OpenCV & Java

Pavel Jetenský
www.upce.cz|www.hooyu.com

Co nás čeká

K-means algoritmus

Cvičení

Instalace OpenCV pro Javu

Základní práce s obrazem – read, write, show

Background substraction

Pomocí edge detection

Pomocí thresholding

Pomocí K-means clustering

Cvičení

Instalace OpenCV pro Javu

Základní práce s obrazem – read, write, show

Background substraction

Pomocí edge detection

Pomocí thresholding

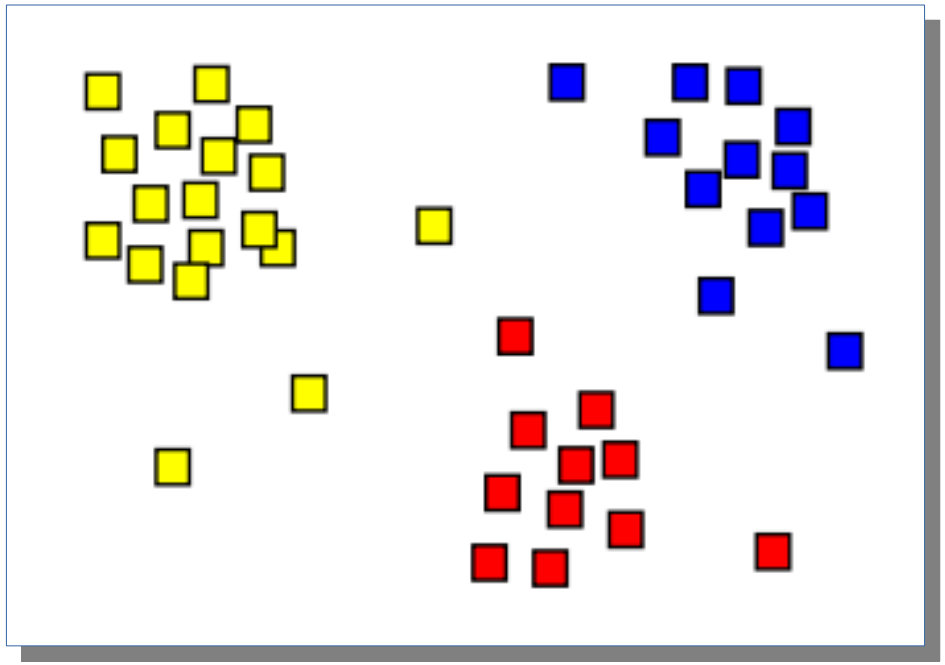
Pomocí K-means clustering

Co vás zajímá?

Programátorské tipy pro práci s:

- IntelliJ ideou
- Verzovacím systémem GIT
- Buildovacím systémem Gradle

K-Means clustering: v8



K = 3

K-Means: vstup

- N d-dimenzionálních vektorů
- k počet klusterů

K-Means: výstup

Centers: d-dimenzionální středy clusterů (k)

- Minimalizované vzdáleností bodů od středů vektorů ↴
- Klustery vytvoří voronoivy regiony

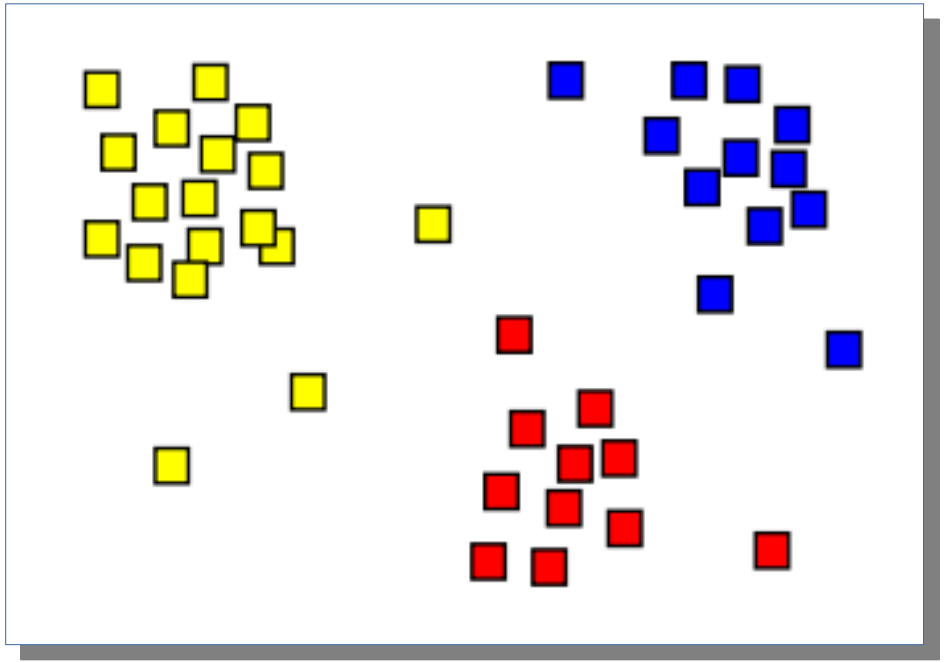
- **Labels**: Příslušnost vektorů ke klusterům

K-Means: použití

- Geografická data (2D,3D)
- Statistická n-rozměrná data (n-D)
- Color quantization (HSV, LAB)

