Algorithm
First step is to understand what is happaning from the samples.
For each sample image we:

1. split the image into RGB channels;
2. subtract G and B components from the doubled R component to measure the prevalence of red component over green and blue ones.

   $$lightness_{RED} := 2R - G - B$$

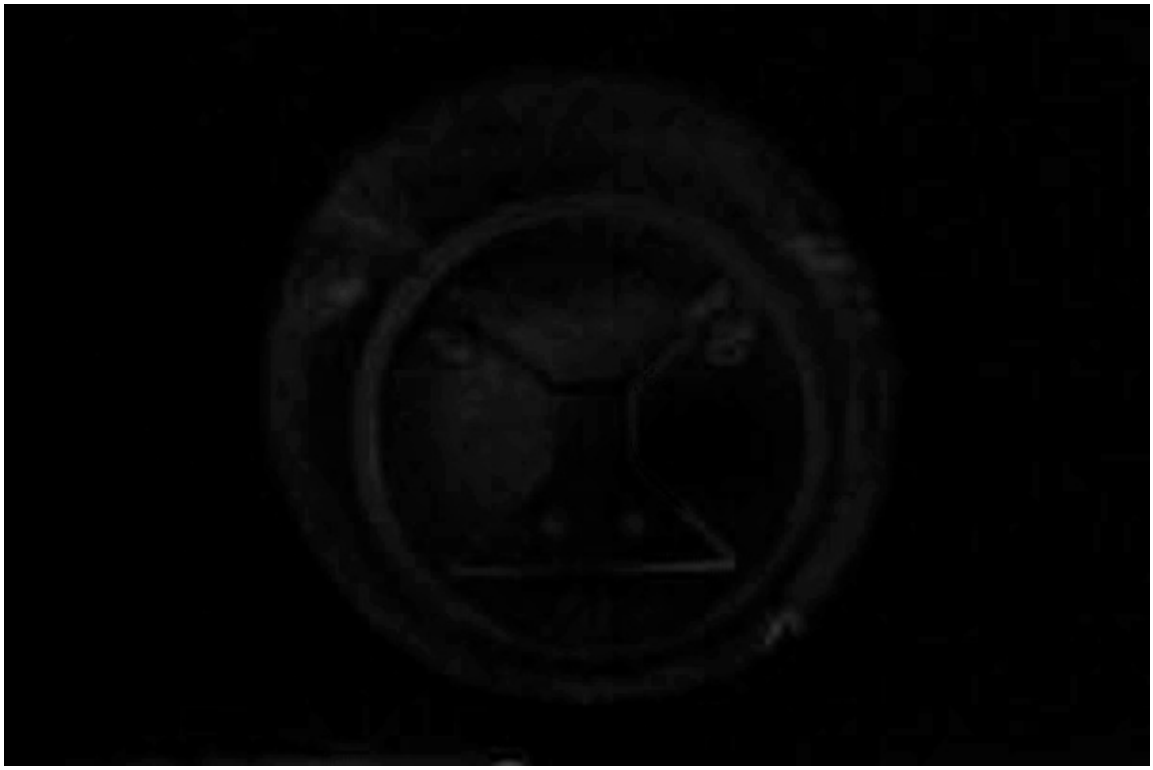
   Experiments show, that due to constant parameters of the light we can distinct every of three classes from the brightness of red channel.
3. Calculate the average value of all pixels in $lightness_{RED}$.
4. Compare average lightens with learn thresholds.

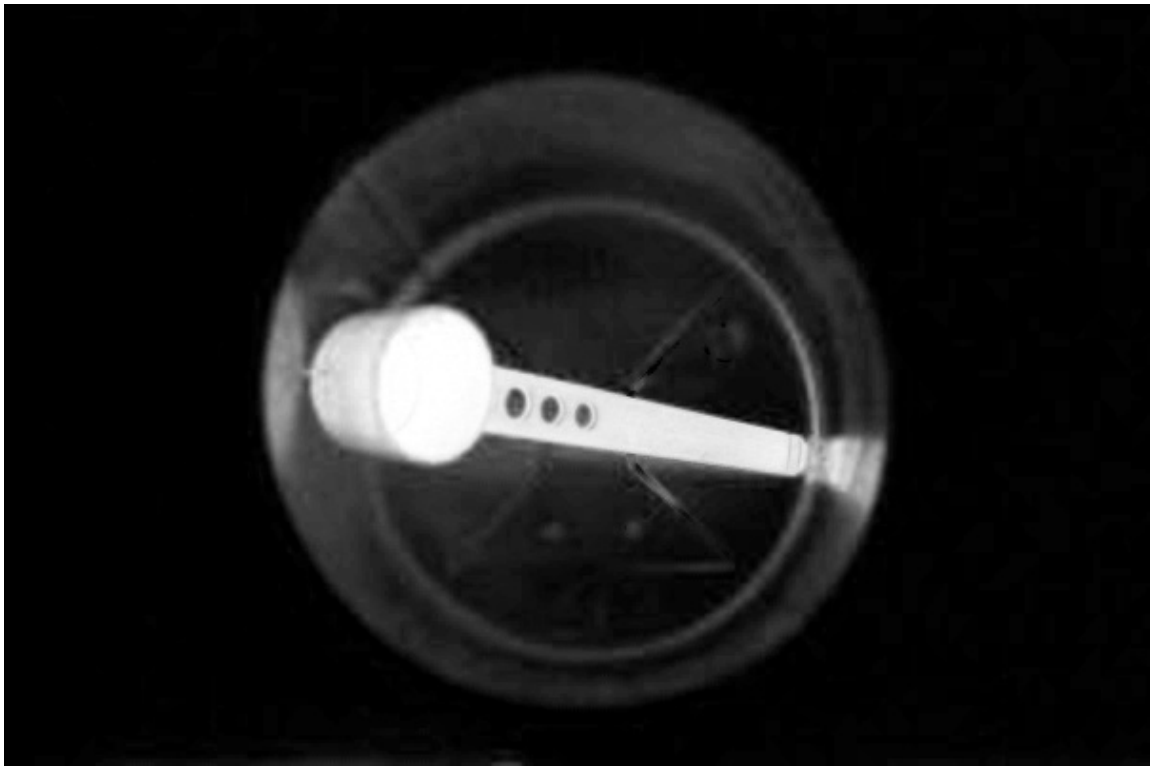
After the train step we do the following:
For each image in test set:

