11077009 資訊碩一 林冠良 HW1 - libsvm

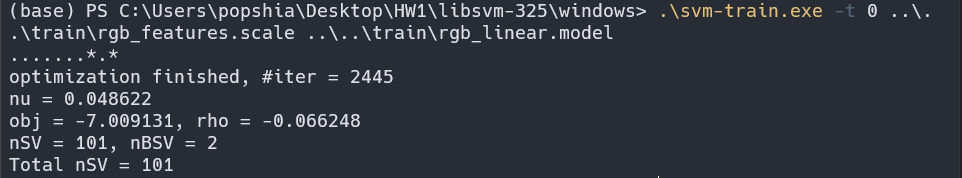
1. **Using linear kernel and RGB color feature**

**(400 × 3 dims for each image)**

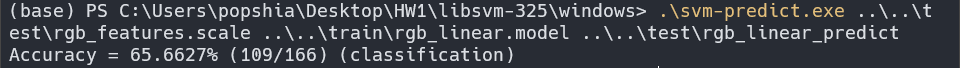
1. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
2. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.



1. Use the scaled file as input and send it to svm-train.exe in the same directory.



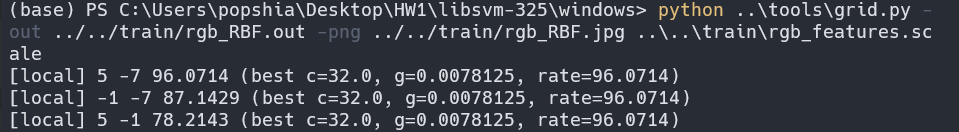
1. Predict the test dataset using the model file from previous step and get an accuracy of 65.6527%.



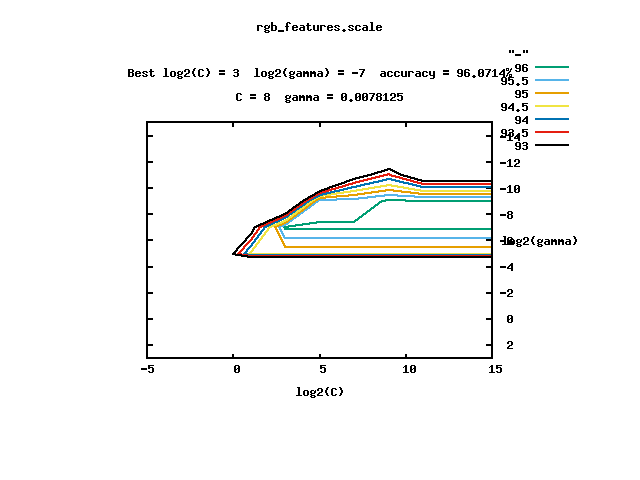
1. **Using RBF kernel and RGB color feature**
2. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
3. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.



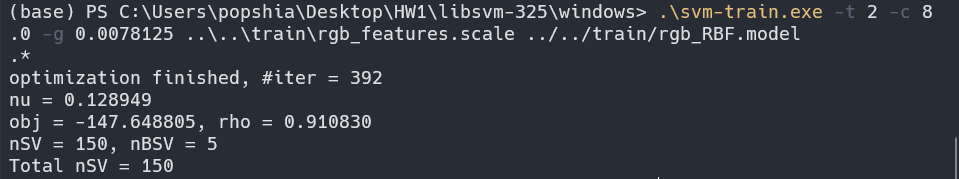
1. Use grip.py in the tools directory to find the best parameters.

****

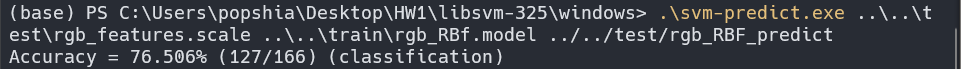
****

****

1. Take the best c and gamma from previous step and use it as parameters in the svm-train.exe command.

****

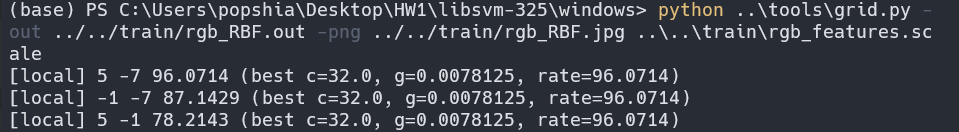
1. Predict the test dataset using the model file from previous step and get an accuracy of 76.506%

