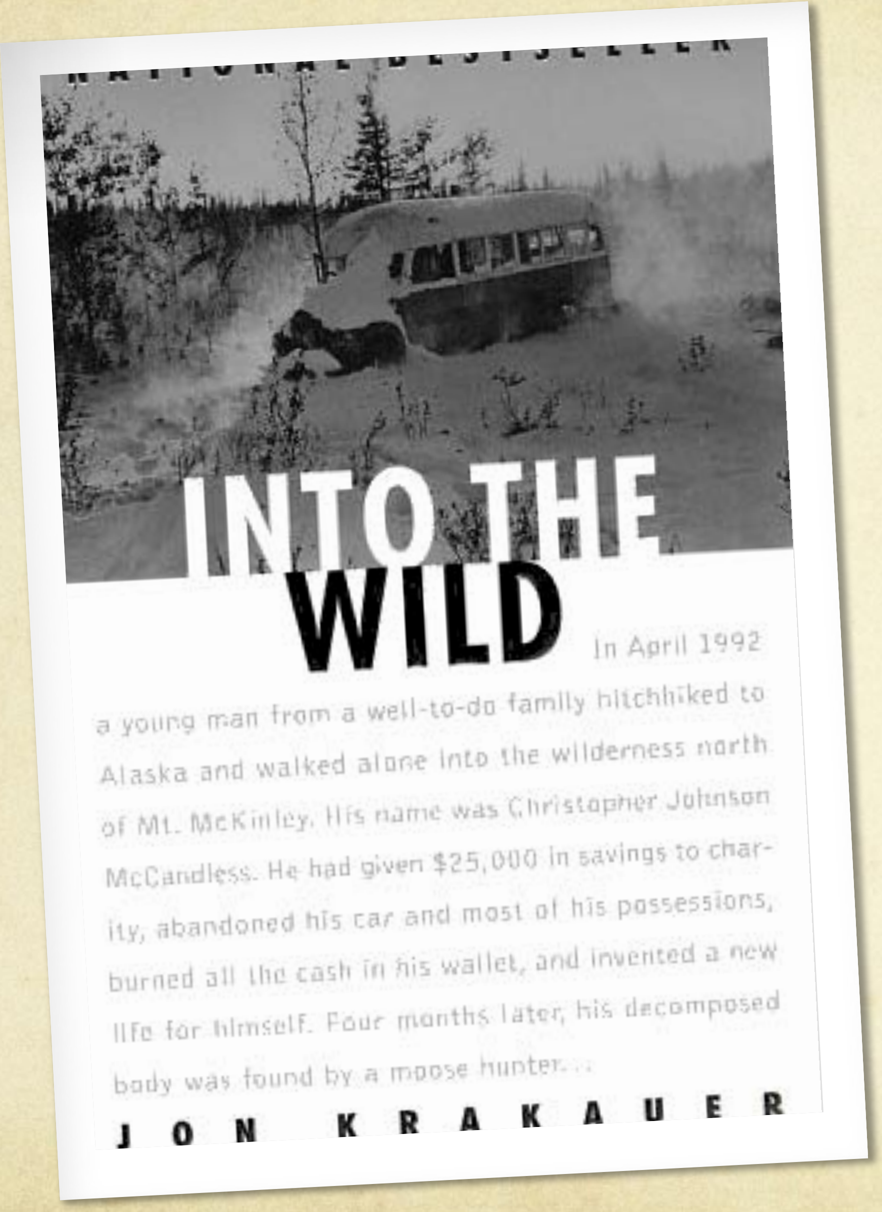
INTO THE WILD

Characterness: An Indicator of Text in the Wild

Fortunati Stefano
Gagni Mirko
Gargiulo Claudia

INTRODUCTION

Text attracts instinctively human attention and it is useful in several contests.



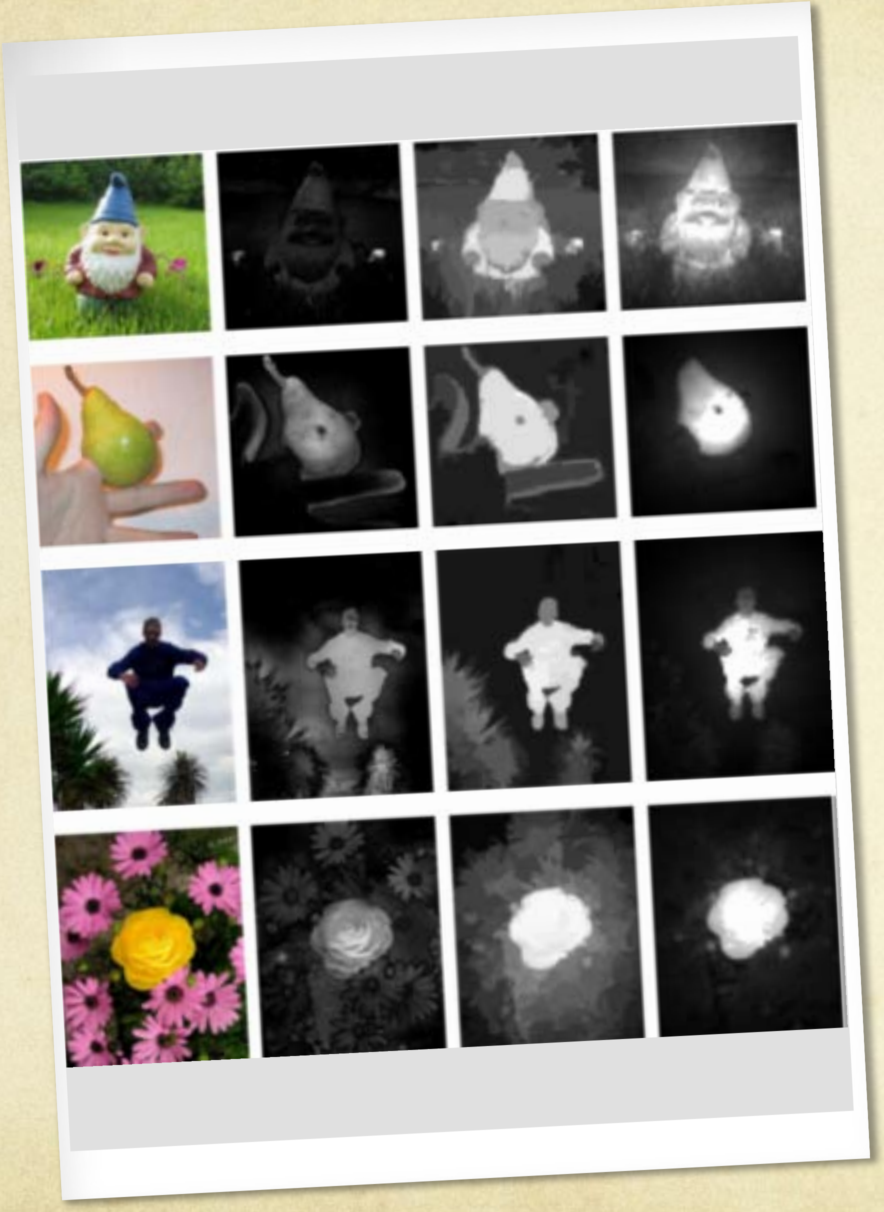
Algorithm Working Principle:

- Characterness:

  Salient Objects Detection

- Objectness:

