pearlite
Micrograph Image 3
One vs One Predicted: spheroidite
Micrograph Image 4
One vs One Predicted: spheroidite+widmanstatten
Micrograph Image 5
One vs One Predicted: pearlite
Micrograph Image 6
One vs One Predicted: spheroidite
Micrograph Image 7
One vs One Predicted: pearlite
Micrograph Image 8
One vs One Predicted: spheroidite
Micrograph Image 9
One vs One Predicted: spheroidite
Micrograph Image 10
One vs One Predicted: pearlite
Micrograph Image 11
One vs One Predicted: pearlite
Micrograph Image 12
One vs One Predicted: pearlite
Micrograph Image 13
One vs One Predicted: spheroidite
Micrograph Image 14
One vs One Predicted: spheroidite
Micrograph Image 15
One vs One Predicted: pearlite
...

```

Figure 6: Snippet of Predictions for mixed pearlite + widmanstatten

## 4 Part D

- Now we apply the pairwise classifier pearlite vs. spheroidite to the mixed pearlite + spheroidite
- We can see that in almost all cases the two classifiers predict the same outcome for the micrograph images.

```

pearlite+spheroidite
Micrograph Image 0
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 1
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 2
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 3
Pairwise Classifier Predicted: pearlite
OnevsOne Classifier Predicted: pearlite
Micrograph Image 4
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 5
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 6
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 7
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 8
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite
Micrograph Image 9
Pairwise Classifier Predicted: pearlite
OnevsOne Classifier Predicted: pearlite
Micrograph Image 10
Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite

```

Figure 7: OnevsOne compared with Pairwise Classifier for Predicting the mixed pearlite + spheroidite micrographs.

## 5 Part E

- For the untrained martensite microstructure we apply the trained one-vs-one multilabel voting classifier.

```

Micrograph Image 0 Predicted: spheroidite
Micrograph Image 1 Predicted: network
Micrograph Image 2 Predicted: pearlite
Micrograph Image 3 Predicted: spheroidite
Micrograph Image 4 Predicted: spheroidite
Micrograph Image 5 Predicted: network
Micrograph Image 6 Predicted: spheroidite
Micrograph Image 7 Predicted: pearlite
Micrograph Image 8 Predicted: spheroidite
Micrograph Image 9 Predicted: spheroidite
Micrograph Image 10 Predicted: pearlite
Micrograph Image 11 Predicted: pearlite
Micrograph Image 12 Predicted: pearlite
Micrograph Image 13 Predicted: pearlite
Micrograph Image 14 Predicted: spheroidite
Micrograph Image 15 Predicted: pearlite
Micrograph Image 16 Predicted: spheroidite
Micrograph Image 17 Predicted: pearlite
Micrograph Image 18 Predicted: spheroidite
Micrograph Image 19 Predicted: pearlite
Micrograph Image 20 Predicted: pearlite
Micrograph Image 21 Predicted: spheroidite
Micrograph Image 22 Predicted: spheroidite
Micrograph Image 23 Predicted: pearlite
Micrograph Image 24 Predicted: spheroidite
Micrograph Image 25 Predicted: pearlite
Micrograph Image 26 Predicted: pearlite
Micrograph Image 27 Predicted: pearlite
Micrograph Image 28 Predicted: spheroidite
Micrograph Image 29 Predicted: spheroidite
Micrograph Image 30 Predicted: pearlite
Micrograph Image 31 Predicted: network
Micrograph Image 32 Predicted: spheroidite
Micrograph Image 33 Predicted: spheroidite
Micrograph Image 34 Predicted: pearlite
Micrograph Image 35 Predicted: spheroidite

```

Figure 8: Predictions for multilabel OnevsOne Classifier on Martensite

## 6 Part C Full Predictions

Micrograph Type: pearlite+spheroidite Micrograph Image 0 One vs One Predicted: spheroidite Micrograph Image 1 One vs One Predicted: spheroidite Micrograph Image 2 One vs One Predicted: spheroidite Micrograph Image 3 One vs One Predicted: pearlite Micrograph Image 4 One vs One Predicted: spheroidite Micrograph Image 5 One vs One Predicted: spheroidite Micrograph Image 6 One vs One Predicted: spheroidite Micrograph Image 7 One vs One Predicted: spheroidite Micrograph Image 8 One vs One Predicted: spheroidite Micrograph Image 9 One vs One Predicted: pearlite Micrograph Image 10 One vs One Predicted: spheroidite Micrograph Image 11 One vs One Predicted: spheroidite Micrograph Image 12 One vs One Predicted: spheroidite Micrograph Image 13 One vs One Predicted: spheroidite Micrograph Image 14 One vs One Predicted: spheroidite Micrograph Image 15 One vs One Predicted: spheroidite Micrograph Image 16 One vs One Predicted: spheroidite Micrograph Image 17 One vs One Predicted: spheroidite Micrograph Image 18 One vs One Predicted: spheroidite Micrograph Image 19 One vs One Predicted:





spheroidite Micrograph Image 83 One vs One Predicted: spheroidite Micrograph Image 84 One vs One Predicted: spheroidite Micrograph Image 85 One vs One Predicted: spheroidite Micrograph Image 86 One vs One Predicted: spheroidite Micrograph Image 87 One vs One Predicted: pearlite Micrograph Image 88 One vs One Predicted: pearlite Micrograph Image 89 One vs One Predicted: spheroidite Micrograph Image 90 One vs One Predicted: spheroidite Micrograph Image 91 One vs One Predicted: pearlite Micrograph Image 92 One vs One Predicted: spheroidite Micrograph Image 93 One vs One Predicted: pearlite Micrograph Image 94 One vs One Predicted: spheroidite Micrograph Image 95 One vs One Predicted: spheroidite Micrograph Image 96 One vs One Predicted: pearlite Micrograph Image 97 One vs One Predicted: spheroidite Micrograph Image 98 One vs One Predicted: spheroidite Micrograph Image 99 One vs One Predicted: pearlite Micrograph Image 100 One vs One Predicted: spheroidite Micrograph Image 101 One vs One Predicted: spheroidite Micrograph Image 102 One vs One Predicted: spheroidite Micrograph Image 103 One vs One Predicted: pearlite Micrograph Image 104 One vs One Predicted: spheroidite Micrograph Image 105 One vs One Predicted: spheroidite Micrograph Image 106 One vs One Predicted: network Micrograph Type: pearlite+widmanstatten Micrograph Image 0 One vs One Predicted: pearlite Micrograph Image 1 One vs One Predicted: pearlite Micrograph Image 2 One vs One Predicted: pearlite Micrograph Image 3 One vs One Predicted: spheroidite Micrograph Image 4 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 5 One vs One Predicted: pearlite Micrograph Image 6 One vs One Predicted: spheroidite Micrograph Image 7 One vs One Predicted: pearlite Micrograph Image 8 One vs One Predicted: spheroidite Micrograph Image 9 One vs One Predicted: spheroidite Micrograph Image 10 One vs One Predicted: pearlite Micrograph Image 11 One vs One Predicted: pearlite Micrograph Image 12 One vs One Predicted: pearlite Micrograph Image 13 One vs One Predicted: spheroidite Micrograph Image 14 One vs One Predicted: spheroidite Micrograph Image 15 One vs One Predicted: pearlite Micrograph Image 16 One vs One Predicted: pearlite Micrograph Image 17 One vs One Predicted: pearlite Micrograph Image 18 One vs One Predicted: spheroidite Micrograph Image 19 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 20 One vs One Predicted: pearlite Micrograph Image 21 One vs One Predicted: pearlite Micrograph Image 22 One vs One Predicted: pearlite Micrograph Image 23 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 24 One vs One Predicted: pearlite Micrograph Image 25 One vs One Predicted: spheroidite Micrograph Image 26 One vs One Predicted: pearlite

## 7 Part D Pearlite Spheroidite Full Predictions

pearlite+spheroidite Micrograph Image 0 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 1 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 2 Pairwise Classifier Predicted: spheroidite OnevsOne Classi-







fier Predicted: spheroidite Micrograph Image 106 Pairwise Classifier Predicted:  
spheroidite OnevsOne Classifier Predicted: network

