SAPIENZA UNIVERSITY OF ROME

Machine Learning

Second Homework Report

Author

Silvio Severino

ID number 1705967

Date 30, Dec 2018

Contents

Introduction

Venice Boat Traffic

The particularity of Venice is that their peoples live principally on the water, from the famous Gondole, to the public services such as Police, Ambulance and so on. Their waters are really full of boats and so the boat traffic is a real problem for them as well as car traffic is a real problem for a city. For the above reasons, the Municipal Administration of Venice launched the **ARGOS** system since 2006. ARGOS(Automatic Remote Grand Canal Observation System) is an intelligent surveillance system for boat traffic monitoring that is in use in the Gran Canal of Venice 24/7. This system is used for several functionalities, such as: optical detection and tracking of moving targets; computing position, speed, and heading of any moving target observed by a camera; automatic detection of a set of predefined events; transmission of data and video stream to the Control Center [BIPT15]. Moreover, another functionality of ARGOS is that it allows to get some snapshots of the boats, in order to check them. In fact, it was impossible to check all boats without an automatic system.

Machine Learning in Grand Canal of Venice

Another important goal of the ARGOS system is to track data in order to generate snapshots of the boats. This automatic procedure is very convenient because it can provide a large set of samples, without human intervention. However, it introduces some errors according to several situations:

- False positives, the snapshot doesn't contain a boat but static elements, e.g. water.

- Partial acquisition, the snapshot contains a partial view of the boat.
- Multiple boats, the snapshot contains more one boat. It isn't a very error, but it makes hard the classification process.
- False negatives, the system doesn't get the snapshot.

Knowing the above errors is possible to classify a machine learning model in order to recognize both the kind of boat (e.g. Gondola, Speedboat and so on) and its family (e.g. Public utility, People transport, and so on), also considering the false positive snapshots [BIPT15]. The goals of this homework was to implement some machine learning techniques.

Methodologies and Technologies

MarDCT

MarDCT is the acronym of Maritime Detection, Classification, and Tracking. It is a publicly available database which contains a large number of maritime data, including video streams and images with ground truth data, coming from multiple sources and from different scenarios. MarDCT can be used to help in developing an intelligent surveillance system for the maritime environment [uoR]. In particular, for this homework I have used **Sc5** dataset, having 4774 images for training and 1969 images for testing [BIPT15]. Moreover, Sc5 includes 24 different categories of boats, used to classify their kind. However, this dataset was not divided into boat family, problem solved with a little python script.

Classification Methods

Understand if a snapshot contains a boat or not and so its kind and its family, is a classification problem. Thanks to the dataset described in the previous section, it was possible to use some classification to train a machine learning system. Among the many techniques have been chosen three, explain in detail below.

Convolutional Neural Networks

A convolutional neural network is a particular kind of neural network most commonly applied to analyzing visual imagery. It uses a variant of multilayer perceptrons, designed to require minimal preprocessing. Usually its architecture consists of an input layers,

an output layer and multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers. Nowadays, exist many pre-trained architecture of CNN's, such as AlexNet, GoogleNet, LeNet, VGG and so on. I have decided to use the follows, changing a little bit their architecture.

VGG16

The **VGG16** or Visual Geometry Group is a very deep neural network introduced in 2014, having 16 layers and 138 millions of parameters [KS]. It is possible to view its architecture in the picture shown below (1).

