1. Intro

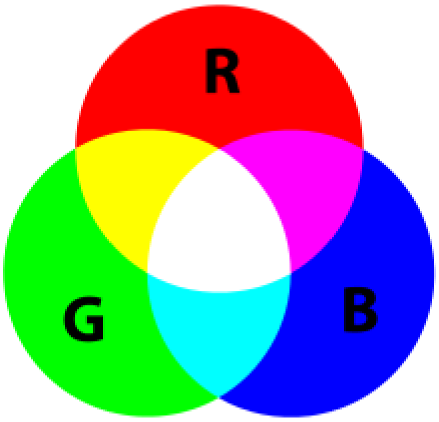
Detection of vegetation in images is a ordinary procedure in remote sensing and is commonly applied to satellite or aerial images. In recent times it has been applied to images recorded from within ground vehicles for navigation in outdoor situations. In this project we present a methods for roadside vegetation detection proposed for separating the flora from the rest of image. Our feature set contains of color features and texture features. One of our detailed goals was to find a useful texture feature set for the problem of flora detection.

2. First solution

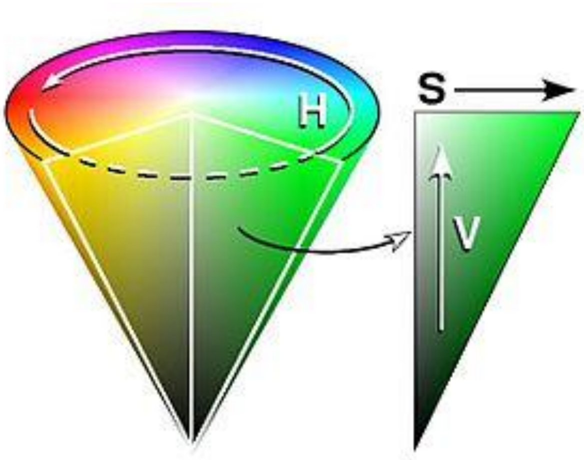
Perception of colours is very subjective and personal. To try to define numbers to the brains reaction is extremely difficult. The point of colour spaces is to assist the process of describing colour between people, machines and programs.

RGB color system, constructs all the colors from the combination of the Red, Green and Blue colors. Each of them use 8 bits, which have values from 0 to 255.

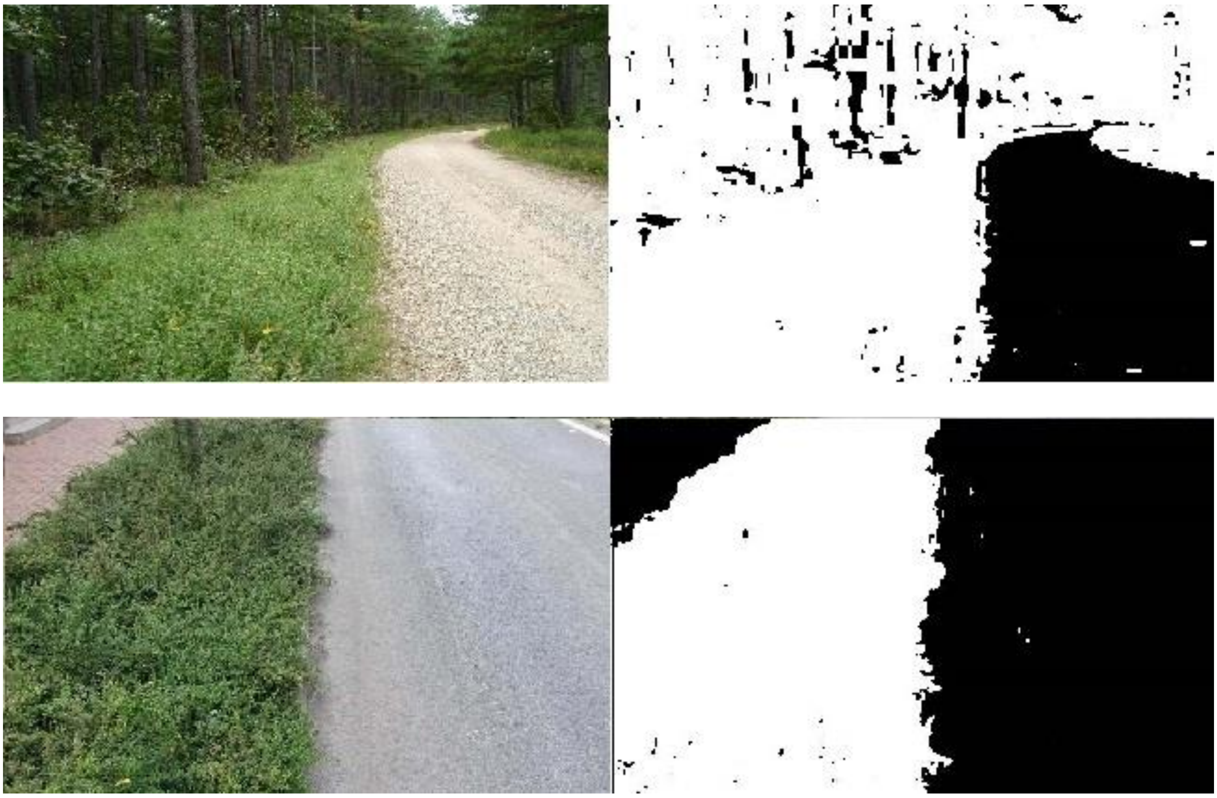
The highest determination of the RGB color model is for identifying representation, and display of images in electronic devices, for example televisions or computers. Before our ages, the RGB model already had a concrete theory behind, because of human perception of hues.



