A new edge detector based on SMOTE and logistic regression

Raquel Fernández Peralta Sebastia Massanet
Arnau Mir

Soft Computing, Image Processing and Aggregation (SCOPIA)
Dept. Mathematics and Computer Science
University of the Balearic Islands, Palma, Mallorca

http://scopia.uib.eu

Outline

1. Introduction and Motivation
2. Edge detector based on SMOTE and logistic regression
   - Formula
   - Training stage
   - SMOTE
   - Binary Logistic Regression
   - Proposed Algorithm
3. Experimental results
   - Objective comparison
   - Optimal training image
   - Quantitative and qualitative comparison
4. Conclusions and Future Work

Introduction and Motivation

Edge detection

Over the last decades, a large number of edge detection algorithms have been proposed making edge detection one of the main lines of research in image processing. The interest in this topic comes from two main reasons:

Edge detection

Over the last decades, a large number of edge detection algorithms have been proposed making edge detection one of the main lines of research in image processing. The interest in this topic comes from two main reasons:

- Edge detection is a preliminary step in high-level operations such as computer vision, pattern recognition or segmentation.

Edge detection

Over the last decades, a large number of edge detection algorithms have been proposed making edge detection one of the main lines of research in image processing. The interest in this topic comes from two main reasons:

- Edge detection is a preliminary step in high-level operations such as computer vision, pattern recognition or segmentation.
- The abstract nature of the edge detection problem. Indeed, there is no clear mathematical definition of what an edge is that is accepted by the whole community. It is widely accepted that *edges are the result of human experience rather than a mathematical definition*.

Edge detection

There has been a lot of edge detectors proposed that rely on the most different approaches and based on diverse theories.

Edge detection

There has been a lot of edge detectors proposed that rely on the most different approaches and based on diverse theories.

- Many edge detectors that rely on the most different approaches and theories have been proposed over the last decades.

Edge detection

There has been a lot of edge detectors proposed that rely on the most different approaches and based on diverse theories.

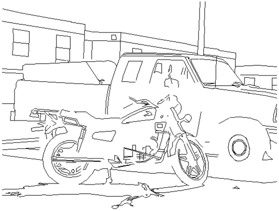
- Many edge detectors that rely on the most different approaches and theories have been proposed over the last decades.
- Some other approaches based on statistics, phase congruency and local energy, vector oriented statistics...

However, up to now, there is no edge detector that is optimal for all kind of images.

The edge detection problem as a classification problem

Among the different approaches for edge detection, a relatively unexplored one is to understand the edge detection problem as a classification problem. According to Canny's restrictions each pixel must be classified as edge or non-edge. Edge detection problems can be faced as binary imbalanced classification problems.

(a) Original



(b) Ground Truth

Figure: An image and its ground truth edge image.

Goals

The goal of this paper is to introduce a novel algorithm for edge detection based on:

- a logistic regression classifier trained using images with an available ground truth, and
- SMOTE technique as an oversampling method to handle the imbalance among the two classes: edge and non-edge.

Also, we want to study the efficiency of this algorithm by comparing the results with other well-known edge detectors.

Edge detector based on SMOTE and logistic regression

Formula

The edge detector based on SMOTE and logistic regression computes a probability image where the value of each pixel represents the probability to belong to the edge class. The probability of a pixel **x** is computed by

$$\pi_{\mathbf{x}}(g_h, g_v, g_{d_1}, g_{d_2}) = \frac{e^{a_1 + a_2 \cdot g_h + a_3 \cdot g_v + a_4 \cdot g_{d_1} + a_5 \cdot g_{d_2}}}{1 + e^{a_1 + a_2 \cdot g_h + a_3 \cdot g_v + a_4 \cdot g_{d_1} + a_5 \cdot g_{d_2}}}$$

where a_i with $i \in \{1, \ldots, 5\}$ are real constants and g_h, g_v, g_{d_1} and g_{d_2} are the four 3×3 Sobel directional gradients:

Formula

1	2	1
-1	-2	-1

1		-1
2		-2
1		-1

	-1	-2
1		-1
2	1	

2	1	
1		-1
	-1	-2

Figure: Convolution masks used to compute the four 3×3 Sobel directional gradients.

Training stage

The performance of this edge detector depends obviously on the choice of the real constants a_i with $1 \leq i \leq 5$. In order to choose good values of these constants, a training stage has been performed.

