



A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery

Juliano E. C. Cruz^{a1}, Elcio H. Shiguemori^{b2} and Lamartine N. F. Guimarães^{c3}

^aCAP, National Institute for Space Research (INPE), São José dos Campos, SP, Brazil

^bEGI-S, Institute for Advanced Studies (IEAv), São José dos Campos, SP, Brazil

^cENU, Institute for Advanced Studies (IEAv), São José dos Campos, SP, Brazil

Received on october 11, 2015 / accepted on december 20, 2015

Abstract

In this article, the three most used object detection approaches, Linear Binary Pattern cascade, Haar-like cascade, and Histogram of Oriented Gradients with Support Vector Machine are applied to automatic runway detection in high resolution satellite imagery and their results are compared. They have been employed predominantly for human feature recognition and this paper tests theirs applicability to runways. The results show that they can be indeed employed for this purpose with LBP and Haar approaches performing better than HOG+SVM.

Keywords: *runway detection, satellite imagery, boosting, boosted cascade, LBP, Haar-like, HOG, SVM, computational mathematics.*

1. Introduction

Experts can usually recognize most of the targets in satellite imagery with high spatial resolution. Some objects are easy to be visually recognized; others however, require a high level of expertise. Due mainly to automatic object detection and identification evolution, today it is possible, to aid human operators, or even substitute them, in tasks dealing with images or videos [1, 2].

There is a wide range of applications that can employ some sort of automatic detection or identification procedures. UAV (Unmanned Aerial Vehicle) platforms are one of them. They are currently widely in use for military, police, and civilian applications [3].

Automatic runway recognition can be an important task for UAVs. They can use it for landing, for air strikes, or even for self-localization procedures [4]. The best runways to perform such procedure are made of concrete and asphalt. They can be found, generally, in medium to large cities and air force bases.

UAVs can have fully-autonomous or semi-autonomous decision-making systems, both of which could use the detection approaches presented in this paper. The semi-autonomous system use is relevant, because human-based recognition is highly susceptible to error. Human recognition processes generally require a huge amount of data processing and it can be a tedious task. Furthermore, in some cases the operator must be previously trained.

This paper analyzes the applicability of the Haar-like cascade classifier [5], the LBP (Linear Binary Patterns) cascade classifier [6] and the HOG+SVM (Histogram of Oriented Gradients with Support Vector

¹juliano.cruz@lac.inpe.br

²elcio@ieav.cta.br

³guimarae@ieav.cta.br

Machine) [7] in concrete and asphalt runway detection. They all are general purpose methods, but they have been employed more often in human feature detection.

1.1. Related work

In the literature, the most used object recognition approaches in digital images are the Haar-like cascade classifier, the LBP cascade classifier, and the HOG+SVM. These three approaches provide good performance, robustness, and, in some cases, real-time performance [8, 9]. Outside the academic community the use of these approaches is also quite high, as seen by the large amount of tutorials and forum threads present on the Internet. Most of the use cases are for human features, although these approaches are for detecting any kind of object.

The combination of HOG and SVM was first employed in [7], where it was used for human profile recognition. Since then other papers have been published. Gritti et al. [10] compared different approaches with parameter variations in order to obtain a good performance in face recognition. Zhang et al. [11] propose modifications to improve performance for use in any kind of object. In [12], a system that has a specific classifier for each part of an object is proposed, which recognizes an object when the specific classifiers are in a particular disposition in a certain area and all of them have positive responses. Dealing with low quality airborne imagery, [13] performs human profile detection. All these studies use only linear SVM.

Viola and Jones [5] were among the first to employ Haar-like cascade classifiers, in their case, for face recognition. Subsequently, [9] show the robustness of this approach for general purposes. Further applications employ the Haar-like cascade exclusively for face recognition. In [14], a three-stage system is used in order to increase robustness and performance for face recognition. In order to decrease the false positive rate, [15] statistically detects where face features most appear and then classifies the image. In [16], a two-stage system for face recognition employs the Haar-like cascade in its first stage. The study [17] proposes some variations in Haar-like features to improve face recognition performance.

There are some papers that work with other types of targets, as in [2], where a two-stage system for hand gesture recognition uses a Haar-like cascade in the first stage. Breckon et al. [18] use imagery captured by UAVs for vehicle recognition using images in the visible spectrum. Likewise, Gaszczaka et al. [8] study UAV imagery in visible and thermal spectrum for recognition of vehicle and people.

The LBP approach, together with the concept of cascaded classifiers, was first used in object recognition in [6], where it was employed for face recognition. Most papers that employ LBP cascade classifiers perform face recognition [19, 10, 20]. In [21], a multi-block LBP approach is proposed for face recognition. Fang and Wang [22] extend the idea to implement gender recognition. There are also studies that detect pedestrians, such as [23], that uses optical and thermal images, [24] that detects and tracks objects, and [1], which detects pedestrians, including recognizing head and face. There are studies for biometric identification, such as [25] which proposes palm identification. Trefny and Matas [26] make changes in the LBP operator and tests it on a face, cars, and gender recognition dataset.

Few studies exist in the literature with reference to runway recognition. Most employ the Canny operator with Hough transform [27, 28, 29] because, from the air, runways are composed of prominent straight lines, and the Hough transform is well known for detecting straight lines in images. But the question remains; can the other recognition methods be used as well? The article follows [30] to report studies on the applicability of LBP cascade, Haar-like cascade, and HOG+SVM for asphalt and concrete runway detection.

1.2. LBP cascade classifier

The Local Binary Pattern operator, also known as LBP, was first introduced by [31] for byte adaptation of a previous study done by [32]. LBP has shown to be a powerful texture descriptor. Texture is defined as a function of spatial variations in the pixel intensity of an image. It has been used in a wide variety of applications [33]. The idea behind this operator is that common features, such as edges, lines, point, can be represented by a value in a particular numerical scale. Therefore, it is possible to recognize objects in an image using a set of values extracted a priori.

