ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Human Detection and Classification of Landing Sites for Search and Rescue Drones

Felipe N. Martins[1], Marc de Groot[2], Xeryus Stokkel[2] and Marco A. Wiering [2] *

1- Federal Institute of Educ., Science and Tech. of Espirito Santo - Serra Campus
Rod. ES-010, km 6,5 - Manguinhos, Serra, ES. 29173-087. Brazil.

2- University of Groningen - Artificial Intelligence and Cognitive Engineering
PO Box 407, 9700 AK, Groningen. Netherlands.

Abstract. Search and rescue is often time and labour intensive. We present a system to be used in drones to make search and rescue operations more effective. The system uses a drone downward facing camera to detect people and to evaluate potential sites as being safe or not for the drone to land. Histogram of Oriented Gradients (HOG) features are extracted and a Support Vector Machine (SVM) is used as classifier. Our results show good performance on classifying frames as containing people (Sensitivity $> 78\%$, Specificity $> 83\%$), and distinguishing between safe and dangerous landing sites (Sensitivity $> 87\%$, Specificity $> 98\%$).

1 Introduction

Finding missing persons is a time consuming and labour intensive task. A swarm of autonomous drones (or Unmanned Aerial Vehicles - UAVs) could potentially increase the chances of finding missing persons while reducing rescue time, thus increasing the overall survival chance. Should those UAVs be able to detect missing persons autonomously, the location of interest could be reported to rescue workers that could then concentrate their search efforts on areas where it is likely that they will find people.

In case of extreme weather conditions and to save battery power (which is an important limiting factor), it might be useful to land the drone and resume operations right after weather conditions are favorable again. Dangerous landing may cause serious damage to the drone and may harm people nearby. It can also become very difficult to retrieve the drone and/or data if it is in an unreachable location, such as under water. Therefore, if the drone is to fly autonomously, it is expected that it lands on a safe place. Moreover, even for remotely operated drones, a safe landing site detection system could be used to assist the pilot.

On the use of UAVs to classify terrain, the authors of [1] utilize a Radio Controlled model helicopter equipped with a downwards pointing camera to classify terrain as safe or unsafe. Their methods consist of a Fuzzy Rule Based classifier and a modified version of the Scale-Invariant Feature Transform (SIFT).

*The authors thank the University of Groningen for the support given to this research. Dr. Martins also thanks the Federal Institute of Espirito Santo and CAPES (a foundation of the Brazilian Ministry of Education) for the support that allowed him to work on his research in the Netherlands (grant BEX 1603/14-0).

In [2] the authors show the use a quadcopter to identify 6 different types of terrain. Their methods make use of random forests and a modified version of Speeded Up Robust Features (SURF) they call Terrain-SURF. They reach promising results, up to a sensitivity of 99.6% at their highest resolution (and highest training time). However, they have not focused on suitable landing sites or minimizing false positives. Regarding human detection, a lot of research has been done from an eye-height perspective [3, 4] while others have used an in-flight perspective [5, 6]. The in-flight perspective resembles a bird's-eye-view while the top-down perspective is more akin to satellite images [5].

In this paper we describe the development of a system that is able to autonomously detect humans in images obtained from a drone's downwards facing camera. With such a system, the drone could automatically send to the rescue teams the GPS coordinates where it detected people. Moreover, using the images from the same camera, we propose another system to allow a UAV to distinguish between safe and dangerous landing sites. The method that is used in both systems is similar to the one presented in [3]. We use the Histogram of Oriented Gradients (HOG) as main feature extractor and a Support Vector Machine (SVM) [7] as a classifier. In this paper we deal only with the computer vision aspect of the system.

2 Methodology

We first created our own datasets: one for the human detection system and another for the landing site classification system. To populate the datasets we used a Parrot AR.Drone 2.0 quadrotor, which has a 60 fps vertical QVGA camera pointing downwards. Video streamed from the camera was obtained via Wi-Fi connection and was processed on an external computer. Instead of storing a video feed, we stored still images at 5 frames per second.

The human detection data set contains 1451 images, of which 186 include people. The training set consists of images from the data set that have been labelled as either negative (no person in the image - 1056 images) or as positive (one or more people in the image - 148 images). During training the positive examples are also flipped either horizontally, vertically or both to obtain 592 positive examples. The images weren't tightly cropped, but there are usually 4 pixels surrounding the person. The test set contains 247 images of which just 38 include people.

The landing site classification data includes 4560 images of terrain, taken from heights ranging from little above the ground up to 5 meters in the air. Of the 4560 images, 2110 represent grass, 1642 represent road, 518 represent water and 109 represent bushes. The remaining 181 images consist of combinations between these (e.g. part grass, part water). Images were manually labeled by stating the types of terrain. Safe sites belong to grass and road terrains.

