Image Operator Learning Based on Local Features

Augusto C. M. Silva, Igor S. Montagner, Roberto Hirata Jr., Nina S. T. Hirata
Institute of Mathematics and Statistics
University of São Paulo
São Paulo, Brazil
augusto.cesar.silva@usp.br, igordsm,hirata,nina@ime.usp.br

***Abstract*—Morphological operators in image processing have a wide range of applications, like in medical imaging and document image analysis. The design of such operators are made, mainly, by a trial and error approach. Another method to design these operators consists in using machine learning algorithms to define a local transformation that represents an operator. Previous works used mainly the intensity values of the pixels as feature vectors in the machine learning algorithms. We propose to extract different features, calculated from the image, to create different feature vectors to be used in the machine learning algorithms. We experiment this approach in four different public datasets, and results show that different features have a significant impact on the learned operators, but, just like the operators, the feature that provides better results also depends on the dataset used.**

I. Introduction

Image processing has a wide range of applications in areas like medical imaging, document image analysis, object segmentation and so forth. Therefore, research in this area are extremely important. In particular, a very interesting field in image processing is the design and applications of morphological operators in images [1] [2] [3].

The use of these operators depends on their design, and that is extremely dependent on the application and dataset used. Manual composition of an operator for a specific dataset is based on a trial and error approach, thus it revolves around the expertise and experience of the professional, and, besides that, it requires a great amount of time and effort.

An alternative approach to the manual design of these operators is the construction of these operators based on pairs of input and output images [4] [5]. This method consists in using training techniques to compose an operator that, given an input, returns the most accurate approximation of the desired output. A great amount of research in this area considers morphological operators that are translation invariant and locally defined in a window W. These operators are called W-operators.

The training technique to design a W-operator, as defined in [5], consists of three steps. First, slide the window W over the image and extract features for each position of the window. The second step is to decide the output of the operator for each observed pattern, and the last step consists in applying a training algorithm to generalize the operator, so it can classify patterns not observed before.

In order to improve the learning of local image operators, many previous works were focused on the selection of window size [6] or in alternative methods of the learning algorithms [7]. Those previous works mainly used, in the first step, the window itself as the feature vector. In this paper we propose a novel approach by extracting different information from the windows, using this information as a feature vector, instead of using the intensity values of the windows' pixels.

In section II we describe in more details the process of training a W-operator from pairs of input and output. Section III contains the description of implemented features that were used in the experiments. A brief explanation of how the experiments were implemented, the datasets used and the results obtained can be seen in Section IV. Section V contains our concluding remarks.

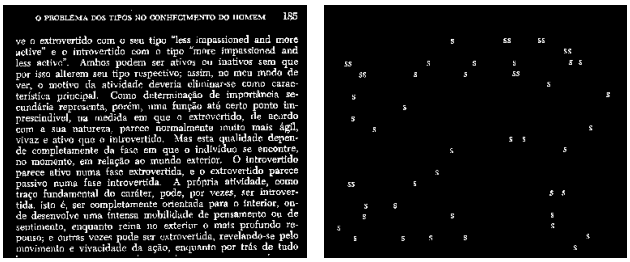
II. Image Operator Learning

Gray-level images can be represented as a function f of the form $f : \mathbb{E} \to K$ where $\mathbb{E}$ is a discrete grid, such as $\mathbb{E} = \mathbb{Z}^2$, and $K = \{0, 1, ..., k-1\}$ denotes a set of gray level values. Given a position $(x, y) \in \mathbb{E}$, the value of $f(x, y)$ corresponds to the gray level value of the pixel at (x, y). An image operator is a mapping of the form $\Psi : K^{\mathbb{E}} \to K^{\mathbb{E}}$, where $K^{\mathbb{E}}$ denotes the set of all images defined on $\mathbb{E}$ with gray-level in K, and the value of $[\Psi(f)](x, y)$ is the gray intensity value of the pixel at position (x, y) of the image f transformed by the operator Ψ.

In this paper we restrict ourselves to operators that are translation-invariant and locally defined within a finite neighborhood W. The output of an operator that is locally defined depends only on the neighborhood W, and the translation-invariant property means that the operator Ψ is the same in every position of the image. The neighborhood W is usually called a window, and the operators that respects these two properties are called W-operators.

