

Fast and efficient face recognition system using Random Forest and Histograms of Oriented Gradients

Abdel Ilah Salhi¹, Mustapha Kardouchi¹ and Nabil Belacel²

¹Computer Science Department
University of Moncton
NB, Canada, E1A 3E9

²Institute for Information Technology
National Research Council of Canada
NB, Moncton, E1A 7R1

abdelilah.salhi@umoncton.ca
mustapha.kardouchi@umoncton.ca
nabil.belacel@nrc-cnrc.gc.ca

Abstract: The efficient face recognition systems are those which are able to achieve higher recognition rate with lower computational cost. To develop such systems both feature representation and classification method should be accurate and less time consuming. Aiming to satisfy these criteria we coupled the HOG descriptor (Histograms of Oriented Gradients) with the Random Forest classifier (RF). Although rarely used in face recognition, HOG have proven to be a power descriptor in this task with a lower computational time. As regards classification method, recent works have shown that apart from their accuracy when compared with its competitors, Random Forest exhibits a low computational time in both training and testing phase. Experimental results on ORL database have demonstrated the efficiency of this combination.

1. Introduction

Biometric systems are increasingly developing in recent years due to their requirement in many areas of commerce, airports and professional companies. As one of the famous biometric systems face recognition has been an extremely active axe of research in both academic and industrial communities over the past decade. A face recognition procedure consists of 2 tasks: feature extraction and classifier designing. Both have a relevant influence on the reliability of recognition method. As regards feature extraction, various approaches have been proposed to deal with this task. These approaches are roughly classified into holistic approaches which extract information from the whole face image, and local approaches which extract information from local facial features [Kw06] [Af07] .

As one of the most successful local approaches, Local Binary Pattern (LBP) has been used within the face recognition task over the last years. LBP [AHP04] computes a histogram with taking into account each pixel in the image and considering the values of its neighborhoods. However LBP suffers from its sensitivity toward noise and its variance to rotation. Another wide and powerful local face descriptor: Gabor features [LW02] [GTG09] which is computed after convolving face image with a family of Gabor kernels at different scales and orientations. Despite the success of this technique the great number of kernels to be applied to the image leads to high feature dimensions and makes it thus high computational and unusable in real-time applications.

In this paper, we used Histograms of Oriented Gradients descriptor [CXC11] [DBS11]. HOG was used in different areas of image processing as human detection [Py11] and hand gesture recognition [LC99]. However, according to our bibliographic researches

very few publications apply this descriptor to face recognition while it achieves comparable results with other powerful descriptors, even better than some of them. Furthermore, HOG have demonstrated to be much lower complexity in term of computing time when compared to its competitors, which allowed it to be used in many areas of image processing as real time applications.

The second key aspect of face recognition method is the performance of the classifier. Many techniques have been developed in this topic based on statistic and artificial intelligence. In this paper we used the Random Forest classifier, which became a wide technique for classification in recent years. Random Forest [B101] is an ensemble of decision trees [PM03], where each tree gives a classification and the final result is the majority vote given by all the trees. Compared to the most popular classifiers, RF has demonstrated to be better or at least provides similar results as some of them in term of accuracy in several task, as well as face recognition task [VGB11]. In fact, Breiman demonstrates with 20 datasets from different domains [B101] that random forest provides highest or same accuracy than the other ensemble methods as Bagging [B196] and Boosting [FS96]. Also, random forest proved to give comparable results to SVM [BZM07],and better than Neural Networks [VGB11]. In our experiments we get the same conclusion in the face recognition task when coupling HOG descriptor with these classifiers cited above.

Except accuracy, random forest demonstrated to be among the fastest states-of-the-art classifiers in both learning and classification phase. In fact [KMM09] implemented this classifier in a real time tracking system where both learning and testing runs on real time and have thus to be very fast. In [PM03] the authors demonstrates that due to speedup of the building of a decision tree,even building thousands of them is cheaper in computing than training one artificial neural network. As well [BZM07] demonstrates that random forest runs faster than SVM in both training and testing phase.

The following section of the paper explains the HOG descriptor in details. In section 3, we describe then random forest classifier. Experimental results are presented in section 4 and conclusions are given in the final section.