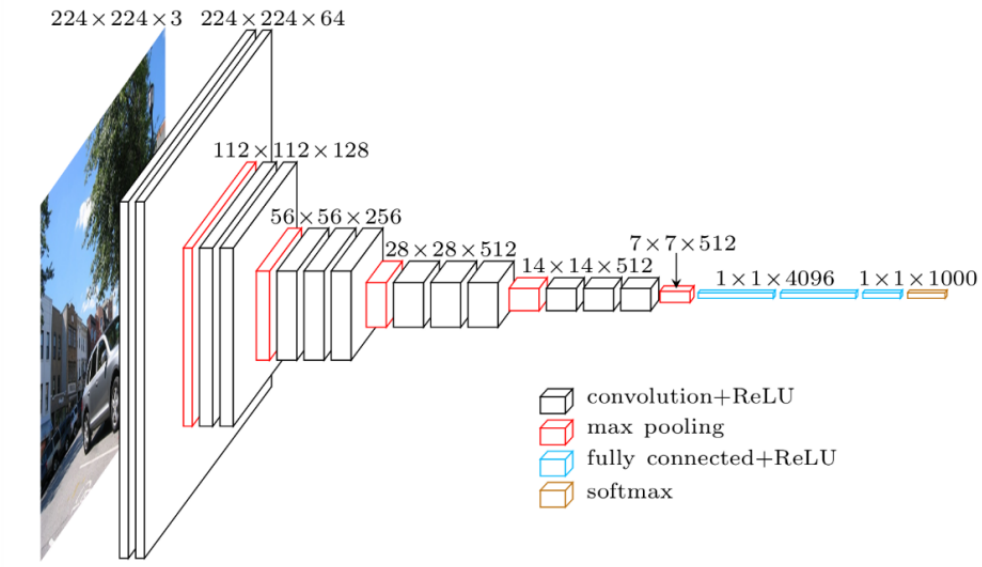


Figure 1: Vgg16 architecture

I've decided to use this network because the sc5 dataset has large pictures (about 200x800), and so I needed a very deep network. It starts from a layer with 64 filters up to double layers with 512 filters. Moreover, I've modified a little bit the network, changing the input shape size (from $224x224x3$ to $180x180x3$), removing one of 512 filter layer and changing the padding of Convolutional layer, from "same" to "valid". I have made these changes for adapting in the best way the network to my dataset.

SmallerVGGNet

It is an unofficial deep network of the Visual Geometry Group networks family. Its architecture is similar to VGG16, with fewer layers. The main difference is that this network starts from a layer with 32 filter up to a layer with 128 . Although it is shorter then VGG16, I've obtained interesting results. Can you see its architecture in the picture shown below (2).

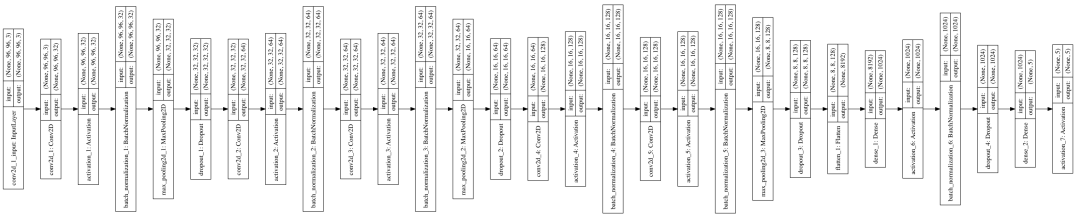


Figure 2: SmallerVGGNet architecture

Random Forest

The **Random forest** is an enseble learning method for classification, regression and other tasks. Their functioning is based on the decision tree. In fact, at training time a multitude of decision trees are built, and outputting the class that is the mode of classes (for classification). RF is a robust algorithm and usually perform very well, but, this algorithm is complicated to tune [wik]. I've tried to use also this method but I have obtained unsatisfactory results, so I have decided to discard it.

Libraries

TensorFlow - Keras

TensorFlow in an open source library for machine learning. It was developed by the Google Brain in 2015. In particular for this homework, I've used **Keras** library, another open source library that use TensorFlow back-end. Keras is principally used for neural networks and deep learning.

Scikit-learn

Scikit-learn is the most open source python library used for machine learning problems, including some algorithm for classification and regression and some metrics.

Numpy

Numpy is the most python library used for the math problems. Is including in that some math functions for vectors, matrices and so on.

Hardware

The hardware used for this homework is the follow:

- **CPU**: Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz
- **RAM**: 16GB
- **GPU**: NVIDIA(R) GeForce GTX 950M

Problems and Performance

Preprocessing Dataset

The first operation that I have done, has been to preprocess Sc5 dataset, in order to adapt this at the CNN's. I started with divide the images into subfolders, obtaining a model like this: $\{family-class\}-\{boat-class\}-images$. It this way it was possible to read the images and to know at the same time their classes. The same model it was applied to the test set. Considering that the test set did not have some boat-classes, I have decided to remove the missing classes from the training set, extracting the 20% of their content. Moreover, I resized each image before to train the CNN's. Finally, I've split the training set into an 80% for training and 20% for testing and after computed the CNN, I've tested the resulting model with the test set.

Boat Classification

In this first part of homework it was necessary to create an application that can classify correctly a given image with its category. The dataset that I've used (see Section **2.1**) is divided into 24 classes:

1. Alilaguna
2. Ambulanza
3. Barchino
4. Cacciapesca
5. Caorlina
6. Gondola
7. Lanciafino10m
8. Lanciafino10mBianca
9. Lanciafino10mMarrone
10. Lanciamaggioredi10m
11. Lanciamaggioredi10mBianca
12. Lanciamaggioredi10mMarrone
13. Motobarca
14. Motopontonerettangolare
15. MotoscafoACTV
16. Mototopo
17. Patanella
18. Polizia
19. Raccoltarifiuti
20. Sandoloaremi
21. Sanpierota
22. Topa
23. VaporettoACTV
24. VigilidelFuoco