```
[11pt]article
[T1]fontenc mathpazo
graphicx caption nolabel labelformat=nolabel
adjustbox xcolor enumerate geometry amsmath amssymb textcomp upquote
eurosym [mathletters]ucs [utf8x]inputenc fancyvrb grffile hyperref longtable book-
tabs [inline]enumitem [normalem]ulem mathrsfs
HighlightingVerbatimcommandchars=
```

```
{ }
HW4_trial5
breaklinks=true, colorlinks=true, urlcolor=urlcolor, linkcolor=linkcolor,
citecolor=citecolor,
verbose,tmargin=1in,bmargin=1in,lmargin=1in,rmargin=1in
[commandchars=
```

```
{ } In [3]: from keras.preprocessing import image from PIL
import Image from keras.applications.vgg16 import VGG16 from
keras.applications.vgg16 import preprocess_input from keras.models im-
port Model
```

```
import numpy as np
[commandchars=
{ } In [4]: import pandas as pd df=pd.read_csv("/home/akshay/Documents/TAMU/Fall
18/660/MSEN660/Programming/Project 4/micrograph.csv")
```

```
[commandchars=
{ } In [5]: df.head() cols=df.columns print(cols) print(cols)
print('size',df.size,'length',len(df)) df[cols[1]][0]
```

```
[commandchars=
{ } Index(['micrograph_id', 'path', 'micron_bar', 'micron_bar_units', 'mi-
cron_bar_px', 'magnification', 'detector', 'sample_key', 'contributor_key', 'pri-
mary_microconstituent'], dtype=object) Index(['micrograph_id', 'path', 'mi-
cron_bar', 'micron_bar_units', 'micron_bar_px', 'magnification', 'detector', 'sam-
ple_key', 'contributor_key', 'primary_microconstituent'], dtype=object) size
9610 length 961
```

```
[commandchars=
{ } Out[5]: 'micrograph1.tif'
```

```
[commandchars=
{ } In [7]: base_model=VGG16(weights='imagenet',include_top=False)
model=Model(input=base_model.input,output=[base_model.get_layer('block1_pool').output,base_model.get_layer('block5_pool').output])
feat=[] for i in range(0,len(df[cols[1]])): feat.append([])
```

