In [2]:
```
from scipy import misc
from PIL import Image
from skimage import exposure
from sklearn import svm

import scipy
from math import sqrt,pi
from numpy import exp
from matplotlib import pyplot as plt
import numpy as np
import glob
import matplotlib.pyplot as pltss
import cv2
from matplotlib import cm
import pandas as pd
from math import pi, sqrt
import pywt
```

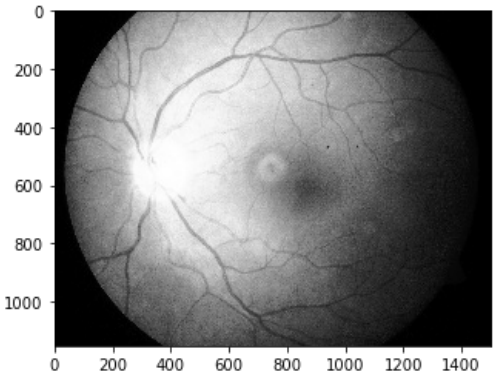
Pre-processing

In [103…
```
immatrix=[]
im_unpre = []


for i in range(1,90):
    img_pt = r'C:\Users\Rohan\Desktop\Diabetic_Retinopathy\diaretdb1_v_1_1\diaretdb1_v_1_1\resources\images\ddb
    if i < 10:
        img_pt = img_pt + "00" + str(i) + ".png"
    else:
        img_pt = img_pt + "0" + str(i)+ ".png"

    img = cv2.imread(img_pt)

    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    equ = cv2.equalizeHist(img_gray)
    immatrix.append(np.array(equ).flatten())
```

In [4]:
```
np.shape(np.array(equ).flatten())
```

Out[4]:
```
(1728000,)
```

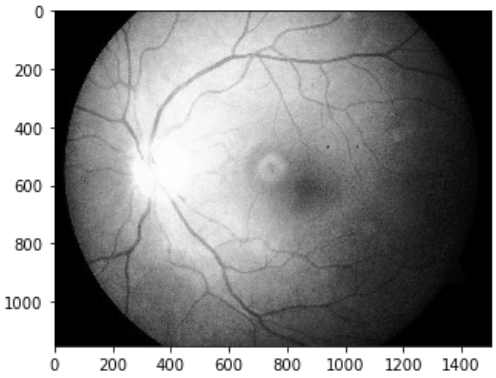
In [111…
```
np.shape(immatrix)
np.shape(equ)
plt.imshow(immatrix[78].reshape((1152,1500)),cmap='gray')
plt.show()
```



In [6]:
```
imm_dwt = []
for equ in immatrix:
    equ = equ.reshape((1152,1500))
    coeffs = pywt.dwt2(equ, 'haar')
    equ2 = pywt.idwt2(coeffs, 'haar')
    imm_dwt.append(np.array(equ2).flatten())
```

Visualising a random image

In [7]:
```
np.shape(imm_dwt)
np.shape(equ2)
plt.imshow(imm_dwt[78].reshape((1152,1500)),cmap='gray')
plt.show()
```



In [38]:
```
e_ = equ3
np.shape(e_)
e_=e_.reshape((-1,3))
np.shape(e_)
```

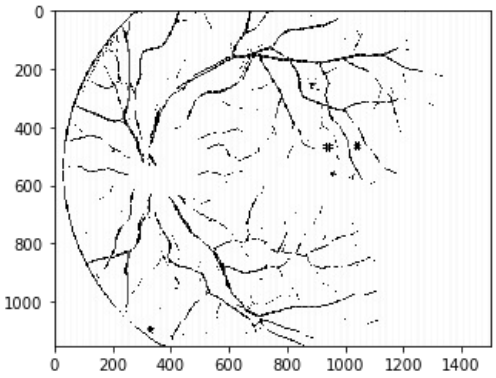
Out[38]:
```
(576000, 3)
```

Performing K-means Clusttering with PP centers(non random) neighbours on the final image

In [ ]:
```
img = equ3
Z = img.reshape((-1,3))


Z = np.float32(Z)

k=cv2.KMEANS_PP_CENTERS



criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
K = 2
ret,label,center=cv2.kmeans(Z,K,None,criteria,10,k)


center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img.shape))
```

In [10]:
```
imm_kmean = []
for equ3 in imm_gauss2:
    img = equ3.reshape((1152,1500))
    Z = img.reshape((-1,3))


    Z = np.float32(Z)

    k=cv2.KMEANS_PP_CENTERS



    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    K = 2
    ret,label,center=cv2.kmeans(Z,K,None,criteria,10,k)


    center = np.uint8(center)
    res = center[label.flatten()]
    res2 = res.reshape((img.shape))
    imm_kmean.append(np.array(res2).flatten())
```

In [113…
```
np.shape(imm_kmean)
plt.imshow(imm_kmean[78].reshape((1152,1500)),cmap="gray")
plt.show()
```



Model training

In [42]:
```
from sklearn.svm import SVC
clf = SVC()
```

In [64]:
```
Y = np.ones(89)
```

In [65]:
```
Y[1]=Y[5]=Y[7]=Y[17]=Y[6]=0
```

In [66]:
```
clf.fit(imm_kmean, Y)
```

Out[66]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [72]:
```
y_pred = clf.predict(imm_kmean)
```

In [1]:
```
k = [1,3,4,9,10,11,13,14,20,22,24,25,26,27,28,29,35,36,38,42,53,55,57,64,70,79,84,86]
```

In [3]:
```
k = k-np.ones(len(k))
```

In [87]:
```
k
```

Out[87]:
```
array([  0.,   2.,   3.,   8.,   9.,  10.,  12.,  13.,  19.,  21.,  23.,
        24.,  25.,  26.,  27.,  28.,  34.,  35.,  37.,  41.,  52.,  54.,
        56.,  63.,  69.,  78.,  83.,  85.])
```

In [92]:
```
k =[int(x) for x in k]
```

In [93]:
```
k
```

Out[93]:
```
[0,
 2,
 3,
 8,
 9,
 10,
 12,
 13,
 19,
 21,
 23,
 24,
 25,
 26,
 27,
 28,
 34,
 35,
 37,
 41,
 52,
 54,
 56,
 63,
 69,
 78,
 83,
 85]
```

In [98]:
```
imm_train = []
y_train = []
k.append(5)
k.append(7)
for i in k:
    imm_train.append(imm_kmean[i])
    y_train.append(Y[i])
```

In [99]:
```
y_train
```

Out[99]:
```
[1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 0.0,
 0.0]
```

In [100…
```
clf.fit(imm_train, y_train)
```

Out[100]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [101…
```
y_pred = clf.predict(imm_kmean)
```

In [102…
```
accuracy_score(Y,y_pred)
```

Out[102]:
```
0.9662921348314607
```

In [114…
```
from sklearn.neighbors import KNeighborsClassifier
```

In [115…
```
neigh = KNeighborsClassifier(n_neighbors=3)
```

In [116…
```
neigh.fit(imm_train, y_train)
```

Out[116]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
```

In [117…
```
y_pred2=neigh.predict(imm_kmean)
```

In [119…
```
neigh.score(imm_kmean,Y)
```

Out[119]:
```
0.9438202247191011
```