Task II: Image Classification

By,

Vinay Jayadev

Abhilash Madathil Karottu Sadasivan

Problem statement:

Task is classification of pictures from image collection Caltech1011, containing images from 101 different classes. Perform a classification using data and determine good parameters for the chosen classifier. Choose any discussed in the lecture (Nearest Neighbour, Bayes, Support Vector Machine, Decision Tree, Random Forest or Neural Network) We will not work on the picture data directly. We will use a feature called Edge Histogram2. You will receive the following files as part of the task:

• Images.csv: Contains the images represented by image ID and corresponding class. The ID and the class are separated by a semicolon. The first line of the file contains the number of images.

• EdgeHistogram.csv: Contains feature data for Edge Histogram feature for the images. Each line here consists of the image ID followed by the feature vector. The ID and the vector are separated by a semicolon. The individual dimensions are also separated by a semicolon. The first line contains the number of images followed by the number of dimensions separated by a semicolon.

When training, the following amounts of training images per class should be tried out: 3, 5, 10, 15. Also, you should choose a suitable strategy for selecting the training and test images. Depending on the amount and your chosen classifier you should also try to find the hyperparameters that yield the highest accuracy. Write or use an appropriate program for your experiments and make a transcript of your experiment setup. Describe, present, evaluate and discuss your results.

Procedure:

We started by using the whole dataset for training and testing, 80% for training and 20% for testing. From this phase itself we started adjusting hyper parameters.

We tested a few different algorithms such as Random Forest, SVM etc and selected SVM based on accuracy. We found SVM to be yielding the highest accuracy from the get-go about 40% without adjusting hyper parameters, while random forest showed a lower accuracy around 36%.

Test images were selected according to requirements as 3, 5, 10 and 15 per class and Hyper parameters were further adjusted to gain greater accuracy.

As expected, we were able to get greater accuracy as no. of images per class increased. There was an exception in the case of 3 image set as it failed to run due to having number of images below case k-split value.

In the function load_data(), some data Engineering techniques were employed for merging the Image set with corresponding Histogram data.

In preprocess_data(), One-Hot encoding was used to assign integer ids to names of various classes, as strings are not acceptable in training data and data is split in to training and test.

In function train_and_predict(), GridSearchCV is used to adjust hyper parameters, and run SVM algorithm.

SVM Parameters:

C: Lower value means that the model will have a higher bias and lower variance. On the other hand, a larger value of C means that the model will have a lower bias but higher variance.

Gamma: The gamma parameter is used to define the shape of the radial basis function (RBF) kernel, which is commonly used as the kernel in SVM algorithms. The gamma parameter controls the "curviness" of the boundary. A small value of gamma will result in a flat boundary, while a large value of gamma will result in a "bumpier" boundary.

Kernel: The kernel is a function that is used to transform the input data into a higher dimensional space. Some of the common kernels used in SVM are linear, radial basis function (RBF), polynomial, and sigmoid.

Over 30 runs were conducted their results are tabulated in below findings.

Findings:

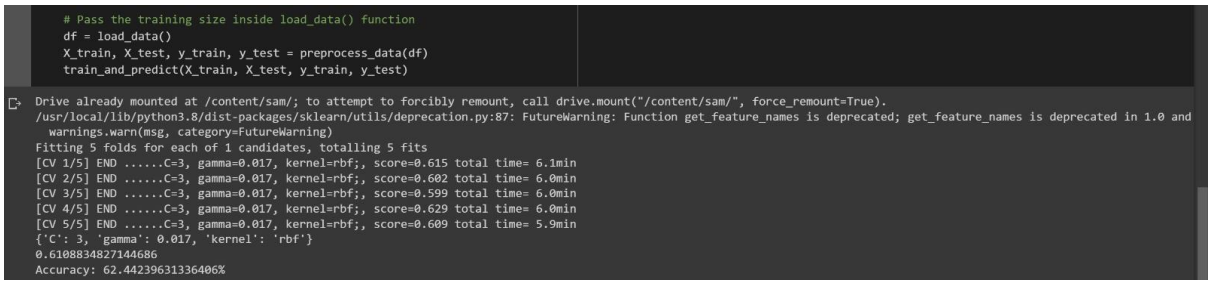
Test Case	C	Gamma	Kernal	Accuracy (%)
1	.1	.001	default	28
2	.1	.01	default	34.93712411153636
3	1	.01	default	55.98687807545106
4	1	.001	default	39.85784581738655
5	1	.1	default	23.332511606886067
6	1.5	.1	default	24.781336724822492
7	1	1	default	09.445063851880073
8	1	.02	default	53.47184253690541
9	2	.01	default	57.13504647348278
10	3	.01	default	58.33788955713505
11	5	.04	default	49.740362456065645
12	8	.1	default	24.959035069043695
13	3	.02	default	60.30617823947513
14	3	.026	default	54.23728813559322
15	3	.03	default	56.75232367413887
16	3	.018	default	58.28321487151449
17	3	.024	default	58.11919081465281
18	3	.018	default	61.17511520737328
19	3	.022	default	58.72061235647895
20	3	.02	Rbf	61.00230414746544
21	3	.018	Rbf	61.40552995391705
22	3	.019	Poly	57.18125960061444
23	3	.019	Sigmoid	8.525345622119816
24	3	.0185	Rbf	61.40552995391705
25	3	.017	rbf	62.44239631336406

