

Homework 2 - Venice Boat Classification

Machine Learning 2018-19 - Sapienza

Luigi Russo 1699981

December 13, 2018

Contents

1	Introduction	3
1.1	Learning goals	3
1.1.1	Classification problem	3
1.1.2	Metrics	3
1.1.3	Many models	3
2	Tools used	4
3	The pretrained model	4
4	The dataset	4
4.1	Preprocessing	5
4.1.1	Extract features	5
5	Classification	5
5.0.1	t-sne features	5
5.1	Running the classifiers	5
6	The results	7
6.1	Metrics	8
6.1.1	SVM	8
6.1.2	KNN	8
6.1.3	RF	8
6.1.4	ET	8
6.1.5	ML	8
6.1.6	GNB	9
7	Conclusions	9

1 Introduction

This is my report for the second homework of Machine Learning course, about the Venice Boat Classification. First of all I will point out which have been the goals of this homework, starting of course from defining the specific classification problem (view section 1.1.1) that I have chosen to focus on. After defining the kinds of algorithms that I have implemented and used, I will describe the dataset (view section 4) and how I have handled the preprocessing phase (view section 4.1). I will finally present the results of the evaluation phase (view section 6).

1.1 Learning goals

1.1.1 Classification problem

I have decided to focus on the problem of classifying **18** different classes of Venice boats. Given an instance of image, I want to be able to compute which is the correct type of boat of the input image. I have decided to use a pretrained model (Inception v3) in order to extract the features of each image: at this point, the feature vector is passed to some classifiers, such as SVM classifier. The model can be finally tested, computing the accuracy and it is possible to plot the confusion matrix.

1.1.2 Metrics

The main goal of this homework is to correctly classify the highest number of input images, for this reason I will take into account both accuracy and precision metrics.

1.1.3 Many models

As in the first homework, also in this one I will give great importance to the comparison between different models. Once the pretrained graph extracts the features vector of the input image, it is passed to some classifier that actually runs the last step of the classification. I have decided to compare these 6 different classifiers:

- Support Vector Machine (**SVM**)
- Extra Trees (**ET**)

- Random Forest (**RF**)
- K-Nearest Neighbor (**KNN**)
- Multi-Layer Perceptron (**ML**)
- Gaussian Binomial (**GNB**)

2 Tools used

I have used Python 3.6 programming language, because of its well-documented modules and libraries: I did not want to reinvent the wheel, so I have used the well-known *Scikit-Learn*, *Numpy*, *Matplotlib* libraries to build, train, and evaluate the classifier. In the next sections I will give further details on the precise methods used. I have provided the source-code as a collection of Python files (extension .py). It is possible to view the source-code in the attached zipped file (it is also present [in this GitLab repository](#)).

3 The pretrained model

Since I have decided to use a pretrained model to extract features, I have downloaded and reused a checkpoint of the Inception v3 model. In particular I have used the checkpoint of 12/05/2015, saving it in the 'pretrained' folder. This file (extension .pb) cannot be directly used, but it has to be imported, when initializing the TensorFlow session: this is possible thanks to some TensorFlow functions: *ParseFromString* and *import_graph_def*; the first one simply transforms a binary file into a complex graph object, instead the second one actually makes the graph available to the current TensorFlow session.

4 The dataset

Both training and test images are taken from the Sapienza MarDCT dataset, a wide collection of Venice boat images. The dataset is already divided into training and test images: however, some preprocessing was needed in order to handle the different structure of the two sets of images and also because there were some multi-labeled images in the test set: in particular, the training set

was organized in 24 folders, each containing a specific class of images; the name of each folder is the name of the class, i.e. the boat type. The test set is provided as a wide collection of images; thanks to the file `ground_truth.txt` it is possible to associate each file to a specific label; however, as already said, some images contain fragments of boats (these images are labeled as partial) or two or more boats of different types (multiple label). I have decided to simply skip these test images.

4.1 Preprocessing

First of all I have decided to limit the number of classes to 18 (out of the total 24), excluding all classes with less than 10 images. Figure 1 lists both all the classes that I have included in the training phase and the number of training images for each label.

4.1.1 Extract features

Once the pretrained graph was taken into main memory from disk, the images have been fed to a TensorFlow implementation of Inception v3 with the classification layer removed in order to produce a set of labeled feature vectors of 2048 floats: this *next-to-last* layer is a tensor called 'pool_3:0'.

5 Classification

5.0.1 t-sne features

I have carried out dimensionality reduction on the 2048-D feature vector using t-distributed stochastic neighbor embedding (**t-SNE**) to transform the input features into a 2-D feature which is easy to visualize. The t-SNE is used as an informative step: if the same color (i.e. label) points are mostly clustered together there is a high chance that we could use the features to train a classifier with high accuracy. Figure 2 shows the result of this phase.

