Metadata of the chapter that will be visualized in SpringerLink

Book Title	Distributed Computing and Internet Technology
Series Title	
Chapter Title	Bees Detection on Images: Study of Different Color Models for Neural Networks
Copyright Year	2019
Copyright HolderName	Springer Nature Switzerland AG

Author	Family Name	**Dembski**
	Particle	
	Given Name	**Jerzy**
	Prefix	
	Suffix	
	Role	
	Division	Faculty of Electronic Telecommunications and Informatics
	Organization	Gdańsk University of Technology
	Address	Gdańsk, Poland
	Email	jerzy.dembski@eti.pg.gda.pl
Corresponding Author	Family Name	**Szymański**
	Particle	
	Given Name	**Julian**
	Prefix	
	Suffix	
	Role	
	Division	Faculty of Electronic Telecommunications and Informatics
	Organization	Gdańsk University of Technology
	Address	Gdańsk, Poland
	Email	julian.szymanski@eti.pg.gda.pl

Abstract

This paper presents an approach to bee detection in video streams using a neural network classifier. We describe the motivation for our research and the methodology of data acquisition. The main contribution to this work is a comparison of different color models used as an input format for a feedforward convolutional architecture applied to bee detection. The detection process has is based on a neural binary classifier that classifies ROI windows in frames taken from video streams to determine whether or not the window contains bees. Due to the type of application, we tested two methods of partitioning data into training and test subsets: video-based (some video for training, the rest for testing) and individual based (some bees for training, the rest for testing). The tournament-based algorithm was implemented to aggregate the results of classification. The manually tagged datasets we used for our experiments have been made publicly available. Based on our analysis of the results, we drew conclusions that the best color models are RGB and 3-channeled color models: RGB and HSV are significantly better than black & white or the H channel from HSV.

Keywords (separated by '-'): Automatic bee's image classification - Deep neural networks - Bee farming

Bees Detection on Images: Study of Different Color Models for Neural Networks

Jerzy Dembski and Julian Szymański(✉)

Faculty of Electronic Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
{jerzy.dembski,julian.szymanski}@eti.pg.gda.pl

Abstract. This paper presents an approach to bee detection in video streams using a neural network classifier. We describe the motivation for our research and the methodology of data acquisition. The main contribution to this work is a comparison of different color models used as an input format for a feedforward convolutional architecture applied to bee detection. The detection process has is based on a neural binary classifier that classifies ROI windows in frames taken from video streams to determine whether or not the window contains bees. Due to the type of application, we tested two methods of partitioning data into training and test subsets: video-based (some video for training, the rest for testing) and individual based (some bees for training, the rest for testing). The tournament-based algorithm was implemented to aggregate the results of classification. The manually tagged datasets we used for our experiments have been made publicly available. Based on our analysis of the results, we drew conclusions that the best color models are RGB and 3-channeled color models: RGB and HSV are significantly better than black & white or the H channel from HSV. AQ1

Keywords: Automatic bee's image classification
Deep neural networks · Bee farming

1 Introduction

One of the important factors in bee farming is an evaluation of the colonie's strength. Usually, it is performed by a beekeeper who, during hive inspection, estimates the number of bees that occupy each of the frames. This method requires opening the hive each time it is done, so it is invasive and of course it is not precise. The estimations may vary depending on the time of the day when the inspection is performed, as well as on the weather – e.g. during a sunny day there can be less bees in the hive as some of them would be out collecting pollen.

In our research we aim to provide more objective measures for evaluating the strength of a bee colony. Our method is based on the automatic counting of bees flying in or out of the hive, evaluated on videos recorded by a mobile camera. This

G. Fahrnberger et al. (Eds.): ICDCIT 2019, LNCS 11319, pp. 1–14, 2019.
https://doi.org/10.1007/978-3-030-05366-6_25

task requires a bee detection system that allows us to analyze a video stream. The detection process itself is based on a machine learning approach that scans an entire image using a scanning window from left to right, top to bottom, and for different window scales. In our approach, we use a binary classifier that distinguishes windows containing bees from windows without bees.

The bee-detection system allows us to measure correlations between bee activity and e.g. weather, as well as open new possibilities for the automatic analysis of bee behavior, e.g. tracing bees at a hive's entrance to detect movement patterns [1].