The original LBP labels the pixels by thresholding the 3x3 neighborhood in relation to the central pixel

value, as shown in Equation 1. But, there are recent studies that extend this version [20, 26, 21].

$$LBP(x_p, y_p) = \sum_{n=0}^7 s(i_n - i_p)2^n, \quad (1)$$

where (x_p, y_p) is the pixel of an image, n represents the neighboring pixel, i_n and i_p the respective gray level of neighboring and central pixel, and $s(x)$ can be described by:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

The bit obtained for each neighboring pixel is used through a pre-defined order to form a final value. Using eight neighboring pixels, the final value will be between 0 and 255. Figure 1 shows the whole LBP operator extraction process and Figure 2 shows the application of LBP operator.

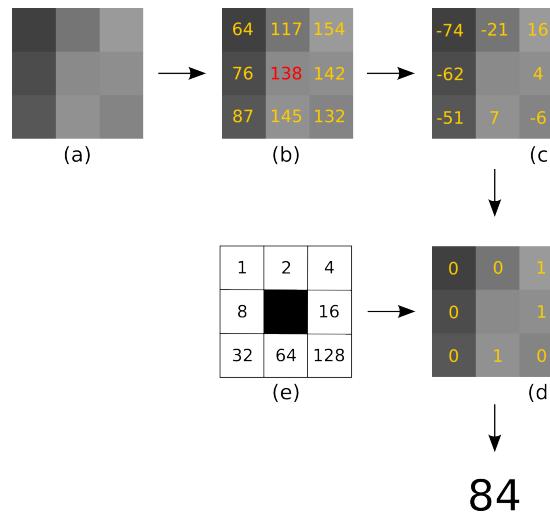


Figure 1: The LBP extraction process, where (a) is the image fragment to be processed, (b) shows gray levels, (c) shows subtraction results among periphery pixels and the central one, (d) shows the assignment of 0 to subtraction results less than 0 and 1 to subtraction results greater or equal 0, (e) shows the binary matrix values. In the final step, the binary matrix values in (e) are added together for every cell with a value of 1 in (d).

An extension to LBP was proposed by [21], called multi-block LBP, or simply MB-LBP. Instead of using single pixels, MB-LBP applies the LBP operator to pixel blocks. All blocks must have the same size and must respect the 3x3 formation, as seen in Figure 1. Then, what the LBP operator uses for its computation is the pixel block mean value. The advantage of MB-LBP is to have a faster and more precise classifier [21].

MB-LBP binary patterns can be used to detect different structures such as edges, lines, spots, flat areas and corners at different scale and location. Since MB-LBP features are non-metric values, it is not possible to use a threshold-based function as a weak classifier.

Thus, a decision tree is used for this purpose. These weak classifiers are slightly better than a random guess, but when set in cascade, they become a strong classifier (Figure 3) with high discrimination power, capable of detecting structures despite illumination, color, or scale variation [9]. The cascade is a combination of successively weak classifiers, starting with simple ones and ending with the most complex ones. In spite of the fact that it seems to be an exhaustive search, the classifier building characteristic allows for early rejection with minimum evaluation and consequently, it has a low computational cost. It relies on the fact that the majority of the detection windows are negative and there are few windows that go through all stages. Therefore, the computational power will be focused on windows that have a higher probability of being positive, once they have already passed through the initial stages [5, 2].

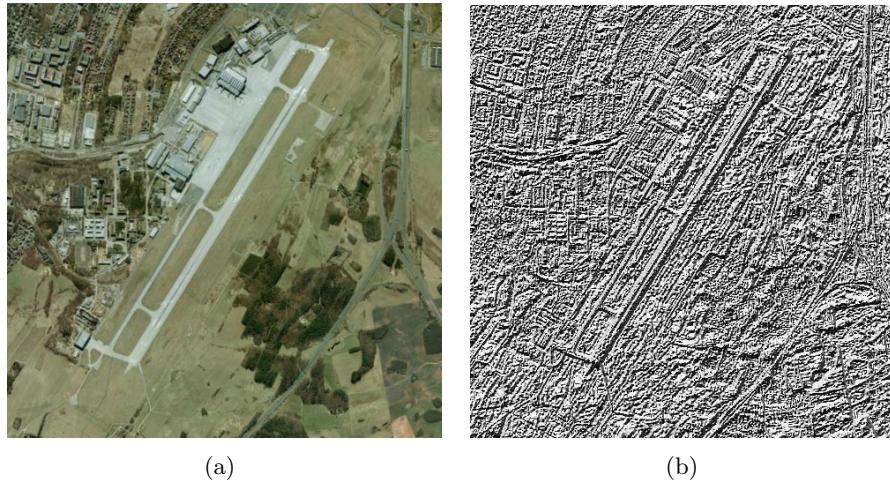


Figure 2: (a) Original image [34] and (b) resultant image after applying LBP operator.

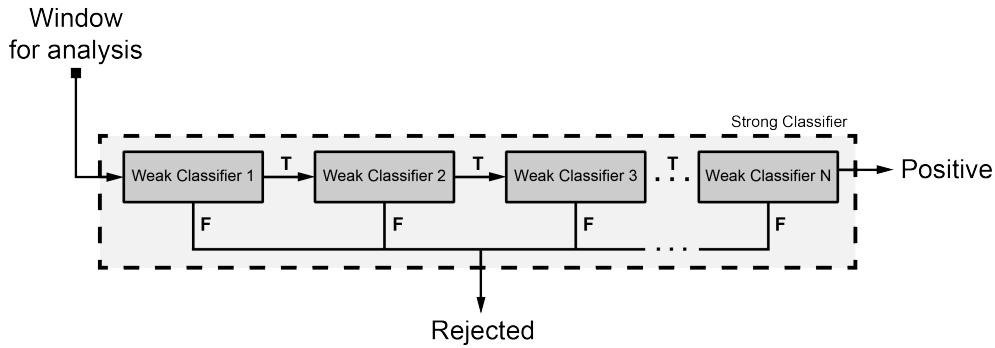


Figure 3: Cascade classifier concept.