```
#try extracting features from a single image for im-
age i in range(0,len(df[cols[1]])): img_path=df[cols[1]][image_i]
img=image.load_img(img_path) x=image.img_to_array(img) x=x[0:484,:,:]
#crop the bottom subtitles x=np.expand_dims(x,axis=0) x=preprocess_input(x)
print('image',image_i)
block_poolfeat=model.predict(x)
```

```

    #print('elements in feature list',len(block'poolfeat')) #for i in range(0,5):
    #print(block'poolfeat[i][0][0][0][0]) #print(block'poolfeat[i].shape) #find mean
    across channels for block in range(0,5): s=block'poolfeat[block].shape
    tot=0 for i in range(0,s[0]): for j in range(0,s[1]): for k in range(0,s[2]):
    tot+=block'poolfeat[block][i][j][k] feat[image'i].append(tot/(s[0]*s[1]*s[2]))
    #print(feat[block].shape,'block',block+1,'feature vector shape')
    [commandchars=
    {}] /home/akshay/.local/lib/python3.6/site-packages/ipykernel_launcher.py:2:
    UserWarning: Update your 'Model' call to the Keras 2 API:
    'Model(inputs=Tensor("in...", outputs=[tf.Tensor(...))'
    [commandchars=
    {}] image 0 image 1 image 2 image 3 image 4 image 5 image 6 image 7 image
    8 image 9 image 10 image 11 image 12 image 13 image 14 image 15 image 16
    image 17 image 18 image 19 image 20 image 21 image 22 image 23 image 24
    image 25 image 26 image 27 image 28 image 29 image 30 image 31 image 32
    image 33 image 34 image 35 image 36 image 37 image 38 image 39 image 40
    image 41 image 42 image 43 image 44 image 45 image 46 image 47 image 48
    image 49 image 50 image 51 image 52 image 53 image 54 image 55 image 56
    image 57 image 58 image 59 image 60 image 61 image 62 image 63 image 64
    image 65 image 66 image 67 image 68 image 69 image 70 image 71 image 72
    image 73 image 74 image 75 image 76 image 77 image 78 image 79 image 80
    image 81 image 82 image 83 image 84 image 85 image 86 image 87 image 88
    image 89 image 90 image 91 image 92 image 93 image 94 image 95 image 96
    image 97 image 98 image 99 image 100 image 101 image 102 image 103 image
    104 image 105 image 106 image 107 image 108 image 109 image 110 image 111
    image 112 image 113 image 114 image 115 image 116 image 117 image 118 image
    119 image 120 image 121 image 122 image 123 image 124 image 125 image 126
    image 127 image 128 image 129 image 130 image 131 image 132 image 133 image
    134 image 135 image 136 image 137 image 138 image 139 image 140 image 141
    image 142 image 143 image 144 image 145 image 146 image 147 image 148 image
    149 image 150 image 151 image 152 image 153 image 154 image 155 image 156
    image 157 image 158 image 159 image 160 image 161 image 162 image 163 image
    164 image 165 image 166 image 167 image 168 image 169 image 170 image 171
    image 172 image 173 image 174 image 175 image 176 image 177 image 178 image
    179 image 180 image 181 image 182 image 183 image 184 image 185 image 186
    image 187 image 188 image 189 image 190 image 191 image 192 image 193 image
    194 image 195 image 196 image 197 image 198 image 199 image 200 image 201
    image 202 image 203 image 204 image 205 image 206 image 207 image 208 image
    209 image 210 image 211 image 212 image 213 image 214 image 215 image 216
    image 217 image 218 image 219 image 220 image 221 image 222 image 223 image
    224 image 225 image 226 image 227 image 228 image 229 image 230 image 231
    image 232 image 233 image 234 image 235 image 236 image 237 image 238 image
    239 image 240 image 241 image 242 image 243 image 244 image 245 image 246
    image 247 image 248 image 249 image 250 image 251 image 252 image 253 image
    254 image 255 image 256 image 257 image 258 image 259 image 260 image 261
    image 262 image 263 image 264 image 265 image 266 image 267 image 268 image

```







```

959 image 960
[commandchars=
{}] In [8]: len(feats)
[commandchars=
{}] Out[8]: 961
[commandchars=
{}] In [73]: df[cols[9]][0]
[commandchars=
{}] Out[73]: 'pearlite'
[commandchars=
{}] In [17]: # Store the features #/media/akshay/OS/Users/aksha/OneDrive
- Texas A&M University/Documents/TAMU/Fall
18/660/MSEN660/Programming/Project 4/ feat`backup=feat import pickle
filename="feat.pickle" with open (filename,'wb') as f: pickle.dump(feats,f)
[commandchars=
{}] In [18]: with open(filename, 'rb') as f: feats = pickle.load(f) len(feats)
[commandchars=
{}] Out[18]: 961
[commandchars=
{}] In [240]: y=[] x`i=[] for i in range(0,len(df[cols[9]])):
if df[cols[9]][i]=='spheroidite': y.append(0) x`i.append(i)
elif df[cols[9]][i]=='network': y.append(1) x`i.append(i)
elif df[cols[9]][i]=='pearlite': y.append(2) x`i.append(i) elif
df[cols[9]][i]=='spheroidite+widmanstätten': y.append(3) x`i.append(i)
elif df[cols[9]][i]=='pearlite+spheroidite': y.append(4) x`i.append(i)
elif df[cols[9]][i]=='pearlite+widmanstätten': y.append(5) x`i.append(i)
elif df[cols[9]][i]=='martensite': y.append(6) x`i.append(i) else: pass
print(len(y),y[0:10],len(x`i),x`i[0:10])
[commandchars=
{}] 961 [2, 0, 4, 2, 0, 3, 1, 1, 0, 0] 961 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[commandchars=
{}] In []: #training data
[commandchars=
{}] In [260]: #pairwise combinations of labelled data from itertools
import combinations combs=list(combinations([0,1,2,3],2)) print(combs)
enc=0:'spheroidite',1:'network',2:'pearlite',3:'spheroidite+widmanstätten',4:'pearlite+spheroidite',5:'pearlite+
from sklearn.svm import SVC model1=SVC(gamma='auto',kernel='rbf')
layers=np.zeros(len(combs)) layer`score=np.zeros([len(combs),5])
layer`testscore=np.zeros([len(combs),5]) com`count=0 import pdb
test`score=np.zeros(len(combs)) models=[]
for com in combs:
#training set X`train=[] Y`train=[] X`test=[] Y`test=[] print (com) for lab-
data in com: if labdata==0: count=0 i=0
while(count<100): if y[i]==0 : X`train.append(feats[x`i[i]])
Y`train.append(0) count+=1 i+=1 while(i<761): if y[i]==0 :
X`test.append(feats[x`i[i]]) Y`test.append(0) i+=1 if labdata==1: count=0 i=0

```

