ALESSANDRO SAVIOLO MATRICOLA: 1186771

PROJECT T2: SHADOW DETECTOR

The shadow detector is developed under the following assumptions:

- One strong source of light (dataset images are all outdoor images where the only source of light is the sun)
- Focus on ground shadows

The features considered by the shadow detector for the comparison of different segments of the image are:

- **Intensity**: segments under shadows become darker as they are blocked from the illumination source. Furthermore, since there is also ambient illumination (such as sky illumination in outdoor images), there is a limit on how much darker they can become. This assumption is used to predict the range of intensity reduction of a segment under shadow and is the first stage used to reject non-shadow segments
- **Color information**: segments under shadow become darker but retain their chromaticity. Since RGB color space offers no separation between intensity and chromaticity, it's better to convert the image to HSV color space, where H represents Hue, S the saturation and V the intensity
- **Texture**: segments under shadow retain most of their texture. Each shadow candidate is classified as either a dark object or a shadow by comparing the texture of the shadow candidate with the texture of the neighboring segments. Texture information is captured by using Local Binary Patterns and edges

The shadow detector method is divided in the following steps:

0. **Preprocessing**: filters are applied to the image in order to get better results in the segmentation step
1. **Segmentation**: $k$means is used to segment the image
2. **Intensity thresholding**: image is converted to HSV color space and the intensity channel is thresholded
3. **Candidates computation**: segments belonging to the white components of the thresholded image (at point 2) are selected as shadow candidates
4. **Candidates merging**: "similar" candidates are merged
5. **Candidates comparison**: candidates are compared with their non-candidate neighbors and selected as shadow based on chromaticity and texture

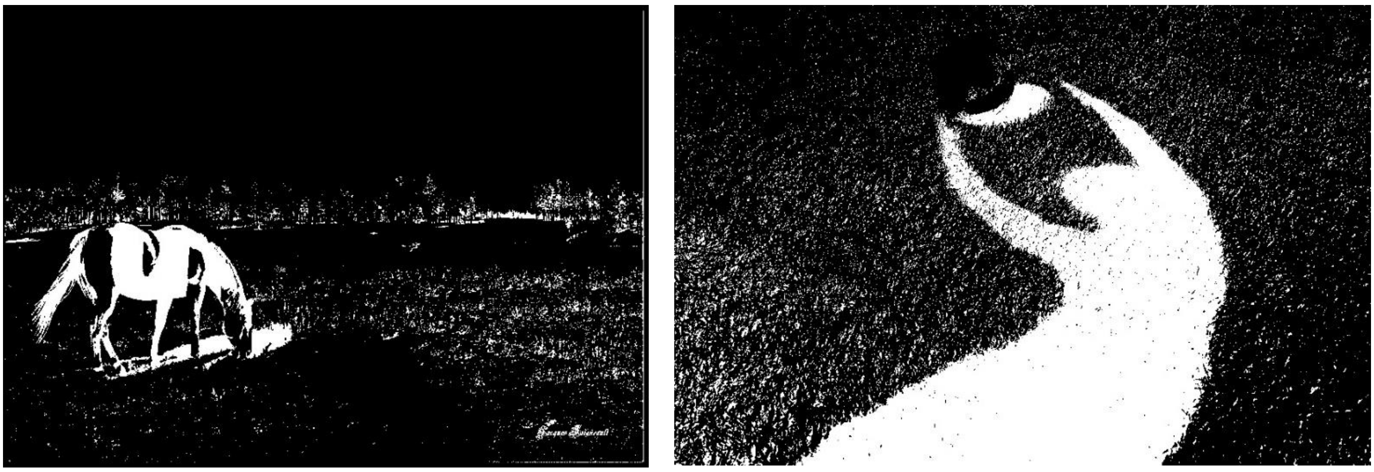
PREPROCESSING OF THE IMAGE

Before segmenting the image, a preprocessing is applied to prepare it for the segmentation process.

