

Homework 2 - Venice Boat Classification

Machine Learning 2018-19 - Sapienza

Luigi Russo 1699981

December 12, 2018

Contents

1	Introduction	3
2	Image classification problem	3
2.1	Learning goals	3
2.1.1	Metrics	3
2.1.2	Many models	3
3	Tools used	4
4	The pretrained model	4
5	The dataset	4
5.1	Preprocessing	4
5.1.1	Extract features	6
6	Classification	6
6.0.1	t-sne features	6
6.1	Running the classifiers	7
7	The results	7
7.1	Metrics	7
7.1.1	SVM	7
7.1.2	KNN	7
7.1.3	RF	8
7.1.4	ET	8
7.1.5	ML	8
7.1.6	GNB	8
7.2	Confusion matrices	8
8	Conclusions	8

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 ??) 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 5). After that, I will finally present the results of the evaluation phase (view section 7).

2 Image 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 is then tested, computing at the end the accuracy and plotting the confusion matrix.

2.1 Learning goals

2.1.1 Metrics

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

2.1.2 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 6 different classifiers:

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

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

3 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](#)).

4 The pretrained model

I have decided to use 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.

5 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.

5.1 Preprocessing

First of all I have decided to limit the number of classes to 18 (out of the total 24), because of the number of sample images of some classes: in some

cases less than 10 images. This led me to the following list of classes:

- Alilaguna
- Ambulanza
- Barchino
- Gondola
- Lanciafino10m
- Lanciafino10mBianca
- Lanciafino10mMarrone
- Lanciamaggioredi10mBianca
- Motobarca
- Motopontonerettangolare
- MotoscafoACTV
- Mototopo
- Patanella
- Polizia
- Raccoltarifiuti
- Sandoloaremi
- Topa
- VaporettoACTV

As for the test set, it provided a ground truth text file, containing the list of all the images and their label: in some cases, however, this label was not present, or it was simply an invalid label (not present in the training set) such as multiple, partial, etc. I had to handle these issues, filtering the images with correct labels!

5.1.1 Extract features

Once the pretrained graph was taken into main memory from disk, they 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.

6 Classification

6.0.1 t-sne features

Dimensionality reduction is carried out on the 2048-D features using t-distributed stochastic neighbor embedding (**t-SNE**) to transform them 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.

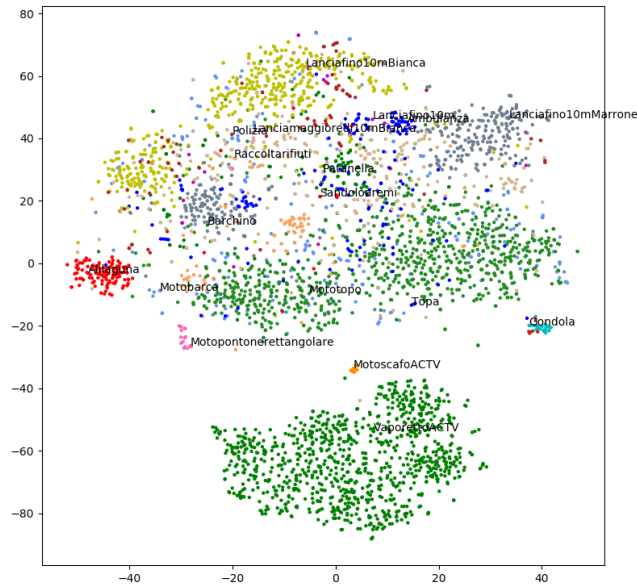


Figure 1: t -SNE features

6.1 Running the classifiers

For every model (see models listed above) I have run the proper classifier: in most cases I simply used the default configuration.