2. Histograms of Oriented Gradients

The HOG descriptor is a local statistic of the orientations of the image gradients. It is characterized by its invariance to rotation and illumination changes. Add to that, the simplicity of its computation technique makes it among the faster object's descriptor in state-of-the-art. The main idea behind this descriptor is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions. In its simplest form, the HOG feature divides the image into many cells, in each of them a histogram counts the occurrences of pixels orientations given by their gradients.The final HOG descriptor is then built with combination of these histograms.

In practice, four major steps are involved:

- Image derivative Computing.
- Magnitude and Gradient Orientation computing.
- Partial histograms building.
- Normalization of partial histograms.

2.1 Computing the derivative in the horizontal and vertical directions

This step is performed by convolving the input image with a filter mask such as the 1-D centered kernel $[-1 \ 0 \ 1]$, the 2×2 Robert diagonal masks or the Sobel filter. We used in this work the simple 1-D centered kernel filter which performed well, besides it's the fastest among these cited filters. The kernels are applied separately in each pixel of the input image to produce separate measurements of the gradient components in the vertical and horizontal orientations. We call the output images G_x and G_y .

2.2 Computing magnitude and orientation of the gradient

After dividing the image into N cells, the next step of HOG feature is to compute the magnitude $|\nabla I(x, y)|$ and the orientation $\theta(x, y)$ of the gradient. In such way that each pixel is represented by a gradient vector which consists of magnitude and direction.

The magnitude is given by:

$$|\nabla I(x, y)| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

While gradient is given by:

$$\theta = \varphi\left(\frac{G_y}{G_x}\right) \quad (2)$$

where φ is a function that returns the direction of the vector $[G_x, G_y]$ in the range $[-\pi, \pi]$ by taking into account $\arctan\left(\frac{G_y}{G_x}\right)$ and the signs of G_x and G_y .

2.3 Generating the histogram from orientations and magnitudes

In this step, the gradient angles in each cell are quantized into a number of bins B of regularly spaced orientations and the magnitudes for identical orientations are accumulated into a histogram. In other words, for each pixel with coordinates (x, y) we determine which of the B orientations is the closest to its orientation $\theta(x, y)$ and we add then its magnitude $|\nabla I(x, y)|$ to the corresponding bin. The number of bins B used indicates the length of the histogram vector for each cell. It influences the accuracy of the histogram and more is larger the number of bins, the more detailed the histogram is. In this work, we used 12 bins.

2.4 Normalization

In order to make the histogram invariant to illumination and contrast, a local normalization step in each cell is performed after calculating the histogram vector. For this purpose a simple Euclidian norm is applied:

$$V_n = \frac{V}{\sqrt{\|V\|^2 + \epsilon^2}} \quad (3)$$

Where V is the vector to be normalized and ϵ is a small positive value needed when evaluating empty gradients.

Finally, the facial image feature is obtained by concatenating histograms of each cell. Fig. 1 shows the whole process of the HOG computing.

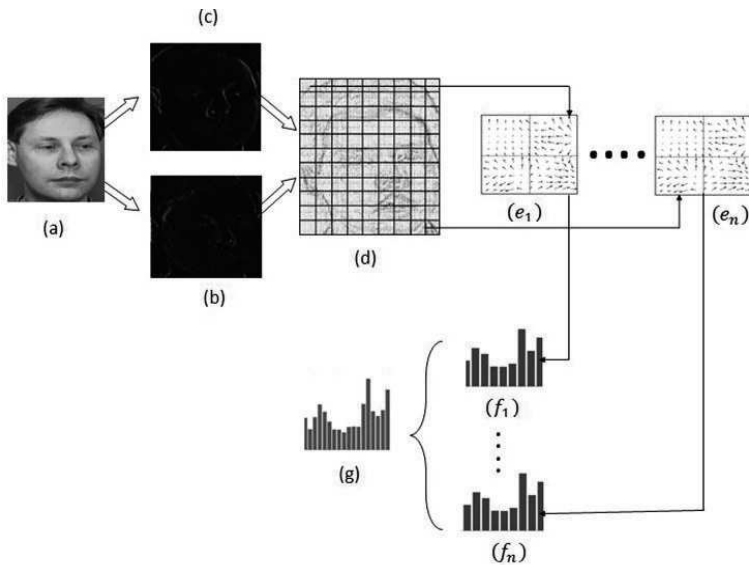


Figure 1. Main steps of HOG computing : (a) Original image, (b) result of convolution of face with horizontal kernel, (c) result of convolution of face with vertical kernel, (d) Gradient orientation of the whole face, (e₁)... (e_n) gradient vectors in each cell, (f₁)... (f_n) histograms corresponding to cells computed based on gradient and magnitude, (g) final HOG computing.