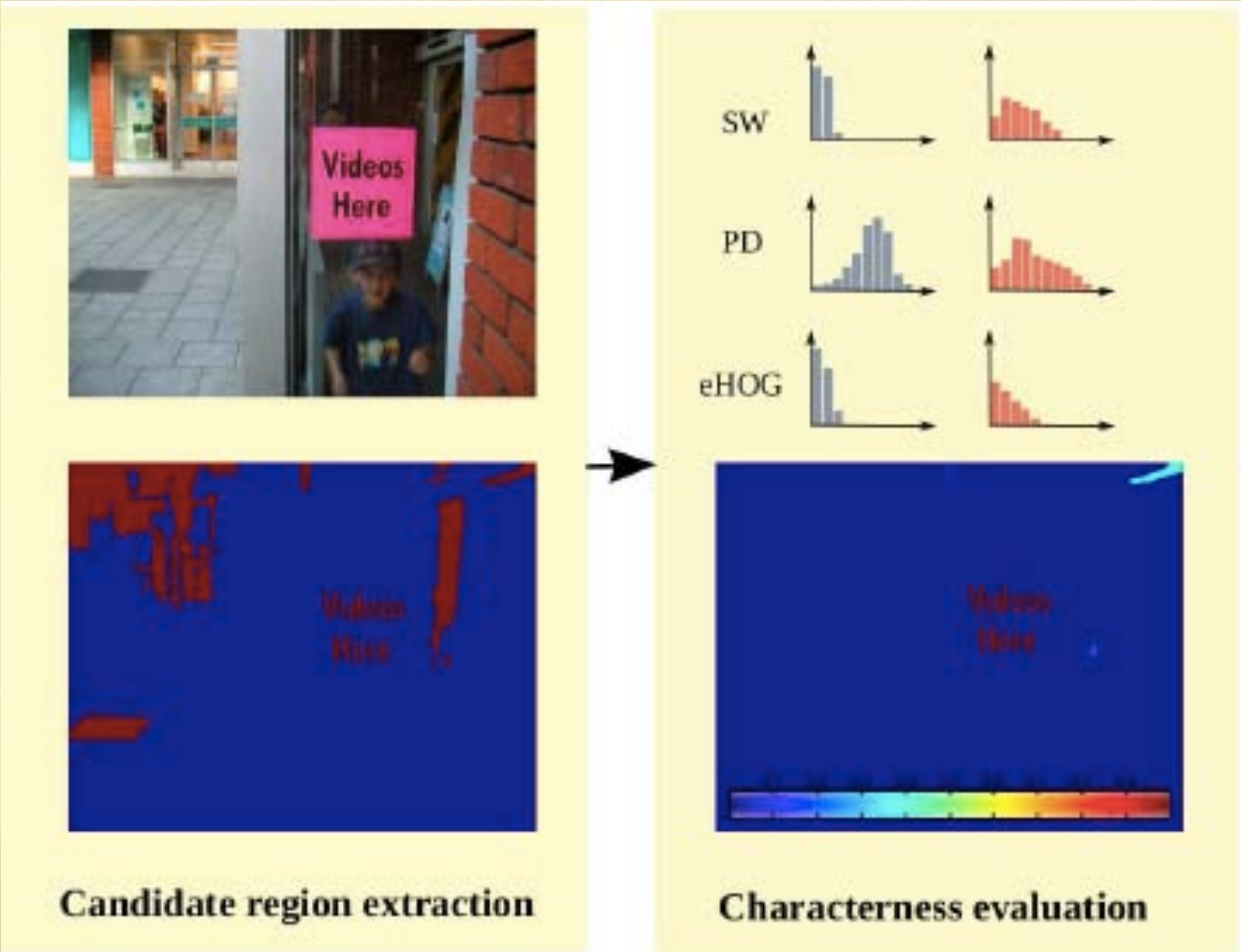
  Interesting Features Identification

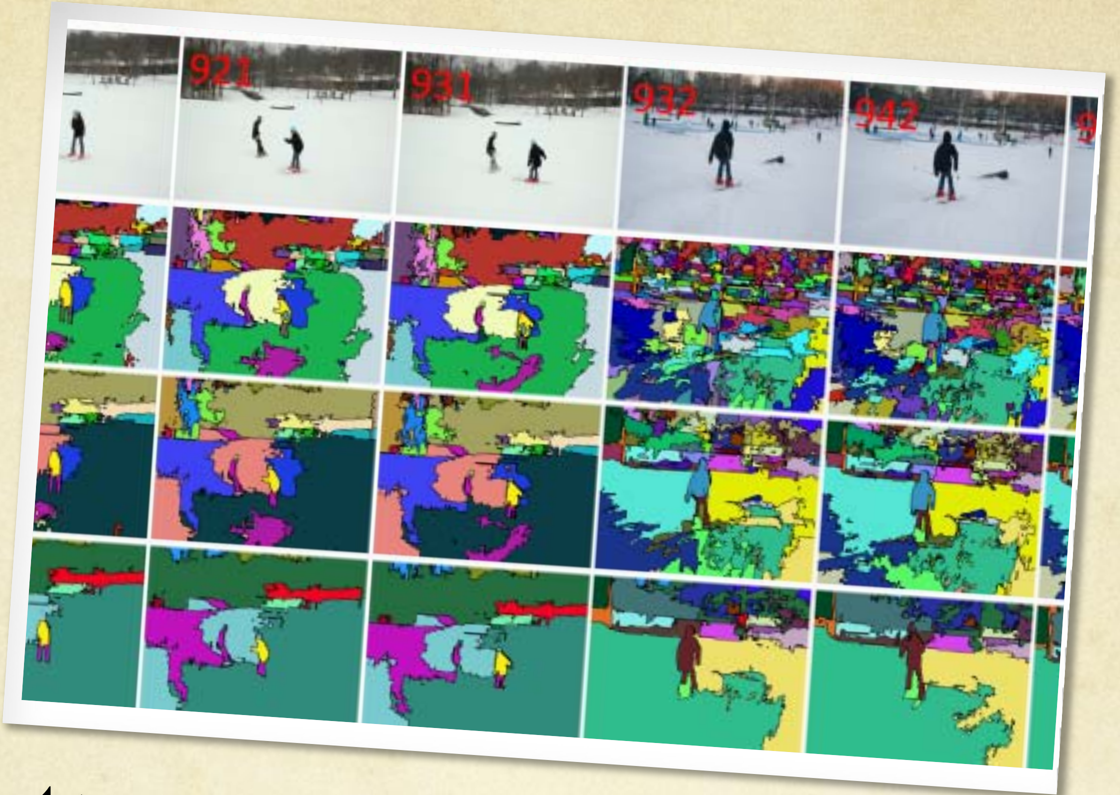


Characterness

3 possible approaches:

- ✓ Texture-based: extract distinct texture properties from sliding windows, and use a classifier to detect individual instances.
- ✓ Region-based: first extract potential characters low level features based on geometric and shape constraints are used to reject non-characters.
- ✓ Hybrid

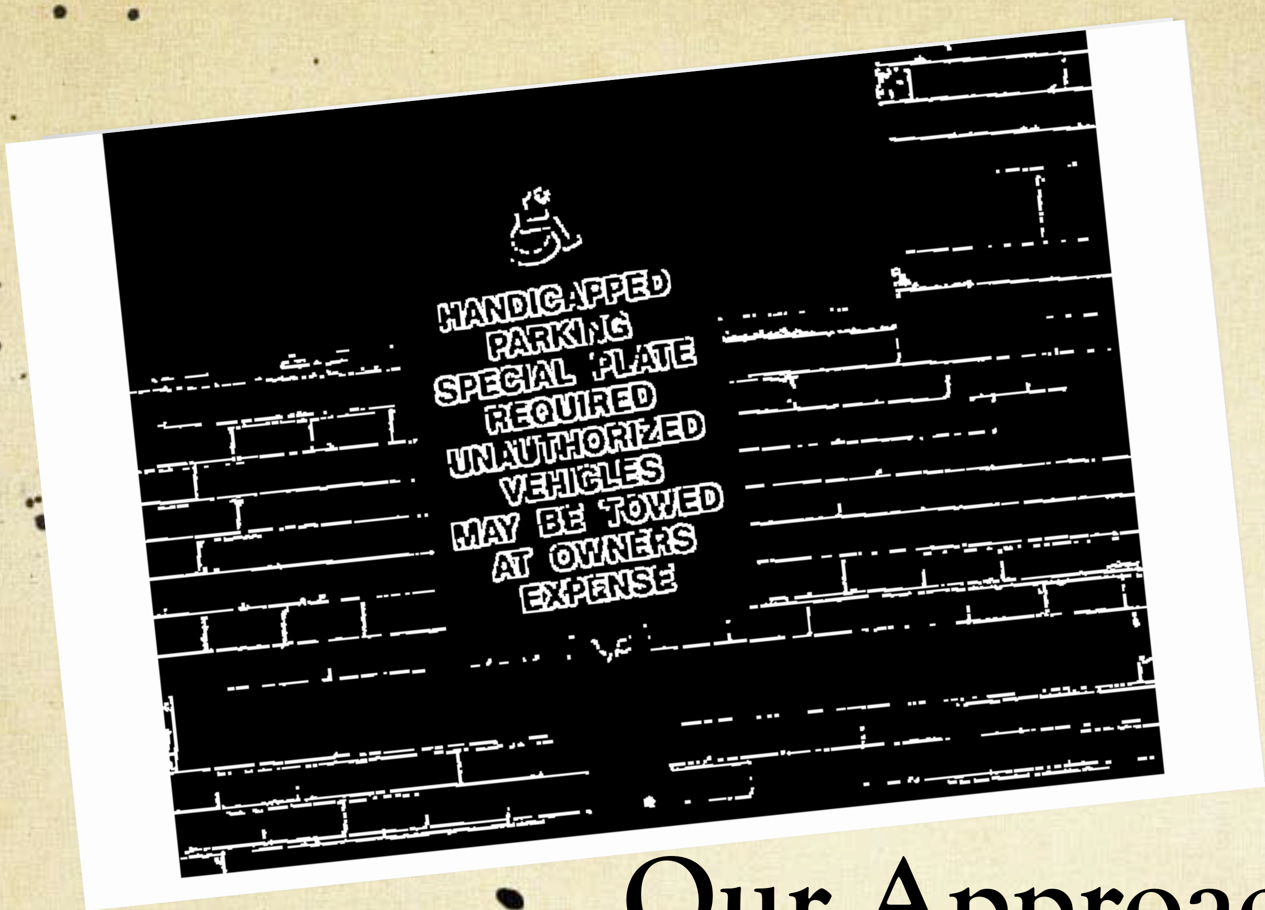

Videos
Here
SW
PD
eHOG
Candidate region extraction
Characterness evaluation



Objectness

Evaluate the contrast between the salient object and the background.
2 methods categories:

- ✓ Local: estimate saliency of an image patch according to its contrast against its surrounding patches.
- ✓ Global: take the entire image into account when estimating saliency of a particular patch.



Our Approach:

① Use EMSER to identify salient regions
② Evaluate with three different characterness cues the regions content
③ Sum up the results using Bayes probability theorem

MSER

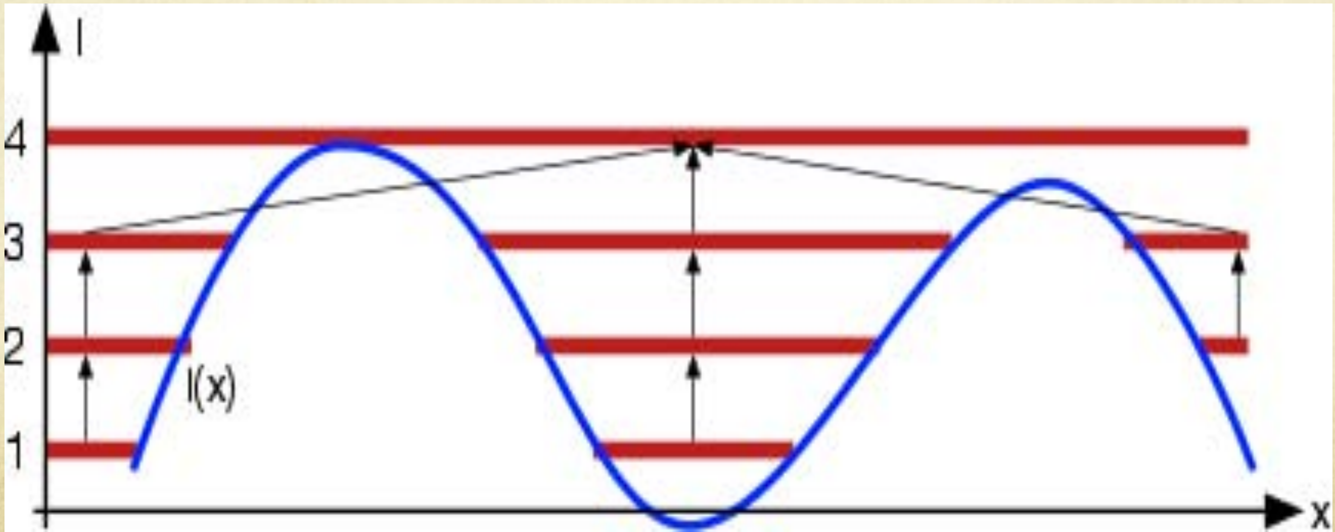
Standard Method

Maximally Stable Extremal Regions

- ✓ It's an effective region detector
- ✓ It's particularly well suited to identify regions with almost uniform intensity surrounded by contrasting background.

Stable Regions [2]

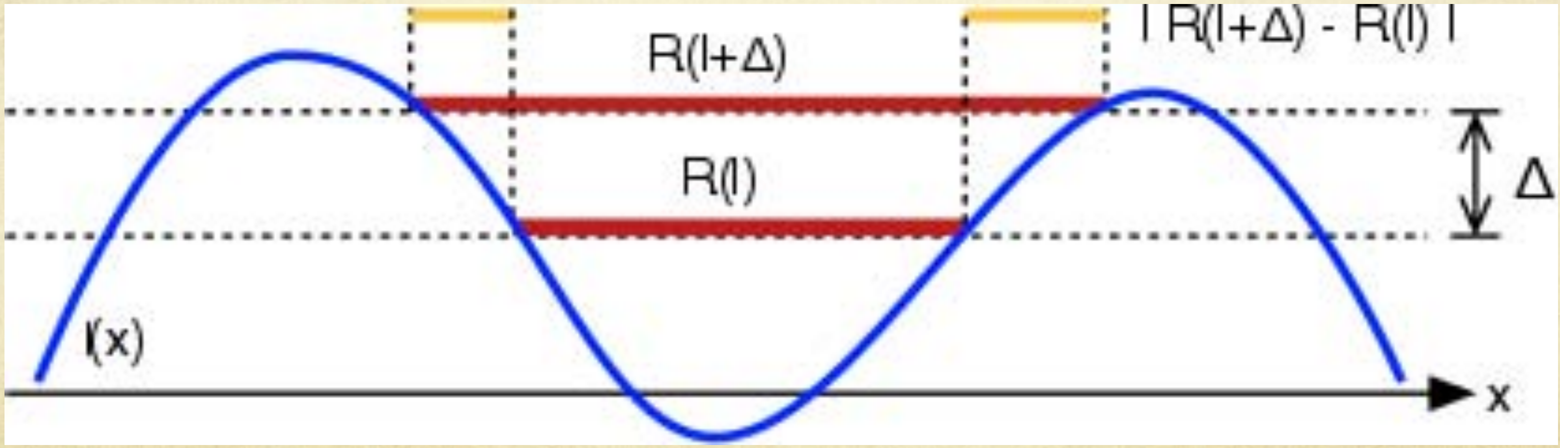
- ✓ An Extremal Region is a connected components of the level sets S_l={ x : I(x) ≤ l }, l ∈ R of the image I(x).
- ✓ They can be arranged in a tree, where a region R_l is a children of a region R_l+1 if $R_l \subset R_l$+1.
- ✓ A Stable Extremal Region is an extremal region that does not change much as the index l is varied



Score of instability - MSER

- Let $B(R_l) = (R_l, R_{l+1}, \ldots, R_{l+\Delta})$ be the branch of the tree $R_l \subset R_{l+1} \subset \cdots \subset R_{l+\Delta}$ rooted at R_l.
- We associate to the branch the (in)stability score:

$$v(R_l) = \frac{|R_{l+\Delta} - R_l|}{|R_l|}$$



MSER - Refine

Other parameters improve the performance of the algorithm:

- ✓ Max Area e Min Area : exclude MSERs too small or too big
- ✓ Max Variations : exclude MSERs too unstable.
- ✓ Min Diversity : remove duplicated MSERs.