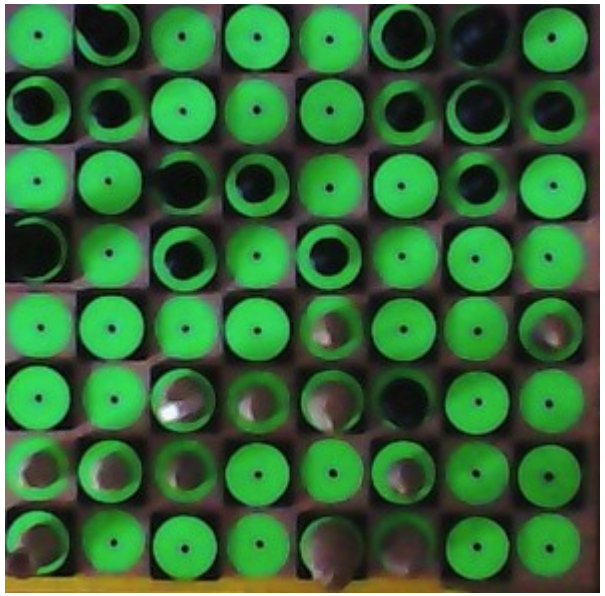
K-Means: použití

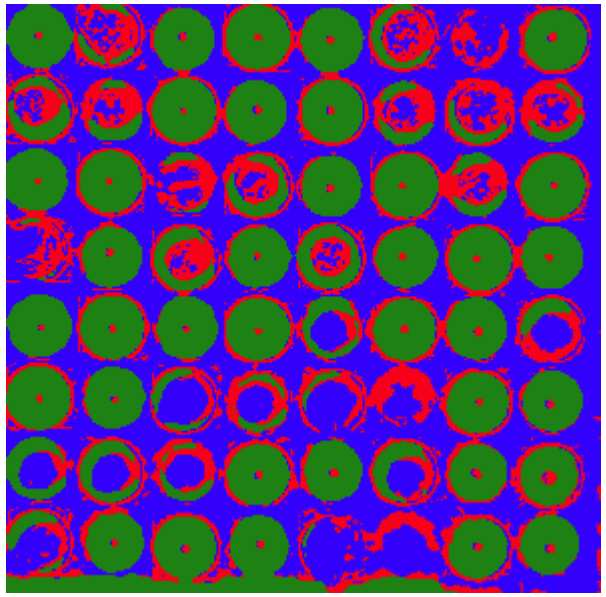
- Navrhněte umístění 3 skladů pro existující prodejny
-



K-Means: použití

- Roztřiďte pixely do 3 skupin stejné barvy (RGB, HSV, LAB)
-



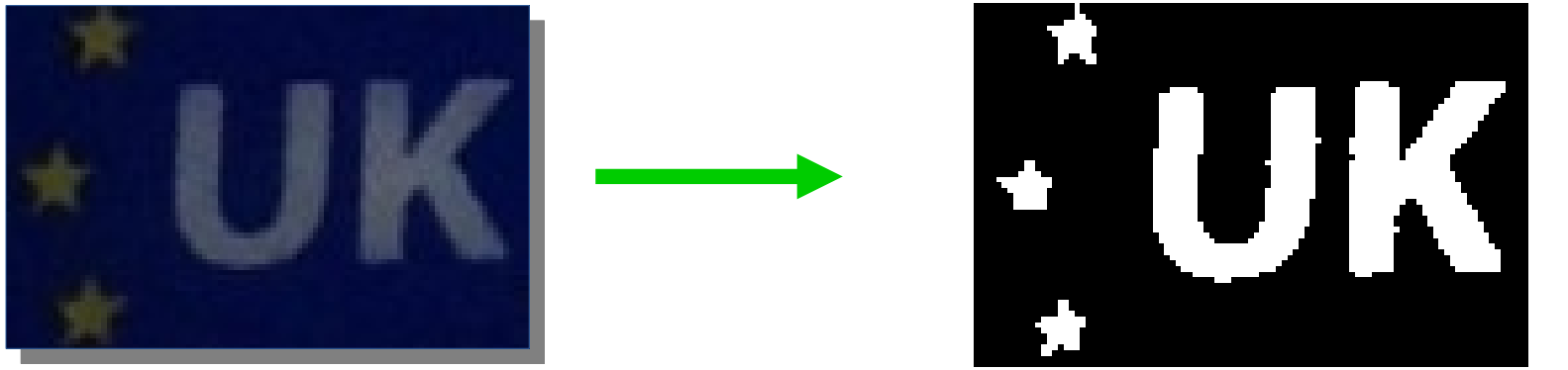


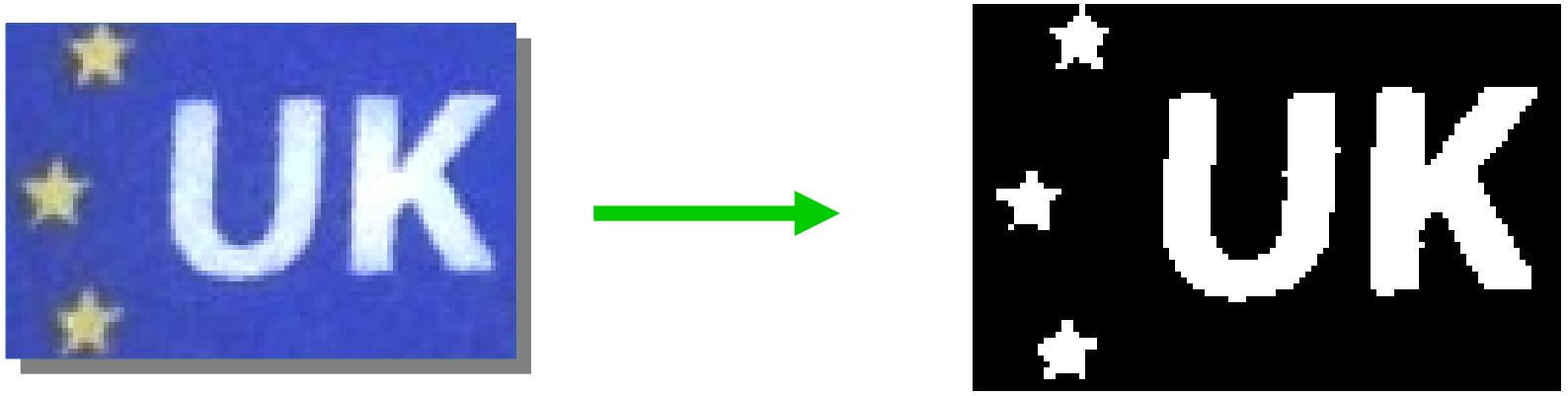
K-Means: color quantisation

K-Means: v9

```
Mat samples32f = new Mat();
Mat allPixelsInOneRow = sourceImg.reshape(1, sourceImg.cols() * sourceImg.rows());

allPixelsInOneRow.convertTo(samples32f, CvType.CV_32F, 1.0 / 255.0);

TermCriteria criteria = new TermCriteria(TermCriteria.COUNT, 100, 1);
Mat labels = new Mat(sourceImg.width()*sourceImg.height(), 1, CvType.CV_32SC1);
labels.setTo(new Scalar(0));
Mat centers = new Mat();
Core.kmeans(samples32f, k,        labels,                    criteria,          1,
Core.KMEANS_PP_CENTERS + Core.KMEANS_USE_INITIAL_LABELS, centers);
```

K-Means : v9





Instalace OpenCV pro Javu

- Opencv.org - release 3.3.1
  - Opencv-331.jar
  - **Windows**
    - opencv_java331.dll
  - **Unix**
    - libopencv_java320.so
    - libraries
    - cmake

GITHUB

https://github.com/JetyCZ/javadays2017.git

v0 – empty gradle & groovy project
v1
v2
v3
...