```

while(count<100):
    if y[i]==1 :
        X'train.append(feats[x'i[i]])
Y'train.append(1) count+=1 i+=1
while(i<761):
    if y[i]==1 :
        X'test.append(feats[x'i[i]]) Y'test.append(1) i+=1
if labdata==2: count=0 i=0
while(count<100):
    if y[i]==2 :
        X'train.append(feats[x'i[i]])
Y'train.append(2) count+=1 i+=1
while(i<761):
    if y[i]==2 :
        X'test.append(feats[x'i[i]]) Y'test.append(2) i+=1
if labdata==3: count=0 i=0
while(count<60):
    if y[i]==3 : X'train.append(feats[x'i[i]]) Y'train.append(3)
count+=1 i+=1
while(i<761):
    if y[i]==3 : X'test.append(feats[x'i[i]])
Y'test.append(3) i+=1
#pdb.set_trace() #for layer in range(0,5):
model=SVC(gamma='auto',kernel='rbf')
#kfolds for each layer
from sklearn.model_selection import KFold
kf=KFold(n_splits=10,shuffle=True) avgscore=np.zeros(5)
testscore=np.zeros(5) #cross validation 10 fold
for train_index,test_index in kf.split(X'train):
    x'train=[] x'test=[] y'train=[] y'test=[]
    for i in train_index:
        x'train.append(X'train[i]) y'train.append(Y'train[i])
    for j in test_index:
        x'test.append(X'train[j]) y'test.append(Y'train[j])
    for layer in range(0,5):
        x'train1=[] x'test1=[]
        for i in range(0,len(x'train)):
            x'train1.append(x'train[i][layer])
        for i in range(0,len(x'test)):
            x'test1.append(x'test[i][layer])
        model.fit(x'train1,y'train)
        sco=model.score(x'test1,y'test)
        avgscore[layer]+=sco
#layer 5 performs best for all pairs of labels #first train the
6 pairwise models based on best layer
y'train=[]
for i in range(0,len(Y'train)):
    y'train.append(Y'train[i])
x'train=[]
for i in range(0,len(X'train)):
    x'train.append(X'train[i][4])
models.append(model.fit(x'train,y'train))
avgscore=avgscore/10
layer'score[com'count,:]=avgscore
#pdb.set_trace()
com'count+=1
[commandchars=
{}] [(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 3)] (0, 1) (0, 2) (0, 3) (1, 2) (1, 3) (2, 3)
[commandchars=
{}] In [259]: layer'score[0]
[commandchars=
{}] Out[259]: array([0.445, 0.445, 0.44 , 0.44 , 0.96 ])
[commandchars=
{}] In [265]:
for i in range(0,6):
    print ('Pairwise Classifier',i+1,'-','enc[combs[i][0]],'vs','enc[combs[i][1]]')
for j in range(0,5):
    print('Layer ',j+1,'10 Fold CV Score:',layer'score[i][j])
[commandchars=
{}] Pairwise Classifier 1 - spheroidite vs network Layer 1 10 Fold CV Score: 0.395
Layer 2 10 Fold CV Score: 0.39 Layer 3 10 Fold CV Score: 0.395 Layer 4 10 Fold CV Score: 0.4000000000000001
Layer 5 10 Fold CV Score: 0.9650000000000001
Pairwise Classifier 2 - spheroidite vs pearlite Layer 1 10 Fold CV Score: 0.42000000000000004
Layer 2 10 Fold CV Score: 0.42000000000000004 Layer 3 10 Fold CV Score: 0.42000000000000004
Layer 4 10 Fold CV Score: 0.425 Layer

```

5 10 Fold CV Score: 0.9550000000000001 Pairwise Classifier 3 - spheroidite vs spheroidite+widmanstatten Layer 1 10 Fold CV Score: 0.625 Layer 2 10 Fold CV Score: 0.625 Layer 3 10 Fold CV Score: 0.625 Layer 4 10 Fold CV Score: 0.625 Layer 5 10 Fold CV Score: 0.74375 Pairwise Classifier 4 - network vs pearlite Layer 1 10 Fold CV Score: 0.43500000000000005 Layer 2 10 Fold CV Score: 0.44500000000000001 Layer 3 10 Fold CV Score: 0.43000000000000005 Layer 4 10 Fold CV Score: 0.44000000000000006 Layer 5 10 Fold CV Score: 0.975 Pairwise Classifier 5 - network vs spheroidite+widmanstatten Layer 1 10 Fold CV Score: 0.625 Layer 2 10 Fold CV Score: 0.625 Layer 3 10 Fold CV Score: 0.625 Layer 4 10 Fold CV Score: 0.625 Layer 5 10 Fold CV Score: 0.9875 Pairwise Classifier 6 - pearlite vs spheroidite+widmanstatten Layer 1 10 Fold CV Score: 0.625 Layer 2 10 Fold CV Score: 0.625 Layer 3 10 Fold CV Score: 0.625 Layer 4 10 Fold CV Score: 0.625 Layer 5 10 Fold CV Score: 0.8875