Greater accuracy was obtained after updating hyper parameters C and gamma. Manipulating kernel also showed significant change in accuracy.

Highest efficiency of 62.44239631336406% was observed with C: 3, gamma: .017 and kernel: ‘rbf’.

Outputs:

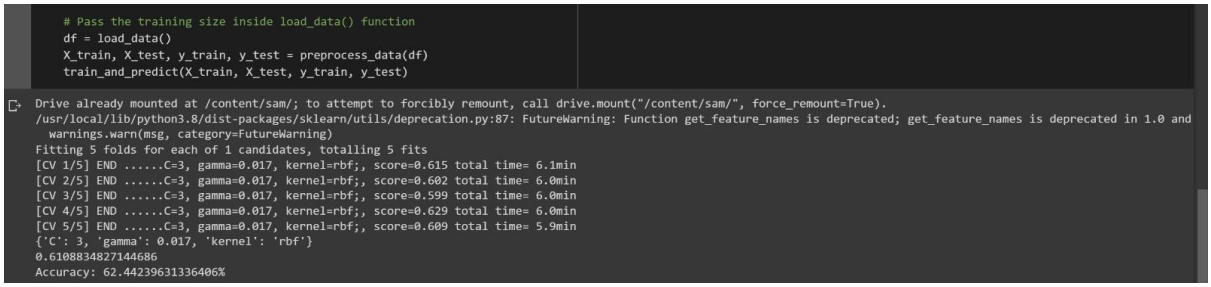
On full data set accuracy obtained: 62.44%



```
# Pass the training size inside load_data() function
df = load_data()
X_train, X_test, y_train, y_test = preprocess_data(df)
train_and_predict(X_train, X_test, y_train, y_test)
```

```
Drive already mounted at /content/sam/; to attempt to forcibly remount, call drive.mount("/content/sam/", force_remount=True).
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
  warnings.warn(msg, category=FutureWarning)
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV 1/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.615 total time= 6.1min
[CV 2/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.602 total time= 6.0min
[CV 3/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.599 total time= 6.0min
[CV 4/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.629 total time= 6.0min
[CV 5/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.609 total time= 5.9min
{'C': 3, 'gamma': 0.017, 'kernel': 'rbf'}
0.6108834827144686
Accuracy: 62.44239631336406%
```