First a strong bilateral filter is applied to remove noises and smooth the image without affecting the edges of the objects and of the shadows. Closure is then applied to remove small holes. Dilation bridges the gaps while erosion removes unwanted details.

Preprocessing the image before segmentation is important for those images that have strong textures.

Images from the Random dataset.



*Results of applying the thresholding technique **without** the preprocessing of the images before the segmentation.*

*Results of applying the thresholding technique **with** the preprocessing of the images before the segmentation.*

AGGLOMERATIVE HIERARCHICAL SEGMENTATION

The segmentation of the image starts by applying $k$means algorithm to a set of precomputed features based on color information and position of the pixels. Then, segments that are considered as candidates are merged based on color information and texture.

The method presented is similar to an agglomerative hierarchical clustering (segments will be merged in the next steps of the program) and limits the need of choosing the right number of segments for different images, since a large number would work good for all the images.

Many color spaces have been tried to obtain the best segmentation with $k$means. Overall, HSV color space seems to offer the best results.

Image from the Random dataset.

Image segmented with 25 segments in the BGR color space.

Image segmented with 25 segments in the HSV color space.

Image segmented with 25 segments in the Lab color space.

MERGING SIMILAR CANDIDATES

After segmentation, first a threshold is applied to the intensity channel of the image, then the segments corresponding to the white components of the binary image returned by the thresholding are selected as shadow candidates. Here is where the agglomerative hierarchical clustering starts. Segments selected as shadow candidates are merged together based on color information (RGB) and texture (LBP) similarity.

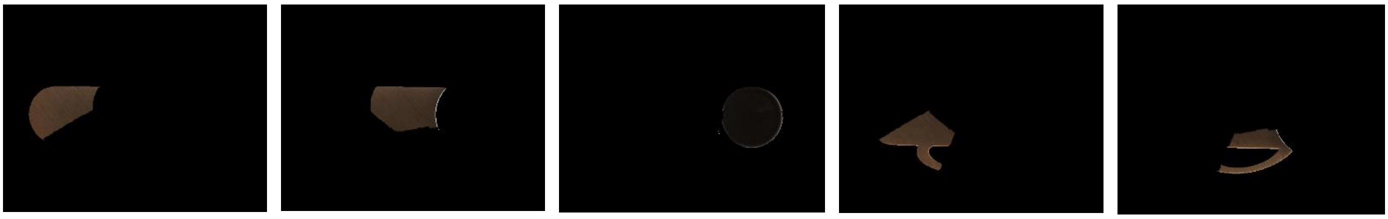
Image from the Black dataset.

```
[100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 100, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Merge table: the segments are represented by the indexes of the rows. Each point with value 100 correspond to the merge of the corresponding segments identified by the row/column indexes. In blue and yellow are highlighted the two iterations that will be computed for the image considered.

```
[100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 100, 0, 0, 0, 0, 0, 200;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 200;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 200;
100, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 200;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Merge table after one iteration. The last column keeps track of the already seen segments (value 200 highlighted by the red rectangles).



*Segments selected as shadow candidates **before** the merging.*

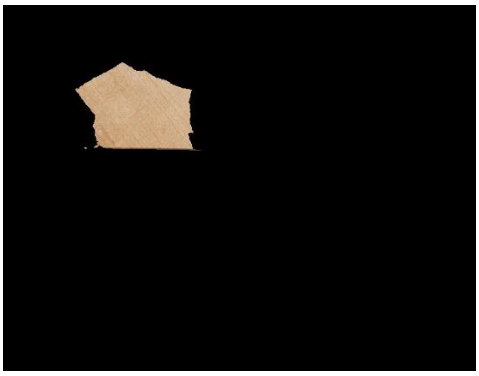
*Segments selected as shadow candidates **after** the merging.*

This step is important since the final comparison of the shadow candidates is based on their neighbors which are non-candidates. This means that if the image is segmented with a huge number of k for $k$means, then it can happen that some shadow segments have as neighboring segments that are only candidates. If this happen, then the shadow candidate is rejected, even if it belongs to a shadow.

