Vision Team Project Proposal : What the hell is that ?

November 3, 2017

1 Background

With the development of automatic transport means (trains, cars, boats, ...), it is necessary to identify the elements detected by the cameras in order to act accordingly. Indeed, the device will not act in the same way if it detects a frog or if it detects a horse for example. The idea of this project is to be able to detect entities that can be found when traveling by rail, sea or road. The recognition of these entities is therefore essential if we want to achieve intelligent and safe autonomous transport.

Labels	Train	Test	Validation	Total
airplane	4000	1000	1000	6000
automobile	4000	1000	1000	6000
bird	4000	1000	1000	6000
cat	4000	1000	1000	6000
deer	4000	1000	1000	6000
dog	4000	1000	1000	6000
frog	4000	1000	1000	6000
horse	4000	1000	1000	6000
ship	4000	1000	1000	6000
truck	4000	1000	1000	6000

Table 1: *Distribution of each set : train, validation, set*

2 Material and Methods

2.1 Data description

The dataset used for this project is CIFAR10 [**1**](Figure **??**) which is composed of 60 000 images of size 32 x 32 representing 10 labels of animals and means of transport. The datasets is divided as follows: 40 000 for the training step, 10 000 for the testing step and 10 000 for the validation step. Each set is balance distributed (Table **??**).

2.2 Preprocessing

Before classifying the different images of our dataset, we need to preprocess the images, extracting the features, which will permits us to classify them more precisely. In fact, classifying directly the images will gives us bad results (Table **??**), which is why extract features are necessarily. Different features are used like HoG. But we decide to use convolutional neural network used in Deep Learning [**2**]. We use the MobileNet [**3**] architecture of Google on the Keras Framework, and we added some layers so as to reduce the size of the features to 256. We do not want to take features bigger than 256 because the computation time will increase for the different algorithm of classification. We do not want to reduce too much the features because we think that we might lose some potential features which can be useful for the classification task. Regarding the architecture, MobileNet is a good feature extractor because it is a fast and soft model (so our personal computer is powerful enough) with a good accuracy on ImageNet. This is why we use it. We save 256 features for each image in CIFAR10 in the AutoML format.

2.3 Method

We consider our problem as a multi label classification therefore each image is associated to a label value between 0 and 9 included. The final step is to use a machine learning algorithm(using skicit learn for example) so as to be able to predict the class of an image. There are a lot of machine learning algorithm permitting to solve this problem so student have to classify the features with different models and choose the best they obtain. (Naives bayes, SVM, tree, . . .). If the student is familiar with the keras framework, he/she can also classify features with a simple neuronal network ! The set of data (features) have 256 dimensions and

Model	Accuracy	Recall	Precision	F1
Tree	0.22	0.22	0.22	0.22
Random Forest	0.21	0.20	0.21	0.15
Naives Bayes Gaussian	**0.27**	**0.28**	**0.27**	**0.25**
Naives Bayes Multinomial	0.24	0.25	0.24	0.22

Table 2: *Classification's result without extraction features*

can be subject to a reduction of dimension in order to limit the training time of the models and to limit the difficulty associated to the Curse of Dimensionality. As we can see on the figure **??**, using principal component analysis (PCA), it is possible to lessen the number of features (about 73% is conserved keeping only 10 dimensions, and about 80% is conserved keeping only 20 dimensions). The reduction of dimension is an important pre-processing step and even if it is not essential to get results on this dataset, it permits to save learning time on model which is non-negligible.

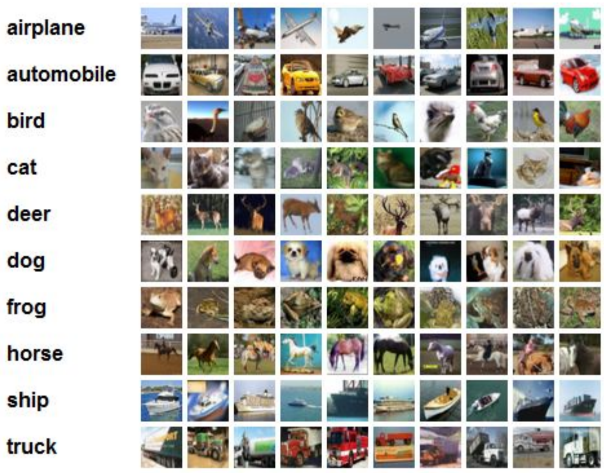

Figure 1: *Representation of the dataset Cifar10*

3 Preliminary results

In this experiment, we tested differents models and compared their performance (Table **??**). We note that SVMs models are very efficient on the features extracted by the convolutional neural network but it takes some time to construct the model contrary to the others models like tree or random forest which have lowest performances but they are faster to learn. However, the computing time of the prediction is fast for all the models.

Model	Accuracy	Recall	Precision	F1	Time (sec)
Tree	0.797	0.798	0.797	0.797	16.73
Random Forest	0.831	0.829	0.831	0.830	33.42
Naives Bayes Gaussian	0.853	0.856	0.853	0.854	0.198
Naives Bayes Multinomial	0.866	0.869	0.866	0.867	**0.06**
SVM rbf	**0.888**	**0.888**	**0.888**	**0.888**	87.49
SVM Linear	0.871	0.872	0.871	0.872	173.15
SVM Poly	0.886	0.887	0.886	0.886	66.24

Table 3: *Classification's result with extraction features on the test set*

The results associated with the learning on the data processed by PCA are available on the figure **??**. As we can see, performances are very similar to learning on the initial features (see table **??** and figure **??**) when the dimension is superior to 10. The use of PCA (or other method of reducing dimension) is left to the student's sensitivity because this stage of pre-processing is a matter of good understanding about how to learn a dataset. However this step is not mandatory considering the potential lacking experience of the student.

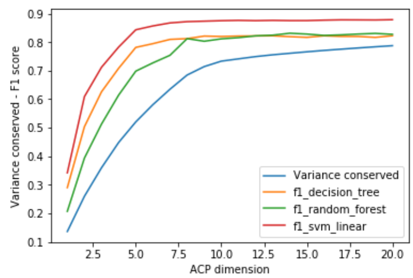

Figure 2: *Graph representing F1 score depending of the number of dimension*

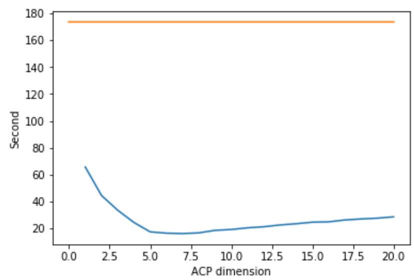

Figure 3: *Graph representing the training time depending of the number of dimension. In blue, the training time of the SVM linear model depending of the number of features, in orange the training time of the SVM linear on the original features(256)*

References

[1] https://www.cs.toronto.edu/ kriz/cifar.html

[2] "CNN Features off-the-shelf: an Astounding Baseline for Recognition", Ali Sharif Razavian et al, 2014.

[3] "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" ,Andrew G. Howard, Menglong Zhu et al, 2017.