3. Random Forest

3.1 Basic theory

Once a feature vector was extracted from the face, the next step is to use this vector in classification. Several methods have been proposed in this topic. Since few years *ensemble-learning algorithms* are receiving more on more interest in the field of classification methods. Ensemble methods are learning algorithms that construct a set of many individual classifiers called weak learners to form a unique classification system. Random Forest belongs to this ensemble method category; they correspond on combination of decision tree-type classifier, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. RF can be seen as combination of two types of ensemble-method, boosting and bagging [VGB11] [P08]. In fact, it is built by randomly sampling a feature subset for each decision tree as boosting, and by randomly sampling a training data subset for each decision tree as in Bagging. Random forest represents many advantages, we cite below the main among them:

- RF performed efficiently with problems with high or low dimensions.
- Trains rapidly even with thousands of input variables.
- It gives an estimation of discriminate variables for the classification.
- It allows adding new data without re-training.
- It is computationally lighter than many other classification methods.

3.1.1 Training

During the training step, the random forest algorithm generates multiple decisions trees, each one trained on a subset of bootstrapped data from the original data, as well only a subset of randomly selected variables will be used to determinate a split at each node. In the original random forest [B101], the authors used the *GINI index* [P08] to perform this split, which is a very fast and efficient heuristic method for this purpose. In fact, all the randomly selected variables are evaluated with the GINI index criterion and the one which have the most increased value is used to split this node.

The *GINI-INDEX* formula is given by the following expression:

$$Gini(A_i) = - \sum_{i=1}^c p(y_i)^2 + \sum_{j=1}^{m_i} p(v_{i,j}) \sum_{i=1}^c p(y_i/v_{i,j})^2 \quad (4)$$

where $A_i(v_1, v_2, \dots, v_{m_i})$ is the vector on which we compute the GINI index, y_i represents its class. $p(v_{i,j})$ represents the probability that attribute A_i has value V_k . While $p(y_j)$ represents the probability of the class y_j , and $p(y_j/v_{i,j})$ is the probability that the class y_j has the attribute A_i with value v_k .

The procedure to build a random forest with n trees can be thus summarized as follow:

```
repeat n time
  select bootstrap sample from the training data
  select p variables at random in this bootstrap
  repeat
    compute the Highest GINI index h of the samples on
    these p variables
    split on h
  until h < threshold
end (repeat)
```

3.1.2 Testing

At testing, each tree in the forest votes for the appropriate class of the input data. The output of the final classifier is determined by the most frequent value generated by all the trees. The participation of several trees to predict an input data class contributes on accuracy of random forest. In fact, only one decision tree is accurate. In addition, even if this tree gives an error prediction, it's difficult that all the trees do the same error for the same input data.

3.2 Performance of random forest

Recently, researchers demonstrated when performing mining data with random forest [VGB11], that RF is among the best states-of-the-art classification methods in many object classification fields, among others in face recognition. We get the same conclusion in our experiments when combined the HOG descriptor with random forest.

In fact, we demonstrate that this former gives good results in term of accuracies when compared to Neural Networks and SVM. In addition, it should be noted that random forest was much faster in both training and classification when compared to these conventional classifiers cited above. In fact these classifiers are usually based on an exhaustive research through all the data, thus when handling with high dimensional data as face recognition in large databases this require a large time for both training and testing. In opposition to RF which doesn't face this problem because it selects only a subset of the data and only a subset of variables each time. In addition the speedup came also from the fact that each tree is grown independently of the other trees grown. Thus, in a parallel environment, as GPU programming in Matlab, if each one of the n unity of computation is given the job of growing k/n trees in a forest with k trees, the unities should not communicate with each other until the end of trees growing when the results are aggregated.

Another aspect which characterizes RF is its simplicity due to its non-parametric nature; RF can be considered to be black box type classifier since only one control parameter is needed to experiment with, which is the number of trees and all other the parameters of split rules for classification are unknown.

