



SOFTWARE ATELIER: SIMULATION, DATA SCIENCE & SUPERCOMPUTING

SPRING 2019

---

# A High Performance Video Segmentation Framework

---

<b>Liudmila Karagyaur</b>	karagl@usi.ch
<b>Lorenzo Ferri</b>	ferril@usi.ch
<b>Vanessa Braglia</b>	braglv@usi.ch

March 12, 2019

## Introduction

Nowadays edge detection techniques have several applications. For example, a content based video retrieval is widely required for searching digital information in large databases, in order to improve text based retrieval systems [1]. It also has an important role in the development of self-driving vehicles, that need to perform a real-time feature detection [2], and in medical field, in the analysis of digital images of pathological conditions, such as tumors [3].

The goal of this project is to find the main features of a video and to be able to follow their trajectories. In particular, we will analyse single video frames using edge detection, which is an image segmentation technique.

There exists a great variety of clustering algorithms that perform efficient image segmentation, which is a fundamental step in feature extraction. The first step consists in analysing different clustering algorithms and detect the most suitable one to the problem at hand.

Since the frames could be noisy or blurry, the next stage will focus on possible ways to improve the result of the clustering algorithm, by looking for an optimal sharpening operator. In this step some deblurring techniques will be examined.

Finally, the edge detection will be executed in parallel on each frame of the video, in order to improve the performance. The final result will be a video that will only contain its critical features.

# 1 Project tasks

## 1.1 Image clustering

Clustering is one of the most widely used techniques for exploratory data analysis, with applications ranging from statistics, computer science, biology to image segmentation. In the latter case, it aims at extracting meaningful objects lying on the image. The cluster analysis is to partition an image data set into a number of disjoint groups or clusters. The algorithms that will be examined are k-means, DBSCAN and spectral partitioning.

### 1.1.1 k-means

The k-means algorithm is the simplest and most popular clustering algorithm. Its ultimate goal is to partition  $n$  initial datapoints into  $k$  clusters. It requires that the user provides the number  $k$  of clusters that have to be found. The algorithm uses an iterative approach: starting from an initial set of random centroids, at each iteration it generates the clusters by minimizing the Euclidean distance  $D$  between each point  $x_i$  and the centroids  $c_j$ :

$$\arg \min_j D(x_i, c_j) \quad j = 1, \dots, k$$

Then, for each cluster  $C_j$  a new centroid  $c_j$  is computed as the mean of all  $n_j$  points  $x_i$  assigned to the cluster in the previous step:

$$c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

Convergence is achieved when points are no longer interchanged between the clusters.

### 1.1.2 DBSCAN

Density-Based Spatial Clustering of Applications with Noise algorithm needs two parameters in order to perform clustering on data: the maximum radius (*eps*) of the neighbourhood and the

minimum number of points ( $MinPts$ ) in the  $eps$ -neighbourhood of a point. All the points are classified as:

- **core point:** if it has equal or more than  $MinPts$  points in its  $eps$ -neighbourhood
- **border point:** if it has less than  $MinPts$  points in its  $eps$ -neighbourhood, but it is in the neighbourhood of a core point
- **noise point:** any point that is not a core or a border point.

Then the datapoints are assigned to clusters in the following way. Any two core points that belong to each other's neighbourhood (located within a distance  $eps$ ) are assigned to the same cluster; any border point that belongs to the  $eps$ -neighbourhood of a core point is assigned to the same cluster of the core point. Noise points are discarded.

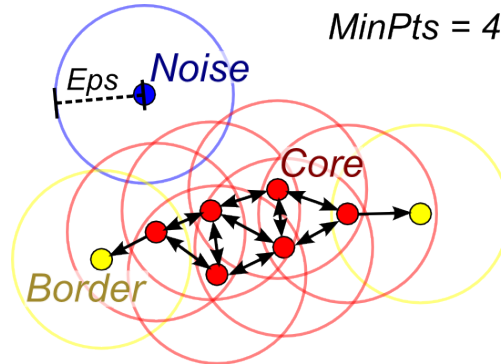


Figure 1: An illustration of DBSCAN algorithm ([www.stats.stackexchange.com](http://www.stats.stackexchange.com))

### 1.1.3 Spectral Partitioning

The spectral clustering makes use of the spectrum (eigenvalues) of the Laplacian matrix  $L$  of the data, that specify all the connections between the image pixels. The eigenvector corresponding to the first eigenvalue  $\lambda_1 = 0$  is the vector of all ones and it does not provide information about the graph structure. The second lowest eigenvector, instead, called the “Fiedler eigenvector,” is used by spectral partitioning to divide into sub graphs. The pixels that correspond to a negative value

in the eigenvector will be in one group and the ones corresponding to a positive value in the other. This clustering algorithm, unlike the ones outlined above, relies on global information of the image, that is encoded in the Laplacian matrix.