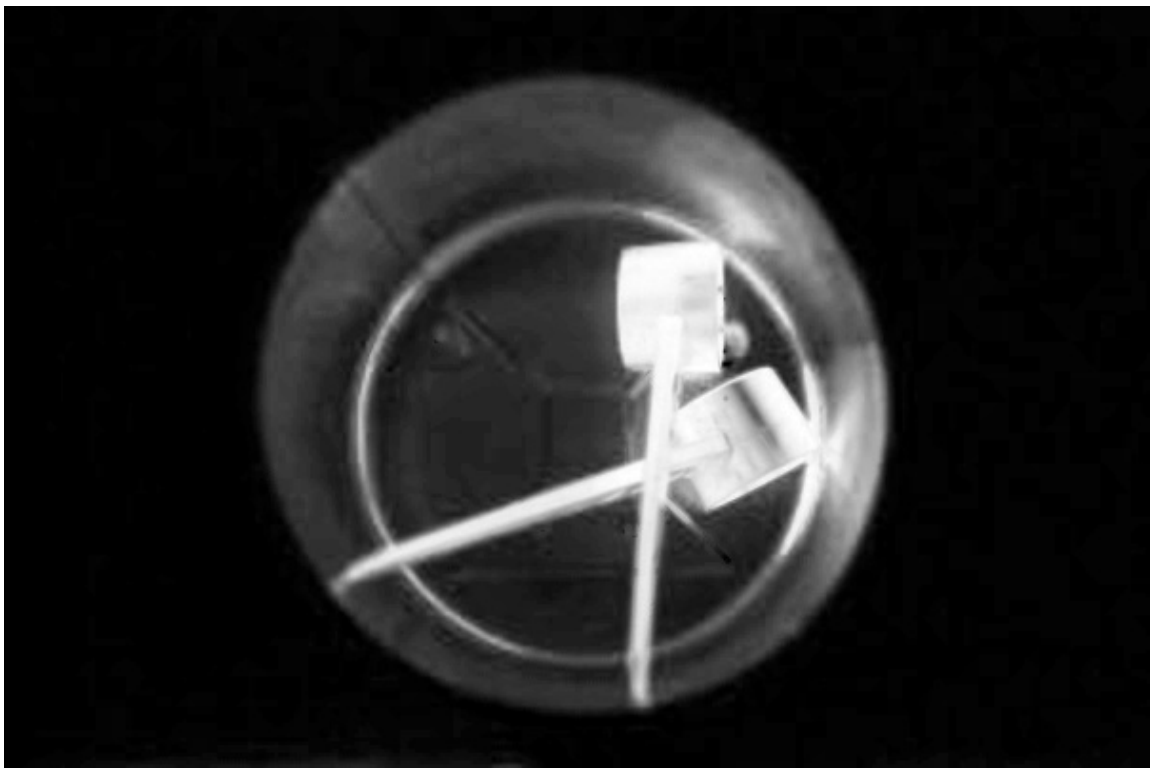
1. Load test image;
2. Split the test image into channels;
3. Find $lightness_{RED}$ for the image;
4. Calculate the average value of all pixels in $lightness_{RED}$;
5. Compare average lightens with learned thresholds.

Illustration

Here are a few results of image spliting:







From upside down: $lightness_{RED}$.

Performance measure According to confusion matrix:

	predicted 1	predicted 0	Total
real class 1	True Positive (TP)	False Negative (FN)	P
real class 0	False Positive (FP)	True Negative (TN)	N
Total	P'	N'	$P+N$

The formulas for precision, recall and accuracy:

1) $precision = \dfrac{TP}{TP+FP}$

2) $recall = \dfrac{TP}{P}$

3) $accuracy = \frac{TP+TN}{P+N}$

The results of the program:
For every class precision = recall = accuracy = 100