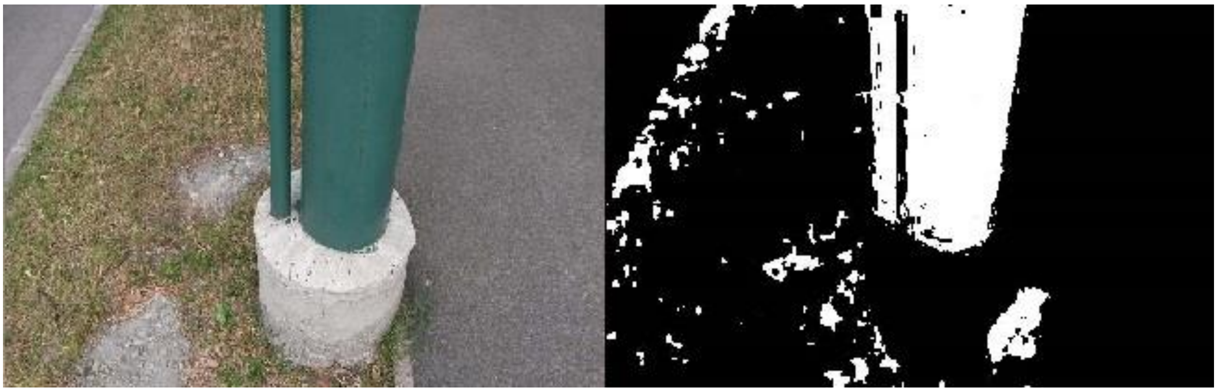
Other model of the colors representation is HSL and HSV. Both models are easy transformation of RGB models. The colors there depend of the basic components (red, green, blue primaries) and also of the gamma correction used to represent the amounts of those elements. Each RGB device has his own HSL representation to accompany it. Both of the models have advantages and disadvantages, but though this both are used extensively in computer graphics.