In any window inside an image, a huge amount of MB-LBP features can be found. So, during the training stage, it is necessary to focus on a small set of critical features, discarding most of the non-critical ones in order to increase classification speed significantly without affecting accuracy. Boosting [35] is the method employed in the training stage to solve this problem [5]. It is an effective learning algorithm and has strong bounds on generalization performance.

Objects can appear in different regions of the image and in different scales. In order to solve this problem, the sliding window method is used [36]. It consists of a detection window that slides over an image extracting regions and classifying them. A Gaussian pyramid [37] is also applied to the image during the classification stage in order to perform a scale search.

Due to the fact that detection windows overlap each other, the same object can be counted more than one time. So, a concept called non-maximum suppression is employed, i.e. windows with a local maximum classifier response suppress nearby windows [36].

1.3. Haar-like cascade classifier

Haar-like features are attributes extracted from images used in pattern recognition. Their name comes from their similarity to Haar wavelets [38, 9]. The utilization of these features instead of handling gray or color level of the pixels directly was proposed in [39, 5]. First, the pixel values inside the black area are added together; then the values in the white area are summed, as seen in Figure 4. Then the total value of the white area is subtracted from the total value of the black area. This result is used to categorize image sub-regions.

As occurs in LBP cascades, weak classifiers become strong classifiers when arranged in sequence in Haar-like cascade (Figure 3). They are able to detect structures despite illumination, color or scale variation

[9, 40].

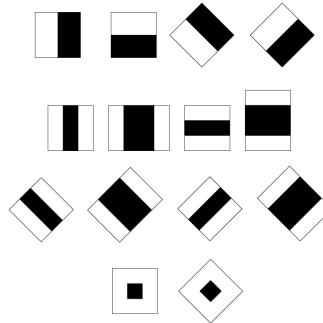


Figure 4: Haar-like features.

1.4. HOG+SVM

This approach is an association between two methods and works as follows: HOG is a local descriptor that uses a gradient vector orientation histogram and SVM is a classifier with good generalization power that uses the features extracted by the descriptor. The Histogram of Oriented Gradients, HOG, was first proposed in [7] and its main idea is that object appearance and shape can be described by pixel gradient distribution [10].

The descriptor extracting process can be divided into four steps: 1) calculate the vector gradient of each pixel; 2) group pixels in cells; 3) group cells in blocks; and 4) assemble the descriptor. First, one-dimensional masks of point discrete derivative, Equation 3, are applied in both horizontal and vertical axis in order to calculate each pixel gradient, as seen in Figure 5 (b) and Figure 6 (b).

$$[-1, 0, 1] \text{ and } [-1, 0, 1]^T \quad (3)$$

Second, the pixels are grouped into cells, as shown in Figure 5 (c), Figure 6 (c) and Figure 7 (a). In the third step, blocks are created by grouping the cells, as seen in Figure 5 (e) and Figure 7 (b). In the final step, Figure 5 (f), the descriptor is assembled. The descriptor is a cell histogram list of all blocks.

Local light variation or high contrast between foreground and background can be an issue in image processing. The way to attenuate this issue is by normalizing histograms according to their neighbors. Some studies [41, 21] claim that the best norm to be employed in HOG is the $L2_{\text{phys}}$. It consists of applying $L2_{\text{norm}}$, Equation 4, once, then limiting the results to a particular ceiling before applying $L2_{\text{norm}}$ once again.

$$L2_{\text{norm}} = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}, \quad (4)$$

where v is the descriptor vector, $\|v\|_k$ its k -norm to $k = 1, 2$ and e a tiny constant [7].

The Support Vector Machine was first described in [42] and is a supervised learning algorithm used mainly for classification and regression analyses. SVM is a binary linear classifier, but there are approaches that enable it to deal with non-linear or multiple class problems [43].

SVM works basically by finding a hyperplane that fits in the middle of two classes. Mathematically, a training dataset can be represented by \mathbf{X} , where \mathbf{x}_i , $i = 1, 2, \dots, N$ are its feature vectors. Considering a linear separability problem, these vectors belong to only two classes, ω_1 or ω_2 . The objective of SVM is to find a hyperplane, Equation 5 that classifies the training vectors correctly.

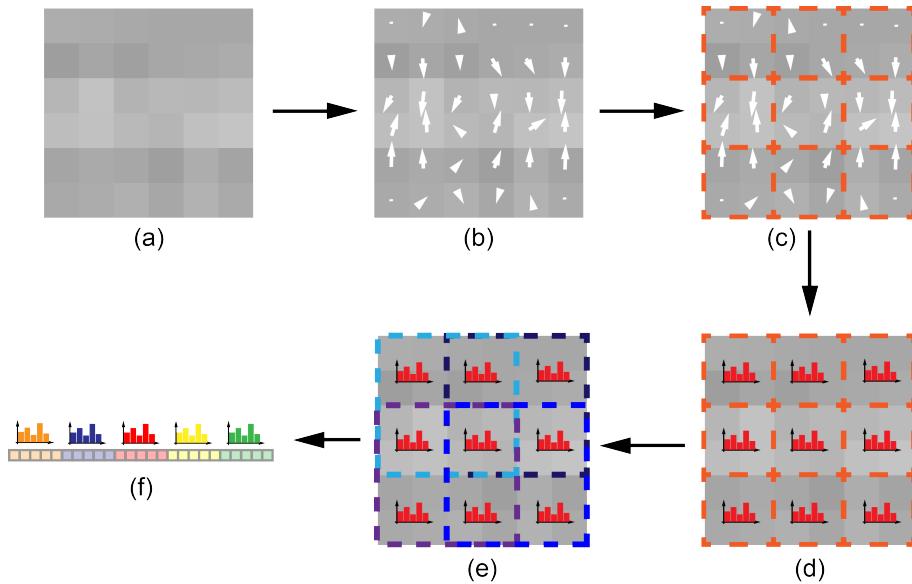


Figure 5: HOG descriptor extraction steps: (a) original image, (b) gradient vector calculatio, (c) pixel grouping in cells, (d) cell histogram calculation, (e) cell grouping in blocks and (f) descriptor assembly.