Parameter	**Symbol**	**Value Used**
Delta	Δ	10
Max Area	a+	0.1
Min Area	a−	0.00002
Max Var	v+	0.2
Min Diversity	d+	0.5

Results

VL Feat [3]

- ✓ **VLFeat** is a cross-platform open source collection of vision algorithms with a special focus on visual features (for instance SIFT and MSER) and clustering
  - MATLAB toolbox,
  - C library
  - command line utilities.

- ✓ VLFeat was created by Andrea Vedaldi and Brian Fulkerson in 2007;

- ✓ Also OpenCV [4] implements the same functions by means of an apposite Class (C++ version)

Why Not OpenCV

(Bright On Dark)

OpenCV

Vl_Feat

Why Not OpenCV

Dark On Bright

OpenCV

Vl_Feat

Problems

- ✓ **BLUR – SENSITIVITY !**
  - may lead to incorrect character boundaries.
  - Some characters can be are either missed or incorrectly connected.

- ✓ Solutions : we use **edge-preserving MSER algorithm**

E - MSER

Modified Method to Detect Regions

Edge-preserving MSER

Algorithm 1: Edge-preserving MSER (eMSER)

Input: A color image, and required parameters

Output: Regions contain characters and non-characters

1. Convert the color image to an intensity image I.
2. Smooth I using a guided filter
3. Compute the gradient amplitude map ∇I, then normalize it to $[0, 255]$.
4. Get a new image $I^* = I + \gamma \nabla I$ (resp. $I^* = I - \gamma \nabla I$).
5. Perform MSER algorithm on I^* to extract dark regions on the bright background (resp. bright regions on the dark background).

NOTE: This implies that for each studied image we will obtain two results: the bright on dark and the dark on bright.

Guided Filter [5]

- Input image p, output image q, guidance image I:

$$q_i = \sum_j W_{ij}(I) p_j$$

- Kernel weights are defined as:

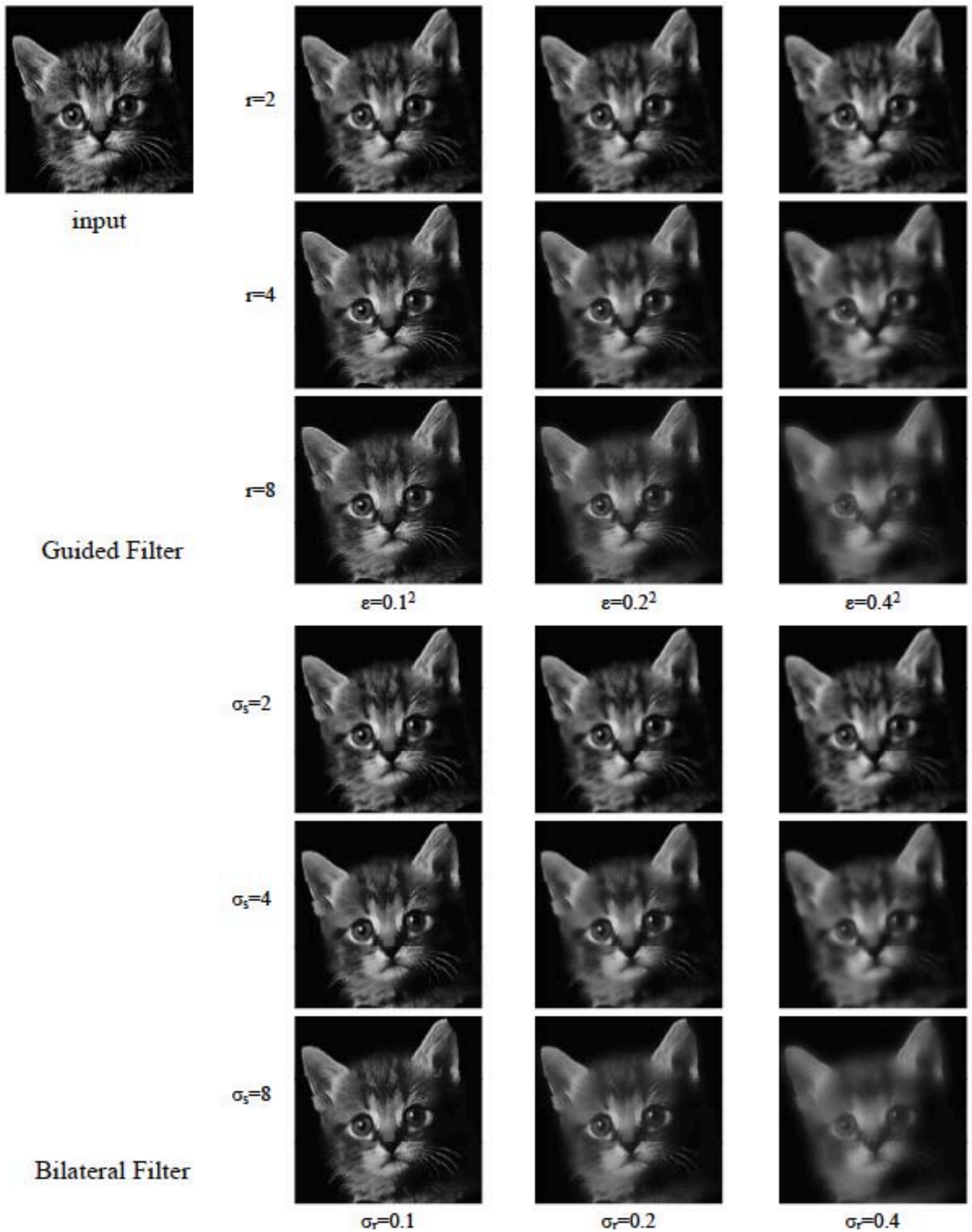
$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k::(i,j)\in\omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}\right)$$

Where μ_k and σ^2_k are mean and variance of I in the window w_k, $|w|$ is the number of pixels in w_k and ε is a regularization parameter preventing coefficients to be too large.

The Filtering Problem

Guided Filter

Bilateral Filter



Results for the two filters with different parameters.

Differences

- ✓ Both filters preserve the edges positions because of their linearity;
- ✓ Bilateral filter is easier to be implemented, the guided one involves matrices operations;
- ✓ Guided filter has been developed to optimize some detection purpose;
- ✓ Edges rendering is very different (main problem!)


Original
Guided Filter
Bilateral Filter


Original HDR
Guided Filter
Bilateral Filter

Better Performance (Bright On Dark)

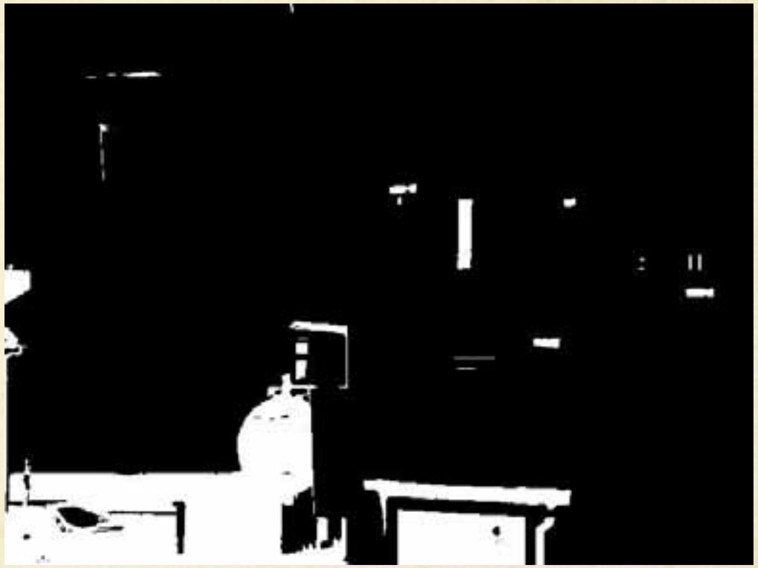
MSER

eMSER

Better Performance (Dark On Bright)

MSER

eMSER



CUES

CUES

Methods to detect text regions

Detection of characters

- ✓ Characters attract human attention

- ✓ Methods to measure their unique properties:
  - Stroke width (SW)
  - Perceptual divergence (PD)
  - Histogram of gradients at edges (eHOG)

- ✓ Bayesian multi-cue integration

LEARNING DISTRIBUTIONS

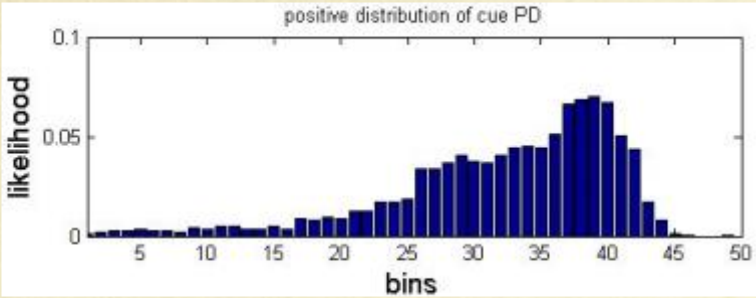
The Training Set

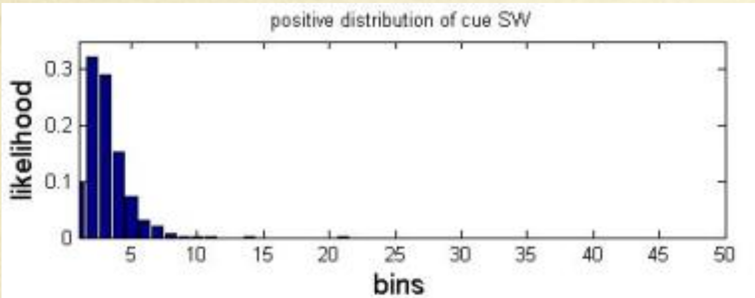
Learning distributions

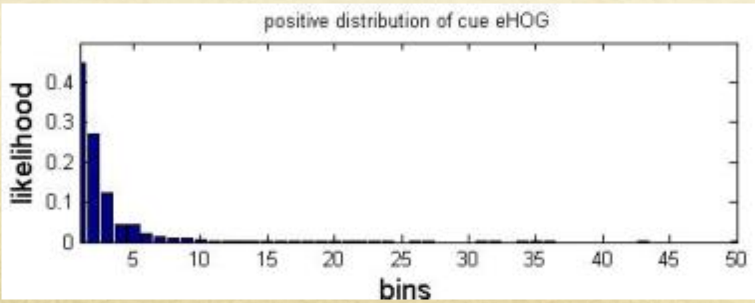
- ✓ ICDAR 2013 contains 229 images harvested from natural scenes
- ✓ Distribution of cue on positive samples:
  - Directly applying three cues
- ✓ Distribution of cue on negative samples:
  - eMSER applied twice
  - Erasing ground truth characters
  - The rest of regions are considered negative on which compute three cues.