Instalace OpenCV pro Javu: v1

- build.gradle

```
compile files('/usr/share/OpenCV/java/opencv-320.jar')
```

- Loading library

```
System.load("/usr/share/OpenCV/java/libopencv_java320.so");
```

Základní práce s obrazem: v2

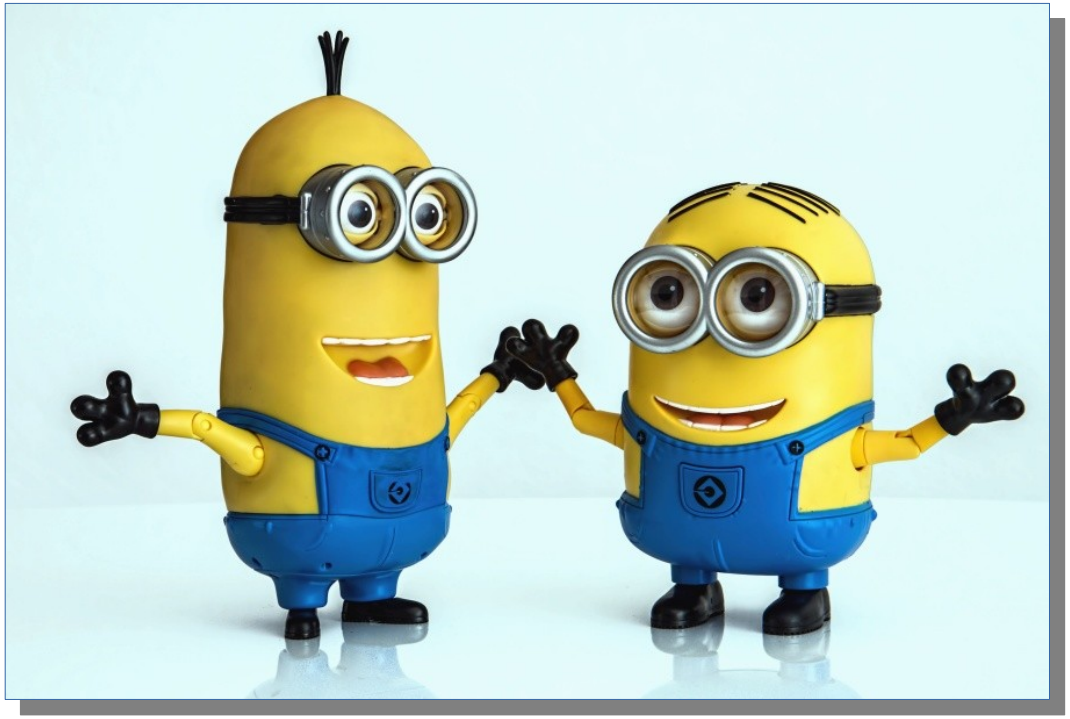
```
Imgcodecs.imwrite("/tmp/a.png", mat);
```

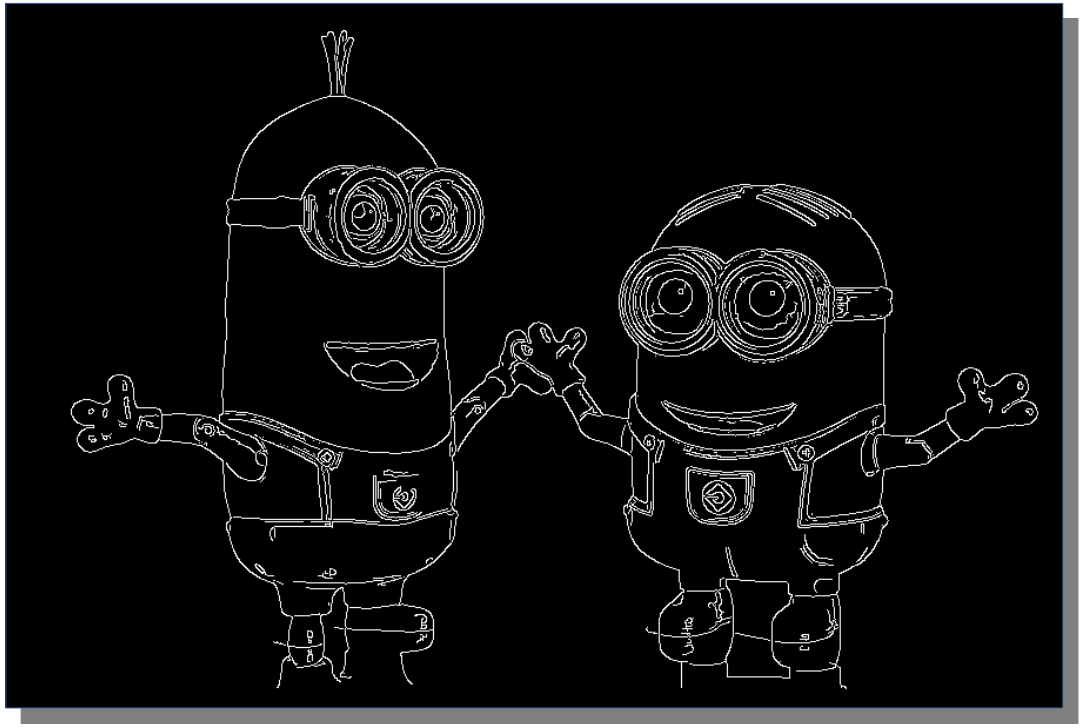
```
Mat mat = Imgcodecs.imread("/tmp/a.png");
```

Zobrazení v jFrame: v3

```
JFrame jFrame = new JFrame();
jFrame.setTitle("Image " + mat.size());
jFrame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
jFrame.setSize(new Dimension(mat.width(), mat.height()));

int type = (mat.channels()==1)? BufferedImage.TYPE_BYTE_GRAY :
BufferedImage.TYPE_3BYTE_BGR;

BufferedImage bufferedImage = new BufferedImage(mat.width(), mat.height(),
type);

DataBufferByte dataBuffer = (DataBufferByte)
bufferedImage.getRaster().getDataBuffer();
byte[] data = dataBuffer.getData();

mat.get(0,0,data);

jFrame.getContentPane().add(new JLabel(new ImageIcon(bufferedImage)));
jFrame.setVisible(true);
```

