Advances in Intelligent Systems and Computing 326

Viet-Ha Nguyen
Anh-Cuong Le
Van Nam Huynh *Editors*

Knowledge and Systems Engineering

Proceedings of the Sixth International Conference KSE 2014

Springer

326 Nguyen · Le · Huynh *Eds.*

Knowledge and Systems Engineering

Advances in Intelligent Systems and Computing 326

Viet-Ha Nguyen · Anh-Cuong Le · Van Nam Huynh *Editors*

Knowledge and Systems Engineering

Proceedings of the Sixth International Conference KSE 2014

This volume contains papers presented at the Sixth International Conference on Knowledge and Systems Engineering (KSE 2014), which was held in Hanoi, Vietnam, during 9–11 October, 2014. The conference was organized by the University of Engineering and Technology, Vietnam National University, Hanoi. Besides the main track of contributed papers, this proceedings feature the results of four special sessions focusing on specific topics of interest and three invited keynote speeches. The book gathers a total of 51 carefully reviewed papers describing recent advances and development on various topics including knowledge discovery and data mining, natural language processing, expert systems, intelligent decision making, computational biology, computational modeling, optimization algorithms, and industrial applications.

Engineering
ISSN 2194-5357

ISBN 978-3-319-11679-2

9 783319 116792

▸ springer.com

Advances in Intelligent Systems and Computing

Volume 326

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

Editors
Viet-Ha Nguyen
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi
Vietnam

Van Nam Huynh
School of Knowledge Science
Japan Advanced Institute of Science and Technology
Nomi
Japan

Anh-Cuong Le
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi
Vietnam

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-11679-2 ISBN 978-3-319-11680-8 (eBook)
DOI 10.1007/978-3-319-11680-8

Library of Congress Control Number: 2014951000

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Semantic Regions Recognition in UAV Images Sequence

Stéphane Lathuilière, Hai Vu, Thi-Lan Le,
Thanh-Hai Tran, and Dinh Tan Hung

Abstract. In this work, we describe a framework to analyze UAV videos content. A multi-class image segmentation approach is proposed considering UAV videos specific characteristics. A static image segmentation is applied on each frame. After a preprocessing step on resulting segments, a SVM classifier is used to recognize regions. A Markov model is introduced to combine the results from the previous frames in order to improve the accuracy. The framework has been designed to be as flexible as possible with an eye to allow to insert holistic information into the model.

1 Introduction

It is commonly accepted that there is a growing need for efficient video content analysis tools. In addition UAVs have been used more and more lately and gained popularity both in the general public and the engineering world. Image sequences captured from a camera attached to a small drone usually contain background such as trees, constructions (buildings and roads), grass and sky. The goal is to partition each frame in multiple segments and to label them with these basic categories. Detecting more special objects as cars is another problem and requires using other techniques. Nevertheless, our model has to be flexible to be able to integrate this kind of methods.

Stéphane Lathuilière · Hai Vu · Thi-Lan Le · Thanh-Hai Tran
Research Institute MICA,
Hanoi University of Science and Technology, Hanoi, Vietnam
e-mail: {stephane.lathuiliere,hai.vu,thi-lan.le,
thanh-hai.tran}@mica.edu.vn

Stéphane Lathuilière
Ensimag - Grenoble INP, France

Dinh Tan Hung
Department of Aeronautical & Space Engineering
Hanoi University of Science and Technology, Hanoi, Vietnam
e-mail: dinhtanhung@gmail.com

V.-H. Nguyen et al. (eds.), *Knowledge and Systems Engineering*,
Advances in Intelligent Systems and Computing 326, DOI: 10.1007/978-3-319-11680-8_25

Videos taken from such devices have their own specific characteristics: moving camera, high distance from objects, view from up high, non constant angle between camera axis and the ground. As a consequence, specific techniques and software have to be developed to analyze their content. Research on image segmentation for the last 30 years has led to various algorithms and methods. Choosing an efficient tool in a particular case is not an easy task. In addition temporal information has to be included at best whereas algorithms on static image are not necessary easily adaptable. In the presented work, a framework is proposed to analyze UAV videos content. In the proposed method, the frames are analyzed one after the other but a temporal model is used to combine results through time and increase the precision on every single frame.