Learning distributions

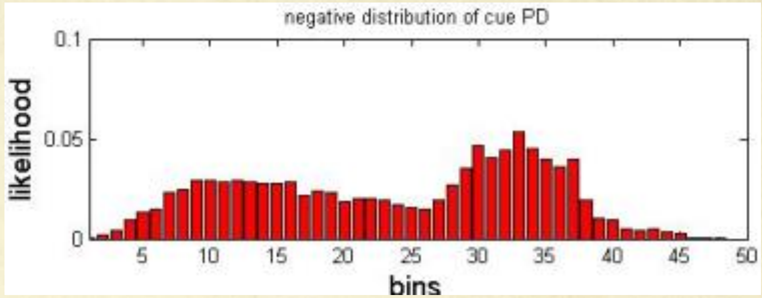
Positive distributions

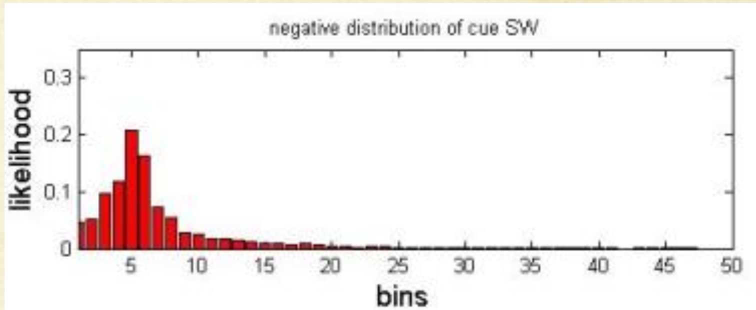


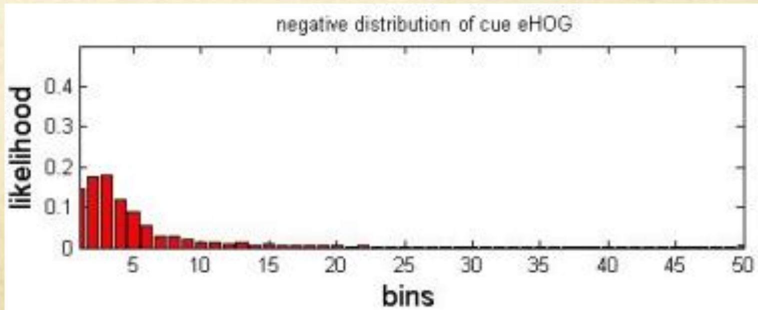




Negative distributions

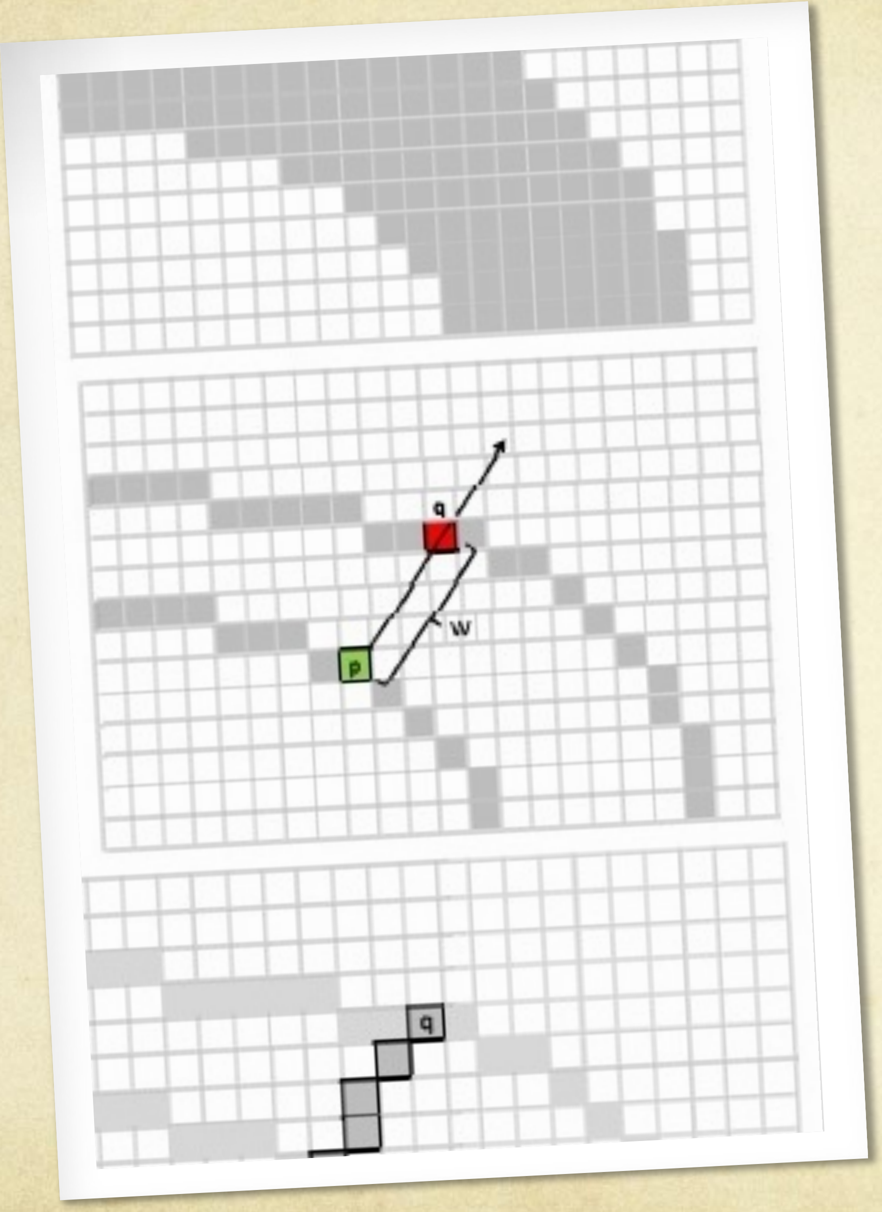






SW

Stroke Width



Stroke width

(general idea) [6]

✓ Compute length of a straight line between two edge pixels in perpendicular direction.

✓ Definition: a STROKE is a connected region with uniform color and half-closed boundary

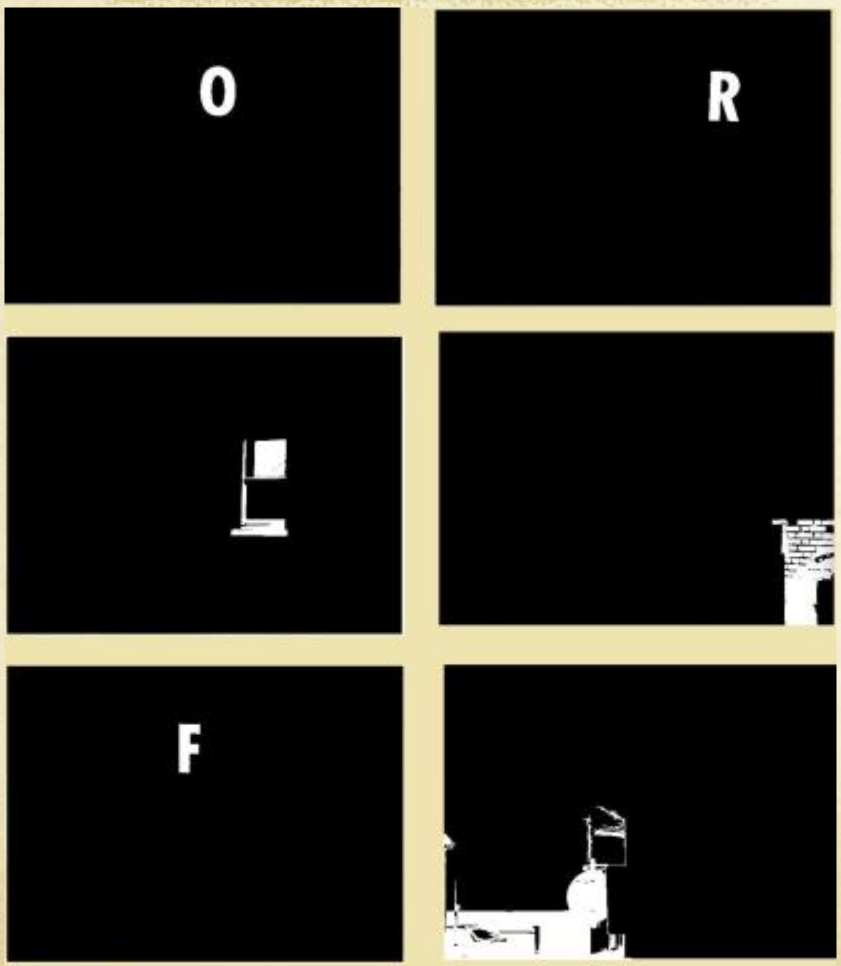
✓ Compute SW(r) = $\frac{var(r)}{E(r)^2}$

or $\frac{std\_dev(r)}{E(r)}$

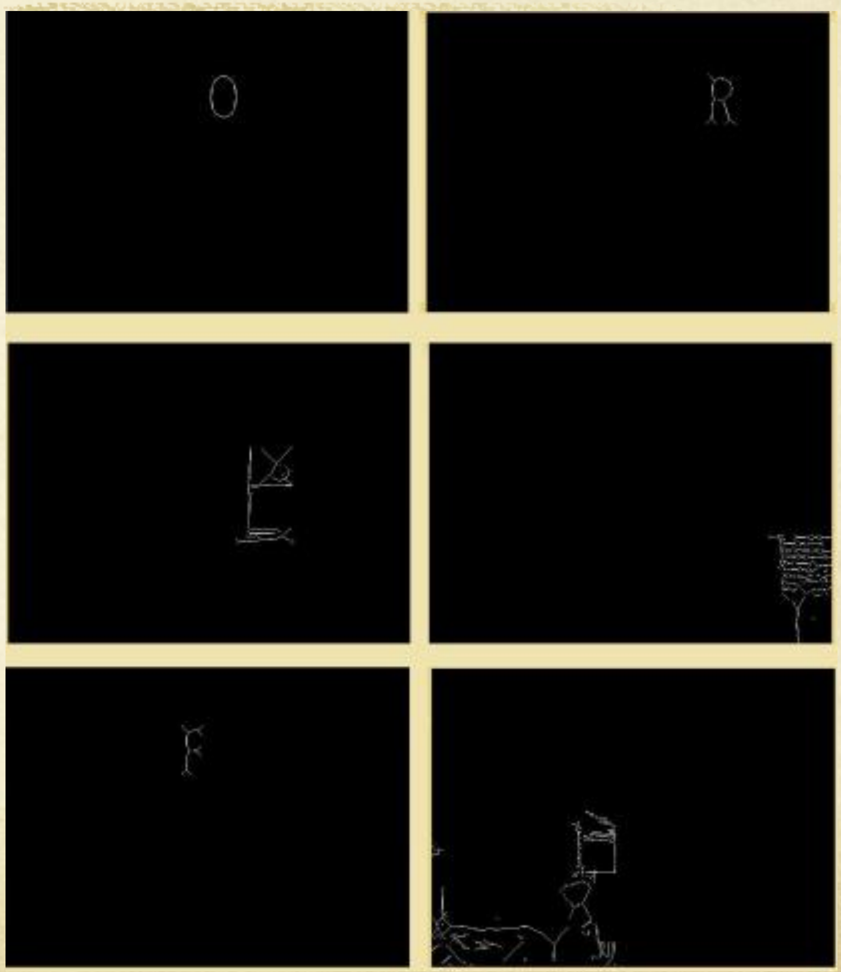
✓ Evaluation of this parameter

Stroke width [7]

- ✓ Select a single region "r"