Our approach employs a mobile RGB camera that is placed at the hive's entrance. Then, a recorded video is analyzed using a dedicated algorithm that enables us to detect bees contained in frames of the video. The results presented in this paper show that the proposed neural network architecture is useful for the task of bee detection. We also publicly provide the dataset that we manually tagged to construct our model.

This paper is constructed as follows: in the next section we give a brief overview of the research and open source projects that use computers for analyzing bee activity. Section 3 describes our data acquisition process: the environment in which we recorded videos and the method for manually tagging the video-frames used as training data for the construction of our classifier. Section 4 describes the usage of different color models for classification algorithms. The classification results have been aggregated using the tournament method presented in Sect. 5. We performed a series of experiments presented in Sect. 6. The results of these experiments show that the proposed approach is useful and can be implemented in a real-world commercial application. We discuss possible extensions of our system and further work in the last section.

2 Automatic Analysis of Bee Activity

In this section we present current works in areas related to the subject of this paper. First, we present projects that aim to automate the process of analyzing bee behavior, with a focus on bee detection and the usage of neural networks with bee-related data.

Human interest in bee farming has a long history. The knowledge gathered during the ages allows us to better understand bee behavior, however a lot is still unknown. In recent years, technological progress has given us additional tools for more detailed environment monitoring [2] and gives us a closer look at the life of bees. The usage of smart sensing technologies and algorithms that analyze data from sensors allows us to draw new conclusions on bee behavior [3]. As the application of Information Technology for environment monitoring rapidly grows, we can find more and more research that employs computers for analyzing different aspects of a bee's life [4]. Below we present some of the research that focuses on bee detection.

Automatic image analysis has been employed for bee larvae detection. The authors of the research employ deep learning convolutional networks that, as is reported, is able to segment a frame containing a bee and differentiate empty cells and that with the laves to a high degree of accuracy. The project does not contain an in-depth description of the methods used in the form of publication, but it offers its code as open source[1].

The analysis of recorded bee sounds allows us to construct a classifier that indicates presence or absence of the bee queen in a hive. This is our initial project that employs IOT technology for bee farming. It shows that classifying bee sounds with an SVM [5] and using linear predictive coding [6] allows us to differentiate between the presence and absence of the queen bee in the hive. A detailed description of this method can be found in [7].

One of the first approaches for using cameras to detect and track bees at the entrance of a hive was proposed by Campbell et al. [8]. In this approach the camera was placed directly over the entrance to the hive. To detect bees on individual image frames, the authors first subtracted an empty background image from the current image frame and then found objects that matched with one of 16 fixed-size elliptic templates. The tracking method from frame-to-frame is based on bi-partite graph matching [9] on a graph composed of the current and previous frames' detections. The system works in a fixed environment and for every environmental change the algorithm requires additional templates.

The usage of 3D cameras to detect and track bees at the entrance of a hive has been also proposed by Chiron et al. [10]. Their method segments both the color and depth portions of images, and then finds bees by matching the segments with ellipses. In order to track bees across frames, the authors used a mixture of global nearest-neighbors and Kalman filtering [11]: individual bees were tracked across frames using the Kalman filter, and global nearest neighbors was used to track the paths of specific bees.

An approach for analyzing bees on video-data has been presented in "A Deep Learning Approach to Recognizing Bees in Video Analysis of Bee Traffic"[2]. The initial results reported [12] indicate that convolutional networks can be useful for the task. Unfortunately, the authors do not publicly provide the data used in their research, which in turn doesn't allow us to perform a comparison with our approach.

In another thesis "Fast detection of bees using deep learning and bayesian optimization"[3] an approach is presented for building a vision-based hive monitoring system by training three deep learning-based object detection models that detect bees and predators using fast, region-based convolutional neural networks (Fast R-CNN). The results achieved on manually a created dataset report an F1 score below 0.5, which is insufficient for commercial applications.

Another project[4] aimed at counting bees using an artificial neural network has been made on Raspberry Pi. The authors use a standard convolutional network trained on half-resolution patches but run against full resolution images.

---

[1] https://github.com/metaflow-ai/hive.

[2] https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=8185&context=etd.

[3] https://mspace.lib.umanitoba.ca/handle/1993/32981.