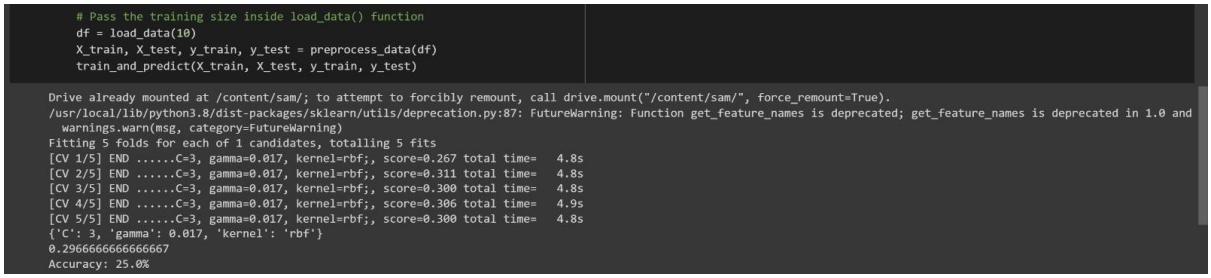
For class size 5 accuracy obtained: 12%



```
# Pass the training size inside load_data() function
df = load_data()
X_train, X_test, y_train, y_test = preprocess_data(df)
train_and_predict(X_train, X_test, y_train, y_test)
```

```
Drive already mounted at /content/sam/; to attempt to forcibly remount, call drive.mount("/content/sam/", force_remount=True).
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
  warnings.warn(msg, category=FutureWarning)
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV 1/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.615 total time= 6.1min
[CV 2/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.602 total time= 6.0min
[CV 3/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.599 total time= 6.0min
[CV 4/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.629 total time= 6.0min
[CV 5/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.609 total time= 5.9min
{'C': 3, 'gamma': 0.017, 'kernel': 'rbf'}
0.6108834827144686
Accuracy: 62.44239631336406%
```

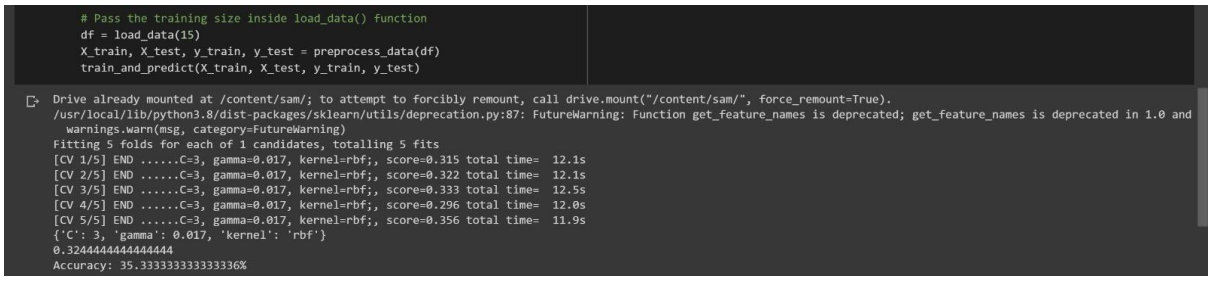
For class size 10 accuracy obtained: 25%



```
# Pass the training size inside load_data() function
df = load_data(10)
X_train, X_test, y_train, y_test = preprocess_data(df)
train_and_predict(X_train, X_test, y_train, y_test)
```

```
Drive already mounted at /content/sam/; to attempt to forcibly remount, call drive.mount("/content/sam/", force_remount=True).
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
  warnings.warn(msg, category=FutureWarning)
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV 1/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.267 total time=   4.8s
[CV 2/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.311 total time=   4.8s
[CV 3/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.300 total time=   4.8s
[CV 4/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.306 total time=   4.9s
[CV 5/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.300 total time=   4.8s
{'C': 3, 'gamma': 0.017, 'kernel': 'rbf'}
0.2966666666666667
Accuracy: 25.0%
```

For class size 15 accuracy obtained: 35%



```
# Pass the training size inside load_data() function
df = load_data(15)
X_train, X_test, y_train, y_test = preprocess_data(df)
train_and_predict(X_train, X_test, y_train, y_test)
```

```
Drive already mounted at /content/sam/; to attempt to forcibly remount, call drive.mount("/content/sam/", force_remount=True).
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
  warnings.warn(msg, category=FutureWarning)
Fitting 5 folds for each of 1 candidates, totalling 5 fits
[CV 1/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.315 total time=  12.1s
[CV 2/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.322 total time=  12.1s
[CV 3/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.333 total time=  12.5s
[CV 4/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.296 total time=  12.0s
[CV 5/5] END ......C=3, gamma=0.017, kernel=rbf;, score=0.356 total time=  11.9s
{'C': 3, 'gamma': 0.017, 'kernel': 'rbf'}
0.3244444444444444
Accuracy: 35.333333333333336%
```