- ✓ Extract the skeleton S for the region



Stroke width

- $\forall\, p \in S$ compute shortest path to boundary region

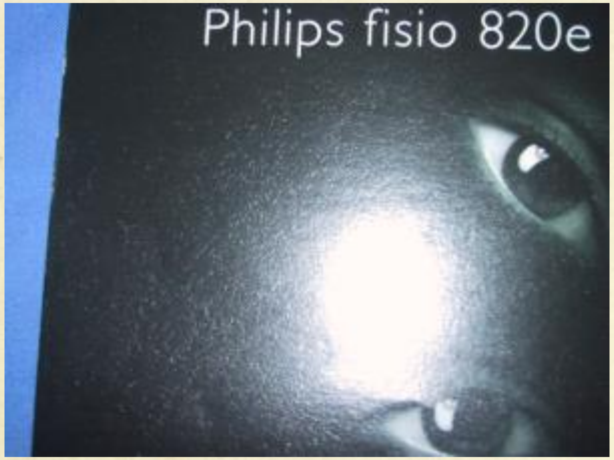
↓

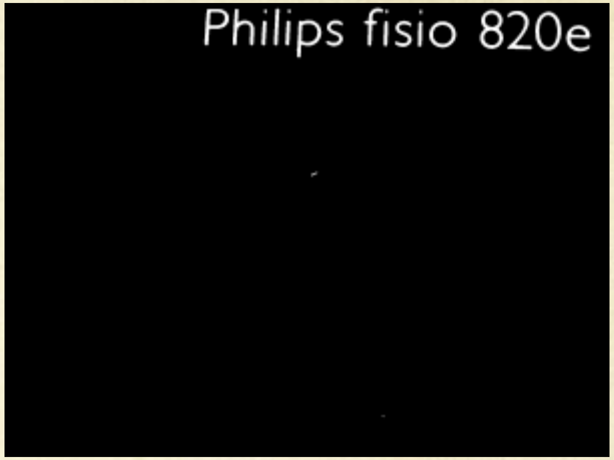
compute stroke width

✓ Output: $SW(r) = \dfrac{var(r)}{E(r)^2}$ or $\dfrac{std\_dev(r)}{E(r)}$

Text region	Non-text region
F = 0.120302	Big region = 1.79409
O = 0.0121291	Bricks = 1.17305
R = 0.0660847	Window = 0.938649

Some stroke width results...





Some stroke width results…

PD

Perceptual Divergence

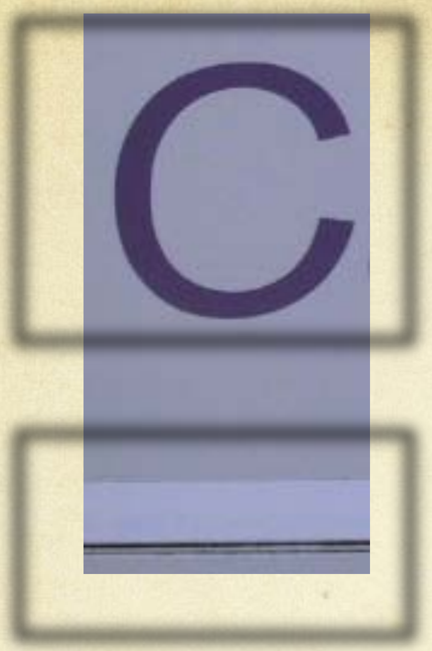
Perceptual Divergence

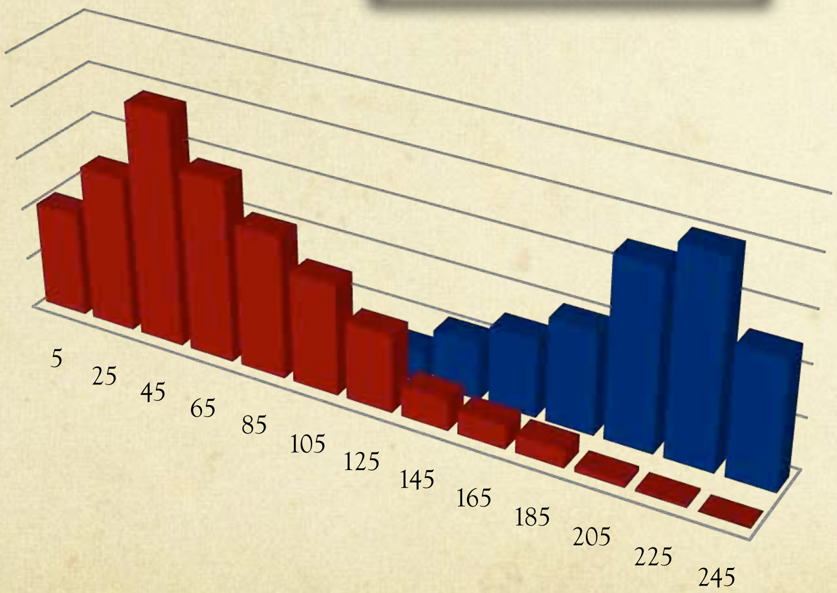
(general idea)

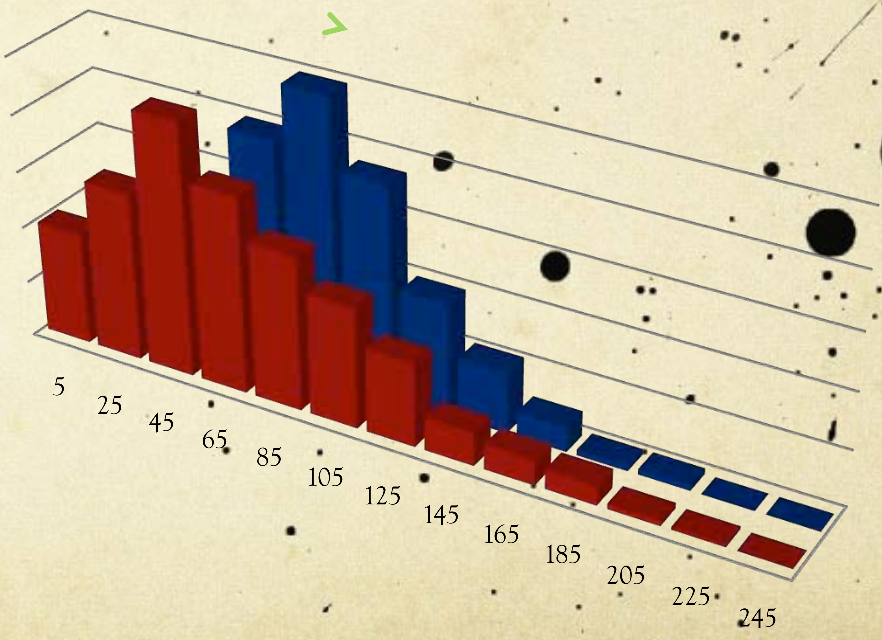
- ✓ Based on Color Contrast
- ✓ It's observable that, in order to ensure reasonable readability of text to a human, the color of text in natural scenes is typically distinct from that of the surrounding area.

Algorithm Step

Compute the Histogram of each colour for the region and the surrounding area



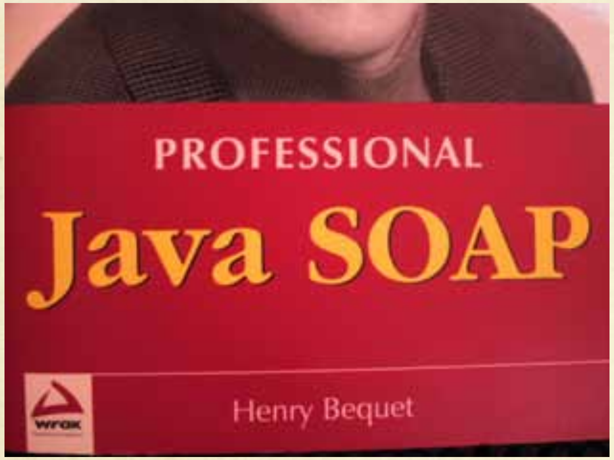




Use KLD to Compute Dissimilarity beetween the Histograms

Some PD results…

Some PD results…



EHOG

Histogram of Oriented Gradients at Edges

Histogram of Oriented Gradients [8]

- ✓ It counts occurrences of gradient orientation in localized portions of an image, called cells.

- ✓ Apply a 1-D derivative mask centered in the point of interest in both horizontal and vertical directions : [-1,0,1] and $[-1,0,1]^T$. [9]

- ✓ Create the cell histograms: each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation.

- ✓ The histogram channels are spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is "unsigned" or "signed".

E-HOG

The HOG feature is applied only on the edges of the image.

Edges can be found applying the Canny method.

What about text.. [10]

- ✓ In letters the orientations of edge pixels is usually equi-divided into 4 groups, defined by 45°+k90°, k=0,1,2,3.

- ✓ The EHOG parameter value is computed for each region found using the following formula:

$$eHOG(r) = \frac{\sqrt{(\omega_1(r) - \omega_3(r))^2 + (\omega_2(r) - \omega_4(r))^2}}{\sum_{i=1}^{4} \omega_i(r)}$$

Where $w_1(r)$, $w_2(r)$, $w_3(r)$, $w_4(r)$ are, respectively, the number of pixel in the four groups previously defined contained in the region.

Some EHOG result…

Some EHOG result…

BAYES

Summing Up Cues

Bayes Theorem

- ✓ It allows to sum up the results of the three cues using the following formula:

$$p(c|\Omega) = \frac{p(\Omega|c)p(c)}{p(\Omega)} = \frac{p(c)\prod_{cue\in\Omega} p(cue|c)}{\sum_{k\in\{c,b\}} p(k)\prod_{cue\in\Omega} p(cue|k)}$$

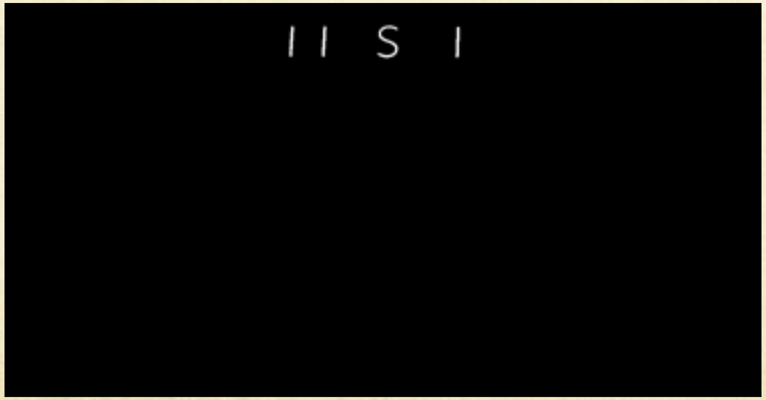
- ✓ Here **Ω** = {SW, PD, EHOG}, p(c) and p(b) denotes the prior probability of characters and background (non-characters) respectively (0.7 and 0.3).