The paper is organized as follows. Section 2 describes previous related work and gives a first overview of our method. Section 3 goes through the segmentation algorithm which makes up the first step of our procedure. Section 4 focuses on the classification step. The temporal model is described in Section 5. Then the results are presented in Section 6. A discussion about future work concludes the paper in Section 7.

2 Background and Overview of the Proposed Algorithm

2.1 Previous Work on Video Analysis and Image Segmentation

UAVs are widely used for different applications. The main one is high quality images generation for applied fields as glaciology [14] or soil surface modeling [8]. UAV videos content has been analyzed to detect and count vehicles [4] but scene taken from UAV understanding is still a domain which needs to be more deeply explored.

Two classical computer vision issues can be compared with our issue: classical scene understanding and aerial images analysis.

First, scene understanding is considered as one of the next great challenges of computer vision. Many approaches are studied: objects detection [7], multiclass segmentation [11, 22] or reasoning about the 3D scenes[15]. Some algorithms try to obtain an holistic scene understanding combining the different approaches [13]. While such methods provide good results on single image, it seems difficult to apply it by now on video because of high time cost [12]. In addition, hypothesis are done about the camera height in most of these techniques [12, 13, 15]. Some methods [13] also compute the distances to the horizon for each pixel whereas such a measure cannot be used in our case. In fact the angle between the camera axis and the ground is nonconstant. Then retrieving the horizon position would be more difficult [2] and not very helpful.

Secondly, the particular camera height makes the studied problem similar to aerial images analysis. Even though, objects are further than in our case, color and texture features have a role as much crutial. Several techniques have been used for this problems: color, texture and structure features [9], 3-D reasoning [18], morphological operators [1].

2.2 Proposed Algorithm Overview

Our algorithm tries to use spatial information and temporal dimension to get a satisfying segmentation. However, a substantial part of our algorithm is applied on each frame ignoring the temporal dimension.

- Static step: this step is illustrated in Fig.1. On each frame a two parts algorithm is performed:
  - First, a segmentation of the frame is computed.
  - Then each segment of the image is labeled thanks to a recognition algorithm. Statistical descriptors of each component are computed and a Support Vector Machine (SVM) algorithm is used to predict the class of each component of the frame. More details about the segmentation step are given in Section 3.

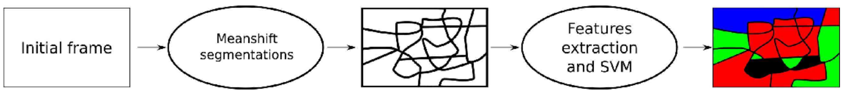


Fig. 1 Outline of the static algorithm step applied on every single frame

- Temporal step: to combine results frame after frame, a Markov chain is used pixel by pixel (Fig.2).

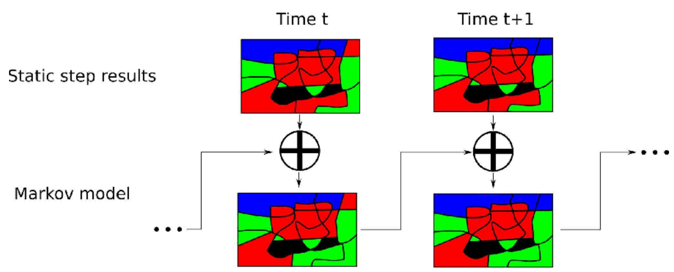


Fig. 2 Outline of the temporal model

More details about the temporal model are given in Section 5.

3 Proposed Approach for Segmentation Step

3.1 Segmentation Algorithm

A really precise segmentation cannot be obtained with a high frame rate. For instance, computing time of sophisticated algorithms such as [12] can reach up to 10 minutes per frame.