Canny edge detection: v3

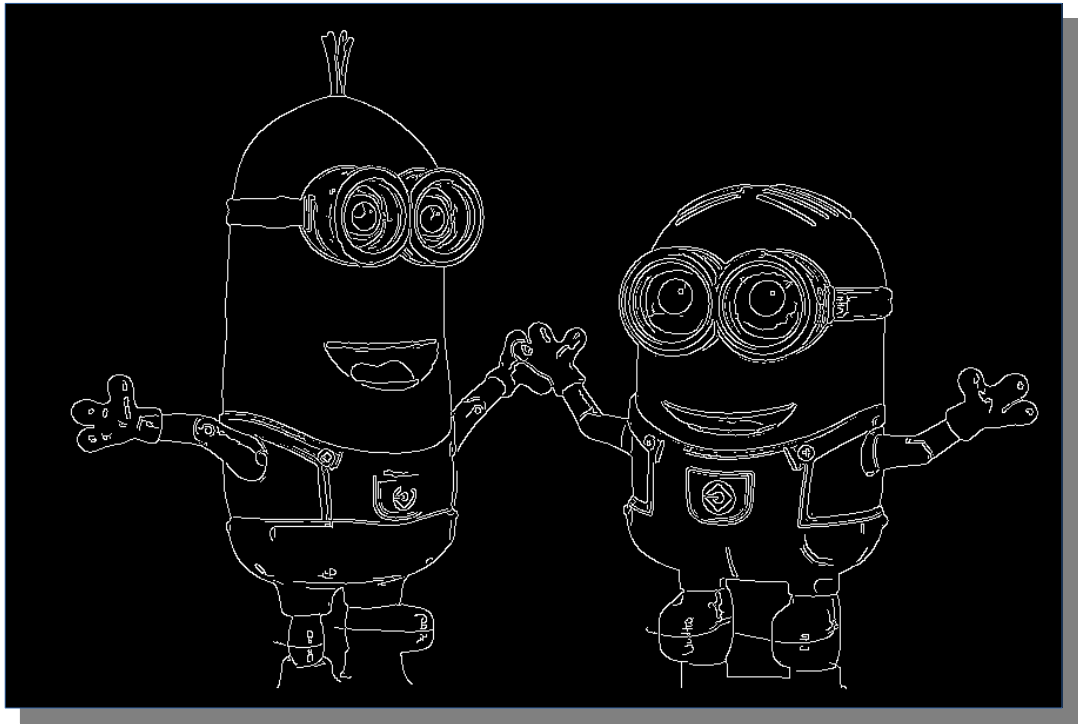


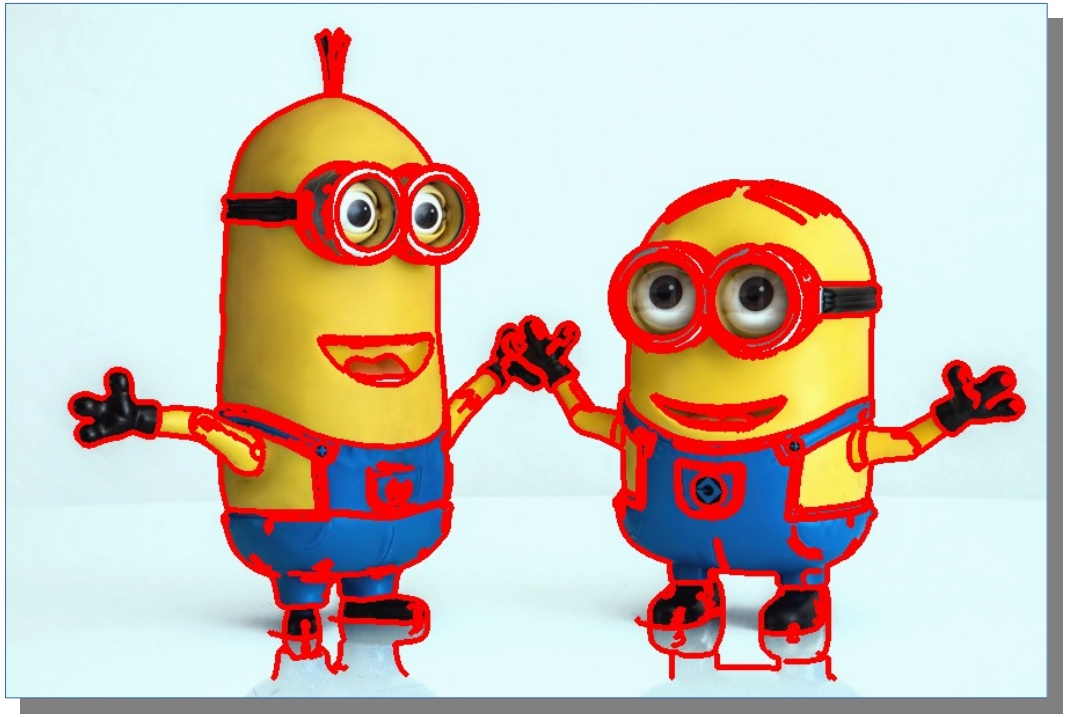


Canny edge detection: v3

```
Mat canny = new Mat();
double mean = Core.mean(sample).val[0];
double threshold1MeanPercentage = 0.3;
double threshold2MeanPercentage = 3*threshold1MeanPercentage;
Imgproc.Canny(sample, canny,
        threshold1MeanPercentage * mean, threshold2MeanPercentage * mean, 3, true);
```

