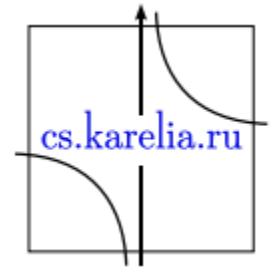




Петрозаводский государственный университет  
Кафедра информатики и математического обеспечения



## Сергей Сергеевич Куренчук

Детектирование и классификация дорожных знаков на цифровых  
графических изображениях

Промежуточный отчет о научно-исследовательской работе

Научный руководитель: ст. преподаватель А. В. Бородин

# Цель и задачи

## Цель работы

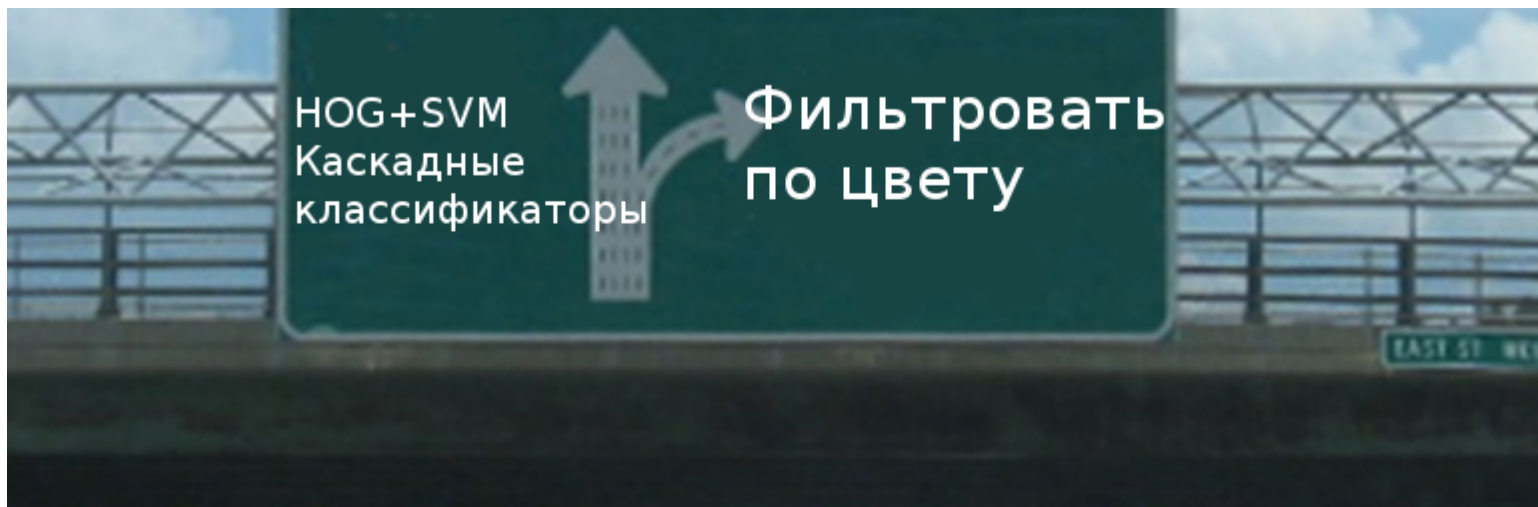
Разработать информационную систему, которая осуществляет детектирование и классификацию дорожных знаков на изображении.

## Задачи

- Изучит различные подходы к решению данной задачи
- Изучить существующие библиотеки электронного зрения и машинного обучения
- На их основе создать информационную систему для данной задачи