```
[commandchars=
{}] In [258]: #now obtain test errors for pairwise classifiers #pairwise
combinations of labelled data test_err=np.zeros(4) from sklearn.svm im-
port SVC com_count=0 x_train1v1=[] x_test1v1=[] y_train1v1=[] y_test1v1=[]
enc=-0:'spheroidite',1:'network',2:'pearlite',3:'spheroidite+widmanstatten',4:'pearlite+spheroidite',5:'pearlite+
for label in range(4):
    #training set X_train=[] Y_train=[] X_test=[] Y_test=[]
    if label==0: count=0 i=0
    while(count<100):
        if y[i]==0 : X_train.append(feats[x_train1v1[i]])
        x_train1v1.append(feats[x_train1v1[i]]) y_train1v1.append(0) Y_train.append(0)
        count+=1 i+=1 while(i<len(y)):
            if y[i]==0 : X_test.append(feats[x_test1v1[i]])
            Y_test.append(0) x_test1v1.append(feats[x_test1v1[i]]) y_test1v1.append(0) i+=1
            if label==1: count=0 i=0
            while(count<100):
                if y[i]==1 : X_train.append(feats[x_train1v1[i]])
                x_train1v1.append(feats[x_train1v1[i]]) Y_train.append(1) y_train1v1.append(1)
                count+=1 i+=1 while(i<len(y)):
                    if y[i]==1 : X_test.append(feats[x_test1v1[i]])
                    Y_test.append(1) x_test1v1.append(feats[x_test1v1[i]]) y_test1v1.append(1) i+=1
                    if label==2: count=0 i=0
                    while(count<100):
                        if y[i]==2 : X_train.append(feats[x_train1v1[i]])
                        x_train1v1.append(feats[x_train1v1[i]]) Y_train.append(2) y_train1v1.append(2)
                        count+=1 i+=1 while(i<len(y)):
                            if y[i]==2 : X_test.append(feats[x_test1v1[i]])
                            Y_test.append(2) x_test1v1.append(feats[x_test1v1[i]]) y_test1v1.append(2) i+=1
                            if label==3: count=0 i=0
                            while(count<60):
                                if y[i]==3 : X_train.append(feats[x_train1v1[i]])
                                x_train1v1.append(feats[x_train1v1[i]]) Y_train.append(3) y_train1v1.append(3)
                                count+=1 i+=1 while(i<len(y)):
                                    if y[i]==3 : X_test.append(feats[x_test1v1[i]])
                                    Y_test.append(3) x_test1v1.append(feats[x_test1v1[i]]) y_test1v1.append(3) i+=1
                                    #pdb.set_trace() #for layer in range(0,5):
                                    y_test=[] for i in range(0,len(Y_test)):
                                        y_test.append(Y_test[i]) x_test=[] for i in range(0,len(X_test)):
                                            x_test.append(X_test[i][4])
                                model=SVC(gamma='auto',kernel='rbf') sco=[] count=0
                                for i in range(0,len(combs)):
                                    if label in combs[i]:
                                        sco.append(models[i].score(x_test,y_test)) pair1=combs[i][0] pair2=combs[i][1]
```