Blob detection: v4

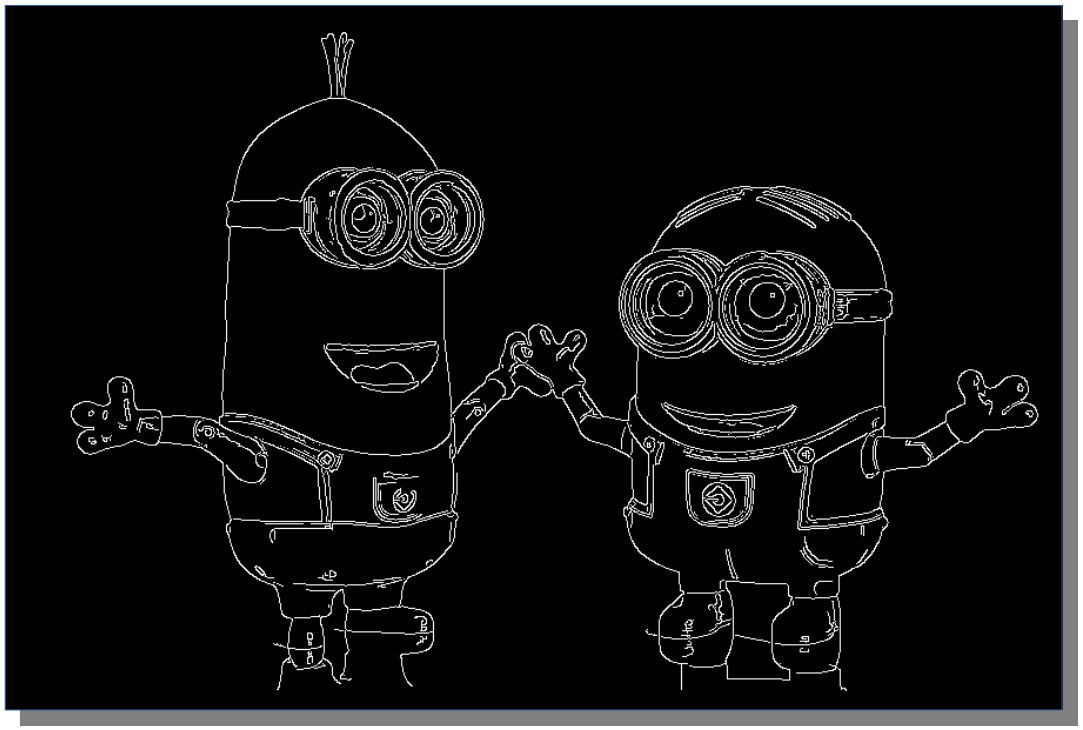


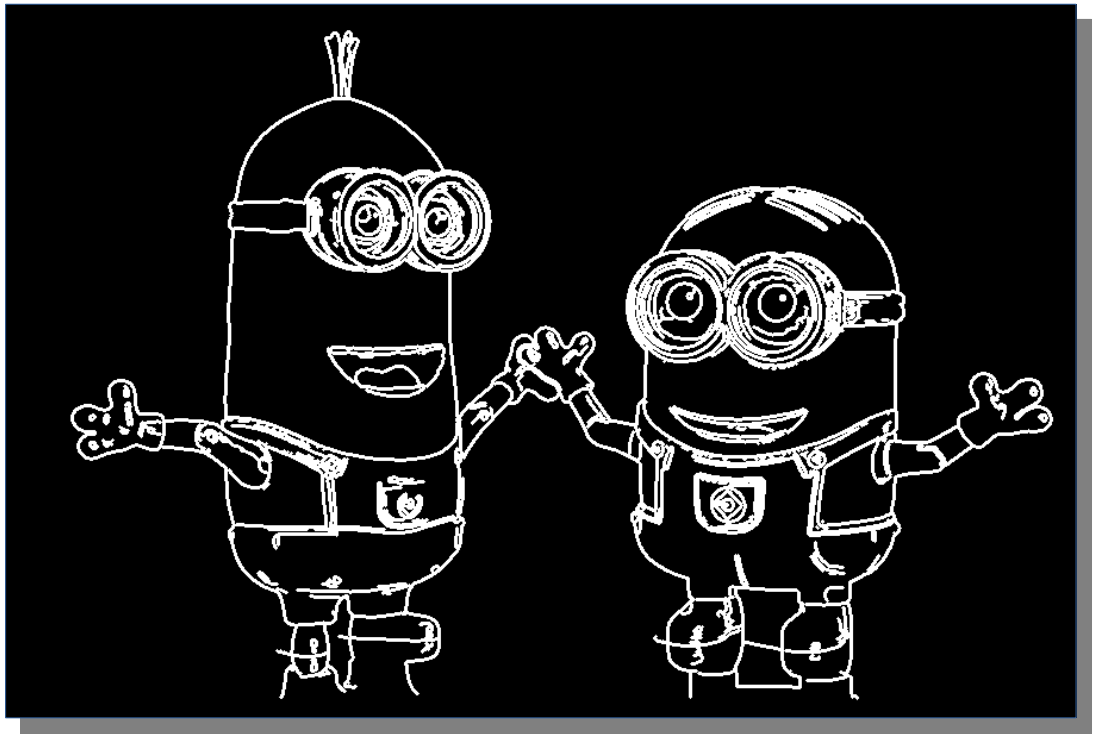


Blob detection: v4

```
List<MatOfPoint> contours = new ArrayList<>();
Mat hierarchy = new Mat();
Imgproc.findContours(dilated, contours, hierarchy,
Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
```