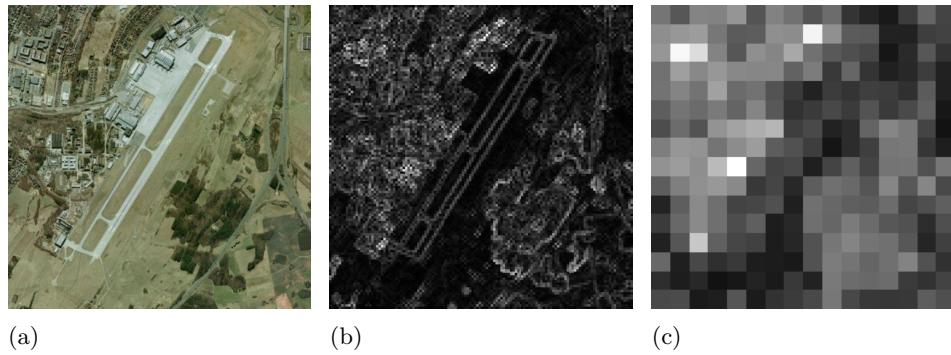


Figure 6: Original image (a) [34], pixel gradient magnitude (b) and cell gradient magnitude (c)

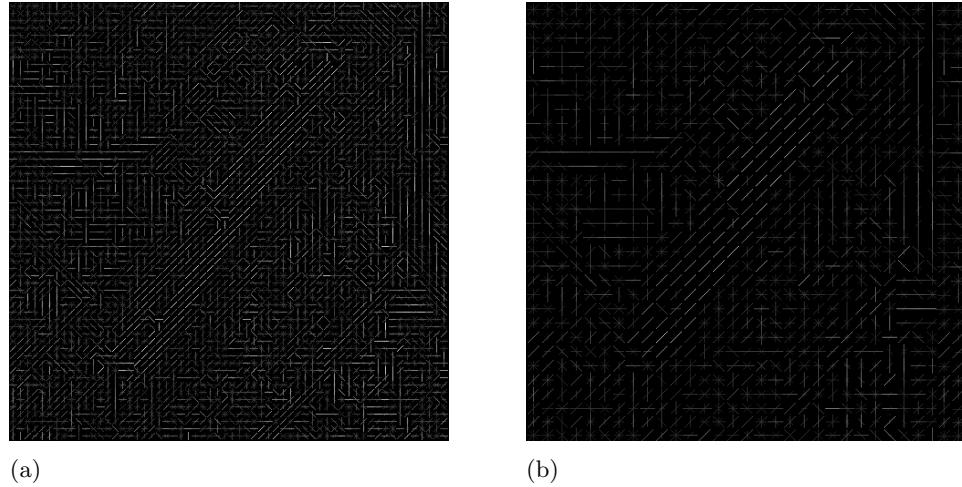


Figure 7: Cell gradient orientation histogram (a) and block gradient orientation histogram (b)

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0, \quad (5)$$

where \mathbf{w} is a matrix containing the support vectors and w_0 the bias.

There are infinite hyperplanes that can be placed between two points, or in this case, two classes. Figure 8 (a) shows three hyperplanes that could separate the two classes classifying the samples correctly. Thus, the classifiers generalization power must be taken into consideration, i.e. the classifiers capacity to work satisfactorily with data outside the training dataset. What SVM does is to choose the hyperplane with the largest margin between classes during the training process, as seen in Figure 8 (b) [43].

Kernel functions can be used with SVM in order to enable the classifier to deal with non-linearly separable classes. These functions modify the feature space trying to transform it into a linear separability problem. Some of the most popular kernel functions in the literature are the sigmoidal and RBF (Radial Basis Function) [44].

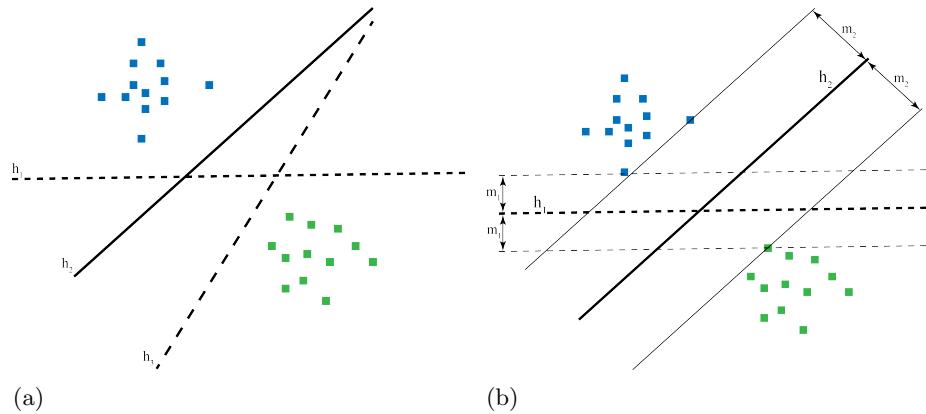


Figure 8: (a) Possible classifiers and (b) hyperplanes and their margins.

2. Methodology

Due to the fact that there is no runway dataset for public use and the previous studies [27, 28, 29] proposing runway detection did not make their own dataset available, it was necessary to create our own.

The training dataset utilized was a high spatial resolution image set composed of 2040 positive (Fig. 9) and 2500 negative samples. All images were obtained from the Google Maps application [34] where the positive samples were runway images from different parts of the world. The positive set was built by rotating the original images forty times, 9 degrees each time, in order to give certain rotation invariance. It is important to highlight that Google Maps images have watermarks with the Google logo and the year of capture scattered throughout the image.