## 1.2 Image sharpening

After the selection of the clustering technique, the video frames will be sharpened with different operators and the most suitable one will be chosen. The sharpening process consists in solving a linear system

$$Ax = b$$

where  $b$  is the blurred image,  $x$  is the clear image that we want to obtain and  $A$  represents the sharpening operator we want to apply. For the solution of the linear system different algorithms will be used (e.g. steepest descent, conjugate gradient). The final goal of this step is to obtain a sharpening operator that will allow to increase the quality of the image clusters, leading to a better feature extraction.

## 1.3 Machine learning

Now that we have clear images, we need to extract the feature of the image that we are interested in. In order to do that we are going to use a convolutional neural network that is going to label all of our clusters, so that we know what we need to process next.

A convolutional neural network is a “deep neural network”, since it’s composed by different layers. The key difference of a CNN in respect to a standard neural network, is the presence of a “convolutional” layer, that scan over the image using a filter. Analysing the image like this will keep the logical connection between the pixel, since each pixel is scanned multiple times when the filter pass over it.

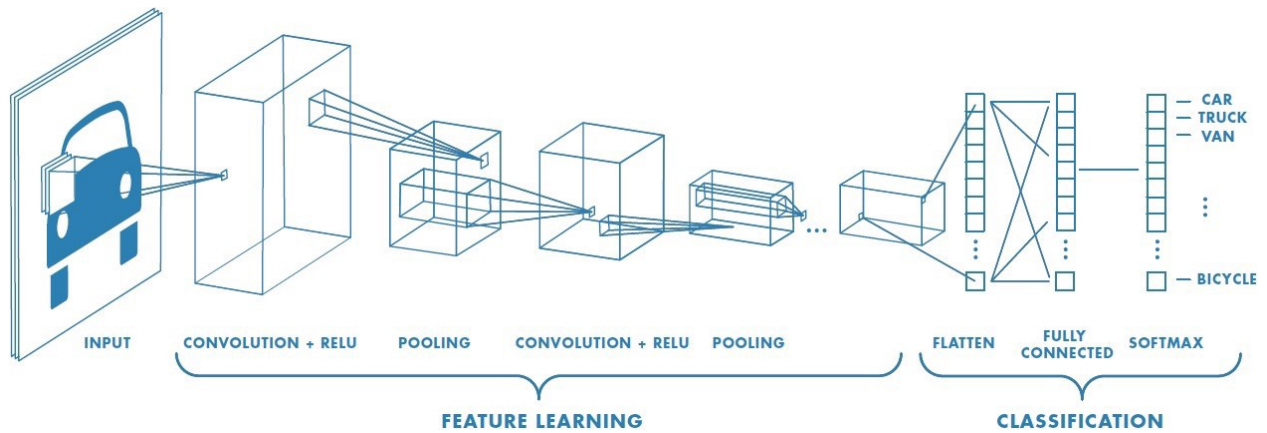
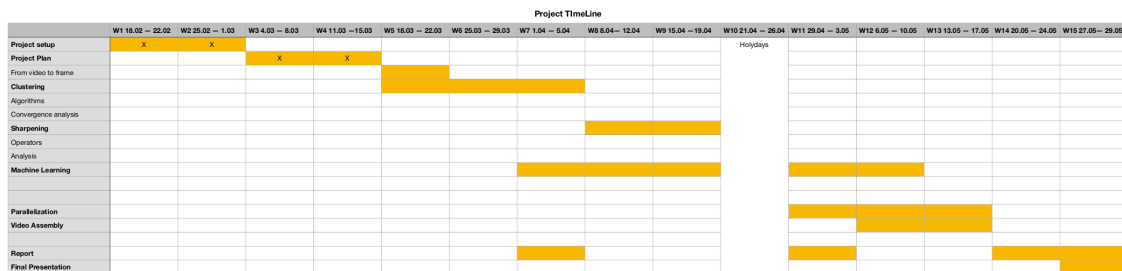


Figure 2: An example of a CNN structure (<https://towardsdatascience.com>)

## 1.4 Parallelization

The last step consists in the parallelization of the program: all frames of the video will be processed simultaneously to improve the performance.

## 2 Project Time Line



## References

- [1] B.V. Patel and B.B. Meshram, *Content Based Retrieval Systems*, International Journal of Ubi-Comp (IJU), Vol.3, No.2, April 2012
- [2] H. Cho, Y. Seo, B. V. K. V. Kumar and R. R. Rajkumar, *A multi-sensor fusion system for moving object detection and tracking in urban driving environments*, IEEE International Conference on Robotics and Automation (ICRA), 2014
- [3] Ed-Edily Mohd, Azhari,Muhd. Mudzakkir Mohd. Hatta<sup>1</sup>, Zaw Zaw Htike<sup>1</sup>andShoon Lei Win, *Tumor detection in medical imaging: a survey* , International Journal of Advanced Information Technology (IJAIT) Vol. 4, No. 1, February 2014