5.1 Running the classifiers

For every model (for the complete list see 1.1.3) I have run the proper classifier. In most cases I have simply used the default configuration, and the

Class	# samples
Alilaguna	113
Ambulanza	85
Barchino	112
Caorlina	1
Gondola	24
Lanciafino10m	22
Lanciafino10mBianca	484
Lanciafino10mMarrone	355
Lanciamaggioredi10mBianca	9
Lanciamaggioredi10mMarrone	5
Motobarca	215
Motopontonerettangolare	18
MotoscafoACTV	11
Mototopo	878
Patanella	279
Polizia	71
Raccoltarifiuti	94
Sandoloaremi	13
Sanpiero	5
Topa	78
VaporettoACTV	949
Vigilidelfuoco	12
Water	907

Figure 1: *Boat classes from dataset MarDCT*

complete list of parameters passed to each classifier can be seen in the file `main.py`.

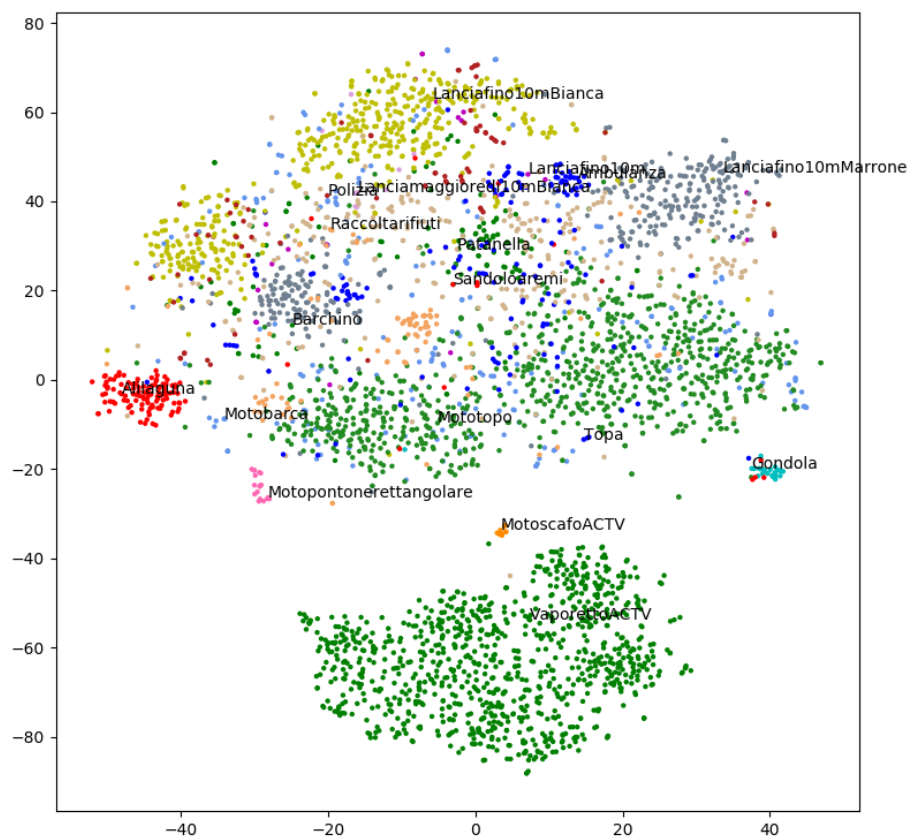


Figure 2: *t-SNE features extracted from Inception v3.*

6 The results

For each model I have computed the accuracy and the precision, and I have finally plotted the confusion (and not normalized) matrix. They are shown in the following images.