4. Experiments

In this section, two experiments will be described. The first one evaluates the HOG descriptor; we tested different variations of HOG and we compared it to others states-of-the-art descriptors. In the second experiment we evaluate the robustness and speedup of the random forest classifier.

The experimental results were evaluated using ORL face database. It contains 40 distinct subjects; each one contains 10 different images which were taken at different times, with taking into account light variations and poses changes. In addition, images are taken with different facial expressions as (open/closed eyes, smiling/non-smiling) and some facial details as (glasses/no-glasses).

A sample of ORL database is shown in Fig.2 which illustrates the different variations cited above. It should be noted that all the experiments were carried out with the same computer performances which is Intel Core 2 duo T6600 2.20 Ghz with 3 GB RAM.

4.1 HOG experiments

4.1.1 Overlapping

In this work, we overlap the blocks on which we computed the histograms. When comparing the results of using HOG descriptor with and without overlapping, it can be seen from Table.1 that overlapping blocks leads to better results. In fact the aim of such technique is to make the majority of the pixels contribute several times in the final descriptor vector. The idea behind this is that redundant information makes the feature vector stronger, thus the performance of the face recognition increases.

Table 1.
Effect of overlapping on recognition rate (%).

HOG with overlapping	HOG without overlapping
95.1	94.5
92.8	90.9
87.5	85.6
86.2	84.6

4.1.2 Angle orientation

One has a choice of defining if the gradient is signed or unsigned. For human detection [SRB06] [Py11] the authors didn't take into account the sign of the gradient i.e a double angle representation was used. This leads to a better performance in their case. However, they said that including sign information does help substantially in some other object recognition tasks. It was the case in the face recognition task as demonstrated in Table 2. This is justified by the fact that the direction of the contrast has no importance. Thus the results with a white object placed on a black background or the reverse are the same.

Table 2.
Effect of angle orientation on recognition rate (%).

Double angle representation	Simple angle representation
95.1	93.4
93.3	91.7
89.5	87.6
87.3	86.1

4.1.3 Comparison with other features

We compare in this experiment the recognition rate of HOG descriptor with those of Gabor and LBP descriptors. For this purpose we used HOG with overlapped cells and signed gradients. Concerning LBP we used the uniform one which takes into account the values of its eight nearest neighbors. As regards Gabor descriptor it used 40 kernels corresponding to five scales. The Random forest was used for classification in this experiment with all descriptors. Fig.3 shows that HOG outperformed LBP and was comparable to Gabor filter in terms of recognition rate. It has to be noted that we considered in this figure the highest recognition rates.

As regarding time computation it's clear from Table. 3 that HOG descriptor has the smallest time computation, in fact we can see from this table that it's 15 times faster than Gabor descriptor and 5 times faster than LBP.

Table 3.
Time computation consisting to the different descriptors.