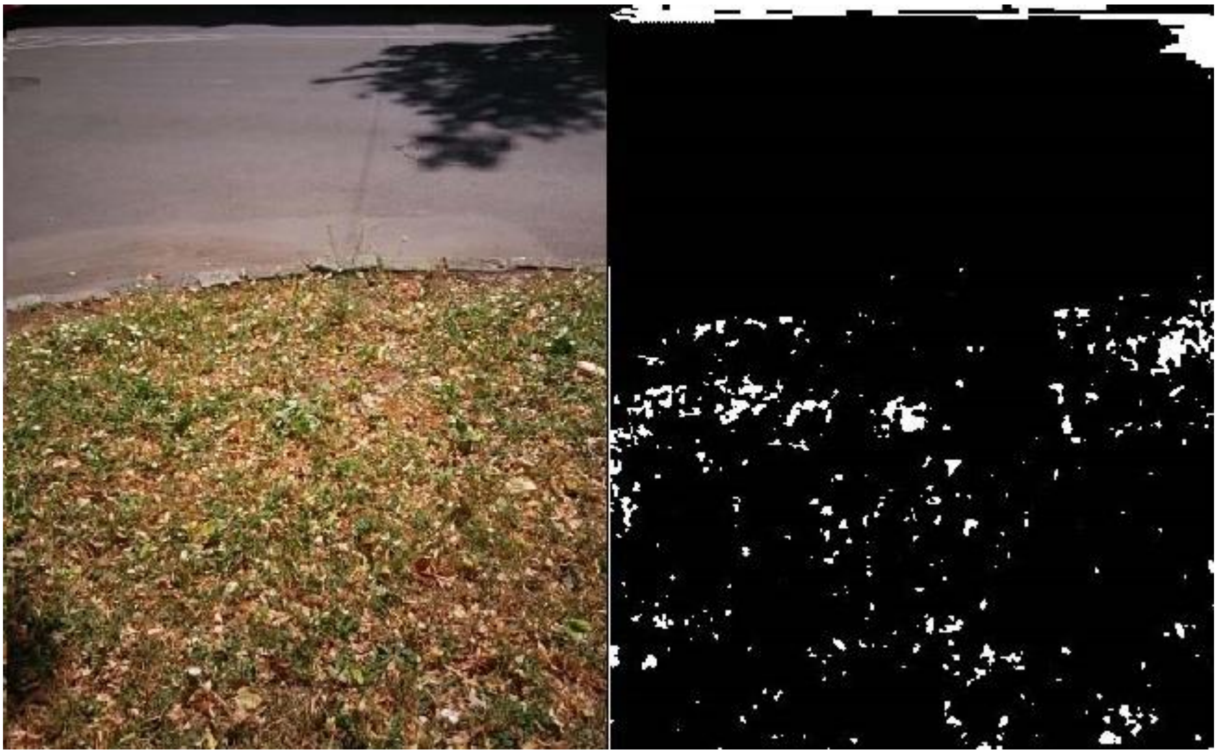


Bearing in mind the informations from above, we created the code in MathLab which recognized the green color from the image. This way is the easiest to distinguish the vegetation from the landscape. Of course, the flora is not always just green, that is why is needed to use post – processing to fill the gaps in defined area. The results can be seen below.



Unfortunately, the easiest solution was not the best one. Here we can present the bad results.





After that, we decided to make it more complicated.

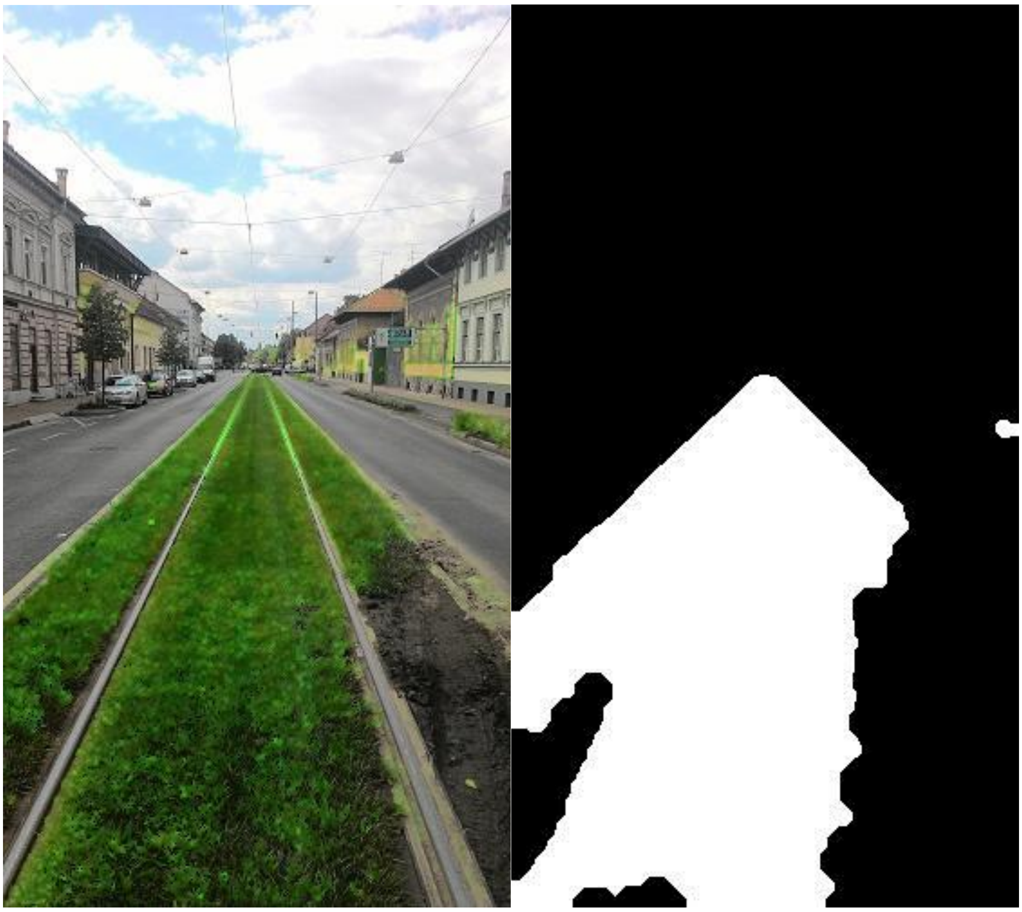
This simple algorithm takes advantage of the fact that the vegetation is usually green colored. (grass, leafs, etc.). First of all we imported RGB image, than we used pre – processing which is: dilate all non-gray scale pixel's green channel. (Each color channels' intensity must not be 'similar'.) This step's goal is to make the detection easier of those pixels which may are a part of the vegetation regions.