We also restrict ourselves to the case where the input is any image, grayscale or binary, but the output is binary. This restriction is not too severe since many important image transformations can be expressed by binary output images. For instance, segmentation or object detection are typical processings where the output image is usually binary. An example of segmentation of a grayscale image can be seen in the DRIVE dataset and an example of recognition in the DIBCO dataset, both described in more details in the experiments section of this paper.

Therefore, given a pair (f, g) of input-output images, the image operator learning process must define a local function $\psi : \mathbb{E} \to \{0, 1\}$ that, when applied to the image f gives



(a) Input (b) Output

Fig. 1. Example of an input-output image pair for the training process

the most accurate approximation of g. To define the optimal operator Ψ, we must calculate a measure of proximity between the output of the operator, $\Psi(f)$, and the desired output g, such as the mean absolute error (MAE). If we consider that the pair (f, g) is drawn from a jointly stationary random process with a probability function $P(\mathrm{y}|X)$, where X is a feature vector extracted from the image f and $\mathrm{y} \in \{0, 1\}$ is the output in image g , it can be shown that the operator that minimizes the MAE is the one characterized by function ψ:

$$\psi(X) = \begin{cases} 1, & \text{if } P(\mathrm{y} = 1|X) > 0.5 \\ 0, & \text{if } P(\mathrm{y} = 1|X) \leq 0.5 \end{cases} \quad (1)$$

Since the probability function is not usually known, our goal is to estimate $P(\mathrm{y}|X)$ using input and output training images, and use the estimated probabilities as the real ones in Equation (1). This approach of training an W-operator by sample images is based on [3] and [4].

Given pairs (f, g) of input and output images, the probability function can be estimated by a process of three steps. First, the window W must be slided over the image and, for each position (x, y) of the window, a feature X must be extracted from the inside of that window and the pair (X, y) recorded, with $\mathrm{y} = g(x, y)$. The second step consists in deciding the function value for each recorded pattern X, i. e., $\hat{\psi}(X) = 1$ if $P(\mathrm{y} = 1|X) > P(\mathrm{y} = 0|X)$, and $\hat{\psi}(X) = 0$ otherwise. The last step is to generalize the operator, so it can classify patterns that weren't seen in the training images. An example of the input and output of this training process can be seen in Figure 1.

Feature extraction plays an important role in this training process. A good choice of feature will maximize $P(\mathrm{y}|X)$, therefore minimizing the MAE, while a poor choice of feature will not be able to differentiate between two different patterns.

III. Local Features

We propose three different local features to be extracted for each pixel of an image. These three features consists of the raw feature, used as a baseline to the experiments, a feature based on Local Binary Patterns [8], and one based on geometric moments.

Each one of these features creates a feature vector that is used on different classification algorithms, such as Decision Trees and Support Vector Machines.

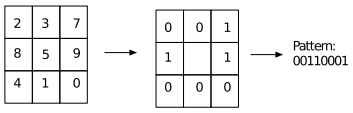


Fig. 2. Basic LBP algorithm

A. Feature Raw

The most basic feature of a neighboring region of a pixel is the region itself. Therefore, the baseline that we use to our experiments is the operator learning that uses the intensity values of all the pixels of the surrounding region as a feature vector.

The length of this feature vector is the amount of pixels in the window W, therefore the size of the window is a determinant factor for the computational cost of the image operator learning process using this feature.

B. Feature LBP

Local Binary Pattern (LBP) is a texture descriptor that labels each pixel according to a predetermined neighboring region of it. LBP descriptor is based on the idea that 2D surface texture can be represented by two measures: local spatial pattern and gray scale contrast.

The texture descriptor, as introduced by Ojala *et al.* [8], labels each pixel by thresholding the surrounding 3x3 neighborhood with the center pixel intensity value and considering the result as a binary number. Then, the histogram of these labels can be used as a texture descriptor. An example of the basic algorithm can be seen in Figure 2.

Our approach applies LBP's basic algorithm to all the pixels of an image, and then, for each pixel, creates a feature vector with all the labels inside the window W.