Flow chart of the training stage

INPUT:

- Image I
- Ground truth edge image of I

Features Construction

For each pixel $\mathbf{x}$ of I

1. Compute its four 3×3 Sobel directional gradients $(g_h)_{\mathbf{x}}, (g_v)_{\mathbf{x}}, (g_{d_1})_{\mathbf{x}}, (g_{d_2})_{\mathbf{x}}$
2. Set

$$y_{\mathbf{x}} = \begin{cases} 1 & \text{if } \mathbf{x} \text{ edge,} \\ 0 & \text{if } \mathbf{x} \text{ non-edge.} \end{cases}$$

SMOTE

Synthetic features vectors belonging to the edge class are created and added to the set $\{((g_h)_{\mathbf{x}}, (g_v)_{\mathbf{x}}, (g_{d_1})_{\mathbf{x}}, (g_{d_2})_{\mathbf{x}}, y_{\mathbf{x}}) | \mathbf{x} \in \mathbf{I}\}$

Logistic Regression

OUTPUT:

Estimates of a_i with $1 \leq i \leq 5$

Features construction phase

Let us compute the features vector of a given pixel.

235	232	230	230	227
229	224	224	209	185
219	202	$\boxed{172}$	146	123
157	132	123	122	126
123	127	129	127	127

Features construction phase

Let us compute the features vector of a given pixel.

235	232	230	230	227
229	224	224	209	185
219	202	172	146	123
157	132	123	122	126
123	127	129	127	127

1	2	1
-1	-2	-1

$(381, \quad , \quad , \quad , \quad)$

Features construction phase

Let us compute the features vector of a given pixel.

235	232	230	230	227
229	224	224	209	185
219	202	172	146	123
157	132	123	122	126
123	127	129	127	127

1		-1
2		-2
1		-1

$(381, 137, \quad, \quad, \quad)$

Features construction phase

Let us compute the features vector of a given pixel.

235	232	230	230	227
229	224	224	209	185
219	202	172	146	123
157	132	123	122	126
123	127	129	127	127

	-1	-2
1		-1
2	1	

$(381, 137, 199, \quad , \quad)$

Features construction phase

Let us compute the features vector of a given pixel.

235	232	230	230	227
229	224	224	209	185
219	202	172	146	123
157	132	123	122	126
123	127	129	127	127

2	1	
1		-1
	-1	-2

$$(381, 137, 199, 361, \quad)$$

Features construction phase

Let us compute the features vector of a given pixel.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	1	1	1	0
1	0	0	0	0

$$(381, 137, 199, 361, 0)$$

Features construction phase

At the end of this phase, we have computed the set

$$\{((g_h)_{\mathbf{x}}, (g_v)_{\mathbf{x}}, (g_{d_1})_{\mathbf{x}}, (g_{d_2})_{\mathbf{x}}, y_{\mathbf{x}}) | \mathbf{x} \in \mathbf{I}\}.$$

This set of features vectors is very imbalanced with the edge class usually representing around a 5-10%. Thus, ordinary classification algorithms are not adequate and for this reason, we apply next the SMOTE technique.

SMOTE

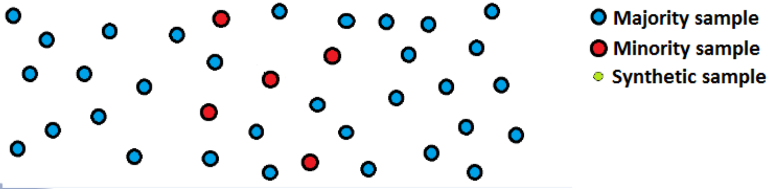
The synthetic minority oversampling technique (SMOTE) is an oversampling method to handle imbalance in highly imbalanced classification problems.

SMOTE oversamples the minority class by creating ”synthetic” examples among the line segments joining a concrete minority class instance with any/all of its k minority class nearest neighbors.

SMOTE

Once a minority class instance and one of its k minority class nearest neighbors are selected, the synthetic instance is created as follows:

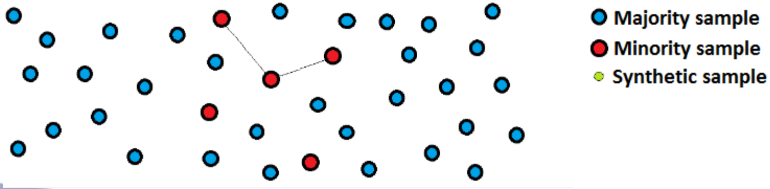
1. Take the difference between both feature vectors under consideration.
2. Multiply this difference by a random number between 0 and 1.
3. The previous result is added to the feature vector of the selected minority class instance leading to a new synthetic instance belonging to the minority class.