Due to memory limitations, the detection window size and, consequently, the size of the positive set images could be, at maximum, 50x50 pixels during the training procedure in the cascade approaches (Haar-like and LBP). Two sizes were used, 50x50 and 32x32 pixels. The negative samples were not resized, because during the training stage, fragments with the same size as the positive samples were systematically extracted from the negative training set images.

In the HOG+SVM approach, windows with three different sizes were utilized: 32, 64, and 128. This approach permits window sizes greater or equal to 16 and they must be a power of 2.

Objects detected by the three approaches may be categorized as: True Positives (TP), like in Figure 10, which are runways that were correctly recognized; False Positives (FP), like in Figure 11 and Figure 12, that are regions of the image that were erroneously classified as runways; and False Negatives (FN), which are non-recognized runways, like in Figure 11 and Figure 12. True Negatives, which are correctly classified non-runway regions, were not used in this study because they appear in a huge number and do not add any substantial information besides that attained with TP, FP, and FN.

Two metrics [45] were utilized in order to measure classification accuracy of an image set obtained from different sensors. The metrics were the TPR (True Positive Rate), Equation 6, also known as Hit Rate, and the Precision, Equation 7.



Figure 9: Positive data set samples from the high resolution image set [34].

$$\text{TPR} = \frac{\text{TP}}{\text{Total Positives}} \cdot 100 \quad (6)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \cdot 100 \quad (7)$$

where Total Positives are the total amount of runways contained in the test dataset, or $TP + FN$.

A third metric was used to measure classification speed: Average Time per Pixel (ATpx), Equation 8.

$$\text{ATpx} = \frac{\text{Total Processing Time}}{\text{Total Pixel Amount}} \quad (8)$$



Figure 10: Cropped region showing a true positive example [34].



Figure 11: Cropped region showing a false positive example [34].

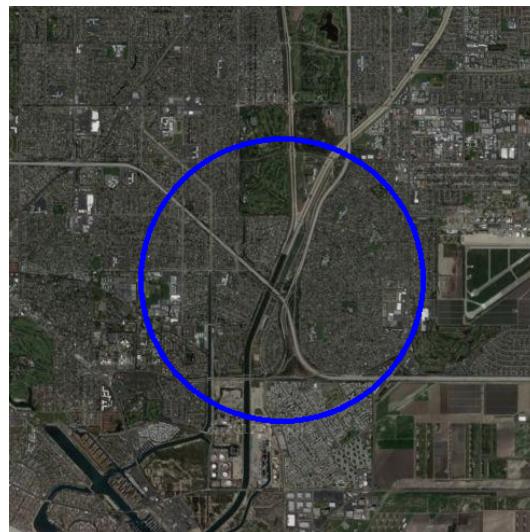


Figure 12: Cropped region showing a false positive example [34].

3. Results

The classification test set was composed of 50 images which were not present in the training dataset. They were not synthetic images, therefore, they were processed just the way they appeared in Google Maps. Each image contained a runway, where the image average size was 1200x600 pixels and the runway length varied from 200 to 400 pixels. The classification process was not parallelized. It was executed in a 2.40 GHz Intel i7- 3630QM processor with 8GB RAM. The classification results and the performance indexes are shown in Table 1, Table 2, and Table 4.

The training parameters used in the cascade approaches were the following: 99.5% for minimum hit rate, 50% for maximum false positive and cascade sizes ranging from 20 to 40 stages. Results and performance metrics can be found in Table 1 and Table 2 as well in Figure 13 and Figure 14.

The results of Haar-like cascade and LBP cascade had similar hit rate, around 80%. However, the precision of the two approaches behaved differently, most of the LBP cascade classifiers had at minimum 78% and half of the Haar-like cascade had precision around 50%. Regarding processing speed, their classifiers behaved alike, if classifiers with same quantity of stages are compared.

Table 1: Classification with Haar-like cascade.

Win.	Stages	TP	TPR	FP	FN	Prec.	ATpx (ms)
32px	20	40	80%	47	9	46%	$4,11 \cdot 10^{-4}$
	25	40	80%	36	9	52,6%	$4,81 \cdot 10^{-4}$
	30	40	80%	37	9	51,9%	$3,76 \cdot 10^{-4}$
	35	41	82%	39	8	51,3%	$3,61 \cdot 10^{-4}$
	40	43	86%	60	7	43%	$2,43 \cdot 10^{-4}$
50px	20	39	78%	13	11	75%	$37,9 \cdot 10^{-4}$
	25	40	80%	10	10	80%	$48,1 \cdot 10^{-4}$
	30	39	78%	12	11	76,5%	$48,66 \cdot 10^{-4}$
	35	39	78%	11	11	78%	$36,16 \cdot 10^{-4}$
	40	40	80%	11	10	78,4%	$33,67 \cdot 10^{-4}$

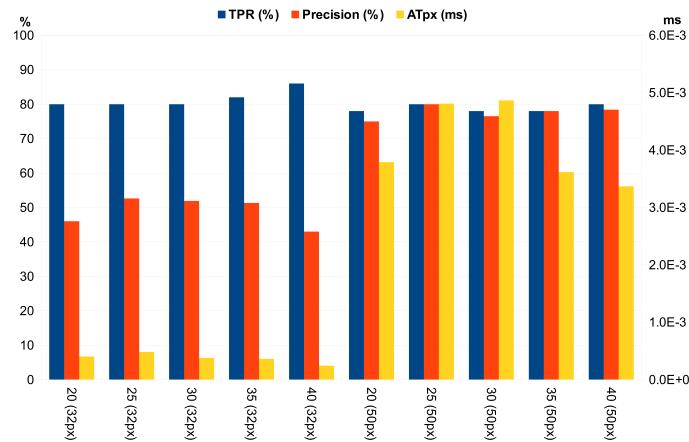


Figure 13: Haar-like cascade results.

