AA PROJECT - #1

Image Segmentation for Object Detection and Target Tracking using the ATLASCAR

Bruno Mendes 68411 Pedro Silva 72645

Professor Petia Georgieva

MIECT

Deti/Dem/Ieeta - UA

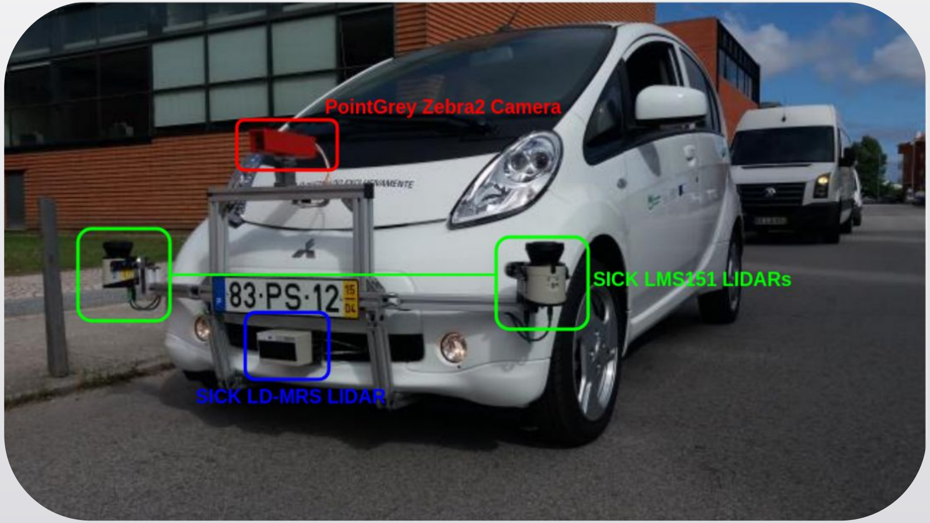
23/04/2019

Scope

- Master's thesis based on developing (using ROS) a driving assistance system that implements an interface to detect, track and label targets present on the road.

- Detection
- Tracking
- Labelling

$\Bigg\}$ LIDAR Sensors + PointGrey Camera



Motivation

- Other similar projects make use of **machine learning**.

  ...but this takes a lot of **image data** and a lot of **time**.

Can we apply **machine learning** then?

- **Machine learning** uses **image data** stored in **datasets**.

  ...but what is the **origin** of these datasets?

Can we create our **own datasets**?

- This **image data** is processed by an **image algorithm**.

  ...but which **image algorithm** do we use?

Is **Template Matching** the right algorithm to use?

- We need to assign **labels** to the **objects models** present in the **dataset**

  ...but what **classification method** do we use?

What is the **best** classification method?

Objectives

- The main **objective** of this project is to develop a system that can perform **image segmentation** on the datasets from the **ATLASCAR** and can be used for:
  - **Camera** Calibration.
  - **Image** Filtering.
  - Semi-Automatic **Detection.**
  - Object **Labelling.**
  - Object **Tracking.**

- Define an **"interest zone"** for a target object in an image.

- Create **new datasets** for learning.

Dataset Tool

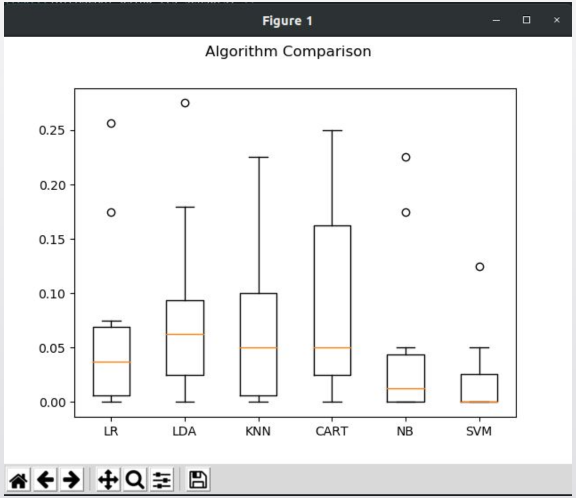
- The tools for loading and storing information, used in this project were developed based on **datasets** created in **previous experiments** with the **ATLASCAR** using its sensors.
- 3 **experiments** ⟹ 3 **datasets**.
- Dataset 1 ⟹ 2589 images in .jpg format with 400x400 res.
- Dataset 2 ⟹ 128 images in .jpg format with 400x400 res.
- Dataset 3 ⟹ 699 images in .jpg format with 400x400 res.