[4] http://matpalm.com/blog/counting_bees/.

The approach has been published as open-source code and the authors claim it allows them to achieve good results that have been tested on a single hive. Unfortunately, the results have not been statistically analyzed and they have not been reported as scientific publication.

A system for the detection, localization, and tracking of honeybee body-parts from video at the entrance ramp of a hive has been reported in [13]. The approach employs a neural network composed of a first feature extraction part based on a pre trained VGG16 network [14] that serves as input for two convolutional network branches: the first branch predicts a confidence map with one channel per part of interest, and the second branch predicts affinity fields with one vectorial channel per connection. The proposed system shows very promising results with over 95entrance, without any prior tracking. It should be noted, however, that in order for the system to work correctly, it requires that the camera be placed exactly in the same position and orientation as it had while training data was acquired.

The problem of bee pose-estimation on single image has been analyzed in [15]. The authors used ConvNet [16] which had previously been employed for human pose-estimation using a fixed number of body joints. The proposed method applied to bee pose-estimation handles cases with a varying number of targets, which allows it to achieve very good results in terms of insect pose-estimation.

The aforementioned mentioned research relating to bee detection does not provide statistically reliable results nor present any publicly available data that would allow for the replication of said results. Some of the research only provides source code without any deeper analysis of the method used or a research paper. Thus, we propose our own approach that has been presented in detail in this paper and we have made the dataset we experiment on publicly available.

The continuous monitoring of bee hives requires the system to match two criteria: speed and accuracy of detection. In the beginning of the 21st century, the Adaboost algorithm [17] with Haar-like rectangular features [18] seems to be the best solution for these two criteria, especially in the face of detection tasks [19,20]. We investigated the usage of these techniques for bee detection and the results were not satisfactory, probably due to a significant difference between the domains of face and bee detection, such as a high percentage of blurry images and a smaller number of details.

In the recent decades, deep learned artificial neural networks with convolutional layers for image classification [21] have been used thanks to a greater generalization measured by the accuracy of test example classification, as well as better hardware allowing for faster calculation speed.

In recent years, newer ideas such as specialized network structures consisting of CNNs have been developed for object detection. One such idea is the region of interest (ROI) extraction from the whole image using very little calculation. Another method uses multitask learning connected with region-based CNN (R-CNN) [22] in which at the output of the CNN two kinds of information are presented: object probability and object position in the ROI area. Yet another idea uses pixel-wise segmentation [23] in which every pixel in an image is classified as belonging to an object or not.

3 Data Acquisition

For building our classifier we use 13 videos, each approximately 40 s long. Each video was shot during sunny weather using a mobile device camera without a tripod or any additional equipment. The videos capture a bee hive from different perspectives and different distance from the hive's entrance. In order to analyze the videos using machine learning classification, the data first needed to be tagged. This task can be completed using tools such as LabelImg[5] but we decided to develop our own software to have additional labeling possibilities such as marking the orientations of a bee's body, individual numbers, negative example areas, and areas excluded from detection during error evaluation, as was shown in Fig. 1. This extended information is not used in the experiments reported in this paper, but the dataset we acquired may be useful for future experiments. Thus, we made our dataset publicly available. The videos are accessible from the web url: https://goo.gl/KNV7sd. Each video has a corresponding metadata text file, wherein each line in the text file describes each video-frame. The text files are available under the url: http://julian.eti.pg.gda.pl/ramki.zip, and contain description of the frames stored in the following format:
<frame no> <x coordinate> <y coordinate> <radius> <1-bee, 0-background> <individual id> [6 fields not used now] <bee direction (0,2π)> <quality>

Our tagging tool allows us to tag video frames in such a way that a user can quickly mark all regions that contain an individual bee from frame-to-frame. Later, the user can append labels to the bee such as an ID number, information on a particular feature, e.g. whether or not the bee is arriving or returning to the hive, if the bee is bringing food to the hive, or if the bee is the queen bee.

Images with blurry objects such as the frame with individual number 3 in Fig. 1 lead to a marking dilemma. Fast moving bees recorded with ordinary cameras without shutter speed control can appear blurry and near transparent, which introduces difficulties in recognizing them, even for humans, who need to look at other neighboring frames for verification. This can also adversely influence classification, due to its proximity to background images. A similar problem is overlapping bee bodies. For these reasons we decided to only tag images containing recognizable bees, or fill in an image quality field as an additional attribute to preserve information of a bees position for tracking purposes.