Table 2: Classification with LBP cascade.

Win.	Stages	TP	TPR	FP	FN	Prec.	ATpx (ms)
32px	20	39	78%	11	11	78%	$5 \cdot 10^{-4}$
	25	39	78%	11	11	78%	$2,85 \cdot 10^{-4}$
	30	39	78%	11	11	78%	$1,69 \cdot 10^{-4}$
	35	40	80%	11	10	78,4%	$1,11 \cdot 10^{-4}$
	40	38	76%	24	12	61,3%	$0,78 \cdot 10^{-4}$
50px	20	39	78%	11	11	78%	$40,8 \cdot 10^{-4}$
	25	39	78%	11	11	78%	$26,4 \cdot 10^{-4}$
	30	40	80%	10	10	80%	$15,3 \cdot 10^{-4}$
	35	40	80%	10	10	80%	$8,82 \cdot 10^{-4}$
	40	40	80%	10	10	80%	$5,35 \cdot 10^{-4}$

Table 3 shows the classifier configurations found during the training process in the HOG+SVM approach. Results and performance metrics can be found in Table 4 and Figure 15.

The HOG+SVM approach had an outcome quite different from the other two approaches. The classifiers varied hit rate from 24% to 94% and precision from around 4% to almost 51%.

The three classifiers that had the best hit rates also had the worst precisions. It can be understood simply looking at the False Positive amount. The classifiers found most of the runways but recognized a

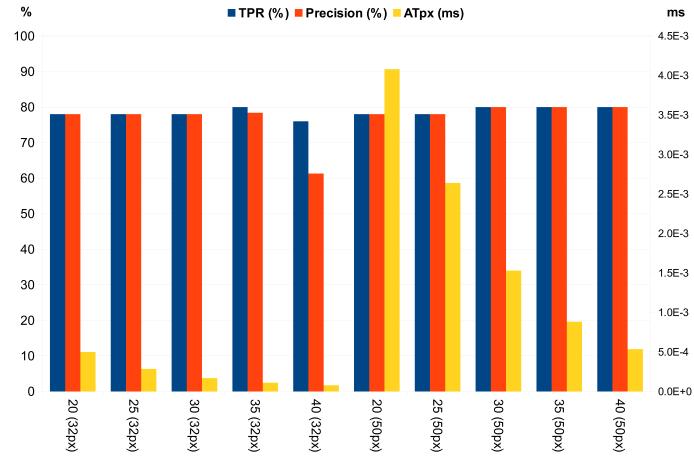


Figure 14: LBP cascade results.

Table 3: HOG+SVM configuration.

Win.	Dim.	Kernel	γ	r	C	SV
32 px	324	Linear	-	-	0,1	1
		RBF	0,50625	-	62,5	5880
		Sig.	$1 \cdot 10^{-5}$	0,1	0,1	4042
64 px	1764	Linear	-	-	0,5	3072
		RBF	$3,375 \cdot 10^{-2}$	-	12,5	3315
		Sig.	$1 \cdot 10^{-5}$	0,1	0,1	4042
128 px	8100	Linear	-	-	0,1	1823
		RBF	$3,375 \cdot 10^{-2}$	-	2,5	3356
		Sig.	$1 \cdot 10^{-5}$	0,1	0,1	3752

Table 4: Classification with HOG+SVM.

Win.	Kernel	TP	TPR	FP	FN	Prec.	ATpx (ms)
32 px	Linear	47	94%	961	3	4,66%	$273,5 \cdot 10^{-4}$
	RBF	45	90%	992	5	4,34%	$845,1 \cdot 10^{-4}$
	Sigmoidal	46	92%	923	4	4,75%	$661,1 \cdot 10^{-4}$
64 px	Linear	23	46%	27	27	46%	$625,5 \cdot 10^{-4}$
	RBF	12	24%	41	38	22,64%	$3463 \cdot 10^{-4}$
	Sigmoidal	23	46%	27	27	46%	$3081,6 \cdot 10^{-4}$
128 px	Linear	27	54%	26	23	50,94%	$792,5 \cdot 10^{-4}$
	RBF	24	48%	95	26	20,17%	$15523,6 \cdot 10^{-4}$
	Sigmoidal	23	46%	27	27	46%	$10913,7 \cdot 10^{-4}$

lot of areas without runway as having one. This kind of classifier is useless, because it will be necessary to employ another method to separate the True Positives of the huge amount of False Positives in order to have good performance.

The linear kernel was much faster than RBF and sigmoidal. It can be explained by the fact that linear kernel require less computation than the other ones.

Figure 16 shows the best classifiers according to hit rate, precision, and classification speed. The performance disparity between the cascade approaches and HOG+SVM can be clearly seen in this figure.

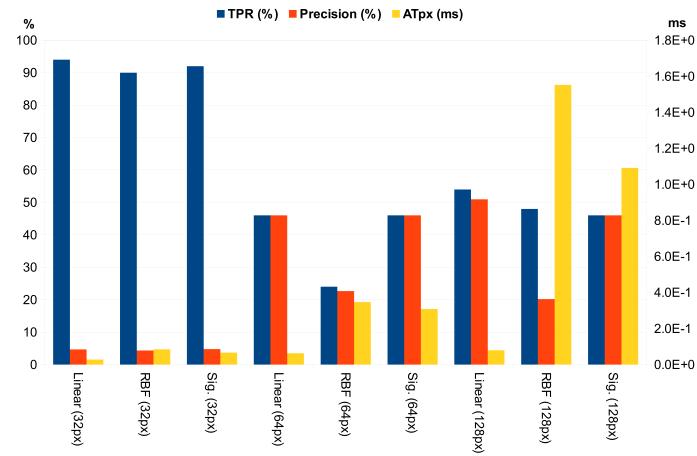


Figure 15: HOG+SVM results.