As a consequence, a mean-shift algorithm has been chosen. In a first time, the segmentation can be only geometric and not semantic. Mean-shift method has already been widely studied [5, 6, 10].

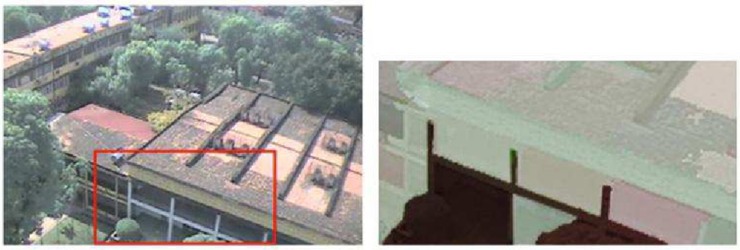
Defining the best parameters for the segmentation step is not an easy task. In fact, segmentation parameters affect the size, the homogeneity of components and border of components. A solution is proposed in Section 5.

3.2 Segmentation Results Pruning

After applying the mean-shift algorithm, the small components are automatically removed. Two reasons for that:

- If an object is made of many parts with different colors, all the parts must be in the same component to be efficiently recognized. For example, a building can consist of many different colors and a similar variety of colors cannot occur in *Tree* or *Sky*. If each color is isolated, segments cannot be recognized efficiently.
- If a component is too small, its descriptors have irregular properties due more by the short size of this sample than by the nature of the object.

However after this step, there is a remaining problem: some components are bigger than the minimal requested size but contain meaningless long branches. This situation is shown in Fig.3.



(a) Original image (b) Zoom on a meaningless segment

Fig. 3 Example of a meaningless segmentation: original image and resulted segmentation

To solve this problem, morphological operators have been used. An opening and a closing algorithms [21] are applied on the matrix containing all the labels. Here morphological operators are applied whereas the classical order over integers has no meaning. So opening and a closing have been applied not to favor components with high or low labels. However, after these operations, a component can be divided in multiple non connected components. The resulting segmentation must be analyzed again to separate not connected components.

3.3 Over-segmentation of Vast Segments

Initially, applying the algorithm on vast and uniform components as sky led to bad results. It could be explained by the presence of some objects in the middle of the zone or a not clear boundary with neighboring segments. An example is shown in Fig.4.

(a) Original segmentation (b) Computed over-segmentation

Fig. 4 Example of object in a vast and uniform segment: computed over-segmentation

To solve this problem the sky must be divided into many rectangular parts as shown in Fig.4. Most of the parts have a uniform color and can easily be recognized as belonging to the sky.

4 Classification Scheme

4.1 Learning from Segmented Images

The learning algorithm follows the scheme below (Fig.5):

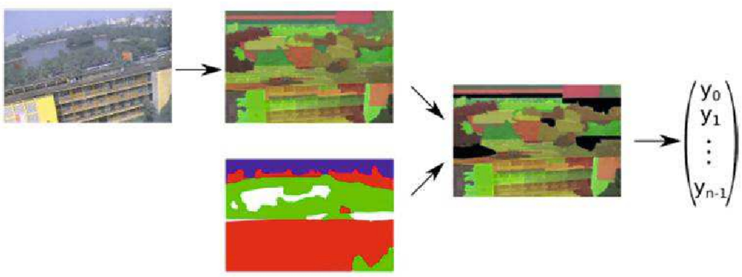


Fig. 5 Scheme of the 3 steps features extraction algorithm: segmentation, comparison with mask resulting of handmade and feature vectors extraction

More precisely, the 3 steps are:

- The segmentation from each training image are computed (see Section 3).
- The result of the segmentation is compared with the handmade labeled image. Only the segments containing $p\%$ of pixels belonging to only one class are kept. As a consequence, the segments which contain pixels which should belong to several classes are ignored. The parameter p allows to increase or decrease the number of descriptors depending on the precision of the segments used for training.
- The features are computed for each correct segment.