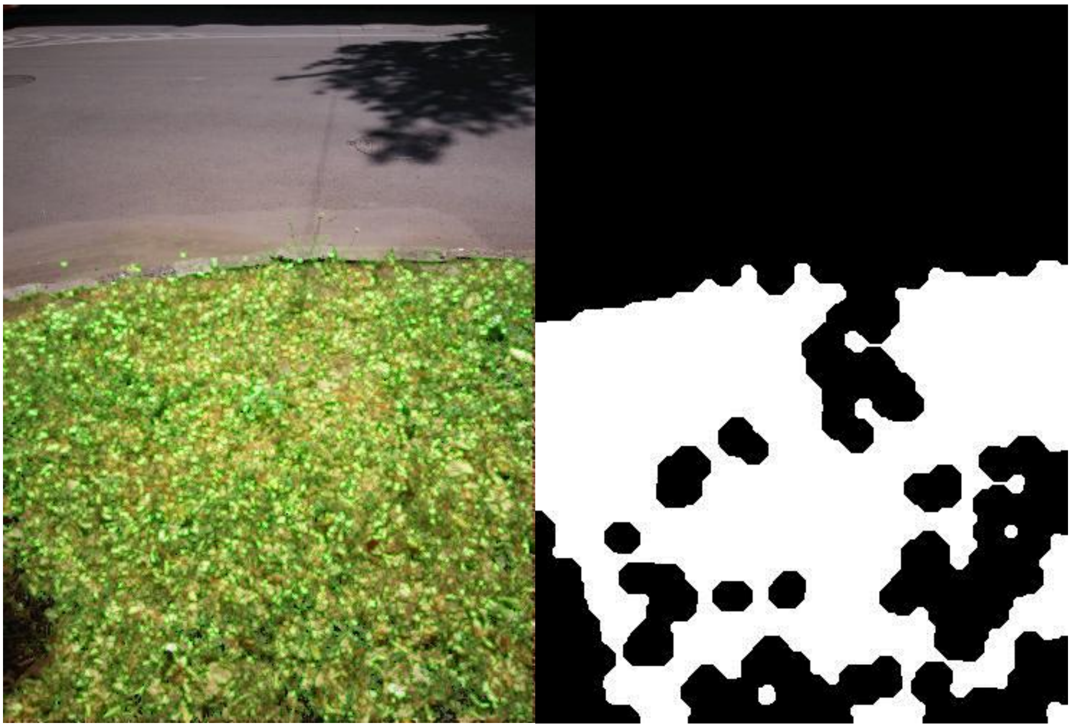
Secondly we had to mark all green pixels. A pixel is considered to be green if the green channel's intensity is greater with at least 10 units than red and blue channel's one. After that we run median filter on the output binary image to decrease the salt-pepper noise which consists of fake detected or missing vegetation pixels. Than was morphologically open then close the current output image to separate the vegetation and non-vegetation regions.

Advantages of this approach are, that it detects precisely the bright green colored vegetation, and clearly skip the gray scale or blue colored objects just like the sky or the road.

Disadvantages are not so significant. The solution detects other (but not all) green objects as vegetation. It also cannot detect the dark or non-green vegetation regions just like shadows.

The results can be seen below.