```

print ('Pair wise Classifier',enc[pair1], 'vs', enc[pair2], 'for', enc[label], ':')
print('Score', sco[count]) count+=1 #pdb.set_trace()
test_err[label]=((sco[0]+sco[1]+sco[2])/3) #pdb.set_trace()
com_count+=1
print ('Pair wise Classifiers Average score for', enc[label], ':')
print(test_err[label])
[commandchars=
{}] Pair wise Classifier spheroidite vs network for spheroidite : Score
0.9927007299270073 Pair wise Classifier spheroidite vs pearlite for spheroidite
: Score 1.0 Pair wise Classifier spheroidite vs spheroidite+widmanstatten
for spheroidite : Score 0.9671532846715328 Pair wise Classifiers Average
score for spheroidite : 0.98661800486618 Pair wise Classifier spheroidite
vs network for network : Score 0.9017857142857143 Pair wise Classi-
fier network vs pearlite for network : Score 0.9107142857142857 Pair
wise Classifier network vs spheroidite+widmanstatten for network : Score
0.9285714285714286 Pair wise Classifiers Average score for network :
0.9136904761904763 Pair wise Classifier spheroidite vs pearlite for pearlite
: Score 1.0 Pair wise Classifier network vs pearlite for pearlite :
Score 1.0 Pair wise Classifier pearlite vs spheroidite+widmanstatten for
pearlite : Score 1.0 Pair wise Classifiers Average score for pearlite
: 1.0 Pair wise Classifier spheroidite vs spheroidite+widmanstatten for
spheroidite+widmanstatten : Score 0.5714285714285714 Pair wise Classi-
fier network vs spheroidite+widmanstatten for spheroidite+widmanstatten
: Score 1.0 Pair wise Classifier pearlite vs spheroidite+widmanstatten for
spheroidite+widmanstatten : Score 0.8095238095238095 Pair wise Classifiers
Average score for spheroidite+widmanstatten : 0.7936507936507936
[commandchars=
{}] In [245]: len(y)
[commandchars=
{}] Out[245]: 961
[commandchars=
{}] In [267]: #one vs one classifier print(len(x_train1v1),len(y_train1v1),len(x_test1v1),len(y_test1v1))
from sklearn.multiclass import OneVsOneClass-
sifier x_train1v1p=[] x_test1v1p=[] for i in
range(0,len(x_train1v1)): x_train1v1p.append(x_train1v1[i][4]) for
i in range(0,len(x_test1v1)): x_test1v1p.append(x_test1v1[i][4])
ovc=OneVsOneClassifier(SVC(gamma='auto',kernel='rbf')) multil-
abel_testerr=ovc.fit(x_train1v1p,y_train1v1).score(x_test1v1p,y_test1v1)
print('One vs One Multilabel Classifier Score :', multilabel_testerr)
[commandchars=
{}] 360 360 431 431 One vs One Multilabel Classifier Score : 0.9280742459396751
[commandchars=
{}] In [276]: #One vs One for each Label
score=[] enc=-0:'spheroidite',1:'network',2:'pearlite',3:'spheroidite+widmanstatten',4:'pearlite+spheroidite'
x_test=[] y_test=[] for i in range(len(x)): if
y[i]=0: x_test.append(feats[i][4]) y_test.append(0)

```

