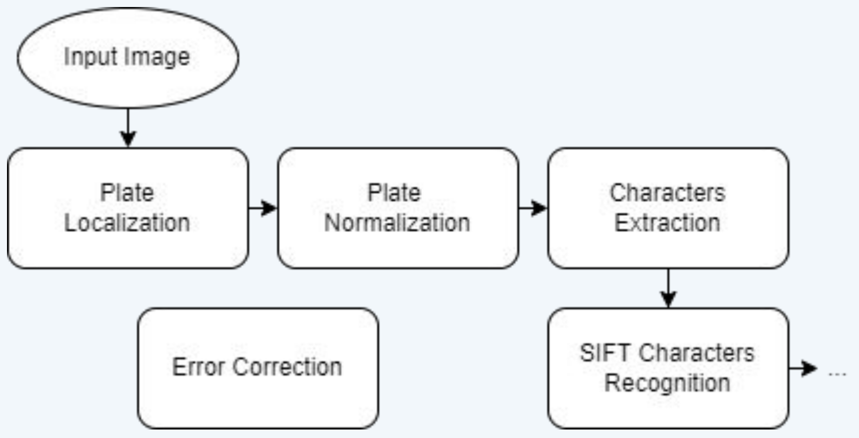
Cirillo Matteo **5616395**
Laporte Lilian **5615208**

Image Processing : Character Recognition

General Details

The main objective of this milestone is to finish off the normalization of the plates and recognize any vehicle's licence plate number.

We first finish off normalizing the localized plates by rotating them to a horizontal position and applying one last color segmentation.



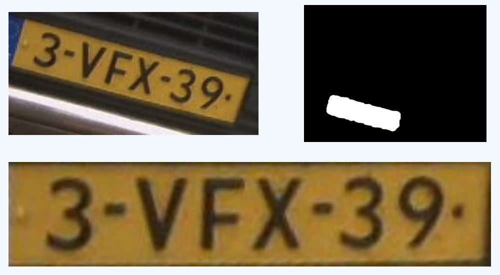
To extract the characters from the grayscale normalized plate, we apply an adaptive thresholding methods, which separates foreground and background into a binary image. That output, however often comes with some unwanted noise and we therefore need to filter out any region on the binary image that does not represent any license plate character

This last step is achieved by comparing bounding boxes' sizes and discarding any elements that are largely out of proportions with respect to the plate.

The last step consists in recognizing the characters sequence with a SIFT method.

Normalization

Once the plate is localized, we still need to rotate and segment it, in order to proceed to character extraction

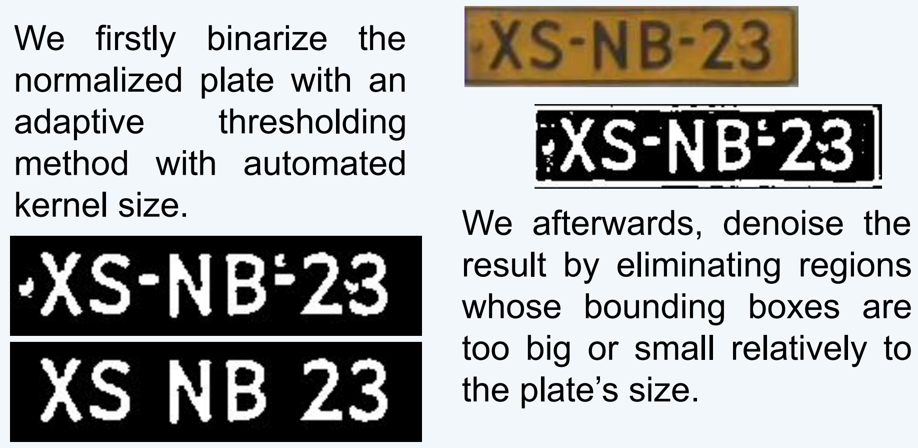


This is achieved bygetting the mask's bounding rectangle that minimises its area.

We indeed then simply get its orientation and compensate it by rotating the image and cropping it on last time to be left only with the plate itself.

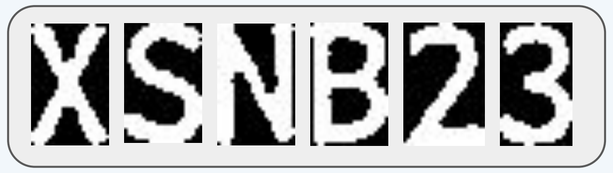
Characters Extraction

We firstly binarize the normalized plate with an adaptive thresholding method with automated kernel size.



We afterwards, denoise the result by eliminating regions whose bounding boxes are too big or small relatively to the plate's size.

The plate goes through a second denoising process (where we only keep the six regions whose areas are the largest), before cropping out each individual character by their respective bounding boxes.



SIFT Characters Recognition

We now apply a SIFT algorithm, which compares descriptors of the plate's characters against a database in order to find the best match using a L2 norm.

The descriptors are the Normalized Histograms of Oriented Gradients (HOG) of 16, 4x4 regions on the rescaled input image.

The database is created at the program's start and stores HOG descriptors for template characters which have the same font as the ones on a license plate.

Note that we crop and scale every characters to its 16x16 bounding box in order to yield the best results

Lastly, we add the dashes into the final recognized sequence, at the spots where the extracted characters had the largest vacant interval!

XSNB23

XS-NB-23

Numerical Evaluation Method

To evaluate the performance of our character recognition method, we simply compare the predicted characters with a ground truth, established by hand for every car and take the ratio.

score = correct / total

In order to avoid developing any bias towards a certain frame of each plate's appearance, we average the results over every possible frame.

Example:

56-JTTS instead of 56-JTT-5 yields 6 correct characters instead of 8

Note that we also implemented a 'overall plate recognition' performance based on the same principle.

Evaluation Results

Plate Recognition			Character Recognition		
score	Training set	Test set	score	Training set	Test set
μ	0,83	0,77	μ	0,92	0,87

Evaluated by split validation. (Tuned all the parameters on ⅔ of the design set's data, and evaluated it on ⅓ of it.)

Notice that while we do display performance on the training set, only the test set performance should be taken into account for unbiased error estimation. We would however want to point out that the amount of training data needed to build a satisfactory classifier only leaves little data for the test set; Which intern leads to a variable error estimate.

The majority of errors made during character recognition come from ambiguity between to similar characters, such as X and K, S and 5, 0 and O, 8 and B, etc..

To improve character recognition we further plan on implementing an error detection algorithm, based on the fact that any group of characters between dashes is either numerical or alphabetical.