4.2 Features Extraction

Statistical descriptors which recognize efficiently textures have to be chosen. In the studied case, textures have to be recognized on not square zones. This specificity has to be taken into account.

Classical descriptors as HOG [7] and SIFT features [19] have not been used for two reasons. They are more suited for close objects recognition than for texture classification and they are quite time-consuming. Local Binary Patterns [20] have not been used as they have not led to good results.

Some of used descriptors have been inspired by [9]. Even though the aim of this article is quite different, the data is similar. Following descriptors have been chosen:

- Histogram of "bicolor" representation: 4 bits of one color space and 4 bits from another base. Best results have been obtained with 4 bits from grayscale and 4 bits from Hue.
- Mean and variance over the all zone of R, G and B dimensions
- Gabor filter: mean over the all zone for different σ (standard deviation of the Gaussian envelope) and θ (orientation of the filter). The chosen kernels are shown below (see Fig.6).

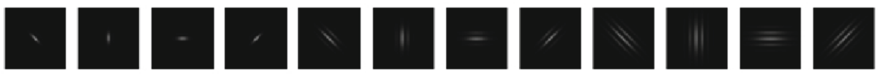


Fig. 6 Used Gabor kernels

- The y coordinate of the component geometric center.
- The size of the component (number of pixels).

The last two descriptors have been added because a large component on the top of the image is more likely to belong to the sky.

4.3 Class Representation

For each segment of the frame, the prediction method of the SVM is performed. It gives us a label corresponding to one of the classes. However labeling segments with a single integer is not flexible. Another class representation has been chosen.

Each pixel of the image matches with a class (*Tree*, *Building*...). Vectors of probabilities are used for each pixel:

$$Y_t = \begin{pmatrix} y_{t,1} \\ y_{t,2} \\ \vdots \\ y_{t,n} \end{pmatrix}. \tag{1}$$

The variable n is the number of considered classes. $y_{t,i}$ denotes the probability at the time t of belonging to the class C_i.

This representation has two advantages:

- More information than only the index of the maximum of the probability are stored. As a consequence, this information can be used to get more stable result through the time.
- Labeling matrix can be seen as a field. If single index representation were used, an interesting norm to compare to pixels could not be defined. The integer order relation is not meaningful in this case. Thanks to this model, the vectors belong to a subset of $[0,1]^4$. This property could allow to use more powerful mathematical tools in the future.

More specifically, two variables are considered in the following paragraphs:

- X_t: real class the pixel belongs to.
- Y_t: observations, resulting from the recognition algorithm.

The model defining the link between X_t and Y_t is described in 5

4.4 Specific Multiclass SVM Use

The basic SVM algorithm can be used only in binary problems. Many techniques have been used to extend SVM to multiclass segmentation.

4.4.1 "One-against-one" Approach

"One-against-one" approach [17] is a classical method to build a multiclass SVM based on several binary SVM. The comparisons of methods between multiclass SVM have shown that is a competitive method [16]. That is the method used in the most widely employed SVM library "libsvm" [3].

As carefully described in [17], the algorithm is divided in two parts:

- First, the procedure tries to find classes linearly separable from all others.
- Then each pair of classes is separated.

In our case, the first step has been ignored for convenience. As four classes are considered (*Construction*, *Tree*, *Grass/field* and *Sky*), 6 classifiers have been used.

4.4.2 From Classifiers Results to Vectors of Probabilities

After applying the 6 classifiers, votes are computed to get the best class. A class can get up to 3 votes. If a class gets 3 votes, a probability α is assigned to this class. Then the other classes get a probability proportional to their vote count. However a minimal threshold probability ε is assigned to class without any vote. The parameters α and ε enable to configure how much the result of our classification can be trusted. If α is close to 1, the prediction gives a really high probability to the class which reaches 3 votes. Conversely, if α is low, probabilities belonging to different classes will be more homogeneous.