2.1 Human Detection

For training, positive labels were created by cropping people out of images and saving them as positive labels. A first round of training was done by generating HOG descriptors for the positive labels (the labelled images were scaled so that each example image is of the same size). From each negatively labelled image 10 random windows were taken and scaled to the common size to extract HOG descriptors. The SVM was trained on the descriptors extracted from the labeled windows obtained during this step.

After training the HOG-SVM classifier, we use it in detector mode by using a sliding window approach. If the output of the classifier is higher than a threshold than a human is supposed to be detected. To improve the detector, we performed a second round of training. For this we picked 10% of the negative images at random. These images were then scanned using the SVM obtained in the previous step, by examining if the SVM detects people. The descriptor of each window that has been given a positive label was collected. It is guaranteed that these windows are actually negatives since the images of the second round of training only contain negatives, therefore we use these descriptors to correct the SVM on the errors that it made. These descriptors were added to the previously obtained descriptors and the combined set was used to retrain the SVM.

The amount of positive windows were tracked on a per pixel basis creating a 2D histogram as shown in Figure 1. This is converted to a binary image by setting a minimum number of per-pixel positives that should constitute a positive. The positive areas in the binary image are converted into rectangular bounding boxes. If any of the dimensions of a bounding box is below a minimum size then it is discarded. This approach ensures that the occasional false positives do not influence the classification of an image.

Three models with different parameters and preprocessing methods were made so that we could compare their classification performance. The ‘Default’ model was used as a baseline. It does no preprocessing on images and uses color-HOG descriptors. It scales windows down to a common size of 64 × 64 pixels before creating the descriptors. The color-HOG first calculates the slopes in each channel of the RGB image and then selects the channel with the largest slope for each pixel. This means that it flattens the information in the 3 color channels to a single channel, but it does it in a way that doesn’t loose as much information as with a straight conversion to grey scale. It uses a cell size of 8 × 8 pixels, blocks are 2 × 2 cells large, the number of orientation bins in the histograms is 9. The second model is the ‘Grey scale’. Its parameters are the same, but the image is converted to grey scale before the extraction of regular HOG features. The third model uses a common window size of 80 × 80 pixels and a cell size of 8 × 8 pixels, which is about the ideal size to describe appendages [3]. This model is refered to as ‘Large window’. All cells are normalized per block using the L2-norm (Euclidian normalization).

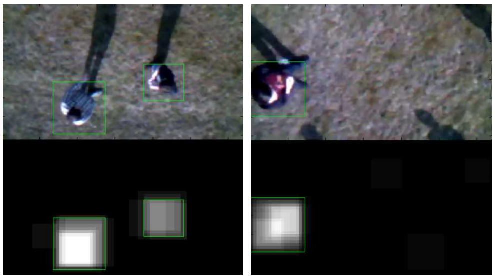

Fig. 1: Images as shown in detector mode. The original image is on top while the 2D histogram is on the bottom. The bounding boxes indicate where the detector has found positives after applying a threshold.

2.2 Classification of Landing Sites

For classifying if the area under the drone is a safe or unsafe landing site, the whole image of the downward pointing camera is used. For extracting HOG features from the dataset for classification of landing sites, 25 non-overlapping blocks of 2 × 2 cells were used, each cell being 72 × 72 pixels in size, and 9 orientation bins (between 0 and 180 degrees). This results in a vector of 900 elements. Cells are normalized per block using the L2-norm.

Adding color as a feature vector might provide an advantage in distinguishing certain types of terrain, and tell safe landing sites from dangerous ones. To incorporate color, a color histogram was made. First, the image is converted from RGB to HSV (Hue, Saturation and Value). The HSV Histogram also has cells of 72 × 72 pixels, blocks of 2 × 2 cells and 9 bins. The hue is divided between the bins, so that every bin represents a spectrum of colors (between $(\frac{\#bin-1}{hue}$ and $\frac{\#bin}{hue}))$. For every pixel in every cell, the hue and saturation are extracted. The value of the saturation (between 0 and 1) is then added to the bin the hue belongs to. Again, every block of cells is normalized using the L2-Norm. The result is a histogram of equal size to the HOG descriptor with similar parameters, represented in a vector of 900 elements.

An example is regarded positive if the drone could land safely on the location represented by the image. If landing may prove dangerous, the example is regarded a negative. For landing safely, an ideal classifier should have a very high specificity (minimizing false positives) and a reasonable sensitivity (and thus retaining a small amount of false negatives). That means the system should avoid labeling a dangerous site as safe, while relaxing on labeling a safe landing site as unsafe. Optimization was done by providing a cost value matrix which states the cost of classifying dangerous areas as safe to be 100, while the cost for classifying false negatives were kept as 1. Several 10-fold cross validations are performed on the dataset. The cross validations vary in the kernel function used (linear, or 2^{nd}, 3^{rd} or 4^{th} order polynomial) and the feature extractors used (HOG, HSV color histogram or both HOG and HSV color histogram).