7 The results

7.1 Metrics

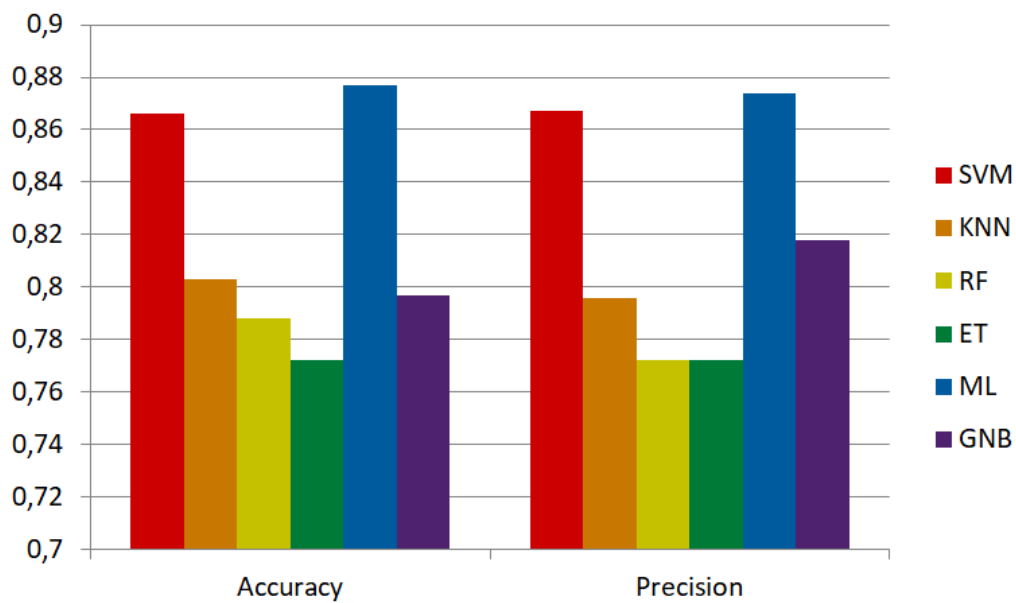


Figure 2: *Precision and accuracy of the classifiers*

7.1.1 SVM

The accuracy was 86.6% and the precision was 86.7%

7.1.2 KNN

The accuracy was 80.3% and the precision was 79.6%

7.1.3 RF

The accuracy was 78.8% and the precision was 77.2%

7.1.4 ET

The accuracy was 77.2% and the precision was 77.2%

7.1.5 ML

The accuracy was 87.7% and the precision was 87.4%

7.1.6 GNB

The accuracy was 79.7% and the precision was 81.8%

7.2 Confusion matrices

For each model I have also plotted the confusion matrix (not normalized!). They are shown in the following images.

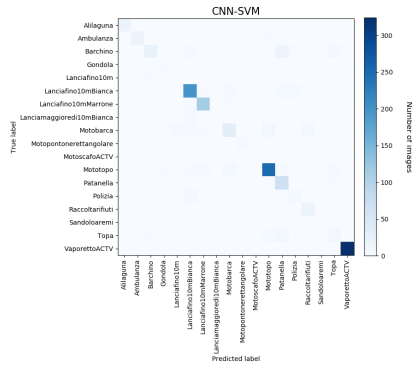


Figure 3: SVM confusion matrix

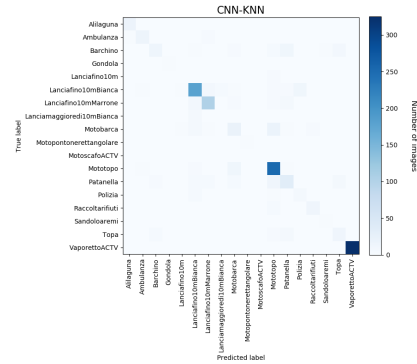


Figure 4: KNN confusion matrix

8 Conclusions

The model with highest accuracy and precision is the Gaussian Binomial.

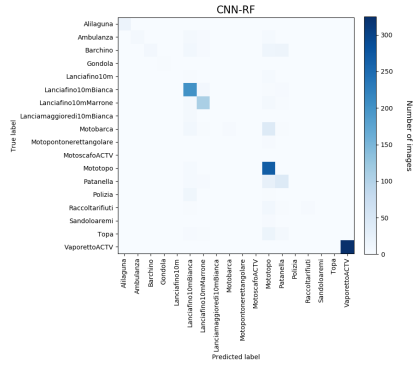


Figure 5: RF confusion matrix

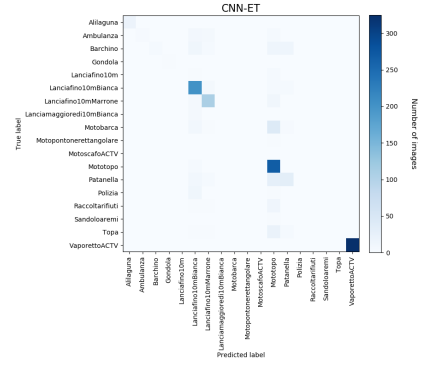


Figure 6: ET confusion matrix

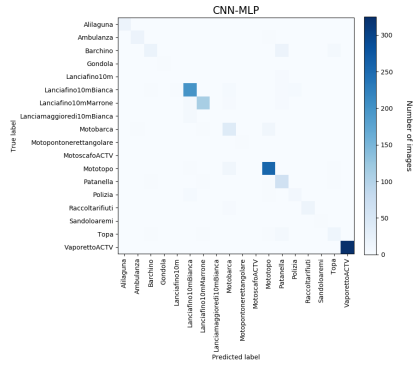


Figure 7: ML confusion matrix

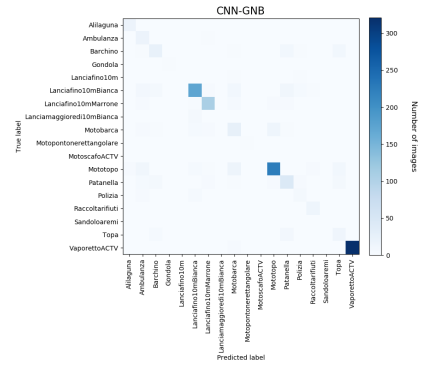


Figure 8: GNB confusion matrix

References

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