Dilate: v4

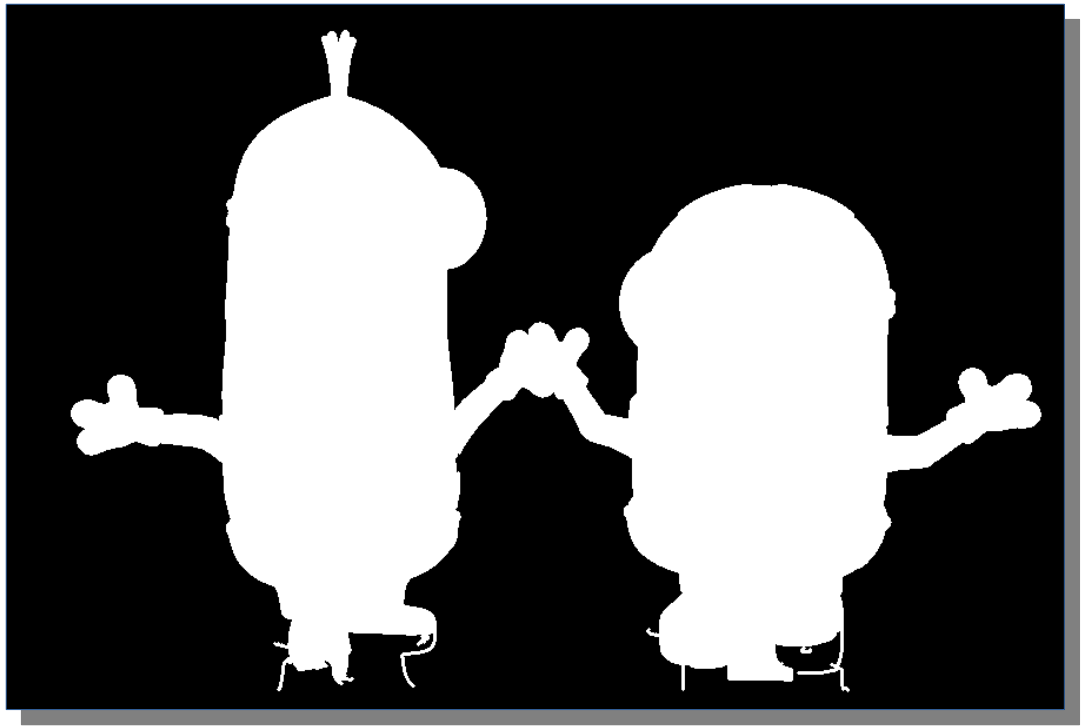


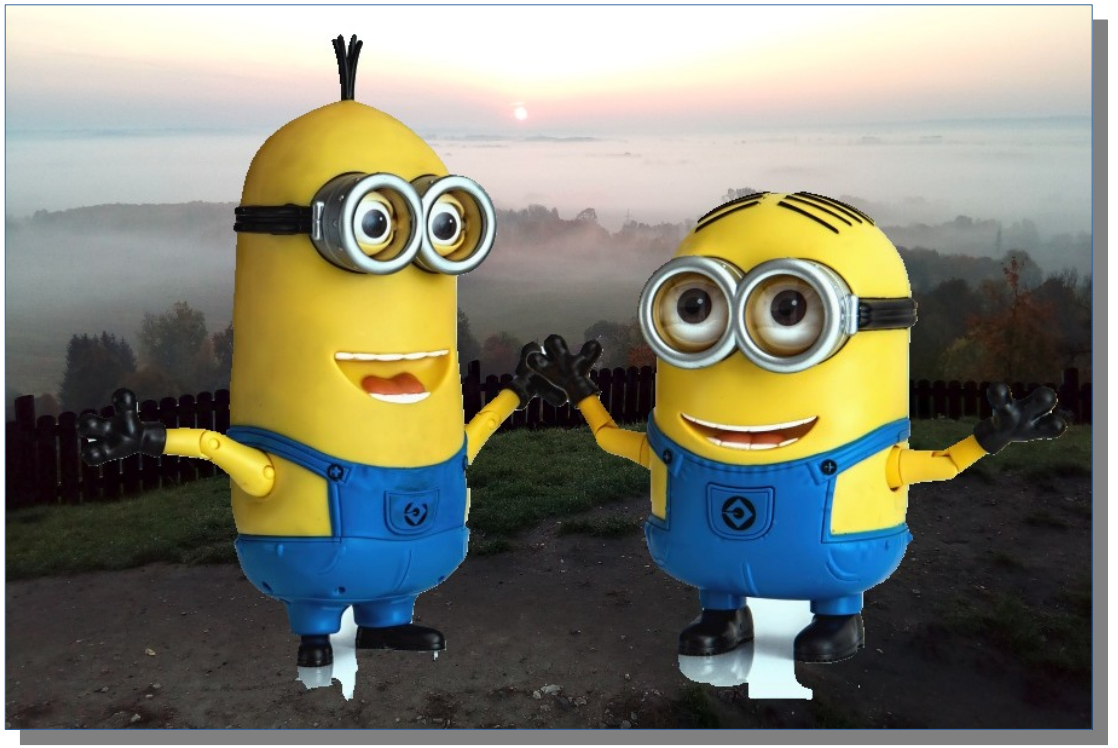


Blob detection: v4

```
List<MatOfPoint> contours = new ArrayList<>();
Mat hierarchy = new Mat();
Imgproc.findContours(dilated, contours, hierarchy,
Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
```