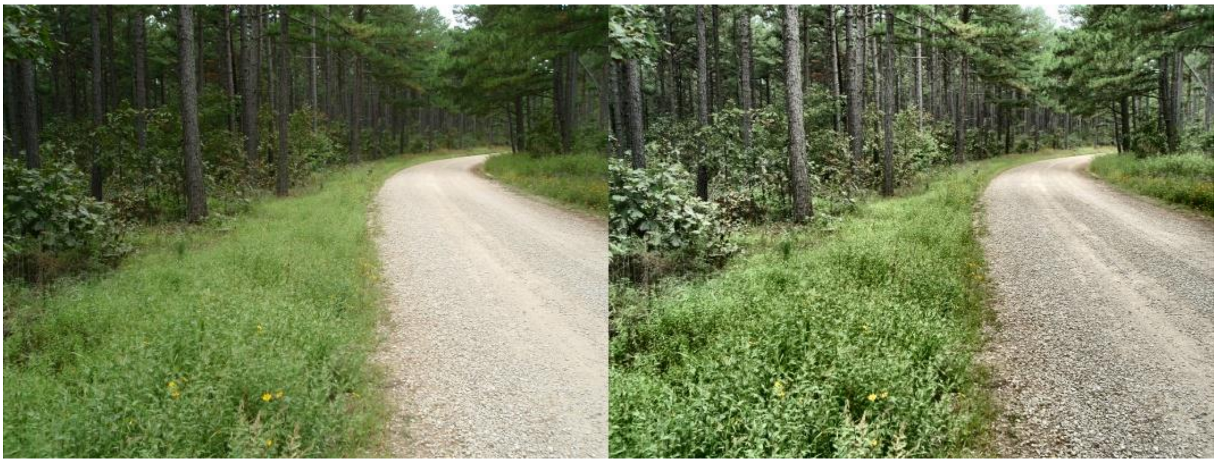
3. Make it difficult

- Pre – processing

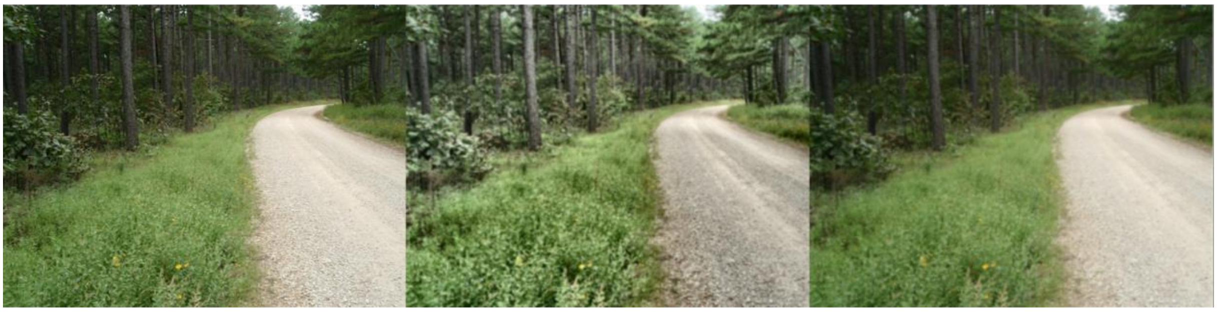
Preprocessing is an significant and diverse set of image preparation steps. It is used for eliminate or minimalize the problems with dataset.

It can distinguish several types of this method. First of all we chose to normalize the image, which can be, for example, by improving brightness of the image. For that we were using image resampling and greyscale contrast enhancement.

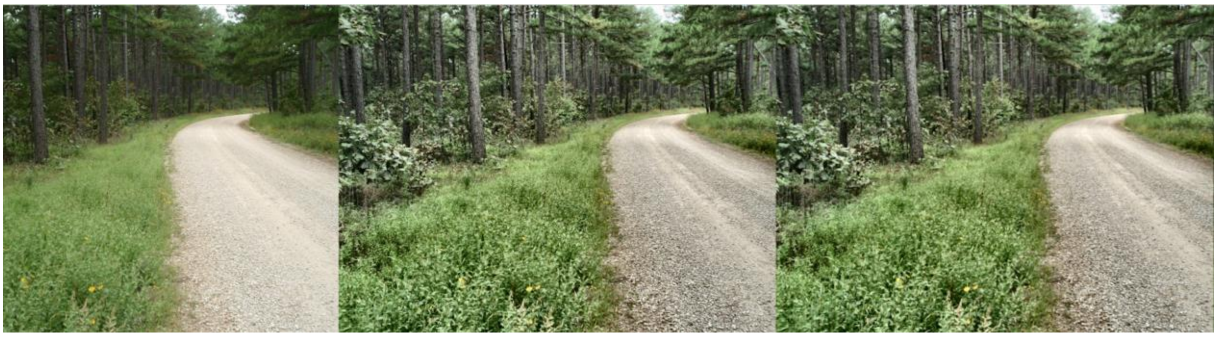
Secondly we decided to denoise the dataset to enhance the quality of input image. For that we used the several types of filtering. The results are shown below.