CANDIDATES COMPARISON

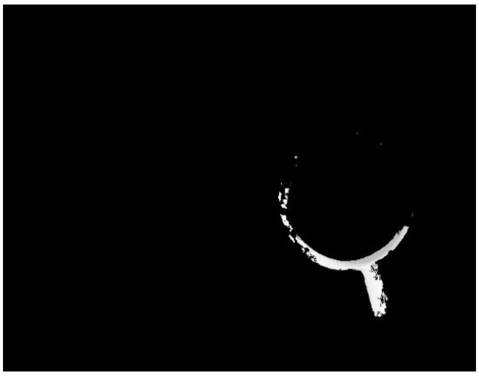
The final step of the program consists of comparing shadow candidates with their non-candidate neighbors. The comparison is based on texture (LBP and Edge Ratio) and chromaticity (Hue and Saturation) similarity. If a candidate has similar properties with one of its non-candidate neighbors (at least one), then it's considered as a shadow segment.

Shadow candidate.



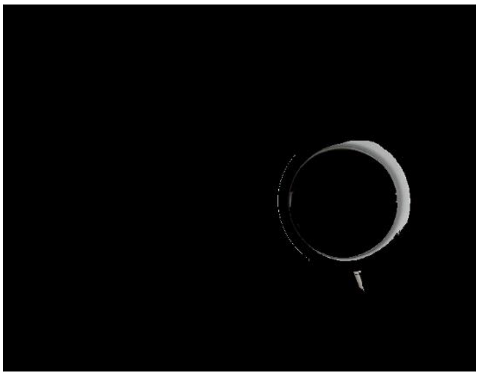
Hue: 2.38, *Saturation:* 28.77, *LBP:* 0.12, *Edge Ratio:* 0.05

Hue: 1,58, *Saturation:* 5.51, *LBP:* 0.12, *Edge Ratio:* 0.05

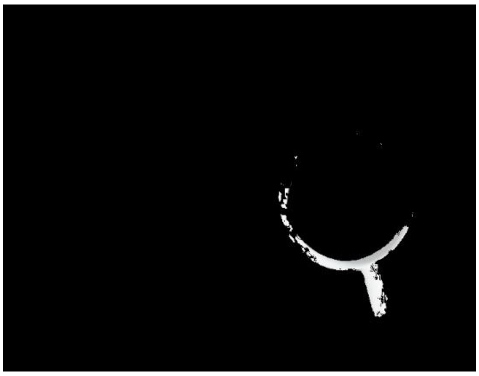


Hue: 66.65, *Saturation:* 114.00, *LBP:* 0.38, *Edge Ratio:* 0.02

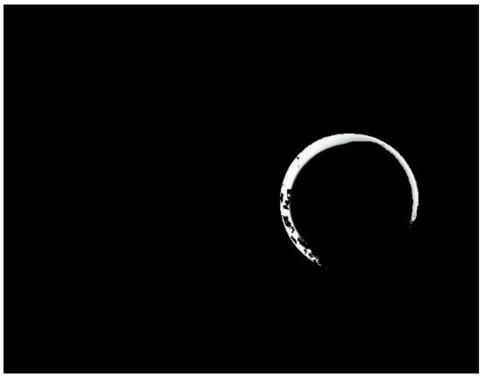
Shadow candidate.



Hue: 52.62*, Saturation:* 83.89*, LBP:* 0.59*, Edge Ratio:* 0.11



Hue: 65.25*, Saturation:* 86.82*, LBP:* 0.45*, Edge Ratio:* 0.02



Hue: 48.17*, Saturation:* 86.80*, LBP:* 0.4*, Edge Ratio:* 0.09

In the example, the first candidate is selected as a shadow segment since it has similar texture (both LBP and Edge Ratio) and chromaticity (Hue and Saturation). Instead, the second candidate is not selected as a shadow segment since it differs significantly by all of its non-candidate neighbors in both texture and chromaticity.