C. Feature Moments

Let $f(x, y)$ be the pixel intensity of the image f at position (x, y). Its *geometric moments* of order $p + q$ are defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y)\, dx\, dy\,. \quad (2)$$

Geometric moments are features that provide rich information due to the uniqueness theorem [9] that states that if $f(x, y)$ is piecewise continuous and has nonzero values only in the finite part of the *xy* plane, moments of all orders exists and the moment sequence $\{m_{pq}\}$ is uniquely determined by $f(x, y)$, and conversely $f(x, y)$ is uniquely determined by $\{m_{pq}\}$.

Our approach is to create, for each pixel, a feature vector based on moments of a determined region of the image around that pixel. The central pixel is considered as the origin point $(0, 0)$ for the x and y in the calculation of the moments. The moments feature vector of order $n = p + q$ is defined as:

$$M_n = [f(x, y), \bar{x}, \bar{y}, m_{00}, m_{01}, m_{10}, ..., m_{pq}] \quad (3)$$

where $\bar{x}$ and $\bar{y}$ are the x and y values of the centroid of the region, defined as $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$, respectively.

IV. Experiments

We experimented the use of these three different feature vectors in the learning process of image operators using TRIOSlib [4], a library that contains state-of-art techniques in image operator learning that was used in many previous works. More details of this library can be seen in https://trioslib.github.io/.

In our experiments, four different datasets were used, two of binary images (*CharS* and *TexRev*) and two of grayscale images (*DRIVE* and *DIBCO*), each one with different goals.

We experimented these different features with three different sizes of windows (5x5, 7x7 and 9x9) in all four datasets. Decision Tree algorithm was used to generalize the operator, so it could predict patterns that weren't previously seen.

A. Datasets

The datasets used are described below:

1) CharS: This dataset consists of scanned pages of the book *"Tipos Psicológicos", Carl Gustav Jung, 1967*, and an example of this dataset can be seen in Fig. 1. The goal of this dataset is to extract a specific character from a binary image. In this dataset, 10 images were used for the training of the operator and 10 others images to test it.

2) TexRev: This dataset consists of scanned pages from the magazine *Revista Veja, "Computador - o micro chega às casas", Special Issue, December, 1995*. The goal of this dataset is text segmentation, where the input is a typical magazine page and the output is a binary image containing only the text inside the page. In this dataset, 5 images were used for training and 5 images for testing.

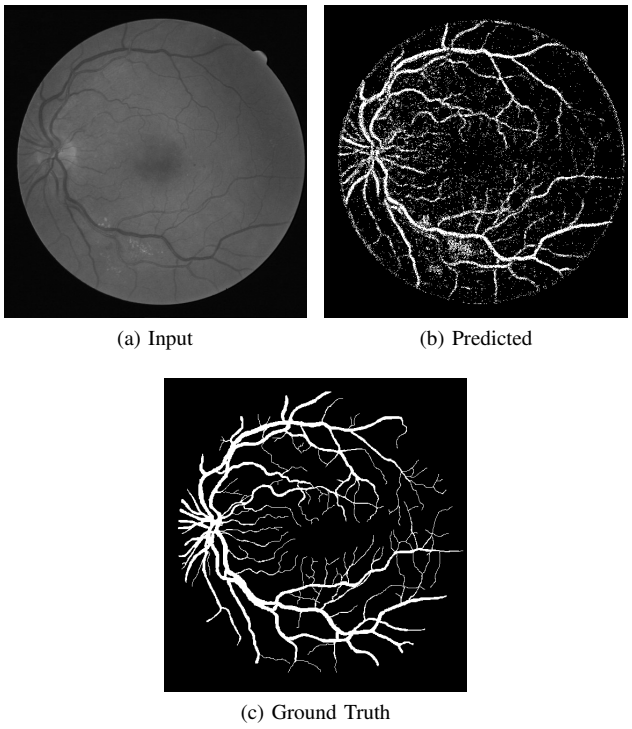
3) DRIVE: This dataset is the one introduced in [10]. It contains digital retinal images, where the main goal is the segmentation of blood vessels inside the retina. A more detailed description of this dataset can be found in [10] and [11]. An example of this dataset can be seen in Figure 3. In this dataset, 10 images were used in the training and 20 images were used for testing.