On the left side we can see the input image (original), and on the right side we applied the CLAHE technique.

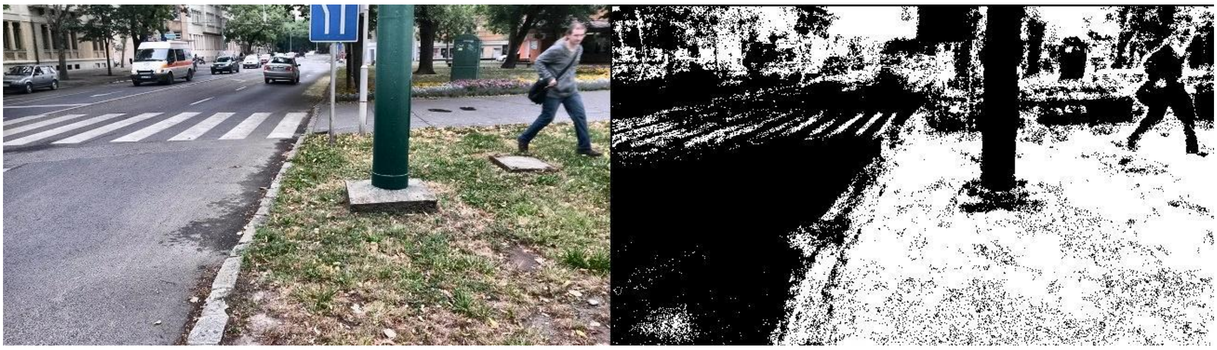


On the image above we compared the original image with ClAHE and Gaussian blur applied, and on the right side just with using the Gaussian blur filter.

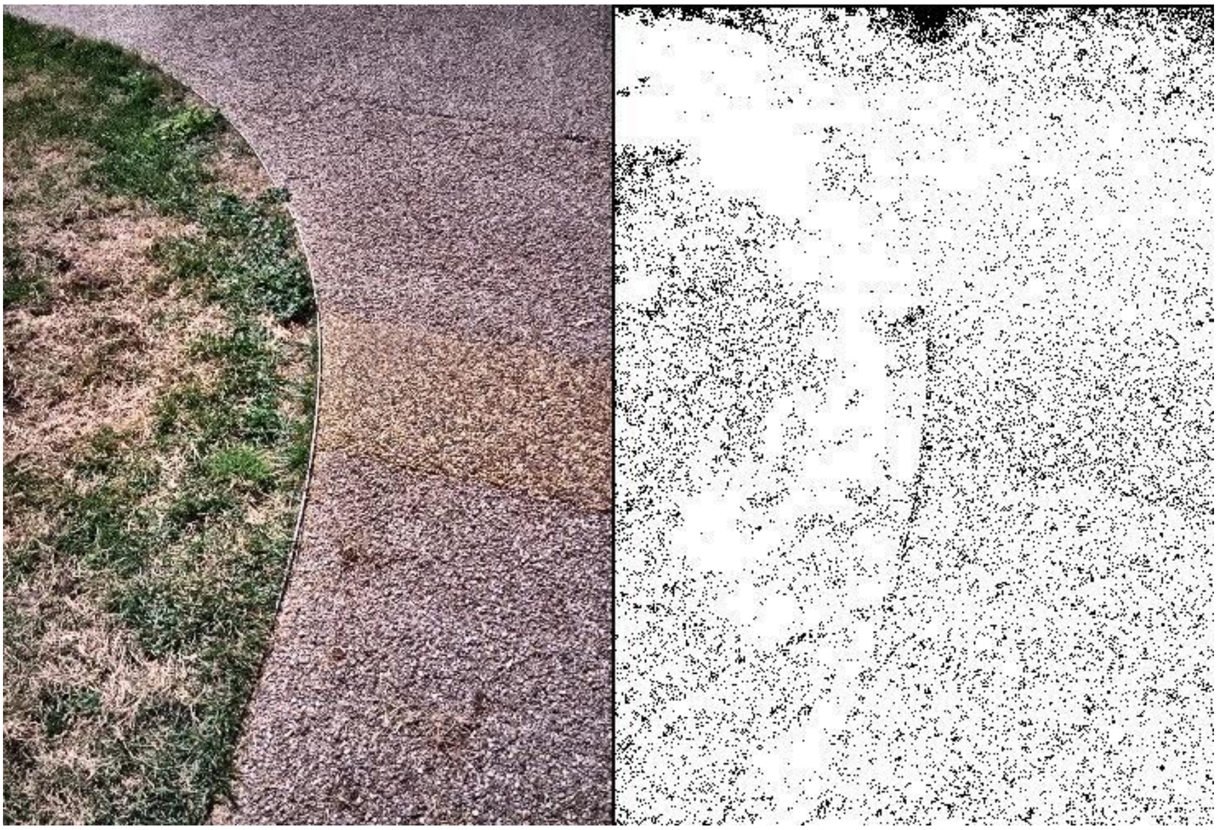


Here we compared the original image with median blur with CLAHE, and on the right side we present the result after using only CLAHE method.