Negative examples (background images) were prepared from the same videos as the positive samples were prepared from, keeping a similar proportion between the two. Such an approach guarantees proper data balance and provides images from the beekeeping environment, or the environment of particular beekeeper. In our acquisition system, a user should mark some ROI which do not contain bees (gray rectangle boxes in Fig. 1). These ROI are marked as negative examples

[5] https://github.com/qaprosoft/labelImg.

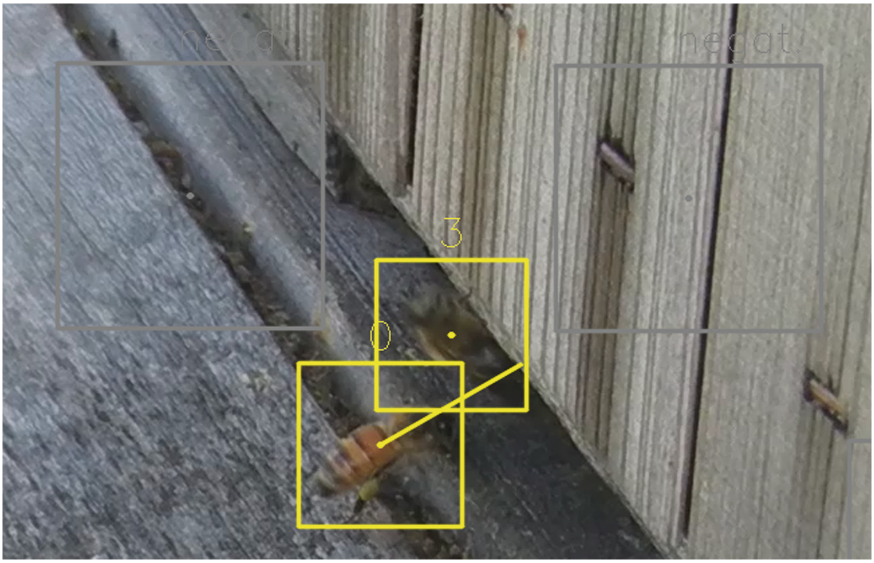

Fig. 1. Typical boxes of negative (gray) and positive (yellow) example images (Color figure online)

and are selected randomly. Additionally, for each negative ROI, a new image is generated from the ROI to enable the user to mark any artifacts that might be classified as false positives, such as bee shadows, wood grain, etc.

4 Training Bees Image Classifiers Using Deep Models

As input for the classifier, we test the usage of four different color models: RGB, HSV, black & white, and the Hue (H) component from the HSV model [24]. The black & white color model was obtained from an RGB model by calculating pixel intensity according to the formula: $J = 0.299R + 0.587G + 0.114B$ where R, G, B are red, green, and blue channel intensities for the same pixel.

Our approach for bee detection on a video frame consists two stages:

- binary classifiers trained to distinguish between images of bees and images of backgrounds,
- a detector algorithm that uses binary classifiers from the previous stage and that also can be trained.

To construct a model for processing the images we used deep convolutional neural networks that are composed of only a few convolutional layers and a few fully connected layers. This model has been used to construct a binary classifier that allows us to distinguish between frames containing bees and background images. The topology of the convolutional neural network was designed by trial and error testing during preliminary experiments, in such a way as to have enough capacity to enable the training error to decrease towards zero and not be too complex, mainly because of two reasons:

– overfitting avoidance in order to obtain a low test error,
– decreasing time complexity which is crucial for real-time detection applications.

Both of the aforementioned reasons were selected as our optimization criteria for creating a neural network topology. The neural network architectures used vary depending on the particular task, color model, and an input image resolution. The typical structure for a 48×48 3-channel RGB image has been shown in Fig. 2.