SMOTE

Once a minority class instance and one of its k minority class nearest neighbors are selected, the synthetic instance is created as follows:

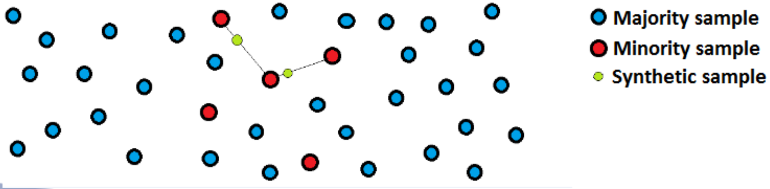
1. Take the difference between both feature vectors under consideration.
2. Multiply this difference by a random number between 0 and 1.
3. The previous result is added to the feature vector of the selected minority class instance leading to a new synthetic instance belonging to the minority class.



SMOTE

Once a minority class instance and one of its *k* minority class nearest neighbors are selected, the synthetic instance is created as follows:

1. Take the difference between both feature vectors under consideration.
2. Multiply this difference by a random number between 0 and 1.
3. The previous result is added to the feature vector of the selected minority class instance leading to a new synthetic instance belonging to the minority class.



Example SMOTE

235	232	230	230	227
229	224	224	209	185
219	202	172	146	123
157	1 132	2 123	122	126
123	127	129	127	127

$$G_1 = (289, 109, 139, 289)$$

$$G_2 = (180, 76, 71, 203)$$

$$d = G_1 - G_2 = (109, 76, 71, 203)$$

$$G = G_2 + d \cdot 0.5 = (234.5, 92.5, 105, 246)$$

$$\Downarrow$$

$$(234.5, 92.5, 105, 246, 1)$$

Binary Logistic Regression

The model of the binary logistic regression is used to estimate the probability of a binary response based on some independent variables.

We consider n independent observations $y_1, \ldots, y_n$ which are realizations of n random variables $Y_i \sim \text{Ber}(\pi_i)$ for all $1 \leq i \leq n$. In our case, given an image of n pixels then

$$y_i = \begin{cases} 0 & \text{if the pixel is a non-edge} \\ 1 & \text{if the pixel is an edge} \end{cases} \qquad \text{for all } 1 \leq i \leq n$$

where the information of what is an edge or not is given by the ground truth.

Binary Logistic Regression

Suppose that we can express the so-called *logit* function of the probabilities as a linear combination of the 4 predictor variables, in our case the value of the 4 Sobel directional gradients, $\{g_h, g_v, g_{d_1}, g_{d_2}\}$. That is,

$$\ln\left(\frac{\pi_i}{1-\pi_i}\right) = a_1 + a_2 \cdot (g_h)_i + a_3 \cdot (g_v)_i + a_4 \cdot (g_{d_1})_i + a_5 \cdot (g_{d_2})_i = S_i$$

for all $1 \leq i \leq n$ where $\boldsymbol{a} = (a_1, \ldots, a_5)^t$ is the parameter vector and $G = ((g_h)_i, (g_v)_i, (g_{d_1})_i, (g_{d_2})_i) \in \mathcal{M}_{n\times 4}$ is a sample of the predictor variables. The goal is to estimate vector $\boldsymbol{a}$. Using maximum likelihood, we obtain the log-likelihood function

$$l(a) = \sum_{i=1}^{n} \Big(y_i \cdot S_i - \ln\left(1 + e^{S_i}\right)\Big).$$

At this point, the estimates of $\boldsymbol{a}$ are obtained applying Newton-Raphson to the non-linear equation system $\frac{\partial l(\boldsymbol{a})}{\partial a_j} = 0$ for all $1 \leq j \leq 5$.

Canny's Restrictions

A logistic regression model generates an image containing for each pixel, the probability to belong to the edge class.

Canny’s Restrictions

A logistic regression model generates an image containing for each pixel, the probability to belong to the edge class.

However, Canny’s restrictions force an edge representation as a binary image with edges of one pixel width.

Proposed Algorithm

Thus, we need to apply to the probability image:

Proposed Algorithm

Thus, we need to apply to the probability image:

- **A *thinning* algorithm**, as Non Maxima Suppression to obtain edges of one pixel width,

Proposed Algorithm

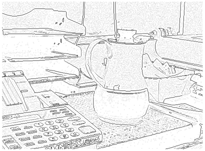
Thus, we need to apply to the probability image:

- **A *thinning* algorithm**, as Non Maxima Suppression to obtain edges of one pixel width,
- **A hysteresis algorithm**, as the one based on the search of the instability zone of the histogram, to obtain binary edges.