Figure 3: Boat classification classes

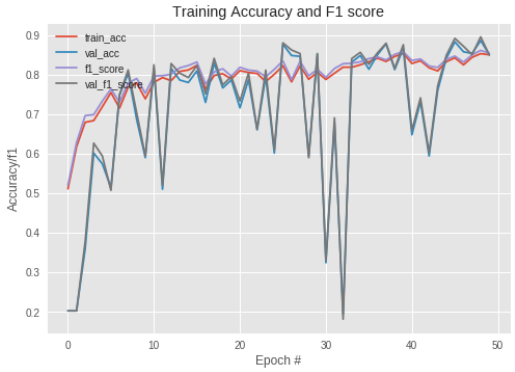
A good metrics for this problem is the F1 score and confusion matrix. F1 score considers both the precision p, where p is the number of correct positive results divided by the number of all positive results and the recall r, where r is the number of correct positive results divided by the number of all relevant samples. While the confusion matrix is a table that permits the performance evaluation of an algorithm. Each row represents the instances in a predicted class, while each column represents the instances in an effective class. I've decided to use this kind of metrics because in the dataset there was a problem of class imbalance. I've also used both SmallerVGGNet than VGG16.

Algorithms and Performance

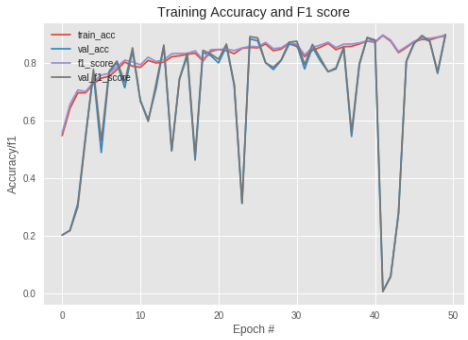
Both SmallerVGGNet than VGG16 were trained with epochs= 50, a batch size = 32, a training set about 3700 samples and a test set about 900 samples. I've also used the Adam Optimizer. The experiment showed that trained the CNN's with the same number of samples, the accuracy of the SmallerVGGNet is resulted greater than the accuracy of VGG16, 0.89 and 0.85 respectively, while the F1 score of the SmallerVGGNet is resulted lower than that of VGG16, 0.84 and 0.87 respectively. Also the confusion matrices are resulted similar.

The details of evaluation are shown below.

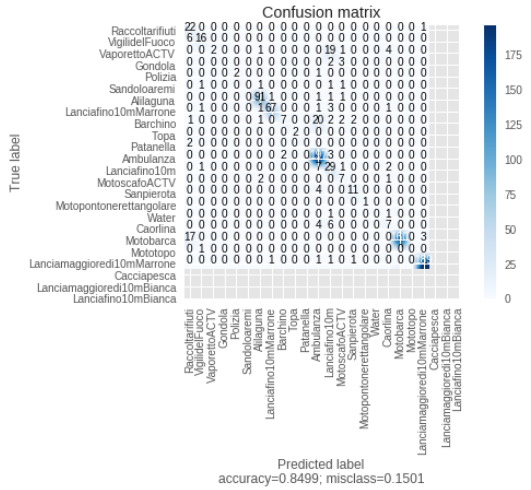
Algorithm	Accuracy	F1 score
SmallerVGGNet	0.89%	0.84%
VGG16	0.85%	0.87%



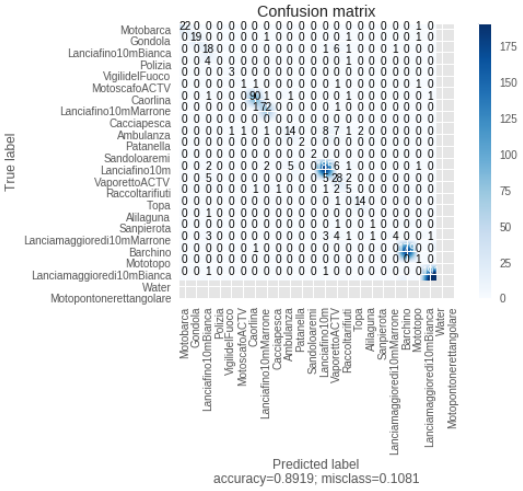
(a) Training with VGG16



(b) Training with SmallerVGGNet



(a) Confusion matrix with VGG16



(b) Confusion matrix with SmallerVGGNet

Family Classification

In the last one part of the homework, it was necessary to create an application that can classify a given boat image which family it belongs to. Also for this application, has been used the same dataset of Boat Classification and the same metrics too. The first

problem that I've encountered, has been that the sc5 dataset did not have the division into family classes, so I had to create a script to do that. The result it was a division into 6 classes:

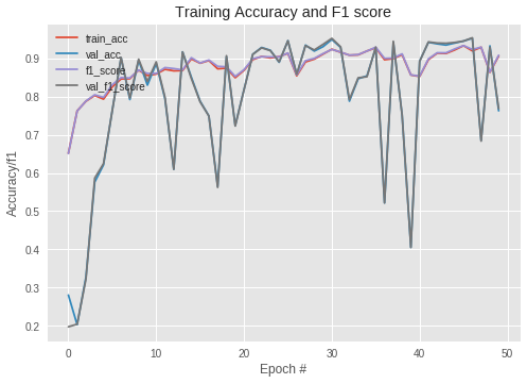
1. Falsepositive
2. Generaltransport
3. Peopletransport
4. Pleasurecraft
5. Publicutility
6. Rowingtransport

Figure 6: Family classification classes

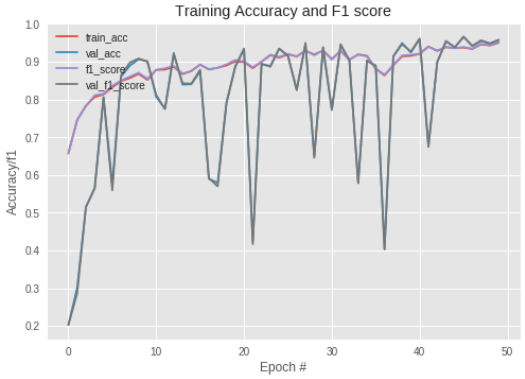
Algorithms and Performance

Also for this classification, I've used both SmallerVGGNet than VGG16, with the same parameters and the same optimizer of Boat Classification. In this case, the experiment showed an interesting result. The classification accuracy and the F1 with the SmallerVGGNet are results much greater than VGG16. In fact, SmallerVGGNet'accuracy was 0.96 against that of VGGNet equals 0.762. The same result is was obtained with the F1 score, with a score of SmallerVGGNet equals to 0.952 against that of VGGNet equals to 0.81. Moreover, another interesting result has been that, although the scores are very different, the confusion matrices and the plots for both CNN's are very similar. The details of the evaluation are shown below.

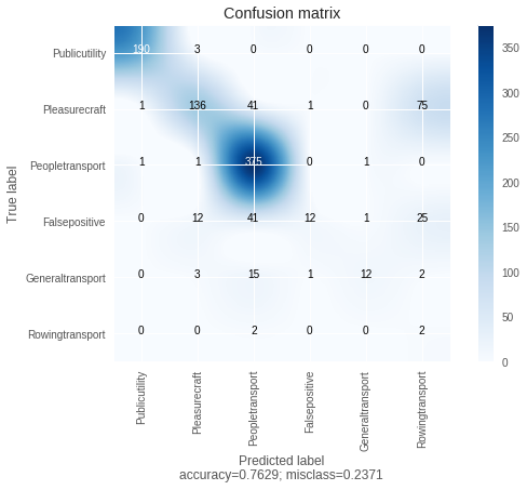
Algorithm	Accuracy	F1 score
SmallerVGGNet	0.96%	0.952%
VGG16	0.762%	0.81%



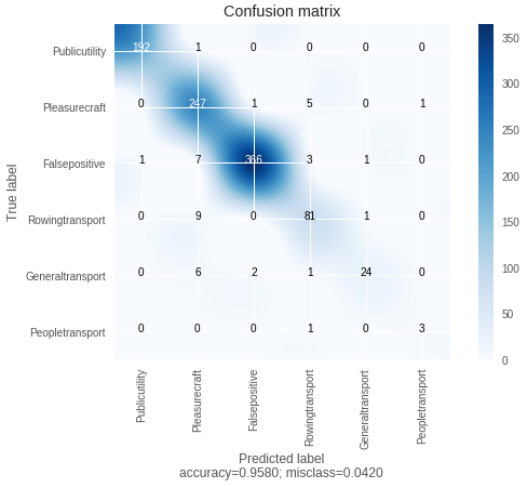
(a) Training with VGG16



(b) Training with SmallerVGGNet



(a) Confusion matrix with VGG16



(b) Confusion matrix with SmallerVGGNet

Conclusions

Results

For the first part of the homework - i.e. Boat Classification - the results are very similar with both the CNN's, a little bit better the training with the VGG16 networks in term of F1 score, but I think that changing a little bit some parameters, the result may be equals. The very interesting data is for the second part of the homework - i.e. Family Classification. In this case, the experiment showed that the SmallerVGGNet is the best result, both in terms of accuracy and in terms of F1 score. So in conclusion, with a low number of classes to classify the best CNN to use is SmallerVGGNet, while with a high number the VGG16.

Final considerations

This homework was very useful for understanding some techniques of machine learning, in particular the Convolutional neural networks and their functionalities.

Bibliography

[BIPT15] Domenico D. Bloisi, Luca Iocchi, Andrea Pennisi, and Luigi Tombolini. ARGOS-Venice boat classification. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6, 2015.

[KS] A. Zisserman K. Simonyan. Very deep convolutional networks for large-scale image recognition.

[uoR] DIAG Sapienza university of Rome. Mardct.

[wik] Random forest.