# Детектирование объектов

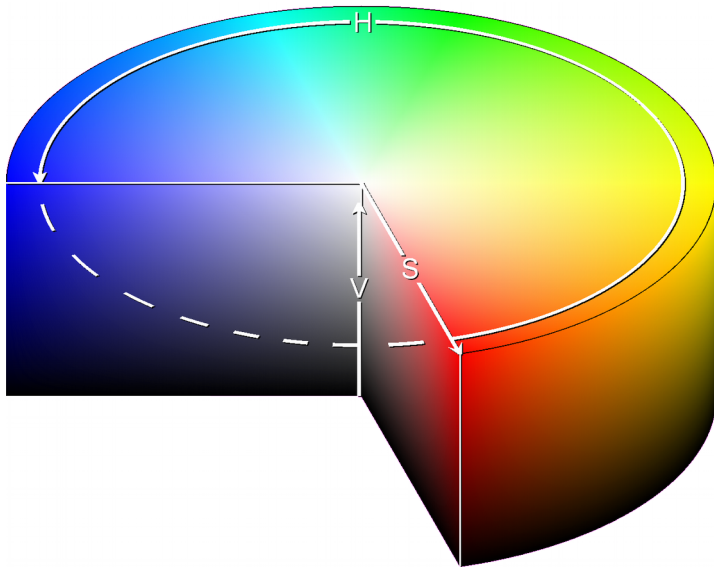


# HSV (цветовая модель)

**Hue** — цветовой тон. Варьируется в пределах  $0—360^\circ$

**Saturation** — насыщенность.

**Value** или **Brightness** — яркость



# Фильтрация по цветам

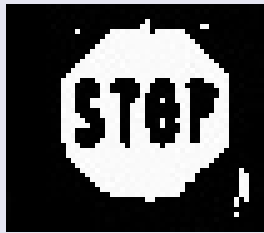


# Thresholding

Const Thresholding

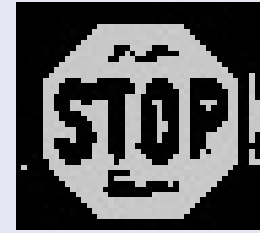


Left



Right

Adaptive Thresholding



Left

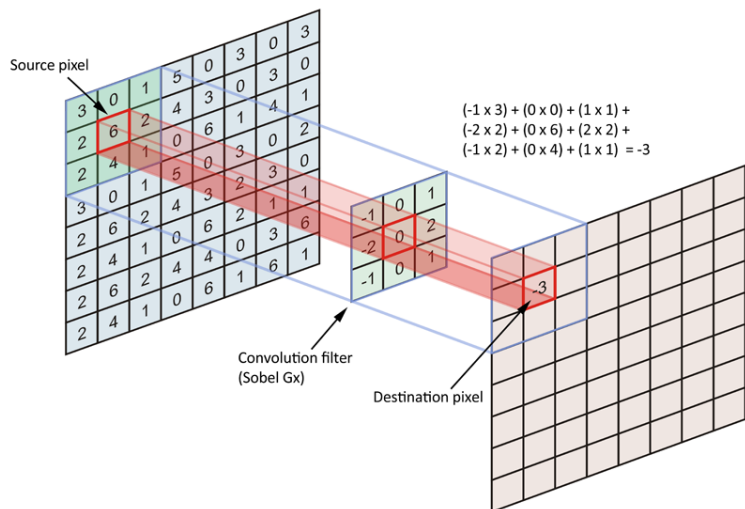


Right

Adaptive Threshold



# Convolution



В крайевых позициях значения:  
 Отражаются  
 Дублируются  
 Имеют рамку одного цвета