TEMPLATES

Machine Learning

- What **machine learning algorithm** do we use?

- Logistic Regression
- Linear Discriminant Analysis.
- K Neighbors Classifier.
- Decision Tree Classifier.
- GaussianNB.
- SVC.



	Mean Accuracy	Standard Deviation Accuracy
LR(Logistic Regression)	0.065641	0.080721
LDA(Linear Discriminant Analysis)	0.082949	0.080421
KNN(K Neighbors Classifier)	0.065128	0.066305
CART(Decision Tree Classifier)	0.095577	0.088758
NB(GaussianNB)	0.050064	0.077439
SVM(SVC)	0.022564	0.037837

Knn-Means Algorithm

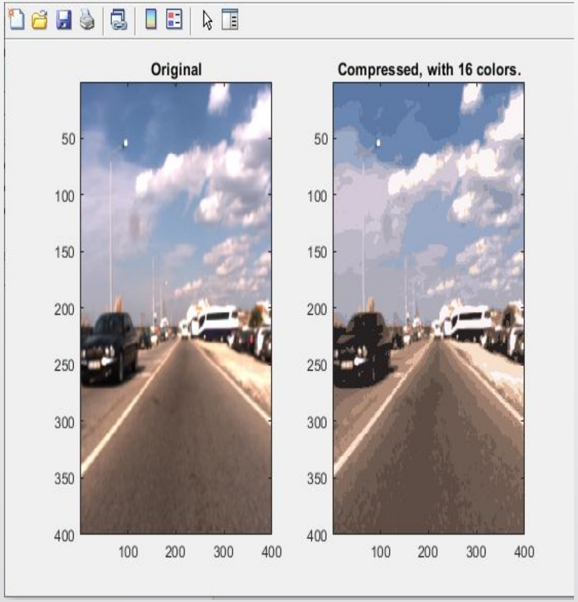
- The **k-nearest neighbors** algorithm is an **unsupervised machine learning** algorithm which is used when you only have input data (X) and no corresponding output variables and the ultimate goal of the algorithm is to model the **underlying structure** or **distribution** in the data in order to learn more about the data.

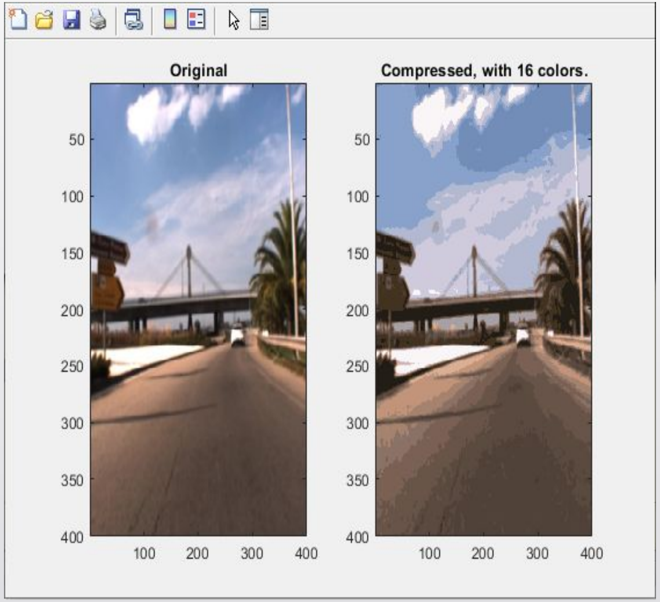
Input:
- K (number of clusters)
- Training set (no labels)

