

MACHINE LEARNING ENGINEER NANODEGREE

CAPSTONE PROPOSAL

Juan José Madrigal Martínez
January 20, 2017

Madrid, Spain
(+0034) 600 86 32 48
juanjomadrigal326@gmail.com
[LinkedIn](#) [GitHub](#) [Kaggle](#) [Udacity](#)

DOMAIN BACKGROUND

This project aims at building a video analysis system for surveillance purposes. It basically finds and indexes the movement events filmed by a fixed camera.

Video surveillance has become a major and widely used tool for multiple issues [1] and is supported by many companies [2] [3]. But the huge amount of information which is usually dealt with has led to the need to use Machine Learning techniques to extract patterns, predictions and other refined information. This approach is being implemented [4] and there is much work for Machine Learning engineers to do in this field.

An efficient implementation of Machine Learning techniques to video surveillance would prevent users from dealing with a sequential (and manual) search through the (perhaps many hours long) video source, which is rather inefficient, boring and error prone, thus providing a major tool for a wide range of purposes.

[1] [Wikipedia - Surveillance / Cameras](#)

[2] [VideoSurveillance.com](#)

[3] [Tyco](#)

[4] [Briefcam](#)

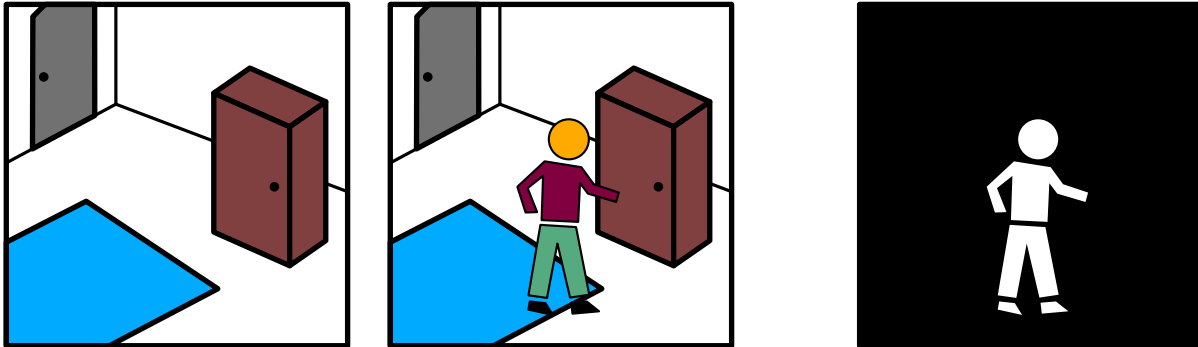
PROBLEM STATEMENT

To apply unsupervised Machine Learning clustering algorithms to detect and quantify the movement events filmed by a fixed camera. The algorithms are to be applied to a tridimensional binary array obtained from the original video after some careful video-preprocessing (see below). Further refinements may also be tackled, such as finding time-parametric curves accurately fitting each movement event.

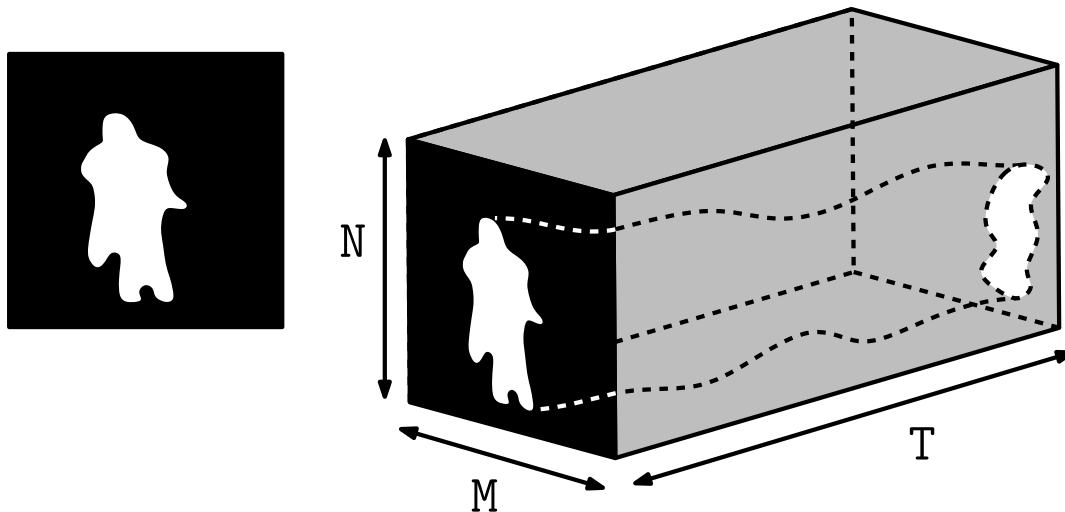
DATASETS AND INPUTS

The datasets and inputs for this project will be obtained after some video preprocessing made on sample videos. The selected videos are located in [capstone_project/data/unprocessed_videos](#) and they have been downloaded from [VIRAT Video Dataset](#) [1]. The video preprocessing is carried out using the free packages [SciKitImage](#) and [SciKitVideo](#) and has been generously provided by [WardenAutomation](#).

The video preprocessing takes as input a .avi $M \times N$ pixels video and a .png $M \times N$ pixels picture (*background*). For each frame in the video, the frame and the background are compared pixel-wise, and the difference is recorded in a $M \times N$ binary array, where 1 denotes difference (above some threshold) and 0 denotes no difference.



If the video has T frames, then the video preprocessing gives as output a $M \times N \times T$ binary array, which is supposed to capture the movement in our scene and that will be the primary input for the project. These arrays are located in [capstone_project/data/tridimensional_binary_arrays](https://github.com/rodrigo1994/capstone_project/tree/main/data/tridimensional_binary_arrays).



[1] **A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video**, Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J.K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xiaoyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai, *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

SOLUTION STATEMENT

We propose the use of [DBSCAN clustering algorithm](#), which is implemented in [SciKitLearn](#). This clustering algorithm is well-suited for our generic clusters, which are uneven in size and non-convex.

BENCHMARK MODEL

One may refer to [Actions as Space-Time Shapes](#) [1] for a remarkable study in this field.

[1] <http://www.wisdom.weizmann.ac.il/vision/SpaceTimeActions.html>

EVALUATION METRICS

The metric(s) used to estimate the quality of our clustering will comprise

- Homogeneity
- Completeness
- V-measure
- Adjusted Rand Index
- Adjusted Mutual Information
- Silhouette Coefficient

The weight given to each measure will be assigned by refined cross-validation process.

PROJECT DESIGN

A theoretical workflow for approaching a solution would include

1. To apply the video preprocessing techniques to the selected videos to get the tridimensional binary arrays, which are the main input for our Machine Learning work.
2. To apply [DBSCAN clustering algorithm](#) to each array for a first contact with the potential results.
3. To compare the previous results with a manual detection and indexing for a small set of training videos. This may lead to a major refinement of the main algorithms. Together with this manual checking, the metrics mentioned above will be used and compared with the latter, thus finding which metrics work best for our video surveillance context.
4. To apply the refined algorithms and selected metrics to a large variety of videos. Further tuning may be implemented.
5. To provide code for spatial visualisation of all the previous processes.
6. From all the extracted data, to make conclusions about the success of our study and devise new ways or fields for further work.