4) DIBCO: DIBCO is a Document Image Binarization Competition that are being held since 2009, and the dataset we have used in this paper contains images from the competition of 2014 [12] and 2016 [13]. This dataset contains images from handwritten documents and the main goal is the text binarization. In this dataset, 10 images were used for training and 7 images for testing.

B. Results

We measured the results of our experiments by calculating the percent error, i. e., the amount of pixels that were mispredicted by our operator divided by the total amount of pixels. These calculated errors are displayed on Table I.

As our operator outputs binary images, other two possible metrics to validate our proposal is the precision and recall. Precision is the percentage of pixels correctly labeled as 1 from all the pixels that our operator predicted as 1, and Recall is the percent of pixels that were correctly labeled as 1 from all the pixels with value 1 in the ground truth. Precision and Recall from all our experiments can also be seen in Table I.



(a) Input (b) Predicted

(c) Ground Truth

Fig. 3. Example of an output of the learned operator using the RAW feature in the DRIVE dataset and its ground truth.

For the binary images, the Raw feature performs better than any other feature in most cases, but the feature based on LBP has a steeper decrease as the window size grows.

As for the grayscale images, Table I shows that the RAW feature performs significantly better than the other features in the *DRIVE* dataset, but in the DIBCO dataset the feature based on moments has a higher accuracy in all window sizes. There is no significant difference between the moments of order 2 and 5. An example of test image and respective result of one experiment with the DRIVE dataset using the RAW feature can be seen in figure 3.

Figure 4 compares the output of operators created by different features and the desired output in an image from the DIBCO dataset. Other images of these experiments are available at: http://vision.ime.usp.br/~augustocms/SIBGRAPI2017/.

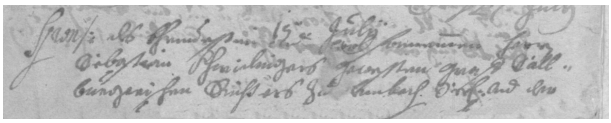
V. Conclusion

In this paper we have explored the use of local features to learn image operators, extending previous works that are predominantly based on raw pixel values. For each pixel, a set of computed features are encoded as a feature vector and then used to predict the value of the output image at that pixel.

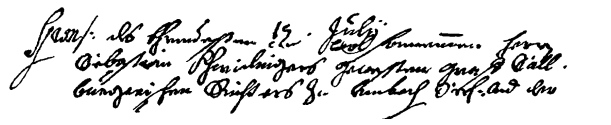
Experiments show that different features can change significantly the accuracy of a learned operator. However, the selection of a feature depends on the dataset used, as depicted by the differences in accuracy in the DRIVE and DIBCO datasets. Therefore, the selection of features is an important task to the process of image operator learning.