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$
Repeat {
Cluster assignment step => for i = 1 to m
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$
Move centroid step => for k = 1 to K
μ_k := average (mean) of points assigned to cluster k
}

- In the context of this project, the **k-nearest neighbors** will be used in order to perform **k-means clustering** on the pixels of the given frames of the dataset, performing a **color compression** on the pixels of the given frames and then map each pixel to its **closest centroid**.

Color Compression with 16 Colors for 10 iterations.





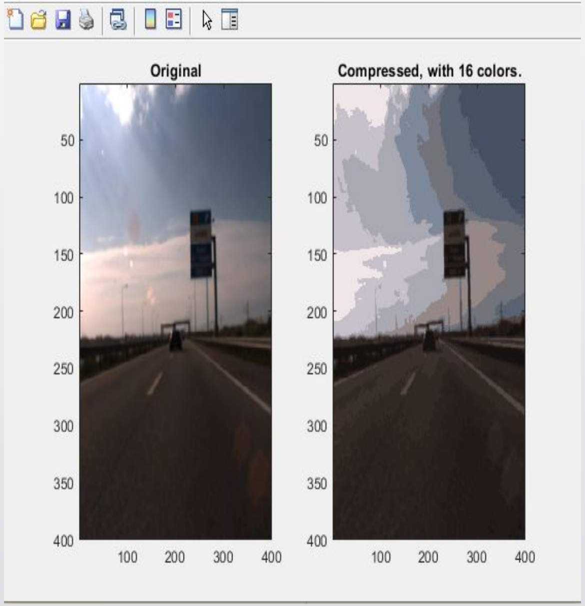


Image Segmentation

- After the **compression** process, we can proceed to the **image segmentation** using the **Template Matching** algorithm in a **spatial domain** by loading the **compressed image** as a **template image** and loading the **actual template** as the **target image**. After this we can construct the **correlation map** by finding the **correlation coefficient** between the images.

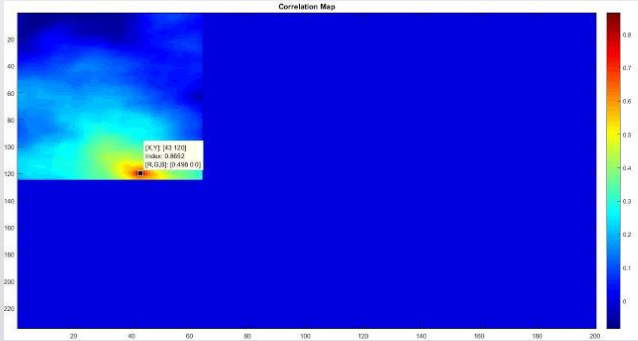
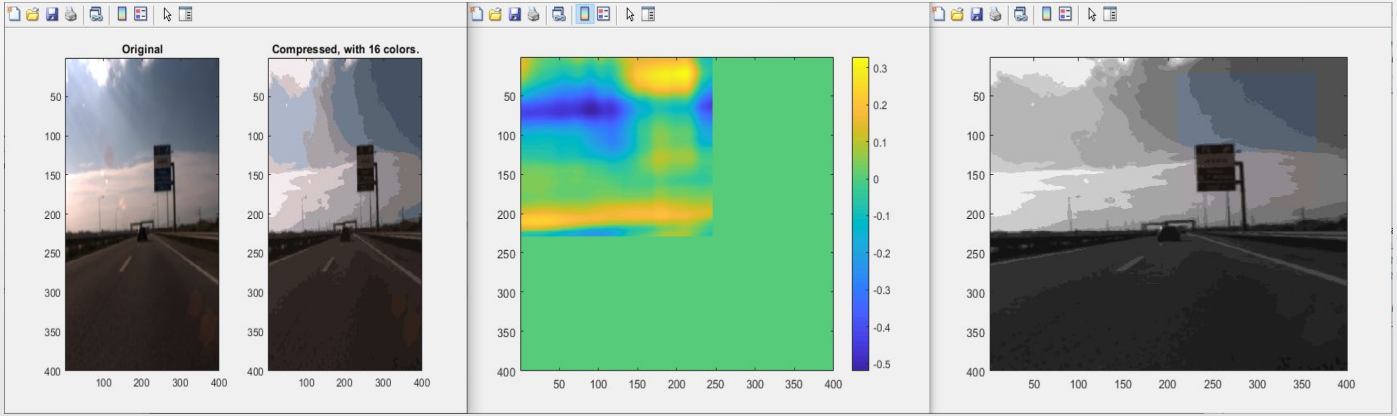