Some results…

An example with 3 techniques

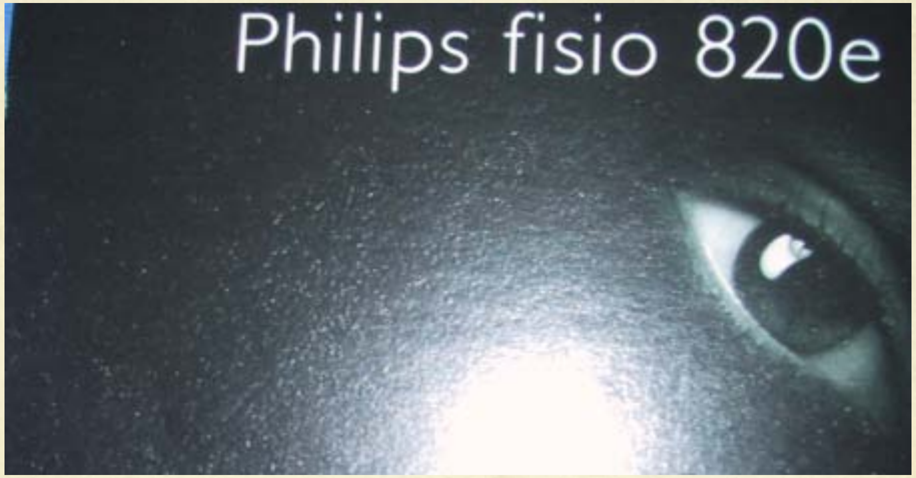
Stroke width

Perceptual divergence



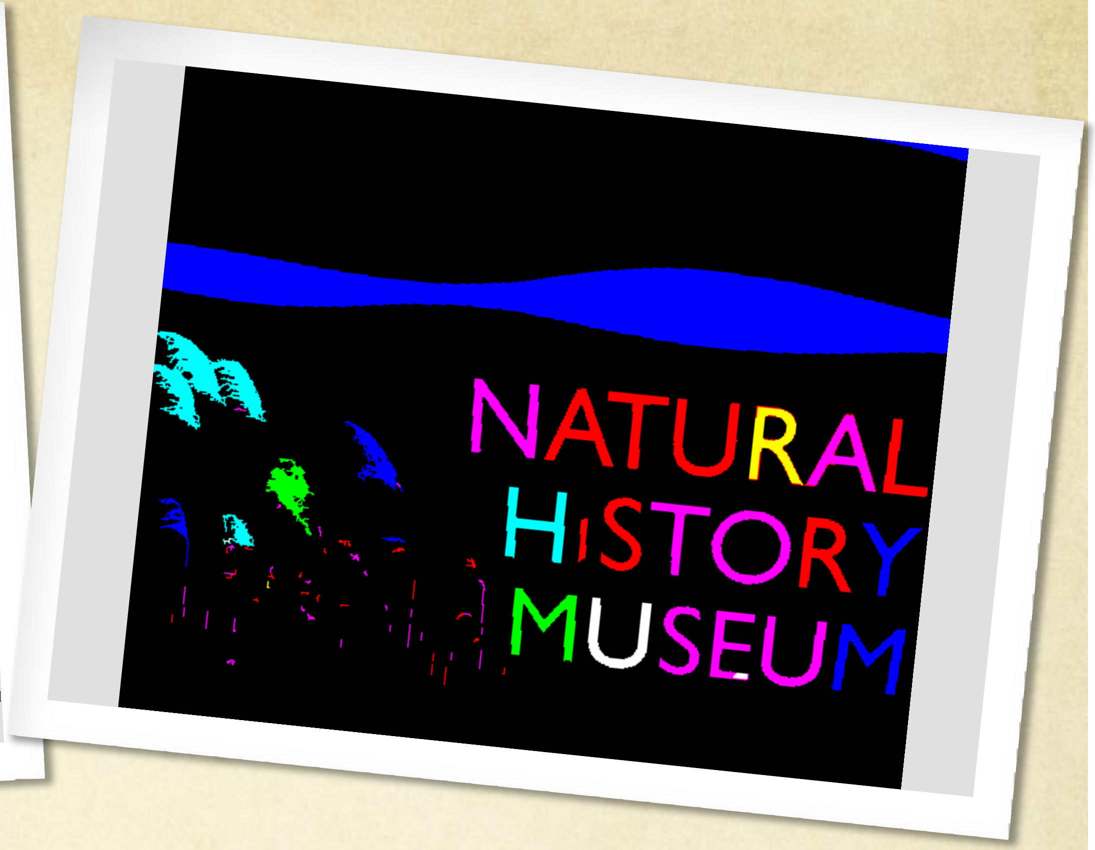
eHOG

The result



Original

Bayes result



Some other COLOURED results..

Key:

- ✓ Red: SW
- ✓ Green: EHOG
- ✓ Blue: PD

→

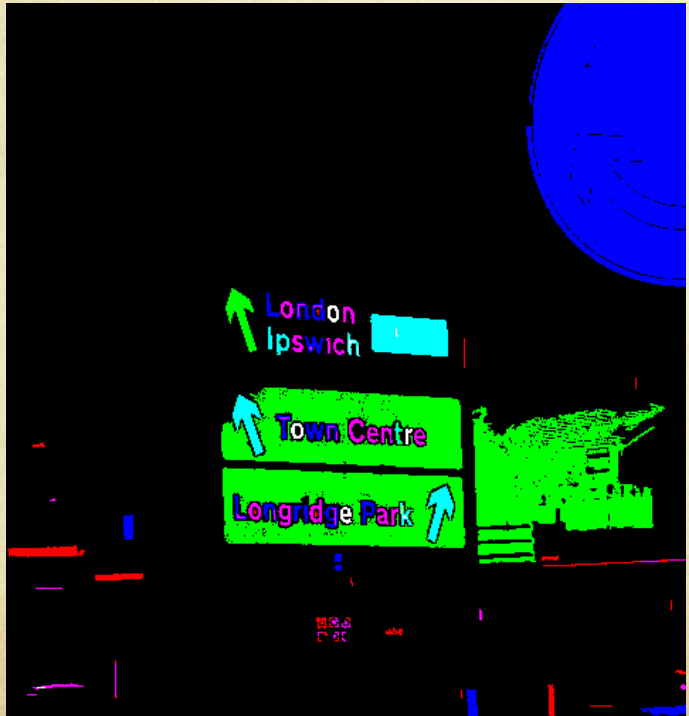
- ✓ Yellow: SW + EHOG
- ✓ Pink: SW + PD
- ✓ Light blue: PD + EHOG

Key:

- ✓ Red: SW
- ✓ Green: EHOG
- ✓ Blue: PD

→

- ✓ Yellow: SW + EHOG
- ✓ Pink: SW + PD
- ✓ Light blue: PD + EHOG



Key:

- ✓ Red: SW
- ✓ Green: EHOG
- ✓ Blue: PD

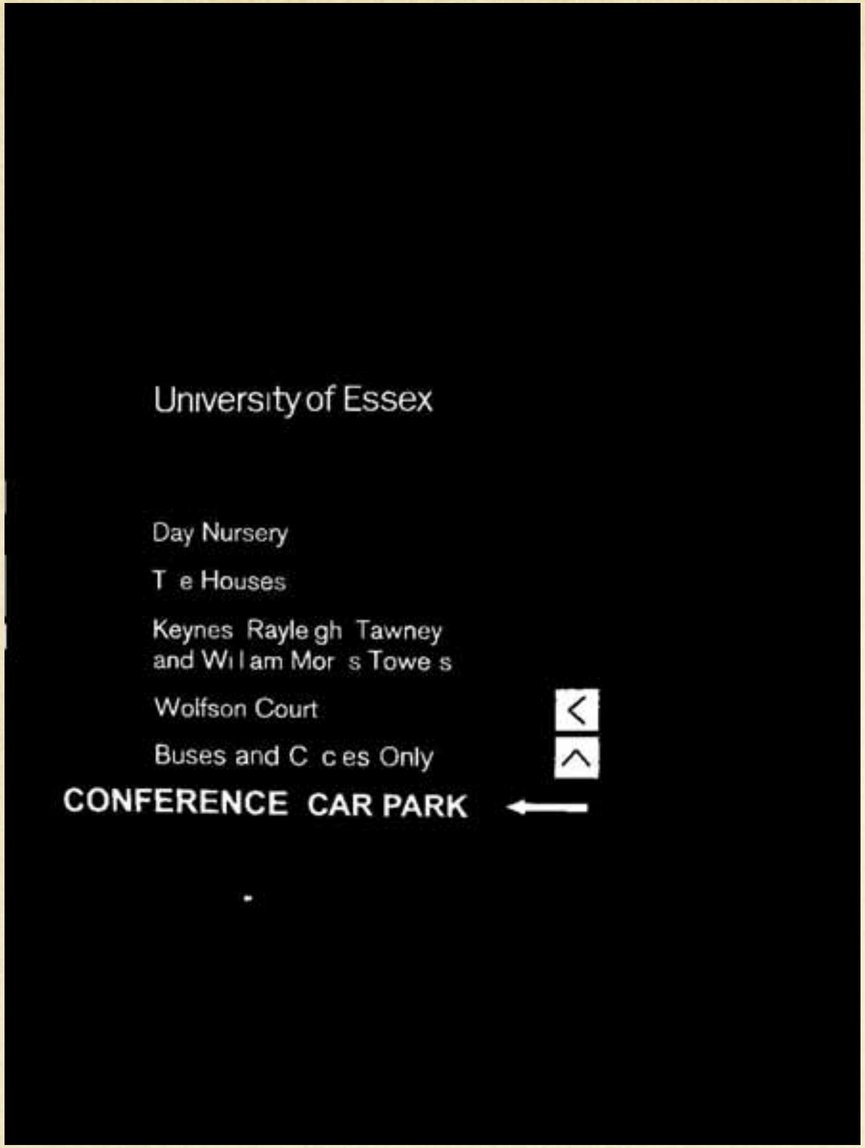
∨

- ✓ Yellow: SW + EHOG
- ✓ Pink: SW + PD
- ✓ Light blue: PD + EHOG

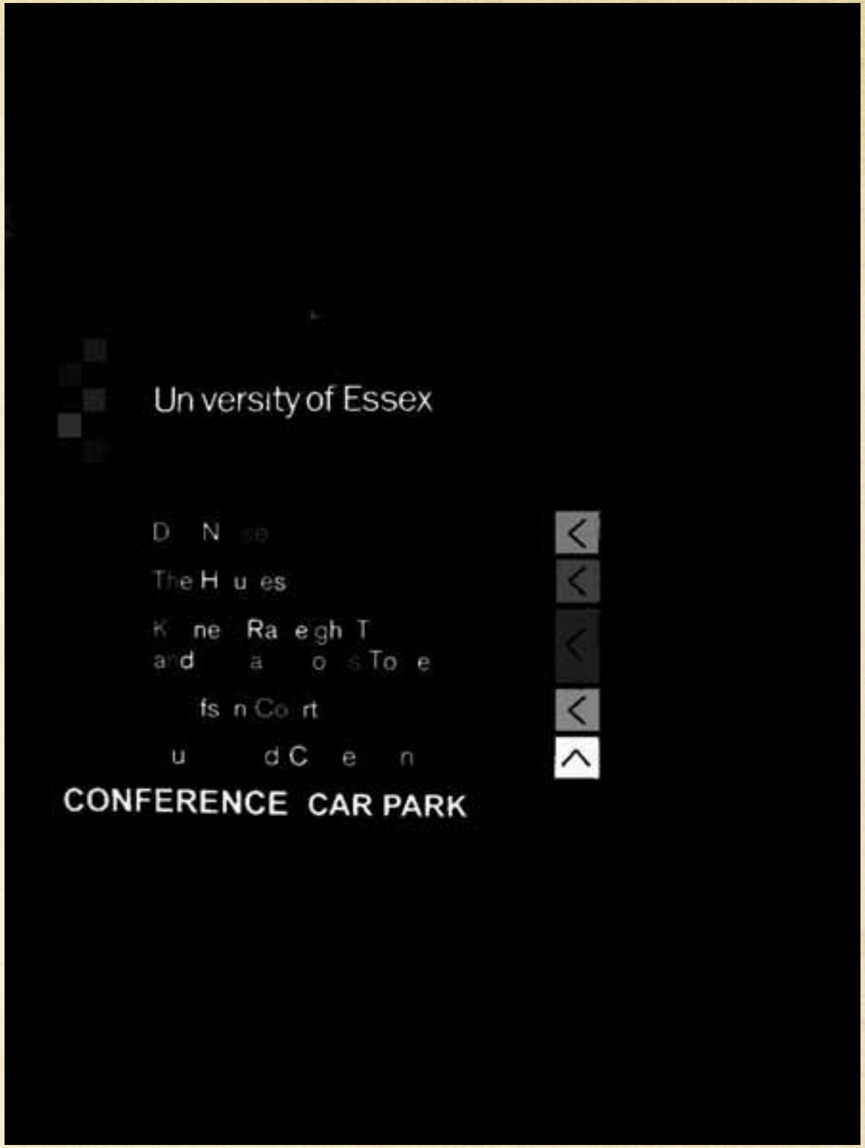
RESULTS

Compare our method with Matlab implementation [11]