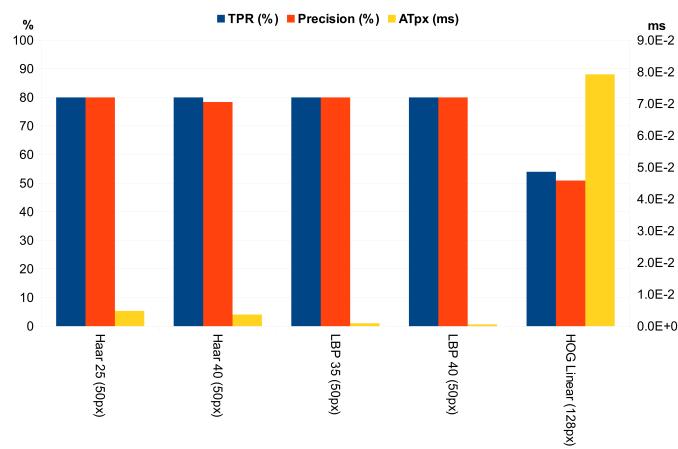


Figure 16: Classifiers with the best performance.

4. Conclusion

Three popular approaches were employed in this study in order to detect asphalt and concrete runways in high resolution satellite images. They had already shown a high performance level with human feature recognition in previous studies and this study confirms their applicability in concrete and asphalt runway detection.

An interesting point to note is that the watermarks present in the high resolution training images had little or no impact on the classifiers performance, due mainly to the classifiers internal architecture that discards features that are not relevant.

The results were very satisfactory, mainly due to cascaded classifier approaches, where the best classifiers were reasonably fast and obtained 80% in hit rate and precision. The best case, shown in Figure 16, processed an image of 1000x1000 pixels in about 535 ms. The result is quite similar to that of Gaszczaka et al. [8], which had a hit rate around 70% for the people profile and around 80% for vehicles. These runway results, however were inferior to those reported for face detection [21, 5, 9] with their 90% to 94% hit rate. These studies on face detection had a much larger training data set than the one used in this paper. So, future studies might increase training data set size trying to have as much as possible unique samples, which could increase the classifier performance.

The HOG+SVM approach had disappointing results, with a hit rate of 51% and accuracy of about 51% for its best classifier. The classification speed was generally much lower than that obtained in the other

approaches. In Table 4, it is possible to find classifiers with a hit rate of 90% or more. However, the accuracy is very low, less than 5%, due to high false positive rates.

As for processing speed performance in cascade classifiers, it can be clearly seen that the more stages there are, the faster it is. And also, the smaller its descriptor is, the faster it gets. To a certain degree this is also true for HOG+SVM, because the smaller the detection window is, the faster the classification speed.

Previous studies about runway detection did not make their own image dataset available. Consequently, it was not possible to compare classification accuracy. Most of them also did not report the processing speed performance of their approach. The only one [28] that stated it, processed a 256x256 pixel image in 27 ms, that means, $ATpx=4.12 \cdot 10^{-4}$ ms. The approaches using the cascade concept and using detection window of 32 pixels showed a much better performance. Future studies could apply Canny with Hough transform on the training and test data set created for this paper and compare the results.

References

- [1] E. Corvee and F. Bremond Haar like and LBP based features for face, head and people detection in video sequences, International Workshop on Behaviour Analysis and Video Understanding, 10, 2011.
- [2] Q. Chen, N. D. Georganas, and E. M. Petriu, Real-time vision-based hand gesture recognition using haar-like features, IEEE Instrumentation and Measurement Technology Conference, 1-6, 2007.
doi:10.1109/IMTC.2007.379068
- [3] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles, Springer, 2010.
- [4] R. Rodrigues, H. Shiguemori, C. Forster, and S. Pellegrino, Color and texture features for landmarks recognition on UAV navigation, Simpósio Brasileiro de Sensoriamento Remoto, 2009.
- [5] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 511-518, 2001.
doi:10.1109/CVPR.2001.990517
- [6] T. Ahonen, A. Hadid, and M. Pietikäinen, Face recognition with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12):2037-2041, 2004.
doi:10.1109/TPAMI.2006.244
- [7] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 886-893, 2005.
doi: 10.1109/CVPR.2005.177
- [8] A. Gaszczaka, T. Breckon, and J. Hana, Real-time people and vehicle detection from UAV imagery, SPIE Conference Intelligent Robots and Computer Vision, 2011.
doi:10.1117/12.876663
- [9] P. Viola and M. Jones, Robust real-time face detection, International Journal of Computer Vision, 57:137-154, 2004.
doi:10.1023/B:VISI.0000013087.49260.fb
- [10] T. Grittì, C. Shan, V. Jeanne, and R. Braspenning, Local features based facial expression recognition with face registration errors, IEEE International Conference on Automatic Face and Gesture Recognition, 1-8, 2008.
doi:10.1109/AFGR.2008.4813379
- [11] L. Zhang, B. Wu, and R. Nevatia, Pedestrian detection in infrared images based on local shape features, IEEE Conference on Computer Vision and Pattern Recognition, 1-8, 2007.
doi:10.1109/CVPR.2007.383452