```

score.append(ovc.fit(x'train1v1p,y'train1v1).score(x'test,y'test))
print('For Micrograph Type',enc[0]) print('OnevsOne has Test
score:',score[0]) x'test=[] y'test=[] for i in range(len(x'i)):
if y[i]==1: x'test.append(feats[x'i][4]) y'test.append(1)
score.append(ovc.fit(x'train1v1p,y'train1v1).score(x'test,y'test))
print('For Micrograph Type',enc[1]) print('OnevsOne has Test
score:',score[1]) x'test=[] y'test=[] for i in range(len(x'i)):
if y[i]==2: x'test.append(feats[x'i][4]) y'test.append(2)
score.append(ovc.fit(x'train1v1p,y'train1v1).score(x'test,y'test))
print('For Micrograph Type',enc[2]) print('OnevsOne has Test
score:',score[2]) x'test=[] y'test=[] for i in range(len(x'i)):
if y[i]==3: x'test.append(feats[x'i][4]) y'test.append(3)
score.append(ovc.fit(x'train1v1p,y'train1v1).score(x'test,y'test)) print('For
Micrograph Type',enc[3]) print('OnevsOne has Test score:',score[3])
[commandchars=
{}] For Micrograph Type spheroidite OnevsOne has Test score:
0.9679144385026738 For Micrograph Type network OnevsOne has Test
score: 0.9481132075471698 For Micrograph Type pearlite OnevsOne has Test
score: 1.0 For Micrograph Type spheroidite+widmanstatten OnevsOne has
Test score: 0.8765432098765432
[commandchars=
{}] In [268]: #apply multilabel classifier for pearlite and wid-
manstatten and pearlite and widmanstatten samples x'test=[] y'test=[]
enc=-0:'spheroidite',1:'network',2:'pearlite',3:'spheroidite+widmanstatten',4:'pearlite+spheroidite',5:'pearlite+
for i in range(len(x'i)): if y[i]==4: x'test.append(feats[x'i][4])
y'test.append(4) pred1=[] for i in range(len(x'test)):
pred1.append(ovc.fit(x'train1v1p,y'train1v1).predict(x'test)) print('Micrograph
Type:',enc[4]) for i in range(0,len(pred1[0])): print ('Micrograph Im-
age',i) print ('One vs One Predicted:',enc[pred1[0][i]]) x'test=[] y'test=[]
for i in range(len(x'i)): if y[i]==5: x'test.append(feats[x'i][4])
y'test.append(5) pred2=[] for i in range(len(x'test)):
pred2.append(ovc.fit(x'train1v1p,y'train1v1).predict(x'test)) print('Micrograph
Type:',enc[5]) for i in range(0,len(pred2[0])): print ('Micrograph Image',i)
print ('One vs One Predicted:',enc[pred2[0][i]])
[commandchars=
{}] Micrograph Type: pearlite+spheroidite Micrograph Image 0 One vs One
Predicted: spheroidite Micrograph Image 1 One vs One Predicted: spheroidite
Micrograph Image 2 One vs One Predicted: spheroidite Micrograph Image 3
One vs One Predicted: pearlite Micrograph Image 4 One vs One Predicted:
spheroidite Micrograph Image 5 One vs One Predicted: spheroidite Micrograph
Image 6 One vs One Predicted: spheroidite Micrograph Image 7 One vs One
Predicted: spheroidite Micrograph Image 8 One vs One Predicted: spheroidite
Micrograph Image 9 One vs One Predicted: pearlite Micrograph Image 10 One
vs One Predicted: spheroidite Micrograph Image 11 One vs One Predicted:
spheroidite Micrograph Image 12 One vs One Predicted: spheroidite Micro-
graph Image 13 One vs One Predicted: spheroidite Micrograph Image 14 One

```



Image 77 One vs One Predicted: spheroidite Micrograph Image 78 One vs One Predicted: pearlite Micrograph Image 79 One vs One Predicted: pearlite Micrograph Image 80 One vs One Predicted: spheroidite Micrograph Image 81 One vs One Predicted: pearlite Micrograph Image 82 One vs One Predicted: spheroidite Micrograph Image 83 One vs One Predicted: spheroidite Micrograph Image 84 One vs One Predicted: spheroidite Micrograph Image 85 One vs One Predicted: spheroidite Micrograph Image 86 One vs One Predicted: spheroidite Micrograph Image 87 One vs One Predicted: pearlite Micrograph Image 88 One vs One Predicted: pearlite Micrograph Image 89 One vs One Predicted: spheroidite Micrograph Image 90 One vs One Predicted: spheroidite Micrograph Image 91 One vs One Predicted: pearlite Micrograph Image 92 One vs One Predicted: spheroidite Micrograph Image 93 One vs One Predicted: pearlite Micrograph Image 94 One vs One Predicted: spheroidite Micrograph Image 95 One vs One Predicted: spheroidite Micrograph Image 96 One vs One Predicted: pearlite Micrograph Image 97 One vs One Predicted: spheroidite Micrograph Image 98 One vs One Predicted: spheroidite Micrograph Image 99 One vs One Predicted: pearlite Micrograph Image 100 One vs One Predicted: spheroidite Micrograph Image 101 One vs One Predicted: spheroidite Micrograph Image 102 One vs One Predicted: spheroidite Micrograph Image 103 One vs One Predicted: pearlite Micrograph Image 104 One vs One Predicted: spheroidite Micrograph Image 105 One vs One Predicted: spheroidite Micrograph Image 106 One vs One Predicted: network Micrograph Type: pearlite+widmanstatten Micrograph Image 0 One vs One Predicted: pearlite Micrograph Image 1 One vs One Predicted: pearlite Micrograph Image 2 One vs One Predicted: pearlite Micrograph Image 3 One vs One Predicted: spheroidite Micrograph Image 4 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 5 One vs One Predicted: pearlite Micrograph Image 6 One vs One Predicted: spheroidite Micrograph Image 7 One vs One Predicted: pearlite Micrograph Image 8 One vs One Predicted: spheroidite Micrograph Image 9 One vs One Predicted: spheroidite Micrograph Image 10 One vs One Predicted: pearlite Micrograph Image 11 One vs One Predicted: pearlite Micrograph Image 12 One vs One Predicted: pearlite Micrograph Image 13 One vs One Predicted: spheroidite Micrograph Image 14 One vs One Predicted: spheroidite Micrograph Image 15 One vs One Predicted: pearlite Micrograph Image 16 One vs One Predicted: pearlite Micrograph Image 17 One vs One Predicted: pearlite Micrograph Image 18 One vs One Predicted: spheroidite Micrograph Image 19 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 20 One vs One Predicted: pearlite Micrograph Image 21 One vs One Predicted: pearlite Micrograph Image 22 One vs One Predicted: pearlite Micrograph Image 23 One vs One Predicted: spheroidite+widmanstatten Micrograph Image 24 One vs One Predicted: pearlite Micrograph Image 25 One vs One Predicted: spheroidite Micrograph Image 26 One vs One Predicted: pearlite

```
[commandchars=
{}] In [217]: print(pred2[0]) print(pred1[0])
[commandchars=
{}] [2 2 2 0 3 2 0 2 0 0 2 2 2 0 0 2 2 2 0 3 2 2 2 3 2 0 2] [0 0 0 2 0 0 0 0 0 2 0 0
```



```

0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 2 0 2 0 2 2 0 0 0 0 0 3 0 2 0 0
2 0 2 0 2 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 2 0 0 0 0 0 2 2 0 0 2 0 2 0 0
2 0 0 2 0 0 0 2 0 0 1]

```

```

[commandchars=
{}] In [231]: #Part D # Now we want to apply the pairwise clas-
sifier pearlite vs spheroidite to the mixed pearlite vs spheroidite micro-
graphs x`test=[] y`test=[] print (combs) for i in range(len(x`i)): if
y[i]==4: x`test.append(feats[x`i][i][4]) y`test.append(4) pred3=[] for i in
range(len(x`test)): pred3.append(models[1].predict(x`test)) print(enc[4]) for
i in range(0,len(pred3[0])): print ('Micrograph Image',i) print ('Pair-
wise Classifier Predicted:',enc[pred3[0][i]]) print ('OnevsOne Classifier Pre-
dicted:',enc[pred1[0][i]])

```