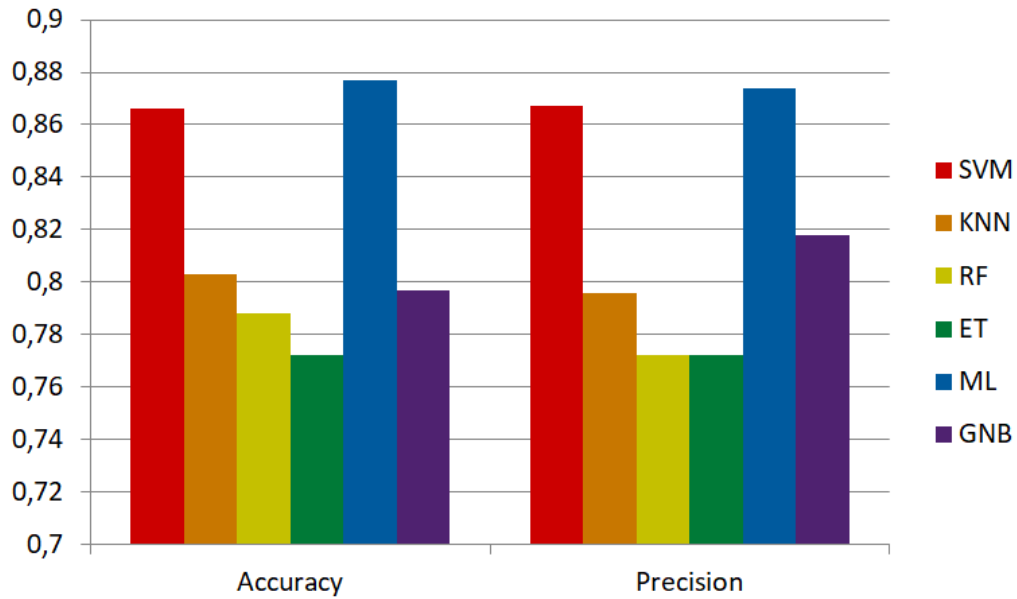


Figure 3: *Precision and accuracy of the classifiers*

6.1 Metrics

6.1.1 SVM

The accuracy was 86.6% and the precision was 86.7%

6.1.2 KNN

The accuracy was 80.3% and the precision was 79.6%

6.1.3 RF

The accuracy was 78.8% and the precision was 77.2%

6.1.4 ET

The accuracy was 77.2% and the precision was 77.2%

6.1.5 ML

The accuracy was 87.7% and the precision was 87.4%

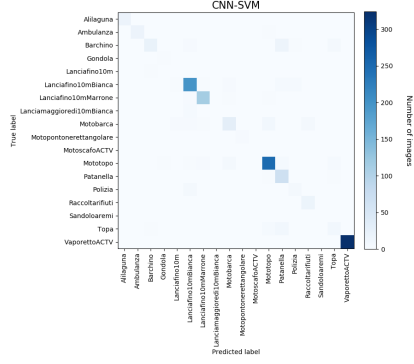


Figure 4: SVM confusion matrix

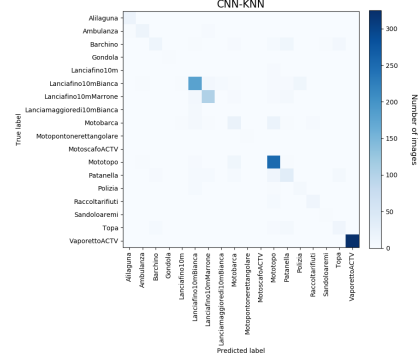


Figure 5: KNN confusion matrix

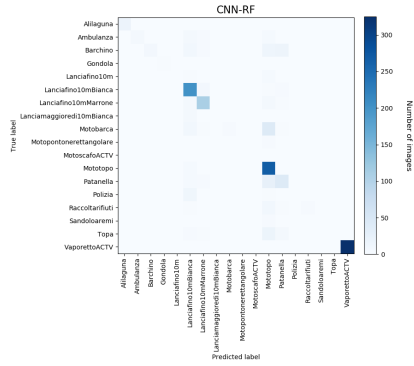


Figure 6: RF confusion matrix

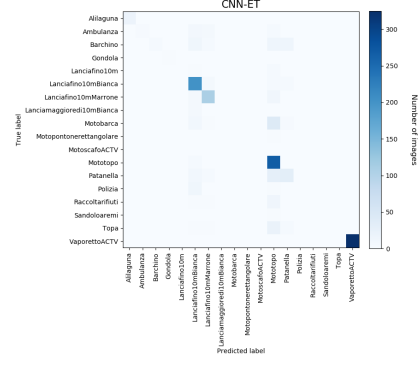


Figure 7: ET confusion matrix

6.1.6 GNB

The accuracy was 79.7% and the precision was 81.8%

7 Conclusions

Figure 3 shows that both Gaussian Binomial and SVM reach very good results in accuracy and precision, with values higher than 0.86. The other models reach levels of accuracy and precision under the 0.8 threshold.

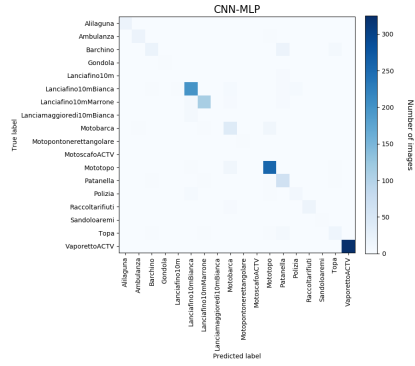


Figure 8: ML confusion matrix

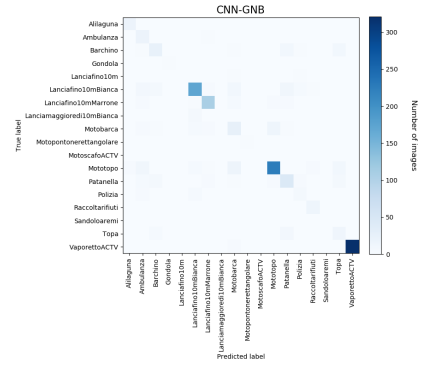


Figure 9: GNB confusion matrix

References

- [1] Tom M. Mitchell *Machine Learning*.