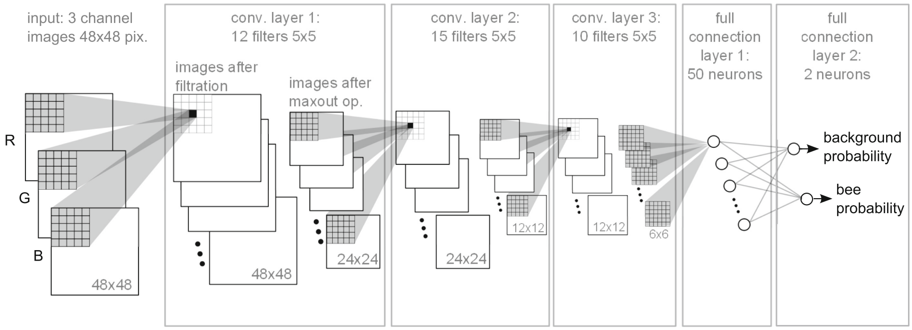

Fig. 2. Convolutional neural network structure used in color models comparison experiment

Each pixel intensity in a convolutional layer is calculated in the standard way as a ReLU activation function from a sum of weighted intensities of a 5×5 pixel area, in each input image from the previous layer. We use the padding option "same" to reduce the size of the input image. The resolution of each image after filtering is reduced two times by a maxout operation. In the last full connection layer we used a softmax activation function. About 27, 000 positive examples (bee-boxes) were obtained for training and test purposes. We chose four types of partition into a training and test set, due to implementation purposes:

1. fully random division corresponding to the specific implementation for one beekeeper in a given season,
2. some bees for training and others for testing (bee based partition),
3. some videos for training and others for testing (video based partition),
4. some hives for training and other for test (hive based partition).

The difficulty grows from levels 1 to 4. Of course, the more representative set of examples is used for training, the small differences between the results are (test examples classification accuracy) because of these partitions.

5 Aggregation of Classification Results

The bee-detector described in the previous section can classify whether or not a window (that is a part of the frame) contains a bee.

In the task of counting the bees in a video frame, we need to scan the whole image assuming a minimal and a maximal size of the scanning window, the size change step, and the vertical and horizontal window shift steps. The typical result of such a scanning process is a huge amount of positive detection boxes around true bee occurrences, due to an inaccuracy of positive example acquisition as is shown in Fig. 3a. Such a cloud of boxes must be reduced to one detection box for each bee, which requires an algorithm for selecting boxes. The challenge becomes more sophisticated when bees are close to each other, which is manifested by the overlapping of positive detection clouds. In our research we use a fast greedy tournament algorithm [25].

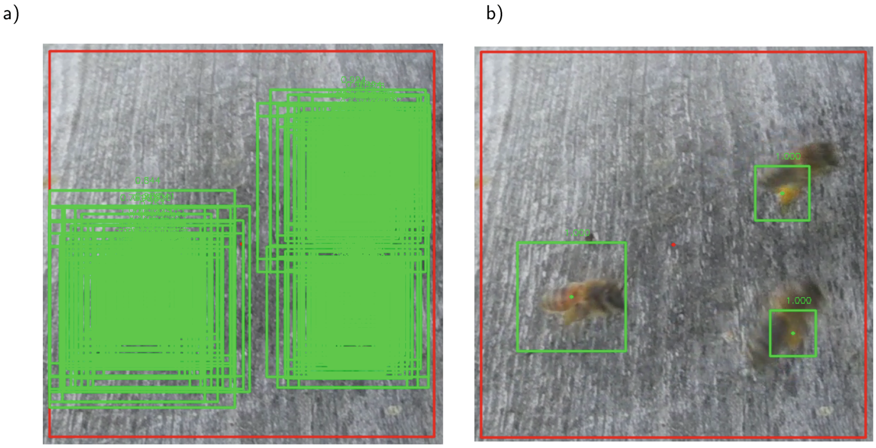

Fig. 3. Greedy tournament detection results: (a) set of frames after scanning the part of the image with a probability of bee appearance greater than 0.5, (b) frames selected based on the probability of a bees appearance. The probability is provided as a number over the box

The greedy tournament algorithm implemented by us is based on an iterative bee appearance probability comparison between overlapping random boxes after scanning a video frame. The overlapping threshold was chosen arbitrarily for 40

We think the aggregation stage can be improved using other approaches, such as:

– clique number minimization [26] from the graph theory domain,
– a weighted greedy C-means clustering algorithm [27].