```

[commandchars=
{}] [(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 3)] pearlite+spheroidite Mi-
crograph Image 0 Pairwise Classifier Predicted: spheroidite OnevsOne Clas-
sifier Predicted: spheroidite Micrograph Image 1 Pairwise Classifier Pre-
dicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Im-
age 2 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted:
spheroidite Micrograph Image 3 Pairwise Classifier Predicted: pearlite Onevs-
One Classifier Predicted: pearlite Micrograph Image 4 Pairwise Classifier Pre-
dicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Im-
age 5 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted:
spheroidite Micrograph Image 6 Pairwise Classifier Predicted: spheroidite Onevs-
One Classifier Predicted: spheroidite Micrograph Image 7 Pairwise Classifier
Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph
Image 8 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted:
spheroidite Micrograph Image 9 Pairwise Classifier Predicted: pearlite
OnevsOne Classifier Predicted: pearlite Micrograph Image 10 Pairwise Clas-
sifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Mi-
crograph Image 11 Pairwise Classifier Predicted: spheroidite OnevsOne Clas-
sifier Predicted: spheroidite Micrograph Image 12 Pairwise Classifier Pre-
dicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Im-
age 13 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted:
spheroidite Micrograph Image 14 Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite Micrograph Image 15 Pairwise Clas-
sifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Mi-
crograph Image 16 Pairwise Classifier Predicted: spheroidite OnevsOne Clas-
sifier Predicted: spheroidite Micrograph Image 17 Pairwise Classifier Pre-
dicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Im-
age 18 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted:
spheroidite Micrograph Image 19 Pairwise Classifier Predicted: spheroidite
OnevsOne Classifier Predicted: spheroidite Micrograph Image 20 Pairwise Clas-
sifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micro-
graph Image 21 Pairwise Classifier Predicted: spheroidite OnevsOne Classi-
fier Predicted: spheroidite Micrograph Image 22 Pairwise Classifier Predicted:
spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 23

```





Image 90 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 91 Pairwise Classifier Predicted: pearlite OnevsOne Classifier Predicted: pearlite Micrograph Image 92 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 93 Pairwise Classifier Predicted: pearlite OnevsOne Classifier Predicted: pearlite Micrograph Image 94 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 95 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 96 Pairwise Classifier Predicted: pearlite OnevsOne Classifier Predicted: pearlite Micrograph Image 97 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 98 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 99 Pairwise Classifier Predicted: pearlite OnevsOne Classifier Predicted: pearlite Micrograph Image 100 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 101 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 102 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 103 Pairwise Classifier Predicted: pearlite OnevsOne Classifier Predicted: pearlite Micrograph Image 104 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 105 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: spheroidite Micrograph Image 106 Pairwise Classifier Predicted: spheroidite OnevsOne Classifier Predicted: network

```
[commandchars=
{}] In [228]: #Part E predict on martensite x'test=[] y'test=[]
for i in range(len(x'i)): if y[i]==6: x'test.append(feats[i][4])
y'test.append(6) pred4=[] for i in range(len(x'test)):
pred4.append(ovc.fit(x'train1v1p,y'train1v1).predict(x'test)) print(enc[6])
for i in range(0,len(pred4[0])): print ('Micrograph Image',i,'Predicted:',enc[pred4[0][i]])
```

```
[commandchars=
{}] martensite Micrograph Image 0 Predicted: spheroidite Micrograph Image 1
Predicted: network Micrograph Image 2 Predicted: pearlite Micrograph Image
3 Predicted: spheroidite Micrograph Image 4 Predicted: spheroidite Micrograph
Image 5 Predicted: network Micrograph Image 6 Predicted: spheroidite Micro-
graph Image 7 Predicted: pearlite Micrograph Image 8 Predicted: spheroidite
Micrograph Image 9 Predicted: spheroidite Micrograph Image 10 Predicted:
pearlite Micrograph Image 11 Predicted: pearlite Micrograph Image 12 Predicted:
pearlite Micrograph Image 13 Predicted: pearlite Micrograph Image 14
Predicted: spheroidite Micrograph Image 15 Predicted: pearlite Micrograph Im-
age 16 Predicted: spheroidite Micrograph Image 17 Predicted: pearlite Micro-
graph Image 18 Predicted: spheroidite Micrograph Image 19 Predicted: pearlite
Micrograph Image 20 Predicted: pearlite Micrograph Image 21 Predicted:
spheroidite Micrograph Image 22 Predicted: spheroidite Micrograph Image 23
Predicted: pearlite Micrograph Image 24 Predicted: spheroidite Micrograph
Image 25 Predicted: pearlite Micrograph Image 26 Predicted: pearlite Micro-
```

graph Image 27 Predicted: pearlite Micrograph Image 28 Predicted: spheroidite  
Micrograph Image 29 Predicted: spheroidite Micrograph Image 30 Predicted:  
pearlite Micrograph Image 31 Predicted: network Micrograph Image 32 Pre-  
dicted: spheroidite Micrograph Image 33 Predicted: spheroidite Micrograph  
Image 34 Predicted: pearlite Micrograph Image 35 Predicted: spheroidite

```
[commandchars=  
{}] In [220]: print(pred4[0])  
[commandchars=  
{}] [0 1 2 0 0 1 0 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 0 2 2 2 0 0 2 1 0 0 2 0]  
[commandchars=  
{}] In []:
```