

# Clear Cut: Algorithm Design

September 30, 2018

## Contents

<b>1</b>	<b>Input data</b>	<b>2</b>
1.1	Image size reduction . . . . .	2
<b>2</b>	<b>Edge detection procedure</b>	<b>2</b>
2.1	Dealing with channels . . . . .	2

# 1 Input data

The input data is an image of arbitrary size,  $M \times N$ . Most images have three channels (RGB = “Red-Green-Blue”), this means an array with three  $M \times N$  arrays. Images with exif data are checked and modified so that the image is imported in the intended orientation, e.g. in case the image is a photo taken from a phone camera.

## 1.1 Image size reduction

To improve the efficiency of the edge detection algorithm, the image size is reduced to that the average image dimension is less than 500 pixels in length. The final image size to be thrown into the algorithm is  $M_{\text{eff}} \times N_{\text{eff}}$ , where

$$\frac{M_{\text{eff}}N_{\text{eff}}}{2} < 500 \text{ pxl.} \quad (1)$$

The image is currently reduced slowly by max pooling. This relies on calculating the smallest kernel size and repeating the max pooling process until the condition in Equation 1 is satisfied. This can be implemented more effectively by calculating the smallest kernel that would satisfy the condition in Equation 1 after just one implementation of max pooling.

As it currently exists, the process consists of determining the smallest number that divides the height (width) of the image to return an integer. If there is no such factor, i.e. the height (width) is a prime number, the image is cropped by removing the single pixel row (column) at the  $M_{\text{eff}}^{th}$  ( $N_{\text{eff}}^{th}$ ) index.

# 2 Edge detection procedure

## 2.1 Dealing with channels

Each of the RGB channels have the edge detection algorithm applied to them.

<b>DevOps</b>	Best practices and administration of the Atlassian Suite, e.g. Jira, Confluence
<b>Non-functional Testing</b>	Performance testing of a variety of applications (e.g. HPE LoadRunner, JMeter)
<b>Automation/Scripting</b>	Developed two software solutions using Java SE 8 and Excel’s VBA macro
<b>API Programming</b>	Utilising RESTful services from online documentation within scripts
<b>Deep Learning</b>	Using Google’s TensorFlow framework (python) for predictive analytics