On the images below we decided to show another examples of using CLAHE method.



On the image above we successfully applied CLAHE technique, what can be seen is that the method gave us satisfying results, but the model has to be better trained.



As it can be seen, this technique can lead to false results, because it emphasize the patterns which also classifies other negative patterns as being positive.

Afterwards by using particular types of mathematical operations we tried to further improve the picture by sharping, smoothing or blurring it. Of course, it was depending of the input image.

- Recognizing textures

Texture is another feature that can help to segment images into regions of interest and to classify them. Texture gives us data about the spatial arrangement of the hues or intensities in the image.

The greatest difficulty in texture analysis is defining what texture is. There are two principal approaches:

- structural approach where texture is a kit of basics texels in regular and repeated relations;
- statistical approach where texture is a quantitative measure of arrangement of intensities in a given area.

We found the database with natural pattern of the vegetation and we added a few more for specifying the patterns of particular parts of the flora. We create the negative and positive examples of the scenes in nature. The few of them are showed below.

We noticed, that this way to recognize the vegetation on the images is dependent of the size of the database and the quality of the images, so it is not the easiest and effective method to achieve the goal. Other obstacle is that the vegetation has no specify pattern.

- Machine learning

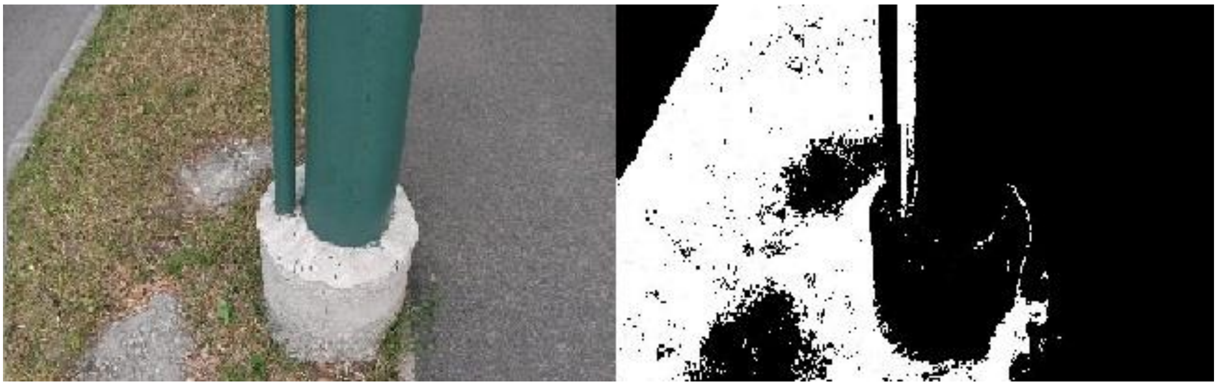
We tried to use different classifiers (Gaussian Naive Bayes, SVM, Decision Tree and Random Forest) to optimize model hyper parameters using 10 fold cross validation on training set. Instead of using all features and keeping possibly irrelevant or redundant features that increases the computation time and possibly in the end lead to worse trained model, we decided to use feature selection techniques (removing low variance features, close to constant values), keeping k-best features with respect to information gain and correlation coefficient score, where k is hyper parameter of model that is being optimized.