Original Image

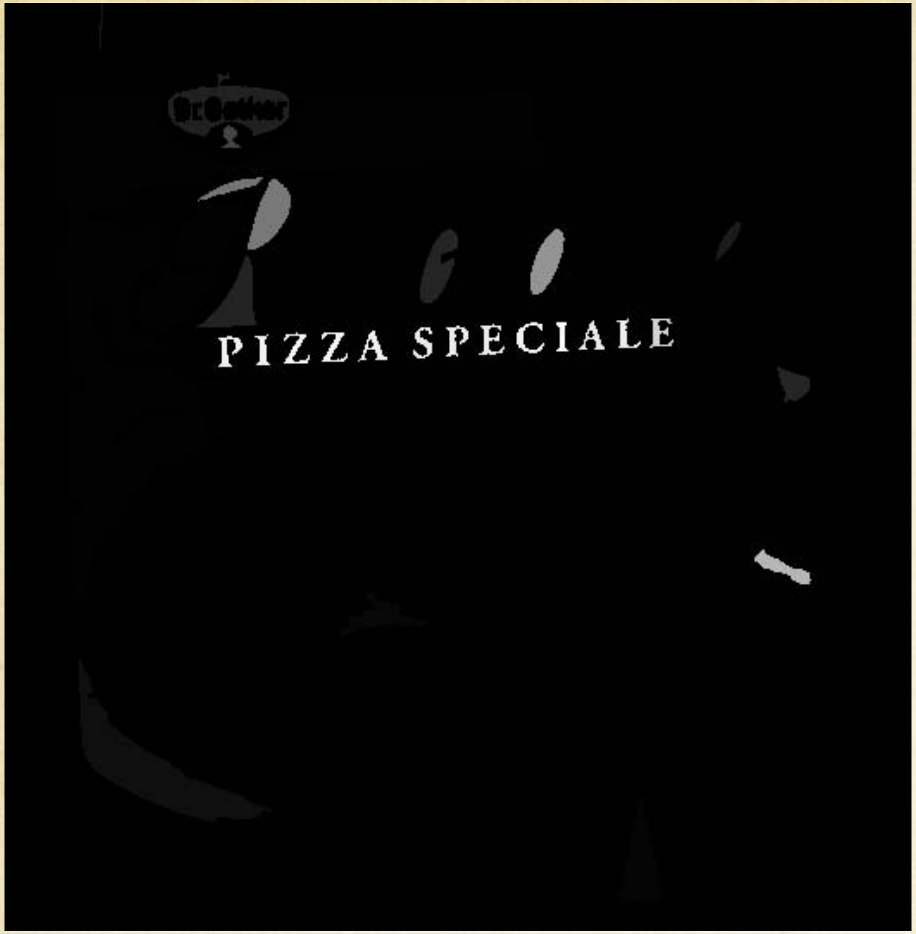


Our Method

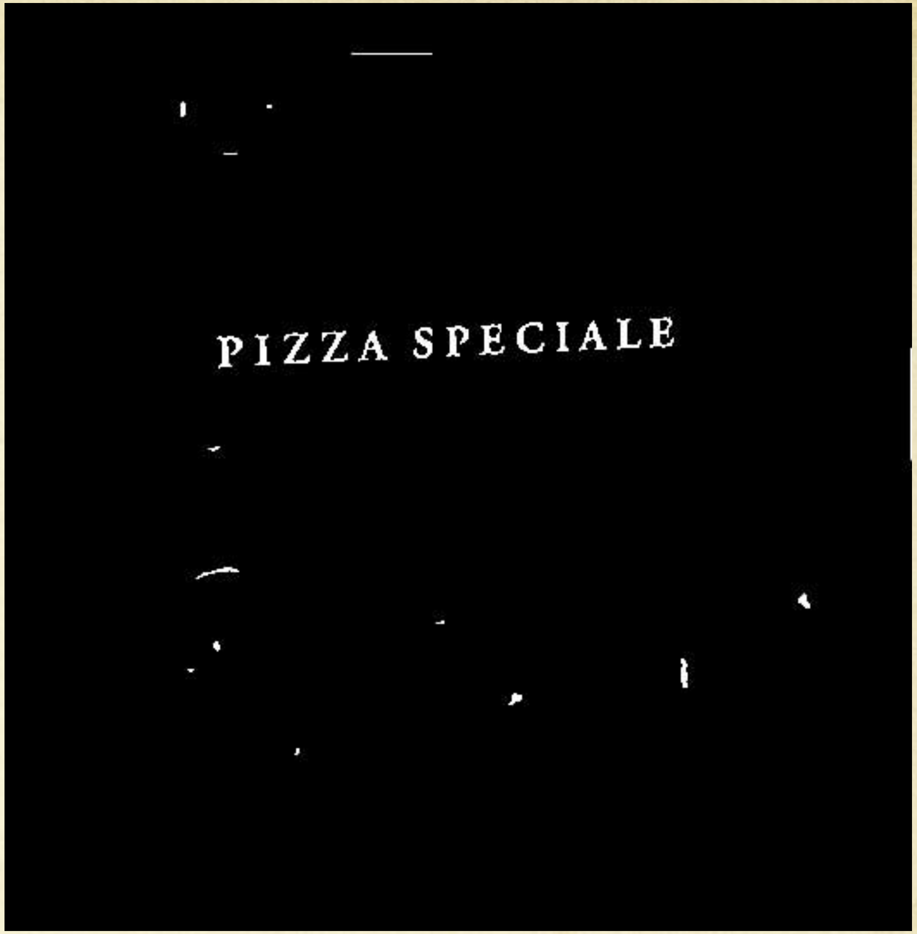


Matlab ®

Original image

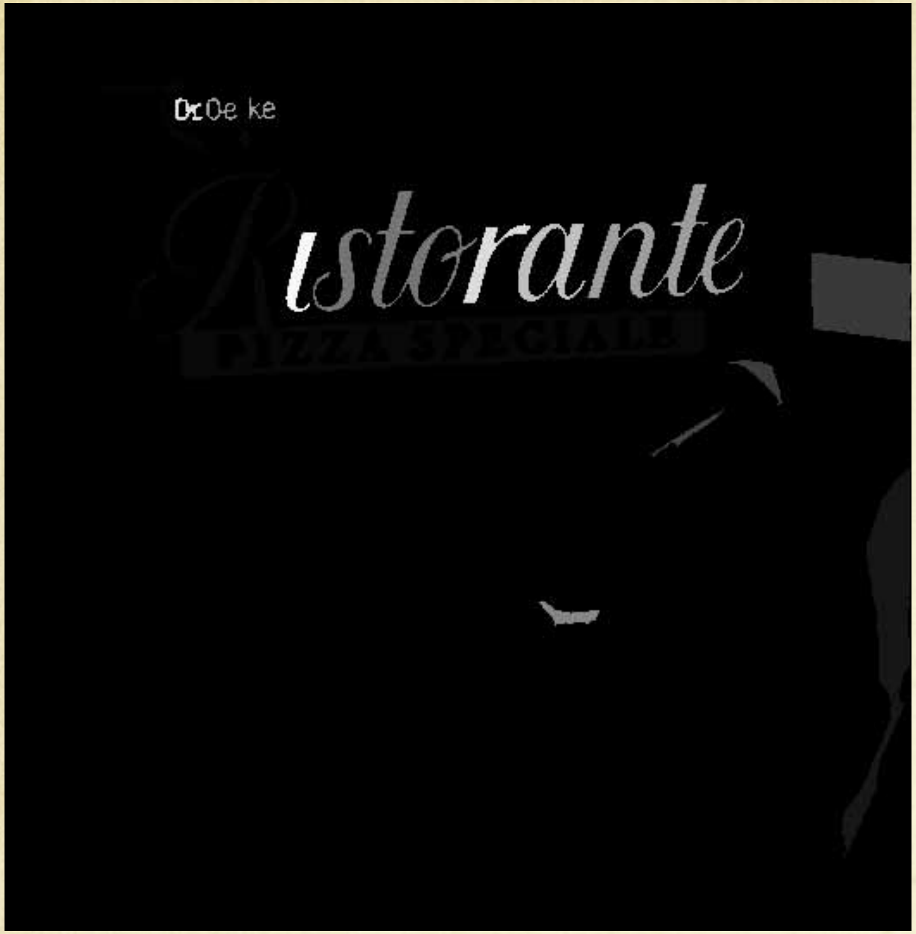


Matlab ®

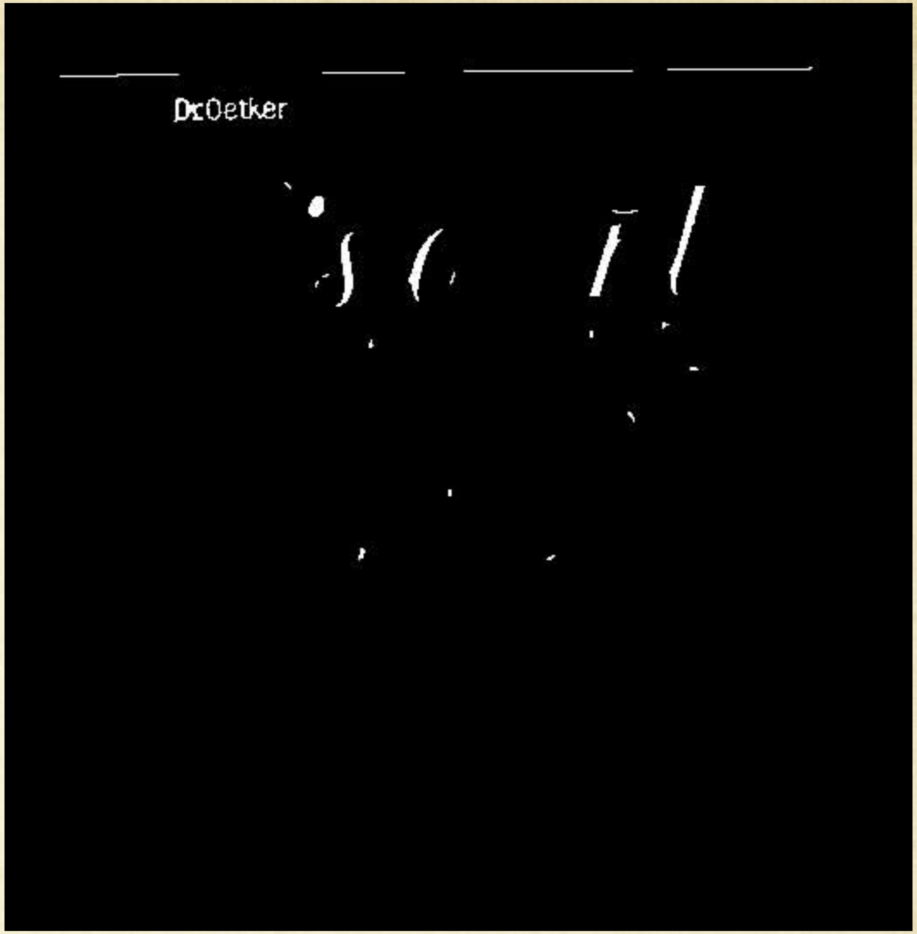


Our Method

Bright on Dark



Matlab ®

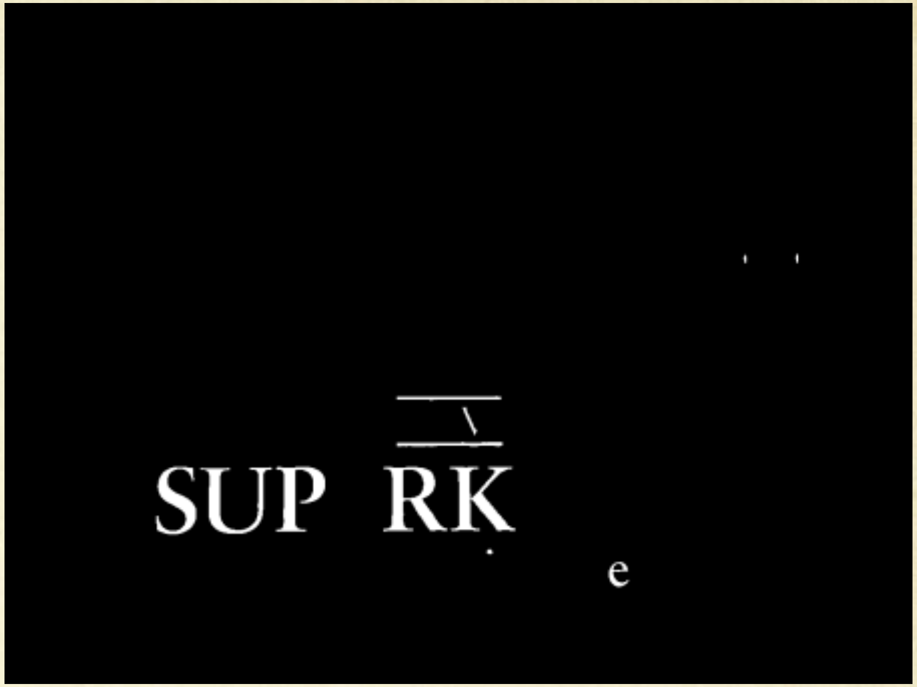


Out Method

Dark on Bright

- Original Image

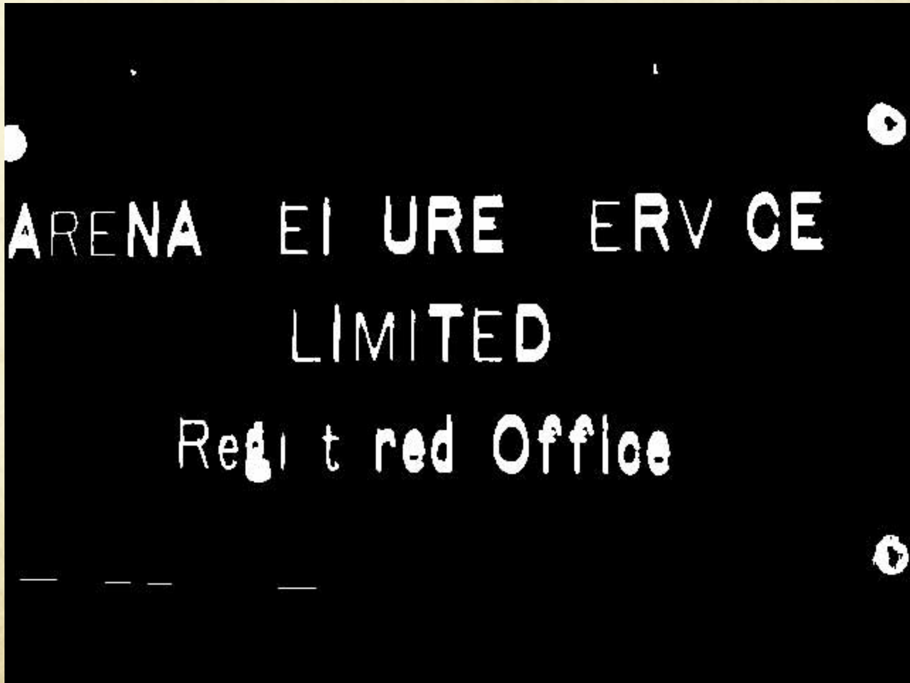
Matlab ®



Out Method

- Original Image

Matlab ®



Our Method

Original Image

Our Method

Matlab ®

Original Image

Matlab ®

Our Method

Results Analysis

General considerations

The main difference is the filter used in the two methods: guided or bilateral.

As a consequence our method works better if:

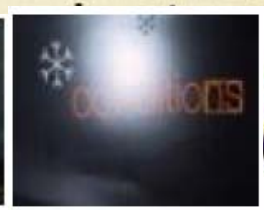
- ✓ Small regions are present;
- ✓ The image is too much blurry.

In general the guided filter is optimized for the text detection approach and so on "standard" images the Matlab code works better.

Problems

✓ This technique doesn't work correctly with:

- Regions with low resolution
- High light intensity
- Characters with uncommon font
- Too small text (solved)
- Transparent text
- Blurred text
- Character that changes colour
- Characters displaced over a certain angle


Improvements

Ideas about how to obtain better results

Not equalized result

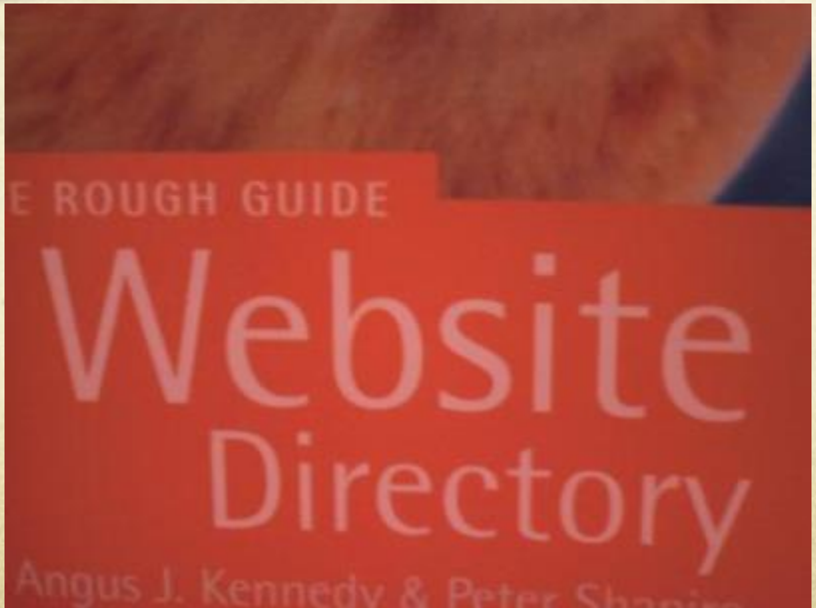
Equalization

- ✓ An equalized image has a better contrast
- ✓ Cues are able to recognize also not well defined letters
- ✓ This partially solve some of the cited problems

Equalized result

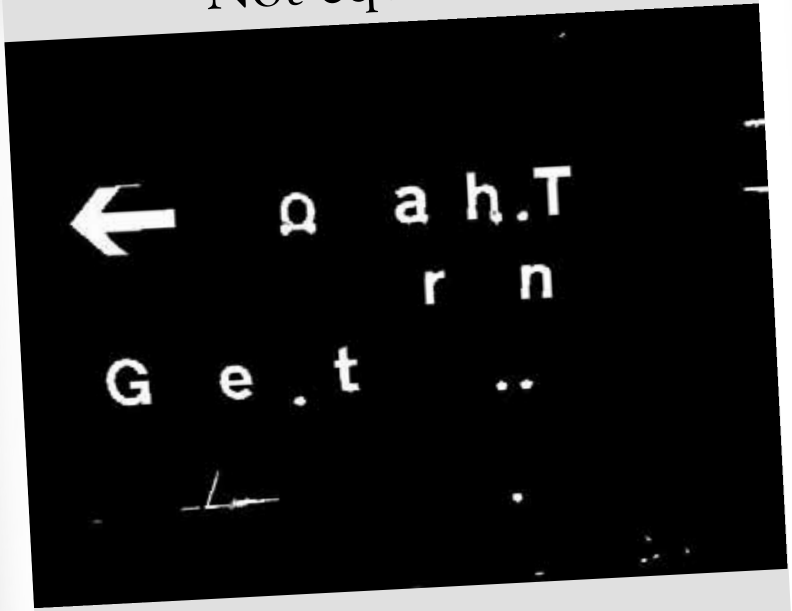
Not equalized