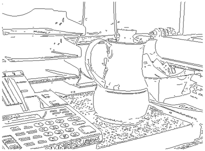
Graphical Steps of the Algorithm

(a) Original

(b) Probabilities



(c) NMS



(d) Hysteresis

Figure: Images obtained in the intermediate steps of the edge detector.

Experimental results

Objective comparison

To evaluate objectively the obtained results, we have considered the quality measure Q that is defined by

$$Q = \sqrt{\left(\frac{|\mathrm{TP}|}{M} - 1\right)^2 + \left(\frac{|\mathrm{FN}|}{M}\right)^2 + \left(\frac{|\mathrm{FP}|}{M}\right)^2 + \left(\frac{\mathrm{F}}{M} - 1\right)^2}$$

where

- $|\mathrm{TP}|$,$|\mathrm{FN}|$ and $|\mathrm{FP}|$ denote the number of true positives, false negatives and false positives, respectively,
- $M = \max\{|\mathrm{TP}| + |\mathrm{FN}|, |\mathrm{TP}| + |\mathrm{FP}|\}$
- F is defined as

$$F = \sum_{i=1}^{|\mathrm{TP}|+|\mathrm{FP}|} \frac{1}{1 + \frac{1}{9}d^2(i)}$$

  with $d(i)$ the distance between the i-th edge pixel of the proposed result and the closest edge pixel of the ground truth.

Smaller values of Q with $0 \leq Q \leq 2$ indicate better capability of edge detection.

Optimal training image

Firstly, we have set the optimal training image to establish our edge detector. For this reason we have used the 50 images of the public image dataset of the University of South Florida which contains the original images with their corresponding ground truth images.

Optimal training image

Firstly, we have set the optimal training image to establish our edge detector. For this reason we have used the 50 images of the public image dataset of the University of South Florida which contains the original images with their corresponding ground truth images.

In order to pick up the best model among the 50 generated models (one for each image of the dataset of the University of South Florida), we have applied each one to the three images of the EUSFLAT 2017 Competition on Edge Detection with their ground truth edge images available.

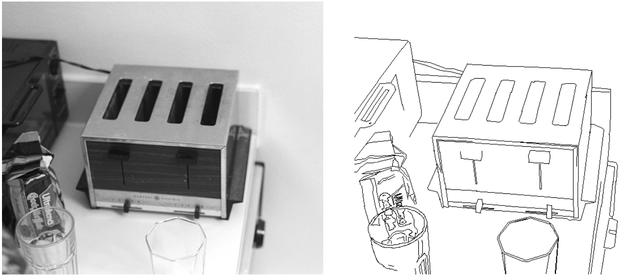
Optimal training image

Firstly, we have set the optimal training image to establish our edge detector. For this reason we have used the 50 images of the public image dataset of the University of South Florida which contains the original images with their corresponding ground truth images.

In order to pick up the best model among the 50 generated models (one for each image of the dataset of the University of South Florida), we have applied each one to the three images of the EUSFLAT 2017 Competition on Edge Detection with their ground truth edge images available.

The best model of our edge detector in the three images of the competition with available ground truth correspond to the model generated from image 131 of the dataset.

Optimal training image



(a) Original (b) Ground Truth

Figure: Image 131 and its ground truth edge image of the dataset of the University of South Florida.

Comparison with other edge detectors

We have compared our edge detector SMOTE-Reg algorithm versus:

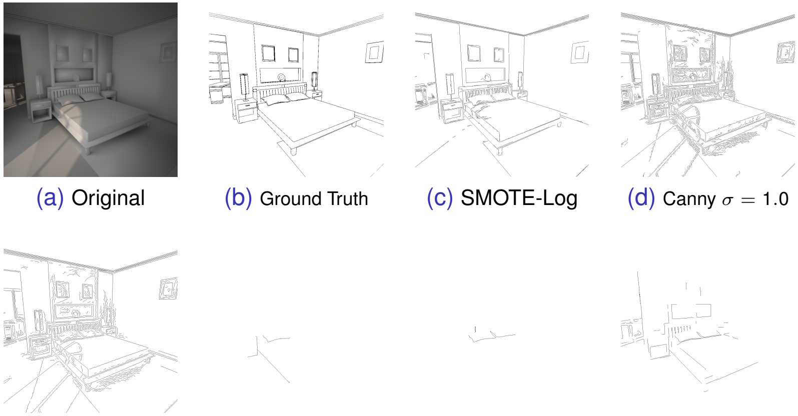
- Canny's edge detector with $\sigma \in \{0.5, 1.0, 1.5, 2.0\}$,
- Gravitational edge detector with the product and minimum t-norms,
- Multiscale edge detector based on Gaussian smoothing and edge tracking.