****

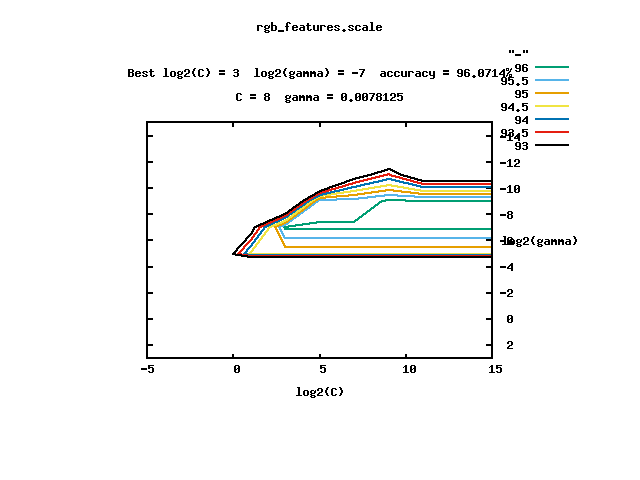
1. **Using RBF kernel and RGB color feature, 5-fold cross validation**
2. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
3. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.

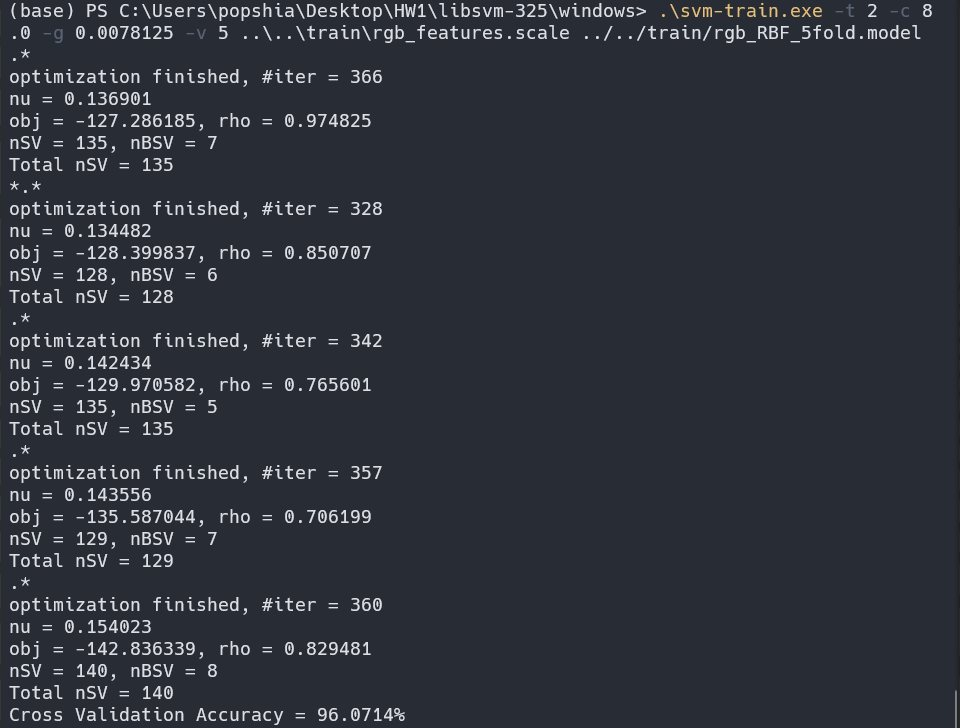


1. Use grip.py in the tools directory to find the best parameters.

****

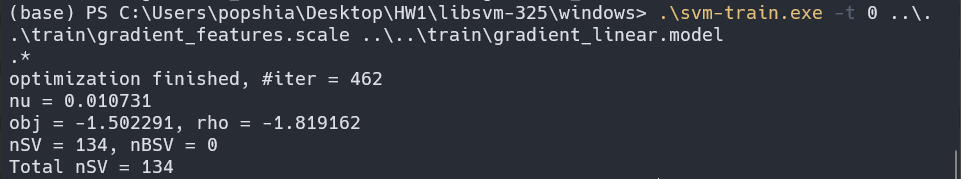
****

****

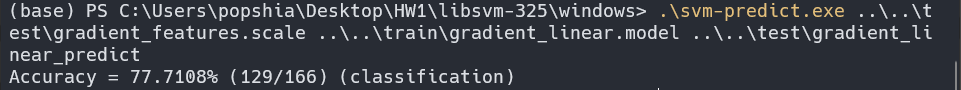
1. Take the best parameters from previous step and add an option of 5-fold cross validation in the training step, in this situation, svm-train doesn’t output a model, only a generalization performance.
2. We can see that in the estimate generalization performance, the cross-validation accuracy is estimated to be 96.0714%.
3. **Using linear kernel and gradient feature**
4. **2 dims for each image)**
5. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
6. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.



1. Use the scaled file as input and send it to svm-train.exe in the same directory.



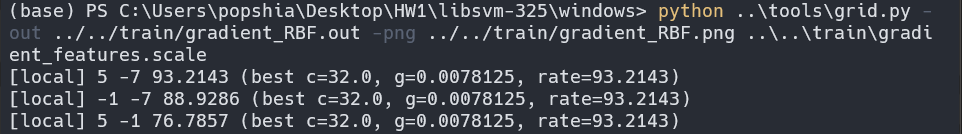
1. Predict the test dataset using the model file from previous step and get an accuracy of 77.7108%.



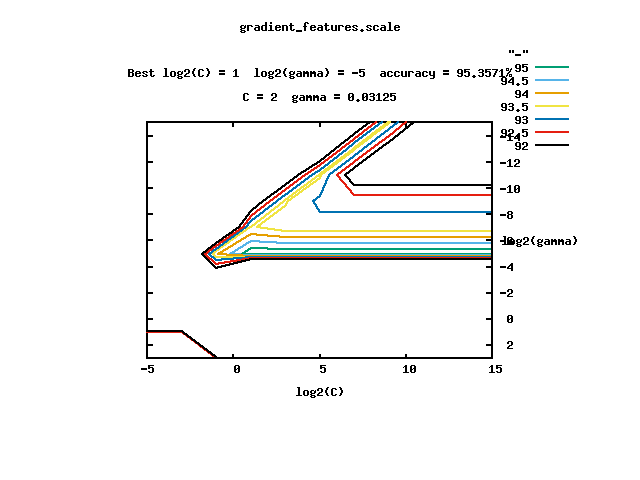
1. **Using RBF kernel and gradient feature**
2. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
3. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.



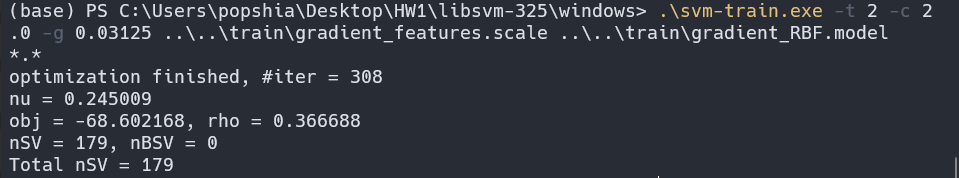
1. Use grip.py in the tools directory to find the best parameters.

****

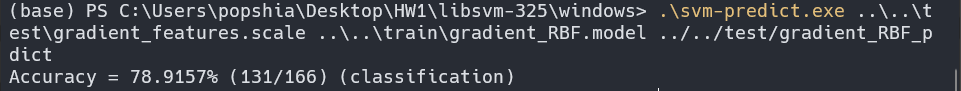
****

****

1. Take the best c and gamma from previous step and use it as parameters in the svm-train.exe command.

****

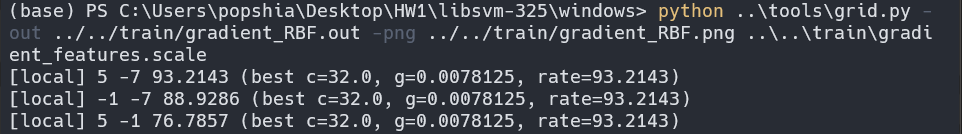
1. Predict the test dataset using the model file from previous step and get an accuracy of 78.9157%