- [12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, Object detection with discriminatively trained part-based models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627-1645, 2010.
doi:10.1109/TPAMI.2009.167
- [13] O. Oreifej, R. Mehran, and M. Shah, Human identity recognition in aerial images, *IEEE Conference on Computer Vision and Pattern Recognition*, 709-716, 2010.
doi:10.1109/CVPR.2010.5540147
- [14] R. Xiao, M. J. Li, and H. J. Zhang, Robust multipose face detection in images, *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):31-41, 2004.
doi:10.1109/TCSVT.2003.818351
- [15] P. I. Wilson and J. Fernandez, Facial feature detection using haar classifiers, *Journal of Computing Sciences in Colleges*, 21(4):127-133, 2006.
- [16] I. Paliy, Face detection using haar-like features cascade and convolutional neural network, *International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, 374-377, 2008.
- [17] N. Jiang, Y. Lu, S. Tang, and S. Goto, Rapid face detection using a multi-mode cascade and separate haar feature, *International Symposium on Intelligent Signal Processing and Communication Systems*, 1-4, 2010.
doi:10.1109/ISPACS.2010.5704623
- [18] T. Breckon, S. Barnes, M. Eichner, and K. Wahren, Autonomous real-time vehicle detection from a medium-level UAV, *International Conference on Unmanned Air Vehicle Systems*, 291-299, 2009.
- [19] C. Ma, T. Tan, and Q. Yang, Cascade boosting LBP feature based classifiers for face recognition, *International Conference on Intelligent System and Knowledge Engineering*, 1:1100-1104, 2008.
doi:10.1109/ISKE.2008.4731094
- [20] T. Ojala, M. Pietikainen, and T. Maenpaa, 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.
doi:10.1109/TPAMI.2002.1017623
- [21] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, Face detection based on multi-block LBP representation, in *Advances in Biometrics*, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 4642:11-18, 2007.
doi:10.1007/978-3-540-74549-5_2
- [22] Y. Fang and Z. Wang, Improving LBP features for gender classification, *International Conference on Wavelet Analysis and Pattern Recognition*, 1:373-377, 2008.
doi:10.1109/ICWAPR.2008.4635807
- [23] D. Xia, H. Sun, and Z. Shen, Real-time infrared pedestrian detection based on multi-block LBP, *International Conference on Computer Application and System Modeling*, 12:139-142, 2010.
doi:10.1109/ICCASM.2010.5622128
- [24] Y. Ma, X. Chen, and G. Chen, Pedestrian detection and tracking using HOG and oriented-LBP features, in *Network and Parallel Computing*, ser. IFIP International Conference, Springer Berlin Heidelberg, 6985:176-184, 2011.

- [25] X. Wang, H. Gong, H. Zhang, B. Li, and Z. Zhuang, Palmpoint identification using boosting local binary pattern, International Conference on Pattern Recognition, 3:503-506, 2006.
doi:10.1109/ICPR.2006.912
- [26] J. Trefny and J. Matas, Extended set of local binary patterns for rapid object detection, Computer Vision Winter Workshop, 10:37-43, 2010.
- [27] D. Liu, L. He, and L. Carin, Airport detection in large aerial optical imagery, IEEE International Conference on Acoustics, Speech, and Signal Processing, 5:761-764, 2004.
doi:10.1109/ICASSP.2004.1327222
- [28] N. Di, M. Zhu, and Y. Wang, Real time method for airport runway detection in aerial images, International Conference on Audio, Language and Image Processing, 563-567, 2008.
doi:10.1109/ICALIP.2008.4590015
- [29] X. Wang, B. Li, and Q. Geng, Runway Detection and Tracking for Unmanned Aerial Vehicle Based on an Improved Canny Edge Detection Algorithm, International Conference on Intelligent Human-Machine Systems and Cybernetics, 2:149-152, 2012.
doi:10.1109/IHMSC.2012.132
- [30] J. E. C. Cruz, E. H. Shiguemori, and L. N. F. Guimarães, Concrete and asphalt runway detection in high resolution images using LBP cascade classifier, BRICS Countries Congress on Computational Intelligence, 465-469, 2013.
doi:10.1109/BRICS-CCI-CBIC.2013.83
- [31] T. Ojala, M. Pietikainen, and D. Harwood, Performance evaluation of texture measures with classification based on kullback discrimination of distributions, International Conference on Pattern Recognition, 1:582-585, 1994.
doi:10.1109/ICPR.1994.576366
- [32] L. Wang and D. He, Texture classification using texture spectrum, Pattern Recognition, 23(8)905-910, 1990.
doi:10.1016/0031-3203(90)90135-8
- [33] C. Chen, Handbook of pattern recognition and computer vision, World Scientific Publishing Company, Incorporated, 2009.
- [34] Google Inc., Google Maps, <http://maps.google.com>, (Acessed in June 2013).
- [35] Y. Freund and R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in *Computational Learning Theory*, ser. Lecture Notes in Artificial Intelligence, Springer, 904:23-37, 1995.
doi:10.1007/3-540-59119-2_166
- [36] D. Forsyth and J. Ponce, Computer vision, Prentice Hall, 2011.
- [37] A. P. Witkin, Scale-space filtering: A new approach to multi-scale description, IEEE International Conference on Acoustics, Speech, and Signal Processing, 9:150-153, 1984.
doi:10.1109/ICASSP.1984.1172729
- [38] A. Haar, Zur theorie der orthogonalen funktionensysteme, Mathematische Annalen, 69(3):331-371, 1910.
doi:10.1007/BF01456326
- [39] C. Papageorgiou, M. Oren, and T. Poggio, A general framework for object detection, International Conference on Computer Vision, 555-562, 1998.
doi:10.1109/ICCV.1998.710772

- [40] R. Lienhart, A. Kuranov, and V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, *Pattern Recognition*, 297-304, 2003.
doi:10.1007/978-3-540-45243-0_39
- [41] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60(2):91-110, 2004.
doi:10.1023/B:VISI.0000029664.99615.94
- [42] B. E. Boser, I. M. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, *Annual Workshop on Computational Learning Theory*, 144-152, 1992.
doi:10.1145/130385.130401
- [43] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th Edition, Elsevier Science, 2008.
- [44] C. W. Hsu, C. C. Chang, and C. J. Lin., *A Practical Guide to Support Vector Classification*, 2010.
- [45] T. Fawcett, An introduction to ROC analysis, in *Pattern Recognition Letters*, special issue: ROC analysis in pattern recognition, 27(8):861-874, 2006.
doi:10.1016/j.patrec.2005.10.010