Equalized



Original

Not equalized



Equalized

Original

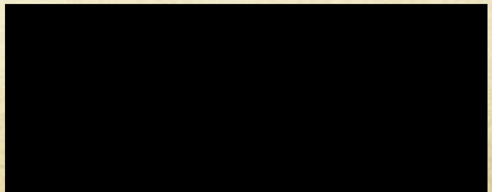
Bayes Modifications

✓ We experimentally see that SW Algorithm works better on small text over large picture...

- PD

- SW

- eHOG

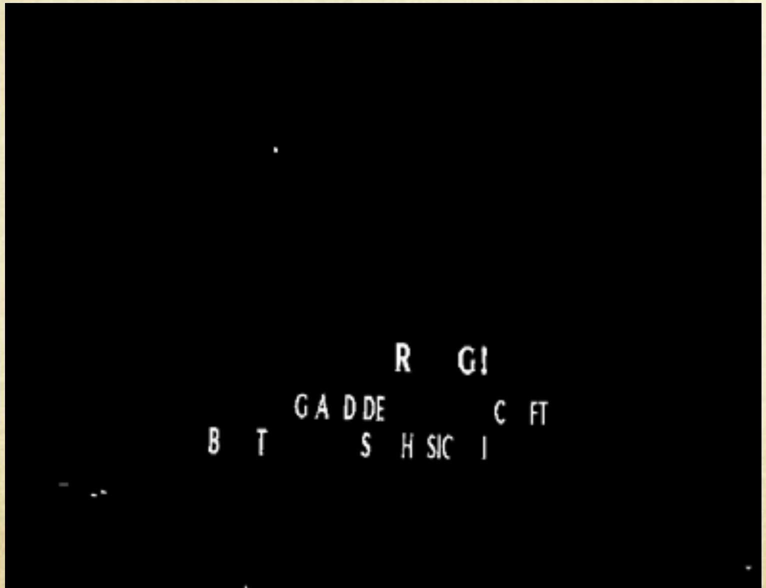


Idea!

→ Play up the role of SW inside the Bayes formula where small regions appear

Original

Bayes



Modified Bayes

Future developments

- ✓ Make the algorithm more robust against problems already exposed
- ✓ Implement a function that detect the words
- ✓ Apply the technique to read car license plate
- ✓ Implement a function that read the text found and then convert it into a file .txt

In particular...

We tried to apply this method to car license plate images.

Car license plate examples

REFERENCES:

1) *Characterness: An Indicator of Text in the Wild, Yao Li*, Wenjing Jia, Chunhua Shen, Anton van den Hengel, 2013.
2) http://www.vlfeat.org/api/mser-fundamentals.html
3) http://www.vlfeat.org/index.html
4) http://docs.opencv.org/modules/features2d/doc/feature_detection_and_description.html
5) *Guided image filtering*, K. He, J. Sun, and X. Tang, 2010.
6) *Detecting text in natural scenes with stroke width transform*, B. Epshtein, E. Ofek, and Y. Wexler, 2010.
7) *Scene text detection via stroke width* , Y. Li and H. Lu, 2012.
8) *Histograms of oriented gradients for human detection*, N. Dalal and B. Triggs, 2005.
9) http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
10) *Text detection using edge gradient and graph spectrum*, J. Zhang and R. Kasturi, 2010.
11) https://github.com/yaoliUoA/characterness