****

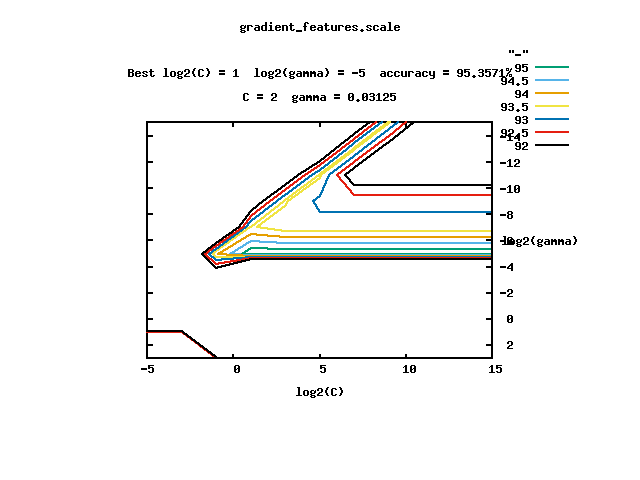
1. **Using RBF kernel and gradient feature, 5-fold cross validation**
2. Use self-written txt\_to\_svm.py to convert the original feature values to svm file format.
3. Take the converted file and scale the data to be in [0, 1] using svm-scale.exe in libsvm-325/windows.



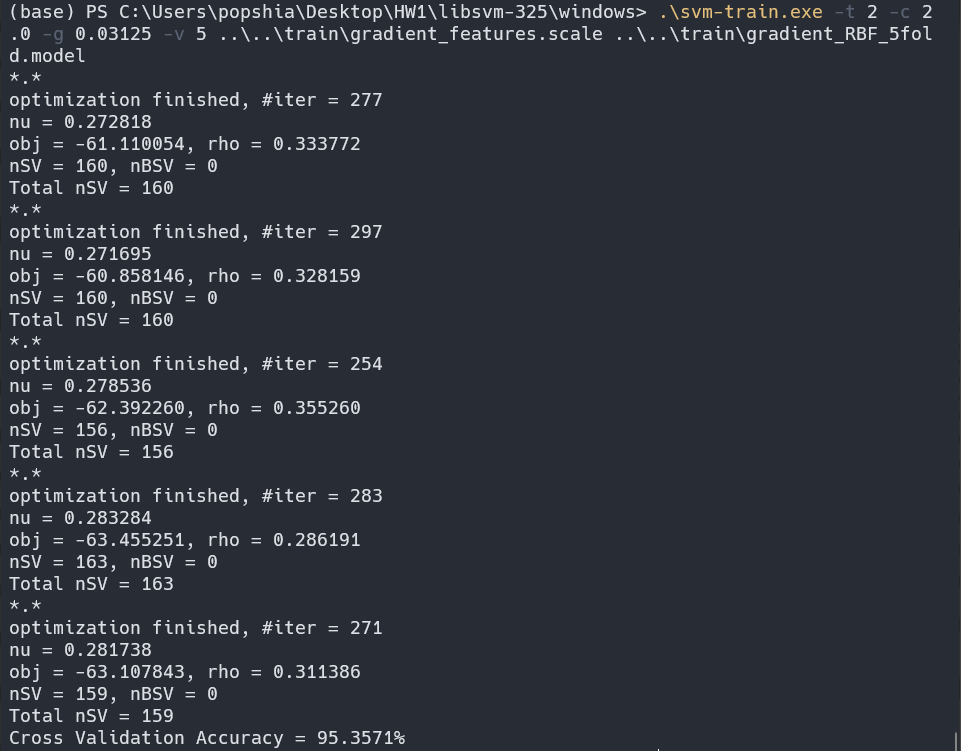
1. Use grip.py in the tools directory to find the best parameters.

****

****

****

1. Take the best parameters from previous step and add an option of 5-fold cross validation in the training step, in this situation, svm-train doesn’t output a model, only a generalization performance.



1. We can see that in the estimate generalization performance, the cross-validation accuracy is estimated to be 95.3571%.