Пример

$\Leftarrow$

-1	-1	-1
-1	7.5	-1
-1	-1	-1

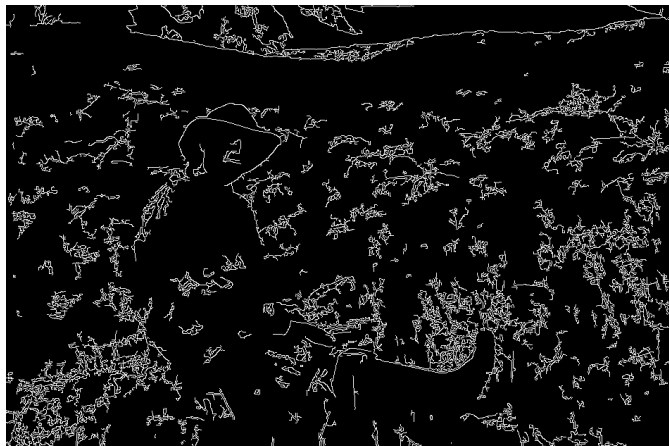




## Sobel operator

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

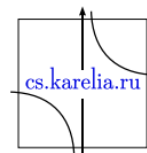
## Canny



Пример

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Gaussian filter





# Canny edge detector

## Пример

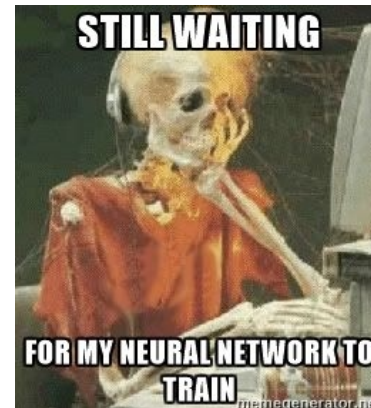
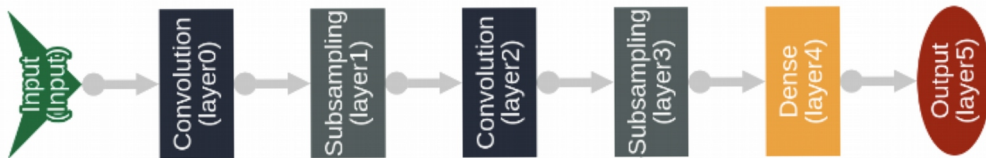
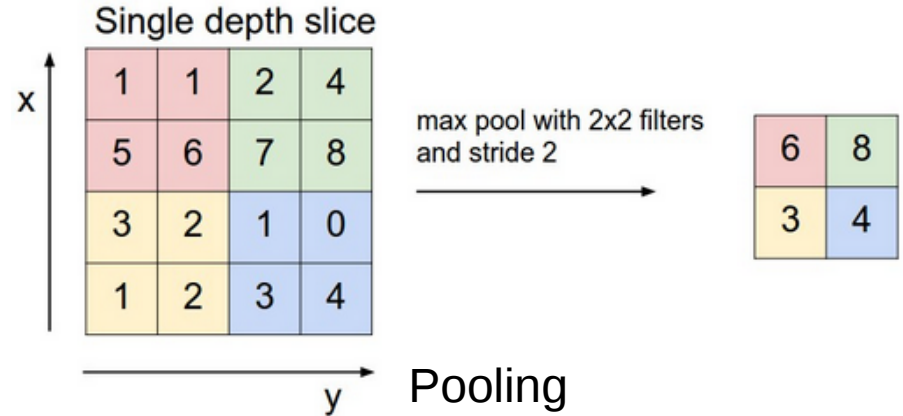
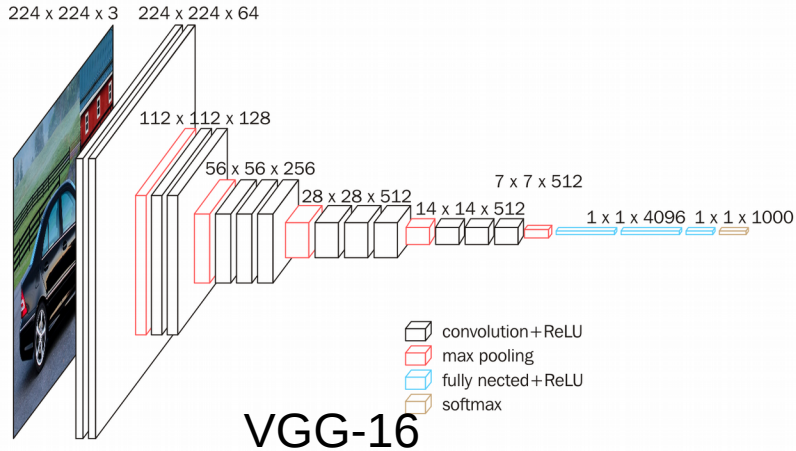