Method	HOG	LBP	GABOR
Time(ms)	0.0071	0.0235	0.1069



Figure 2. Sample of images in ORL face database

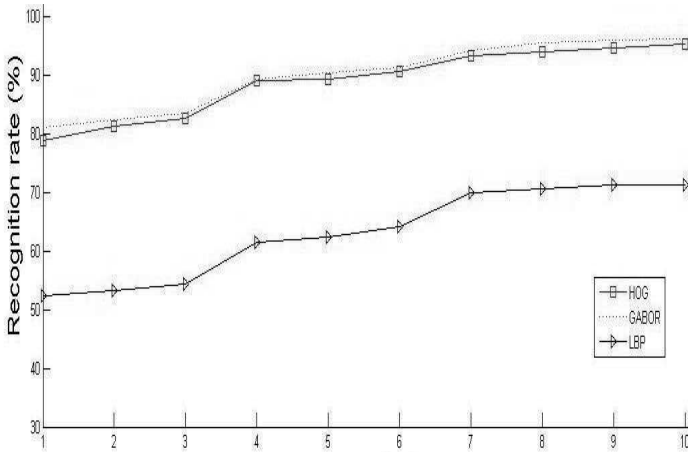


Fig .3: Recognition rate of HOG, Gabor and LBP

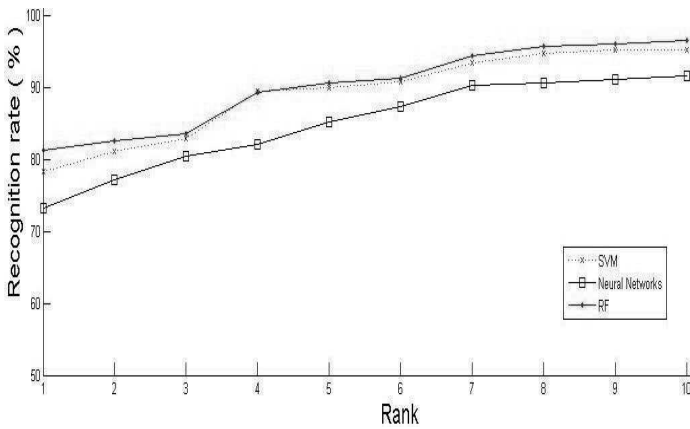


Fig .4: Recognition rate of SVM, NN and RF

4.2 Random forest experiments

4.2.1 Number of trees

As mentioned above, when building the random forest classifier, we need only to specify the number of trees to specify. This number affects the accuracy of results given by the system when predicting a new face. Table .4 shows that increasing the number of trees n provides best recognition rate. However when reaching certain number of trees, which depends essentially on the dimension of the dataset and correspond to 80 in our case, the recognition rate will no longer increase.

Table 4.
Recognition rate according number of trees used to build the random forest.

Number of trees	20	40	60	80	100
Recognition rate (%)	62.5	75.1	78.8	95.1	95.1

4.2.2 Comparison with other classifiers

To evaluate the efficiency of the random forest, we compared it to Neural Network and Support Vector Machine classifiers. The architecture of the Neural Network used is the Multilayered Feed-Forward Network architecture with 20 input nodes, 10 hidden nodes and 10 output nodes, while the SVM used was with Multilayer Perceptron kernel. As regarding the Random Forest, 80 trees were used to build the Forest. When comparing the results of using a HOG features with Random Forest against the results of using SVM and Neural Network with this same feature we can see from Figure .4 that random forest performs much better than Neural Network and achieves almost the same performance as SVM. Only that RF was the cheaper between them in terms of computation time as demonstrated in Table .5.

Table 5.
Time spent to train the classifier with 320 images (112x91). The table does not include time spent to compute the image descriptor.

Classifier	RF	SVM	NEURAL NETWORK
Time(s)	2.875	4.984	8.685

4.3 Whole system results

In this section we explore the different experiments tested above with changing every time the descriptor or the classifier in the aim to show the accuracy of using Random Forest classifier with HOG descriptor. Results in Table .6 show that the recognition rate of our method is superior to the other methods, except Gabor descriptor when combined with Random Forest where results were comparable. Only that our method is extremely cheaper in terms of computing time. From the third column of Table .6 which shows the computation time of the different tested methods we can see the superiority of our method. Note that the time computation was the time spent to compute the feature vector

of 320 face images with size 112x92, add to time spent to train the classifier with these 320 vectors.

Table 6.

Recognition rate and computation cost of different methods. Computation cost includes cost of feature extraction of all the 320 images and time spent to train the classifier.

Method	Recognition rate (%)	Time computation (s)
HOG+RF	95.1	5.376
GABOR+ RF	95.7	39.345
HOG+SVM	94.4	10.368
LBP+RF	67.6	9.173
HOG+NN	92.2	14.368
GABOR+SVM	94.5	47.289

5. Conclusion

A face recognition system based on Random Forest and Histograms of Oriented Gradients was presented in this paper. This system attains very good results in term of accuracy when compared with existing systems. Furthermore this system is characterized by its lower complexity in term of time computing due to the speedup of both face descriptor (HOG) and the classifier (RF). However, it should be noted that, as regarding random forest that it is true that augmentation of tree's number makes the forest more accuracy, but at the same time it makes it slower. Considering the HOG descriptor authors think that introducing fuzzy logic can improve this descriptor. Therefore, our next step consists to improve the HOG descriptor in order to make it more accurate without affecting the computational time.

Acknowledgements

We gratefully acknowledge the support from NSERC's Discovery Award (RGPIN293261-05) granted to Dr. Nabil Belacel. Authors would like also to thank University of Cambridge for providing the ORL face database.