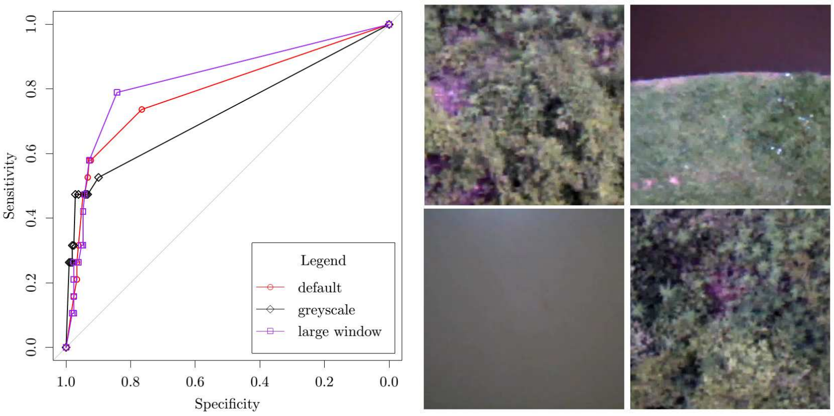

Fig. 2: ROC curves for different models of the human detector (left) and dangerous landing sites misclassified as safe (right).

3 Results and Discussion

3.1 Human Detection

First we show the results of detecting one or more persons in an image or none. Receiver Operating Characteristic (ROC) curves that show the classification performance of the different human detection models can be found in Figure 2 (left). The Grey scale model has the worst performance while the Large window performs the best, with Sensitivity $> 78\%$ and Specificity $> 83\%$. The Area Under the Curve (AUC) of the models are: 0.79 for the Default model, 0.73 for the Grey scale model and 0.83 for the Large window model.

The performance of the models on classifying entire images is quite good. The Grey scale model is slightly faster than the Default model but it does not perform as well, although the difference in performance is not significant. The Large window model has a higher AUC but is not significantly better than the Default model and it is slower than the other models.

An effect of the small amount of data is that the detector does not only detect persons. The models detect high contrast areas which are often relatively wide. One of the common false positives is the heads of shadows cast by people. These false positives do indicate that the detector has not learned to only model persons. Much more data with many other visible objects need to be used in order to improve the people detector in a real-world application.

3.2 Landing Site Classification

The lowest amount of false positives, and the highest true negative rate is reached by a linear SVM trained on HOG-descriptors, without color histogram. It has an average specificity (true negative rate) of 98.7% and a sensitivity of 87.6%. The examples that were misclassified mostly consist of combinations of water and grass. Polynomial SVMs tend to work better with both HOG-descriptors and color histograms. The polynomial SVMs with both HOG-descriptors and color histograms reach high sensitivity (99.6 to 99.8%), while they retain an acceptable specificity (95.2 to 96.0%). In contrary to the linear SVM with HOG-descriptors, the images misclassified by polynomial SVMs do often include purely water and bushes, rather than a transition between grass and water.

Examples of the misclassified images for both linear and polynomial SVMs can be found in Figure 2 (right). The top images are bushes and part water/part grass, and were misclassified by a linear SVM trained on HOG-descriptors only. The bottom images are water and bushes, and were misclassified by the $2nd$, $3rd$ and $4th$ order polynomial SVMs trained on both HOG-descriptors and color histograms.

4 Conclusion

We have presented the development of a computer vision system to be used in drones that assist on search and rescue operations. The system uses HOG features and SVM classifiers to detect people and to evaluate the safety of potential landing sites. Our experiments show that the developed method has good performance on classifying frames as containing people, and a very good performance to correctly identify dangerous landing sites. The developed detectors show great promise for use in search and rescue drones. Future work will focus on improving the run time, implementing the system in an onboard computer and training the system on more data to increase variety and prevent overfitting.

References

[1] A Cesetti, E Frontoni, A Mancini, and P Zingaretti. Autonomous safe landing of a vision guided helicopter. In *Mechatronics and Embedded Systems and Applications (MESA), IEEE/ASME International Conference on*, pages 125–130, 2010.

[2] Yasir Niaz Khan, Andreas Masselli, and Andreas Zell. Visual terrain classification by flying robots. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 498–503, 2012.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, volume 1, pages 886–893, 2005.

[4] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR). Proceedings of the IEEE Computer Society Conference on*, volume 1, pages I–511, 2001.

[5] Paul Blondel, Alex Potelle, Claude Pégard, and Rogelio Lozano. Fast and viewpoint robust human detection for SAR operations. In *Safety, Security, and Rescue Robotics (SSRR), IEEE International Symposium on*, pages 1–6, 2014.

[6] Helen Flynn and Stephen Cameron. Multi-modal people detection from aerial video footage. In *Towards Autonomous Robotic Systems*, pages 190–191. Springer, 2014.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.