## Пример



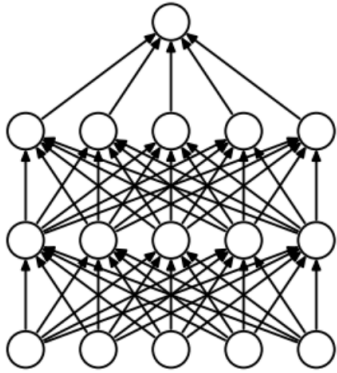
# Classification problem, CNN

## Convolutional neural network

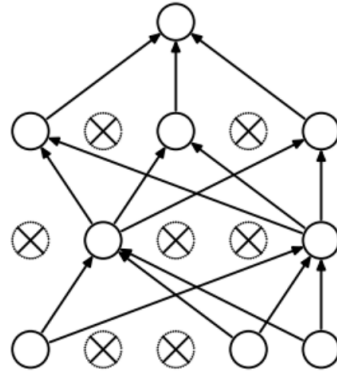


# Classification problem, CNN

## DropOut



(a) Standard Neural Net



(b) After applying dropout.

### Inverted

$$\text{Train: } 1/q * X_i * a\left(\sum_{k=1}^N w_k * x_k + b\right)$$

$$\text{Test: } a\left(\sum_{k=1}^N w_k * x_k + b\right)$$

a — Функция активаций

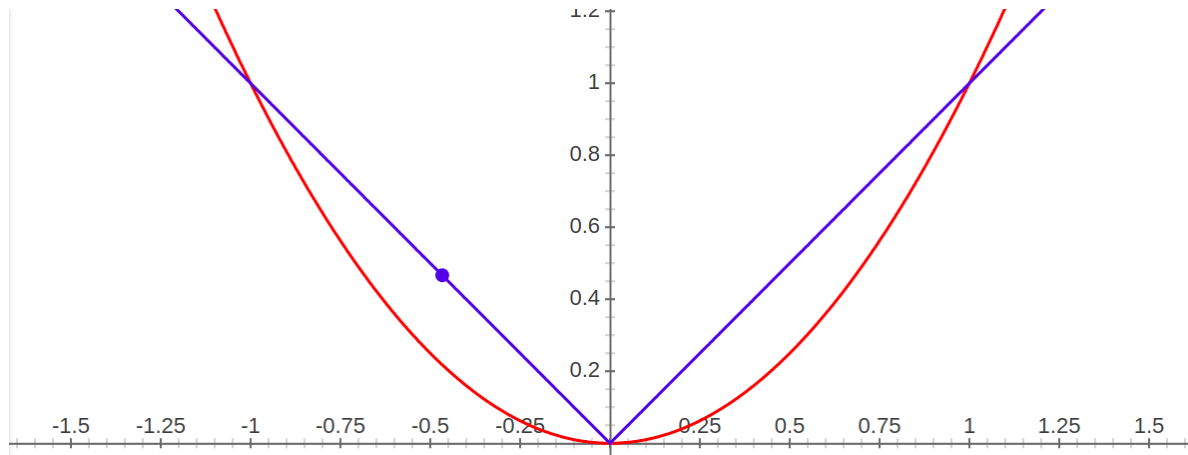
X — Случайное значение

Q — Вероятность использования нейрона в текущей эпохе



# Classification problem, CNN

## L1 & L2 regularization



$$\text{L2: } \sum_{k=1}^N w_k^2$$
$$\text{L1: } \sum_{k=1}^N |w_k|$$

## Normalize data

$$\text{value} = \frac{\text{value} - \text{min}}{\text{max} + \text{min}} \quad 0..255 \Rightarrow 0..1$$

$$\text{data binarization} \quad \begin{cases} 1, \text{value} \geq 128 \\ 0, \text{else} \end{cases}$$

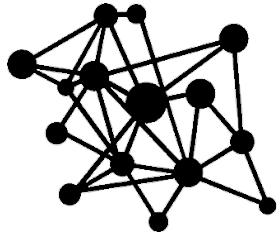


## OpenCV



OpenCV (Open Source Computer Vision Library). It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android.

## Deeplearning4j



# DL4J

Deeplearning4j is a deep learning programming library written for Java and the Java virtual machine (JVM) and a computing framework with wide support for deep learning algorithms.