Image Results

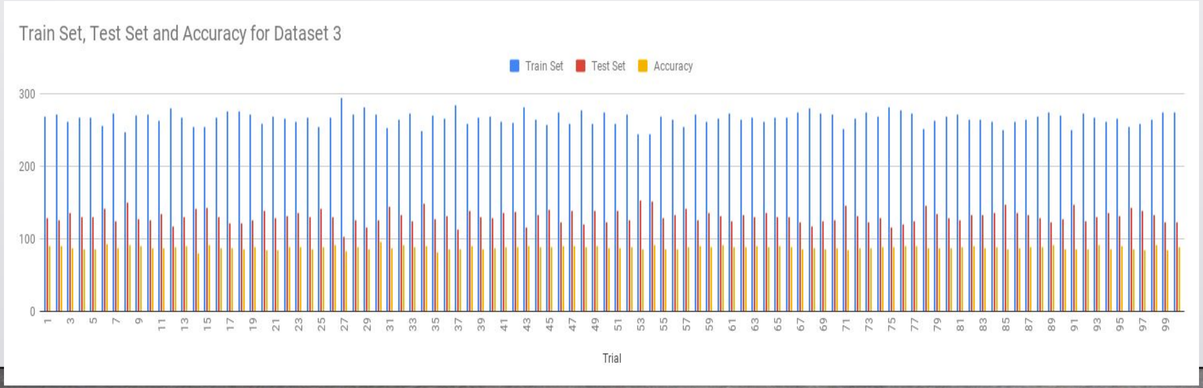
Template

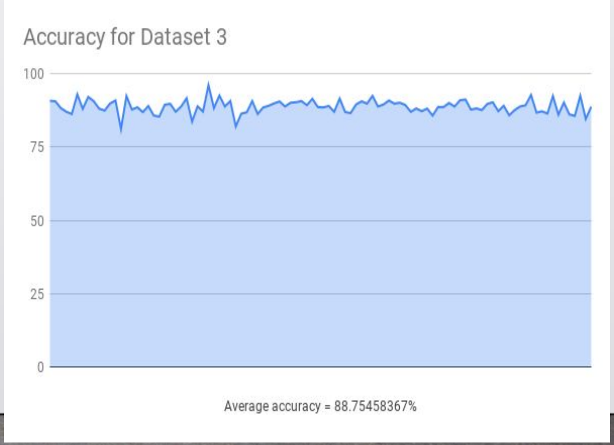


The **correlation map** defines the **“interest zone”** of the **template object**

Train, Test and Accuracy Results

- It’s possible to determine the **accuracy** results of this algorithm by first handling the data by opening the dataset from the **.csv** file and **split** the data randomly into **train** and **test** datasets(in this project a **standard ratio** of **67/33** was used to split the data into train and test datasets).





What's next?

- Improve the **Template Matching** algorithm and the **Correlation Map** construction.
- Define a **set** of **labels** for the **dataset**.
- Try some other **machine learning** type **algorithms**.
- Apply **project logistic** into the **ATLASCAR2** perspective.

Questions?