References

- [Kw06] Kevin W. Bowyer, Kyong Chang, Patrick Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition, *Computer Vision and Image Understanding*, Volume 101, Issue 1, 2006, pp. 1-15.
- [Af07] F. Abate, M. Nappi, D. Riccio, and G. Sabatino, 2D and 3D face recognition: A survey, *Pattern Recognition Letters*, vol. 28, no. 14, 2007, pp. 1885-1906.
- [KMM09] Z. Kalal, J. Matas, and K. Mikolajczyk, Online learning of robust object detectors during unstable tracking, 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 2009, pp. 1417-1424.
- [GTG09] V. Ghosal, P. Tikmani, and P. Gupta, Face Classification Using Gabor Wavelets and Random Forest, *Canadian Conference on Computer and Robot Vision*, 2009, pp. 68-73.

- [LW02] C. Liu and H. Wechsler, Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 11, no. 4, 2002, pp. 467-76.
- [AHP04] T. Ahonen, A. Hadid, and M. Pietikäinen, Face Recognition with Local Binary Patterns, *Proc. Eighth European Conf. Computer Vision*, 2004, pp. 469-481.
- [DT05] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 886-893 .
- [LC99] Lee, H.J. and Chung, J.-H. Hand gesture recognition using orientation histogram. In *Proc Of IEEE Region 10 Conf. Vol. 2*. 1999 .pp .1355–1358.
- [SRB06] F. Suard, A. Rakotomamonjy, and A. Bensrhair, Pedestrian, Detection using Infrared images and Histograms of Oriented Gradients, *Symposium A Quarterly Journal In Modern Foreign Literatures*, 2006, pp. 206-212.
- [CXC11] S. H. U. Chang, D. Xiaoqing, and F. Chi, Histogram of the Oriented Gradient for Face Recognition , *Tsinghua Science and Technology*, vol. 16, 2011, no. 2, pp. 216-224.
- [Py11] Y. Pang, Y. Yuan, X. Li, and J. Pan, Efficient HOG Human detection, *Signal Processing*, vol. 91, 2011 , pp. 773-781.
- [VDC12] N. sonVu, H. M. Dee, and A. Caplier, Face recognition using the POEM descriptor, *Pattern Recognition*, vol. 45, no. 7, 2012, pp. 2478-2488.
- [BI01] L. Breiman. Random forest. *Machine Learning*, vol .45 , 2001 .pp. 1-33.
- [PM03] M. Pal and P. M. Mather, An assessment of the effectiveness of decision tree methods for land cover classification, *Remote Sensing of Environment*, vol. 86, no. 4, 2003, pp. 554-565.
- [GBS06] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, Random Forest for land cover classification, *Pattern Recognition Letters*, 2006 ,vol. 27, no. 4, pp. 294-300.
- [BI96] L. Breiman. Bagging predictors. *Machine Learning*, vol24 ,123–140, 1996.
- [CHD01] Chan, J. C. W., Huang, C., DeFries, R. S. Enhanced algorithm performance for land cover classification from remotely sensed data using bagging and boosting, *IEEE Transactions of Geoscience and Remote Sensing*, 2001, 39, pp. 693–695
- [FS96] Freund, Y, Schapire, R. E. Experiments with a new boosting algorithm, *Machine Learning: Proceedings of the Thirteenth International Conference* ,1996, pp. 48–156.
- [GBS06] Gislason, P. O, Benediktsson, J. A, and Sveinsson, J. R. Random Forest for land cover classification, *Pattern Recognition Letters*, 2006,27, 294–300.
- [CP12] J. C.-wai Chan and D. Paelinckx, Remote Sensing of Environment Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery, *Remote Sensing of Environment*, vol. 112, 2012, pp. 2999-3011.
- [Pi08] A. I. N. Press, Implementation of the Random Forest method for the Imaging Atmospheric Cherenkov Telescope MAGIC, *Nuclear Instruments and Methods in Physics Research*, vol. 588,2008, pp. 424-432.
- [BZM07] A. Bosch, A. Zisserman, and X. Mu, Image Classification using Random Forest and Ferns, In *ICCV*, 2007.
- [VGB11] A. Verikas, A. Gelzinis, and M. Bacauskiene, Mining data with random forest : A survey and results of new tests, *Pattern Recognition*, vol. 44, no. 2, 211, pp. 330-349.
- [DBS11] O. Déniz, G. Bueno, J. Salido, F, Face recognition using Histograms of Oriented Gradients Pattern, *Recognition Letters*, vol. 32, Issue12, 2011, pp. 1598-1603.