Background mask: v5

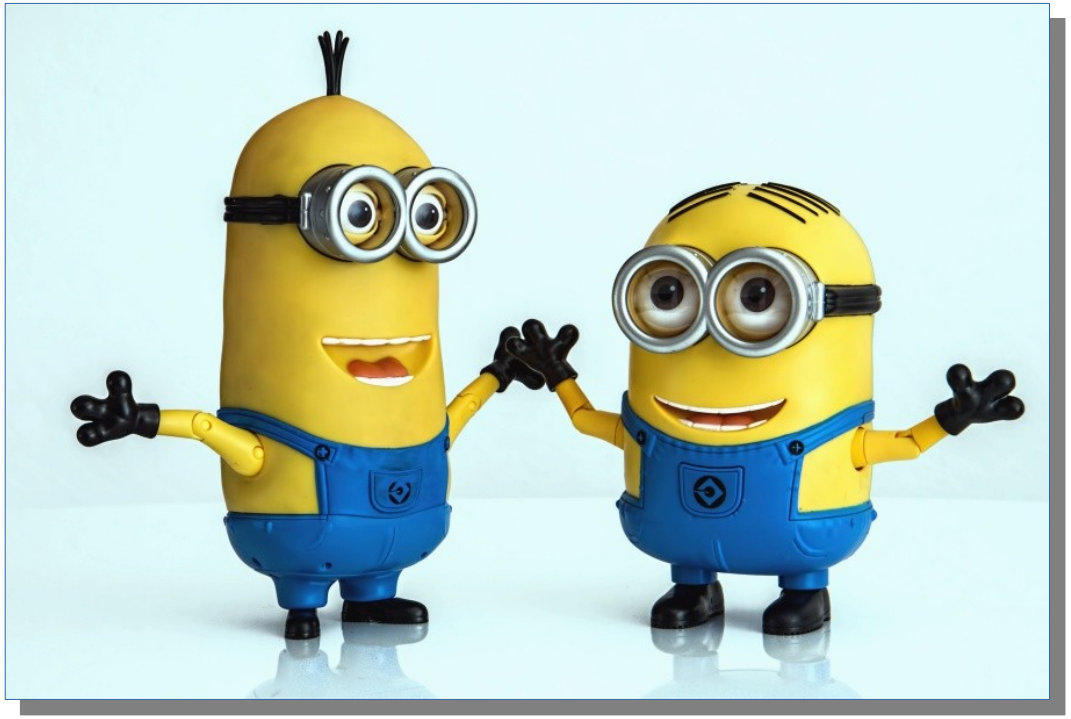


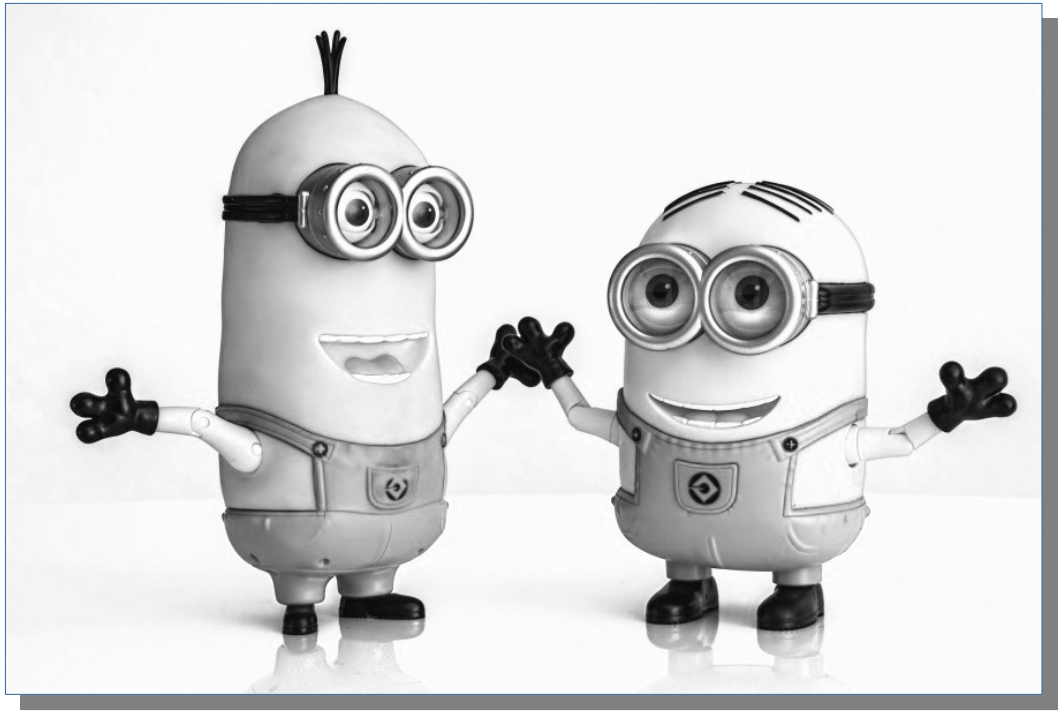


Background mask: v5

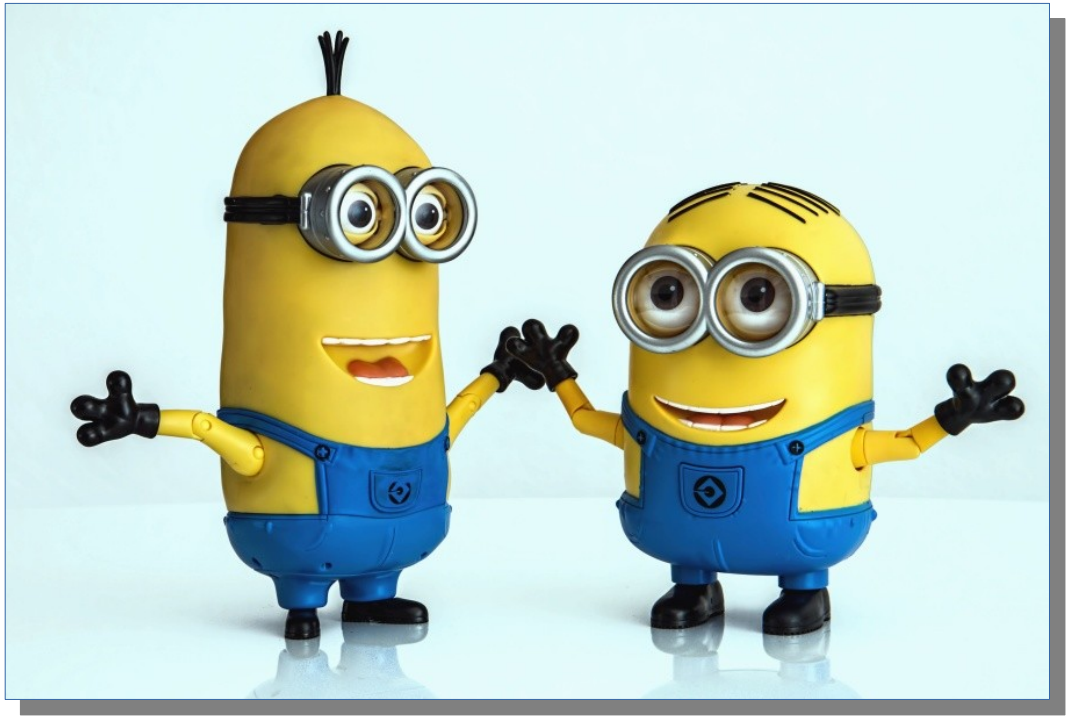
```
sample.setTo(new Scalar(0,0,0),
foregroundMask);
```