5 Temporal Model

5.1 State-Space Model

In this Section, the probabilistic framework is defined and in this way, the notations defined in 4.3 are used. We assume that the model is first-order Markov. Similarly, the observations are modeled as a first-order Markov model. As a consequence only the following probabilities have to be defined:

- $P(X_t|X_{t-1})$: state-transition function. It is the probability of a pixel changing from one class to another between two frames.
- $P(Y_t|X_t)$: observation function, the result of our static recognition algorithm.

Thanks to the first order Markov assumptions, the probability of each state can be written as follows:

$$P(X_t|Y_{1:t} = y_{1:t}) \propto P(X_t|y_t) \left(\sum_{x_{t-1}} P(X_t|x_{t-1}) P(x_{t-1}|y_{1:t-1}) \right) \tag{2}$$

Thus the probability is recursively computed and only the result of the previous frame needs to be memorized. This model is really similar to a hidden Markov model. However, the output Y_i is not known. The observation of the model is a probability vector and not Y_i itself.

In this article, only the following state-transition function is considered:

$$P(X_t = x_{t,i}|X_{t-1} = x_{t-1,j}) = \delta_{i,j}.\alpha + (1 - \delta_{i,j}).\frac{1-\alpha}{n-1} \tag{3}$$

In other words, a pixel has a probability α not to change its class and $(1 - \delta_{i,j}).\frac{1-\alpha}{n-1}$ to change to other classes. The parameter α allows to configure the inertia of the model. The more α is close to 1 the more the model has a high inertia. In future work, more sophisticated state-transition functions will be used. The main idea is to use holistic analysis of the current segmentation to favor some classes.

6 Experimental Results

In order to test the algorithm, some frames are selected in testing videos. They are labeled by hand. The algorithm is applied on the testing videos and the results are compared with the labeled masks of selected frames.

The result and the mask are compared pixel by pixel. If the pixel of the mask is white, the pixel is ignored. In fact, some pixels cannot be labeled by hand with certainty because of the high distance of the camera from the objects or because this class of object is not supported. Then, the precision and recall of the recognition algorithm can be computed.

Training data contains 66 images extracted from UAV videos which have been labeled by hand. The tests were performed using an intel Core i5 3.20GHz. The training step took 48s.

Testing experiments have been conducted on 6 other videos. They have been selected to test the algorithm on varied context: urban, countryside and mountains. Videos have been captured in HD(1920×1080) but have been resized to (480×270). The images sequences total duration is 1m22s at 30 frames per second. 47 frames have been labeled by hand. The algorithm reached 0.84 frames per second. Fig.7 shows examples of returned segmentation.

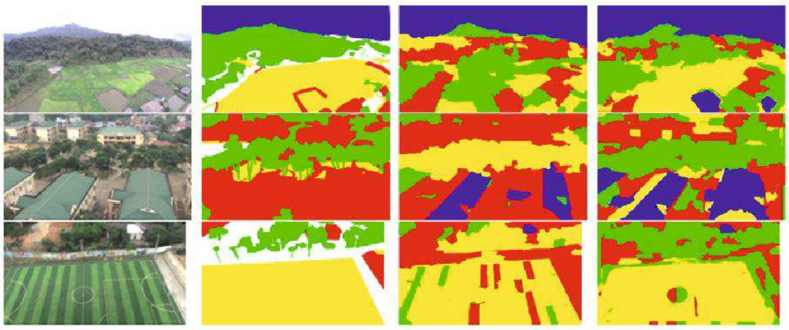


Fig. 7 Examples of result on test frames: original images, expected segmentations, returned segmentations without temporal model and with temporal model: green=*Tree*, red=*Construction*, blue=*Sky* and yellow=*Grass/field*

Table 1 shows precisions and recalls. In order to illustrate the part played by the proposed Markov model, the output of the static step is compared with the output of the temporal model. Table 2 details more precisely errors made by the proposed model. The results are satisfying for all classes. The good results on sky can be noticed. The sky color and geometric characteristics make it easier to detect.