Results accoring to the measure Q

Edge detector	Mean	example1-1.jpg	example1-2.jpg	example1-3.jpg
SMOTE-Log	**1.1499**	**1.0712**	**1.1413**	**1.2372**
Canny $\sigma = 0.5$	1.3859	1.3874	1.2050	1.5654
Canny $\sigma = 1.0$	1.2191	1.1125	1.1626	1.3822
Canny $\sigma = 1.5$	1.2824	1.2109	1.2575	1.3787
Canny $\sigma = 2.0$	1.3715	1.3334	1.3493	1.4318
Gravitational Product	1.6475	1.5223	1.7123	1.7078
Gravitational Minimum	1.6855	1.6635	1.6828	1.7103
Multiscale	1.5314	1.4501	1.5746	1.5695

Table: Q values obtained by the considered edge detectors in the three images of the EUSFLAT 2017 Competition on Edge Detection with available ground truth edge images. The best Q values are depicted in bold.

Visual Comparison

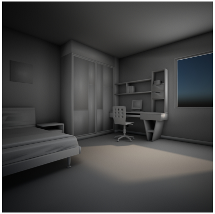


(a) Original (b) Ground Truth (c) SMOTE-Log (d) Canny $\sigma = 1.0$

(e) Canny $\sigma = 1.5$ (f) Grav. Prod. (g) Grav. Min. (h) Multiscale

Figure: Results obtained by the considered edge detectors in image “example1-3.jpg”.

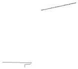
Visual Comparison



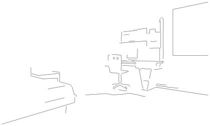
(a) Original

(b) SMOTE-Log

(c) Canny $\sigma = 1.0$



(d) Grav. Prod.



(e) Multiscale

Figure: An original image and the results obtained by some of the considered edge detectors.

Visual Comparison

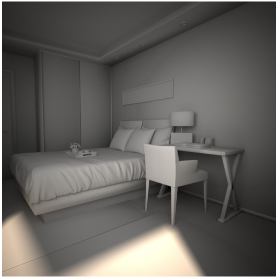
(a) Original

(b) SMOTE-Log

(c) Canny $\sigma = 1.0$

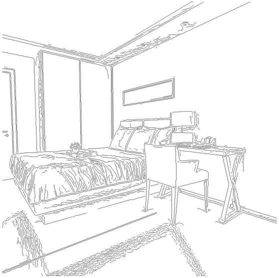
Figure: An original image and the results obtained by some of the considered edge detectors.

Visual Comparison



(a) Original

(b) SMOTE-Log



(c) Canny $\sigma = 1.0$

Figure: An original image and the results obtained by some of the considered edge detectors.

Conclusions and Future Work

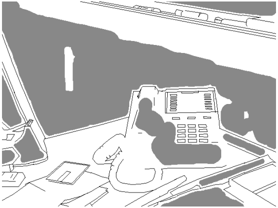
Conclusions

- We have proposed a new edge detector facing the problem of edge detection as an imbalanced classification problem. Specifically, we have trained a logistic regression model applying the SMOTE technique to balance the two classes: edge and non-edge.
- The results show that the algorithm is able to find most of the true edges of the image discriminating them from false edge areas emerged by illumination changes or textures.

Future Work

- Consider other quantitative performance measures such as Pratt's Figure of Merit, the ρ-coefficient or the F-measure which can be adapted to ground truth edge images with 3 zones such as the ones provided in the dataset of the University of South Florida.

(a) Original



(b) Ground Truth

Figure: An image of the dataset of the University of South Florida and its ground truth with 3 zones.

Future Work

- Explore several additional options in order to improve the results:
  - Test other binary classifiers for imbalanced classification problems.
  - Test modifications of the SMOTE technique.
  - Consider another alternatives to oversampling techniques such as cost-sensitive classifiers.
- Compare our results with other recently-introduced edge detectors such as the one based on the F1-transform or the fuzzy morphological gradient.


Thank you for your attention!