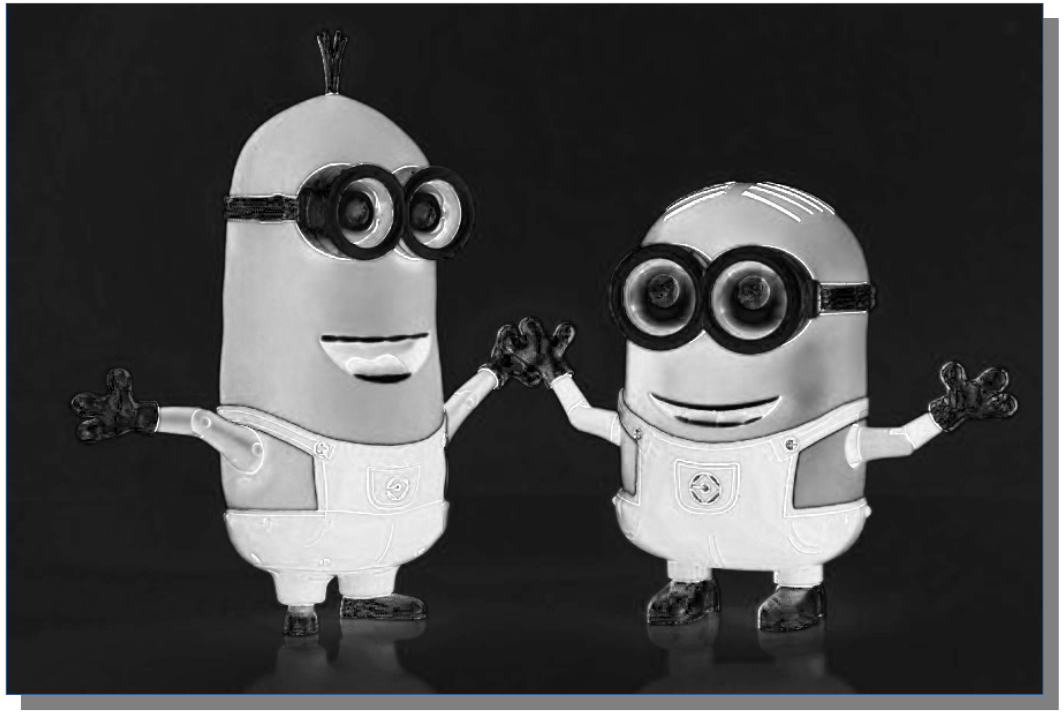
HSV Color model - V : v6



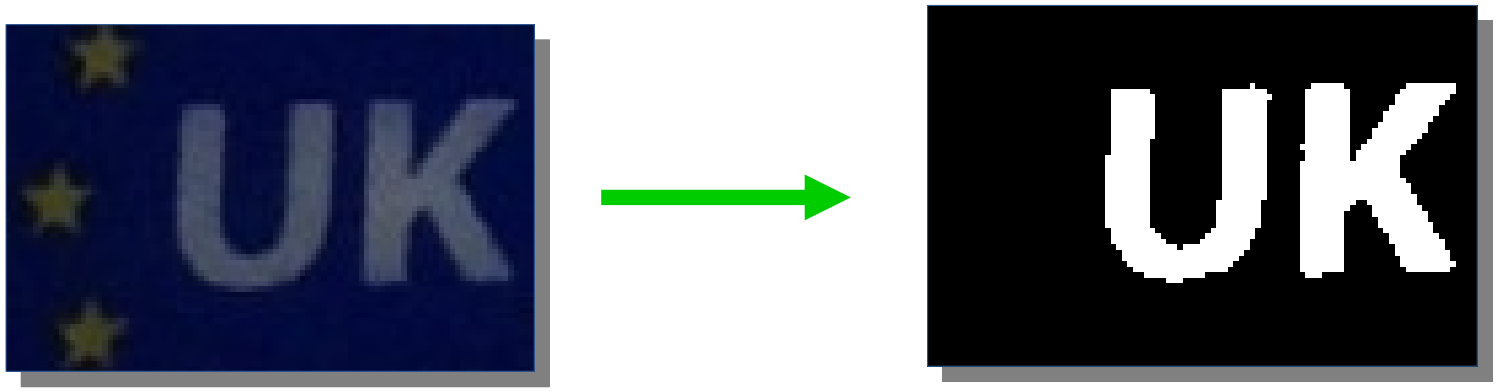


HSV Color model - S : v6

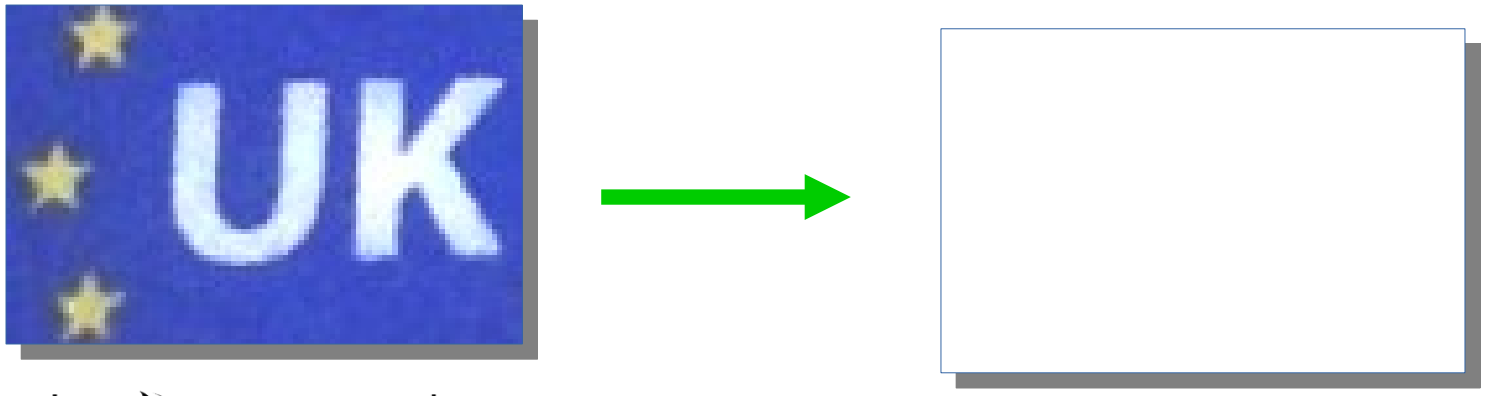




Thresholding : v7



HSV: Value = 80 - 255



Thresholding : v7

```
Core.inRange(
   img.getMat(),
   lowerBoundary, upperBoundary,
   destination);
```

OCR tesseract : v10

- UK

```
jety@jety-17:~$ tesseract /tmp/a.png stdout
UK
```

Memory

```
Mat mat = new Mat();

… DO WORK …

mat.release();
```

```
public static void releaseMats(Mat... mats) {
    for (Mat mat : mats) {
        if (mat!=null) mat.release();
    }
}
```

OCR tesseract : v10

- UK

```
jety@jety-17:~$ tesseract /tmp/a.png stdout
UK
```

Memory

```
Mat mat = new Mat();

… DO WORK …

mat.release();
```

```
public static void releaseMats(Mat... mats) {
    for (Mat mat : mats) {
        if (mat!=null) mat.release();
    }
}
```

Děkuji za pozornost.

Otázky?

www.**JavaDays**.cz

www.**gopas**.cz