TABLE I
PERCENT ERROR, PRECISION AND RECALL IN THE EXPERIMENTS

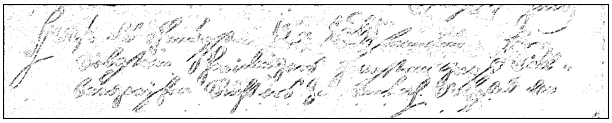
Size	Feature	CharS			TexRev			DRIVE			DIBCO		
5x5	Raw	**0.0142**	**0.8784**	**0.8953**	**0.0546**	**0.9528**	**0.8815**	**0.1165**	**0.5369**	**0.6137**	0.0488	0.0136	0.2644
	LBP	0.0186	0.8731	0.8185	0.0186	0.9375	0.8204	0.1538	0.3747	0.4236	0.1026	**0.0311**	**0.5915**
	Moments (n = 2)	0.0353	0.7708	0.6111	0.0815	0.9260	0.8239	0.1508	0.4190	0.4770	0.0418	0.0160	0.3155
	Moments (n = 5)	0.0341	0.7866	0.6176	0.0811	0.9267	0.8244	0.1543	0.4071	0.4656	**0.0410**	0.0160	0.3155
7x7	Raw	**0.0105**	**0.9038**	**0.9301**	**0.0432**	**0.9671**	**0.9026**	**0.1069**	**0.5707**	**0.6456**	0.0472	0.0134	0.2612
	LBP	0.0111	0.8984	0.9252	0.0499	0.9653	0.8841	0.1538	0.4085	0.4660	0.0942	**0.0293**	**0.5593**
	Moments (n = 2)	0.0507	0.5841	0.6306	0.0727	0.9437	0.8345	0.2248	0.1858	0.2265	**0.0352**	0.0162	0.3220
	Moments (n = 5)	0.0331	0.7232	0.7553	0.0618	0.9517	0.8605	0.2315	0.1691	0.2092	**0.0352**	0.0163	0.3231
9x9	Raw	0.0110	**0.8989**	**0.9259**	**0.0361**	**0.9723**	**0.9192**	**0.1044**	**0.5792**	**0.6549**	0.0464	0.0133	0.2582
	LBP	**0.0106**	0.8800	0.9108	0.0461	0.8891	0.8891	0.1495	0.4232	0.4817	0.1458	**0.0294**	**0.5602**
	Moments (n = 2)	0.0864	0.3287	0.3803	0.0758	0.9439	0.8248	0.2445	0.1333	0.1673	**0.0339**	0.0162	0.3207
	Moments (n = 5)	0.0675	0.4588	0.5062	0.0724	0.9417	0.8377	0.2439	0.1328	0.1657	0.0344	0.0163	0.3220



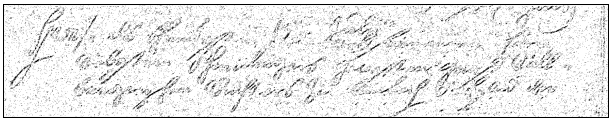
(a) Input



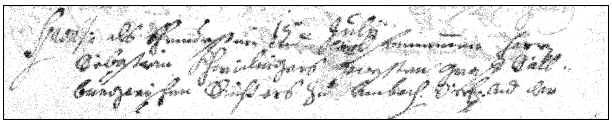
(b) Output



(c) Raw feature



(d) LBP feature



(e) Moments feature

Fig. 4. Comparative results from different features in an image from the DIBCO dataset

Future works will be focused on the implementation of various features based on different techniques, such as filters, like Gabor filters, or transforms, like the Fourier transform. Another task that will be tackled in the future is the automated selection of these features and their combination in the process of learning image operators.

ACKNOWLEDGMENT

This work is supported by FAPESP (2015/17741-9, 2015/01587-0) and by CNPq (484572/2013-0). Augusto C. M. Silva is supported by FAPESP (2017/09137-0), Igor S. Montagner has received support from FAPESP (2014/21692-0, 2011/23310-0), and N. S. T. Hirata is partially supported by CNPq.

REFERENCES

[1] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.

[2] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer Berlin Heidelberg, 2010.

[3] H. J. Heijmans, "Morphological image operators," *Advances in Electronics and Electron Physics Suppl., Boston: Academic Press,— c1994*, 1994.

[4] I. S. Montagner, N. S. T. Hirata, and R. H. Jr, "Image operator learning and applications," in *29th Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 2016.

[5] N. S. T. Hirata, "Multilevel training of binary morphological operators," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 707–720, 2009.

[6] M. M. Dornelles and N. S. T. Hirata, "Selection of windows for W-operator combination from entropy based ranking," in *28th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug 2015, pp. 64–71.

[7] I. S. Montagner, N. S. T. Hirata, R. Hirata, and S. Canu, "NILC: A two level learning algorithm with operator selection," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1873–1877.

[8] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1. IEEE, 1994, pp. 582–585.

[9] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[10] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[11] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *SPIE Medical Imaging*, J. M. Fitzpatrick and M. Sonka, Eds., vol. 5370, SPIE. SPIE, 2004, pp. 648–656.

[12] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "ICFHR2014 Competition on Handwritten Document Image Binarization (H-DIBCO 2014)," in *14th International Conference on Frontiers in Handwriting Recognition*, Sept 2014, pp. 809–813.

[13] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos, "ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016)," in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 619–623.