

# Research project – Master HCID

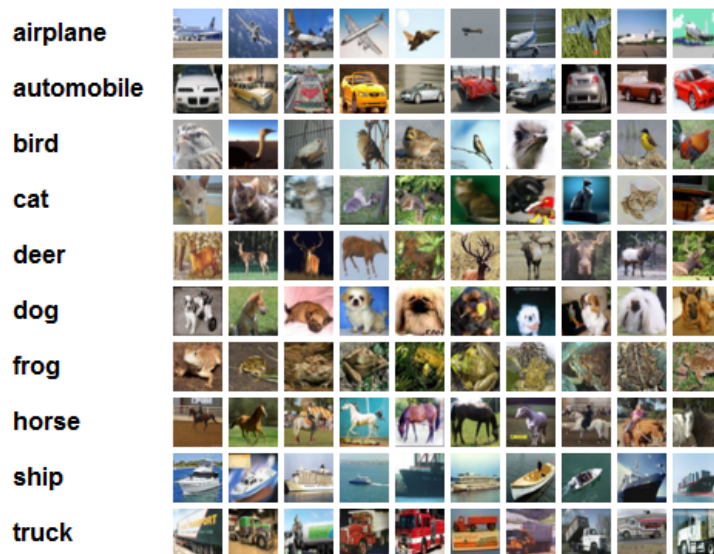
## Object recognition

Alexandre ALLAUZEN

Nicolas PÉCHEUX

March 22, 2015

As the final assignment we take an interest in a more challenging pattern recognition task. This is a small research project, so you have complete freedom to focus on the aspects you prefer, add new ideas or read further scientific publications. Be sure to fully explain your approach in your final report and by viva (guidelines, date, place and time to be announced). You may work by pairs and submit joint work.



## 1 Overview

The goal of this project is to understand how we can make use of the methods seen in class (K-means, perceptron, preprocessing) to deal with a new challenging task. We focus in an object recognition task – CIFAR-10 – a large dataset

of images representing 10 different kind of objects. The main idea is to use the K-means algorithm to define the features of an image, and to train a perceptron on those features, as suggested in [Coates et al., 2011].

## 2 CIFAR-10

CIFAR-10 is a dataset made of labeled tiny images representing objects such as airplanes, dogs, trucks, etc. It is freely available at

<http://www.cs.toronto.edu/~kriz/cifar.html>,

where you can directly find a `pickle` python format and instructions. You should first spend some time loading the dataset, understanding the underlying structure and taking a look at the images. In your report, you may describe briefly the dataset and explain why this task could be interesting. Do you think it is challenging enough? What error rate would you expect for a human been? For a computer program?

You can find on the internet many baselines and results for this data set. Recall we are only using simple learning algorithms, while most research papers use advanced neural network based models. This, do not panic if you don't reach such amazing accuracies!

You do not have to use the whole dataset at first. You may concentrate only on a subset either by restricting the total number of images (the training data are already split in batch, you can start with one of them) or the number of classes (starting with binary classification might be a good idea).

## 3 First steps

As a first step you may try to apply the perceptron algorithm to the dataset, as you did for MNIST digit recognition task, using the pixels of the images as features. How to deal with the fact the images are not grayscale anymore ? What accuracies do you reach? What are the problems/successes this method encounters? Note that the accuracies should be computed on the test data.

You can also apply K-means clustering on the images as in the labs. How do you define the distance here? How does it works? Can you use this clustering in a classification purpose? If yes, what accuracy is reached?

## 4 When scientific literature comes to help

Take a look at [Coates et al., 2011] which can be found here

[http://www.stanford.edu/~acoates/papers/coatesleeng\\_aistats\\_2011.pdf](http://www.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf)

You don't need to fully understand the abstract nor the introduction, but read carefully Section 3 and take a look at the experiments. We only focus on K-means unsupervised feature learning, so you can entirely skip *Sparse auto-encoder*, *Sparse restricted Boltzmann machine* and *Gaussian Mixtures*. It is a good idea to read it many times and to understand in details Figure 1. You may include your understanding of the article in your report.

The main idea is to derive new features for an image instead of using directly the raw pixels<sup>1</sup>. To do so, we first extract many (many) patches from data which are small part of an image (e.g. by chance it could be a cat's ear, a dog's eye, etc.) and train K-means clusters with many clusters (*i.e* 50 or 100) on this patches. Now, for training or prediction, we can use a different representation of an image as the standard bag of pixels representation. We can first extract patches from the image and for each patch find the closest cluster. The patch can be then represented by its affectation to one of the cluster. The image is at the end represented by the concatenation of the patch representations. Let's take an example where we consider that an image can be divided in 4 patches:

- Extract all the patches from the training data and then estimate the K-means
- To use the perceptron learning algorithm on one image:
  - extract the four patches
  - compute the distance to the K clusters for each patch
  - assign each patch to one cluster: a binary vector of K components contains one bit to zero, that corresponds to the closest cluster to the patch
  - create the image representation by concatenating the 4 binary vectors that correspond to the 4 patches

## 5 Unsupervised feature learning

Implement some of the ideas you read in the article. You can try **K-means (hard)** and if you have time **K-means (triangle)**. Explain your methodology, your results, the parameters you choose, etc.

**Remark** [Coates et al., 2011] use SVM with L2 regularization to train the linear classifier to predict the labels given the feature vectors, where you might want to use your perceptron implementation instead.

## 6 Move further on...

If you want to and if you have time, you can experiment any new idea you have or you found in the literature to improve the performances on this task. In

---

<sup>1</sup>Why do you think raw pixels are not good cues to train a classifier for this task?

particular you can look at preprocessing steps. Recall we do not evaluate your final results but the methodology and your analysis.

## 7 Implementation details

Be sure to read again the *numpy* tutorial and to take a look at our solutions to K-means and the perceptron algorithm (provided in the course website). Especially if your implementation is slow.

Additionally you can take a look at `scikit-learn`, a very nice machine learning python module

<http://scikit-learn.org/stable/index.html>

This package provides efficient implementations of K-means, the perceptron algorithm, and many other. You can then compare your own implementation results and/or switch to the provided modules when dealing with large conditions (e.g. if you want to make use of the whole dataset). The scikits documentation is very clear and usefull and many examples are provided.

## References

[Coates et al., 2011] Coates, A., Lee, H., and Ng, A. Y. (2011). An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*.