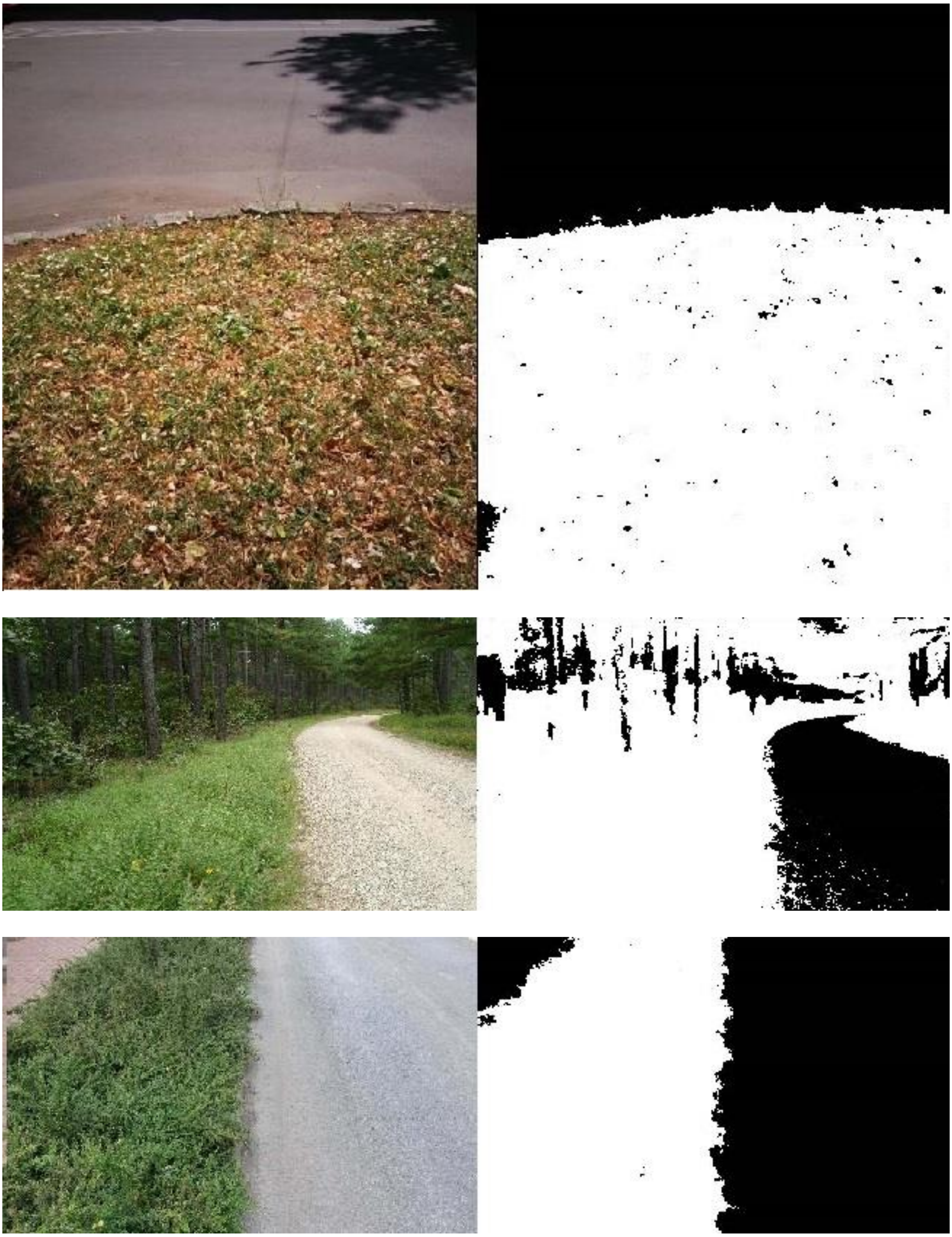
Highest accuracy, recall and F1 score were achieved using RandomForrest Classifier (both precision and recall on test set above 90%). SVM with linear kernel provided a bit worse results (70% in both precision and recall) possibly due to small amount of texture data (70 textures for positive and negative samples).

Downside of machine learning base classifier approach is that is highly dependent on texture dataset.

To obtain accurate model, large dataset has to be obtained so the model could generalize. That especially goes samples for non-vegetation class.

The results of some mechanism can be seen below.





- Post – processing

After classification, produced image is patchy so we are using simple morphology operations (dilation and erosion) to remove areas too small to be treated as vegetation and to fill in the patchy areas.

4. Summary

Our goal was to detect the vegetation from an image of a road side, and we achieve our main request.

We decided not to focus only at one solution, and we were working on several approaches, unfortunately none of them was perfect. Used technologies include MathLab, Python and several image processing and machine learning libraries (OpenCV, skimage, sklearn).

The principal idea fulfilled the task which was placed in front of us, but it did not work well when the vegetation had another determinate color than green. That is why we created more complicated program which used pre – processing and dilated all non-gray scale pixel's green channel.

Later we decided to try some more solution, which was pre – processing, machine learning and recognizing textures. After that, we noticed than none of the approaches has advantages and disadvantages as mentioned above. Though that we are glad of our results in such a short time of working.