The main difficulty is recognizing *Construction*. It can be explained by the fact that boundaries between trees and buildings are not very sharp. As a consequence, the segmentation is not precised enough. Moreover, buildings have varied colors and geometric characteristics. The temporal model does not improve significantly

Table 1 Precision and recall on testing data

	Without temporal model		With temporal model	
	Precision	Recall	Precision	Recall
Tree	0.56	0.39	0.94	0.54
Construction	0.51	0.75	0.56	0.76
Sky	0.96	0.99	0.96	0.99
Grass/field	0.51	0.48	0.65	0.79

Table 2 Confusion matrix of test frames, expected label on rows and returned label on columns. The numbers of pixels have been divided by the image size.

	Tree	*Construction*	*Sky*	*Grass/field*
Tree	5.66	3.51	0.04	1.33
Construction	1.12	5.89	0.21	1.48
Sky	0.00	0.00	6.04	0.01
Grass/field	0.27	1.18	0.00	5.36

the results for *Construction* but it appears to be really efficient at recognizing *Tree* and *Grass/field*. As the results of the static step are unstable through time with *Tree* and *Grass/field*, the proposed temporal model helps results.

7 Conclusion and Perspectives

In this paper, we have described an algorithm for segmentation of UAV videos. We got satisfying results on our test videos. In further work, the proposed framework has to be tested on more varied data sets in order to evaluate the sensitivity to lighting, illumination conditions and video quality. Although the temporal dimension is not included in the segmentation step, the proposed Markov model expands the possibilities of better temporal analysis. The temporal dimension can be used to converge to a geometrically and semantically consistent segmentation. First tries using holistic approaches to define the state-transition function ensure a promising development.

Acknowledgements. This research is funded by Hanoi University of Science and Technology under grant number T2014-130.

References

[1] Benediktsson, J.A., Pesaresi, M., Amason, K.: Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. IEEE Transactions on Geoscience and Remote Sensing 41(9), 1940–1949 (2003)

[2] Boroujeni, N.S., Etemad, S.A., White, A.: Robust horizon detection using segmentation for uav applications. In: 2012 Ninth Conference on Computer and Robot Vision (CRV), pp. 346–352. IEEE (2012)
[3] Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
[4] Cheng, P., Zhou, G., Zheng, Z.: Detecting and counting vehicles from small low-cost uav images. In: ASPRS 2009 Annual Conference, Baltimore, vol. 3, pp. 9–13 (2009)
[5] Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1197–1203. IEEE (1999)
[6] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
[7] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
[8] Eltner, A., Mulsow, C., Maas, H.G.: Quantitative measurement of soil erosion from tls and uav data. In: ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 1(2), pp. 119–124 (2013)
[9] Fauqueur, J., Kingsbury, N., Anderson, R.: Semantic discriminant mapping for classification and browsing of remote sensing textures and objects. In: IEEE International Conference on Image Processing, ICIP 2005, vol. 2, pp. II–846. IEEE (2005)
[10] Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21(1), 32–40 (1975)
[11] Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 670–677. IEEE (2009)
[12] Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1–8. IEEE (2009)
[13] Gould, S., Gao, T., Koller, D.: Region-based segmentation and object detection. In: NIPS, vol. 1, p. 2 (2009)
[14] Hodson, A., Anesio, A.M., Ng, F., Watson, R., Quirk, J., Irvine-Fynn, T., Dye, A., Clark, C., McCloy, P., Kohler, J., et al.: A glacier respires: quantifying the distribution and respiration co2 flux of cryoconite across an entire arctic supraglacial ecosystem. Journal of Geophysical Research: Biogeosciences (2005–2012), 112(G4) (2007)
[15] Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision 80(1), 3–15 (2008)
[16] Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13(2), 415–425 (2002)
[17] Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Neurocomputing, pp. 41–50. Springer (1990)
[18] Lin, C., Nevatia, R.: Building detection and description from a single intensity image. Computer vision and image understanding 72(2), 101–121 (1998)
[19] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
[20] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)