It should be noted that the higher complexity of these algorithms can negatively influence detection-time.

6 Experiments and Results

In the first experiment we compared the four color models not only to find the best one, but also to compare them, taking into account their originality and time complexity.

The structure of our convolutional network was composed of a convolutional part with full-filter connections between neighboring layers and fully-connected layers at the end. During optimization we used the adaptive moment technique (ADAM) and a cross entropy loss function. The dropout technique was used in the fully-connected layers, excluding the last layer with a probability of 0.5. Training examples (input images + class numbers) were presented to the network in the form of packages – minibatches containing 100 randomly selected examples.

The results presented in Fig. 4 were obtained from the best-performing network architecture selected from several architectures tested on each color model, in order to maximize accuracy criterion. The Tensorflow library was used in the implementation of each CNN model.

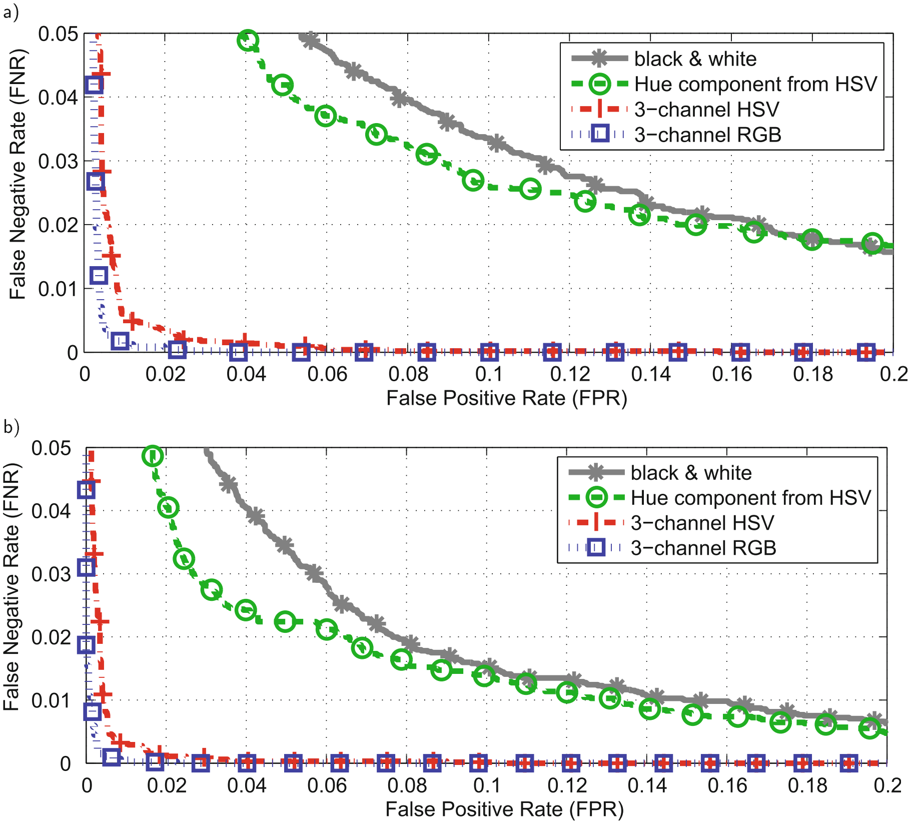

Fig. 4. ROC curves for 4 color models: (a) test example subset taken from videos not included in the training example subset, (b) test example subset with bees unseen in the training example subset. (Color figure online)

The ROC curves presented in Fig. 4 have been obtained by testing classifiers for different positive class probability thresholds. When the threshold is decreased, more images are classified as positive, which decreases the false negative rate (FNR) but increases the false positive rate (FPR). This causes the point in the ROC curve to shift to the right side of the chart.

The RGB color model, for the task of binary bee-detection in images (performed on our deep models), is the best color model by a significant margin, as is also shown in the 3-channel color models: HSV and RGB seem to be significantly better than one-channel models in general.

It should be noted that classifiers trained on one-channel models can be useful for two reasons: the classifier structures can be simpler and less computationally expensive, such a limitation of information can force a network to take into account other features than the ones given in any 3-channel color model, which may be useful in a classifiers ensemble. The black & white model, despite the worst, shows results that can be important in any extraordinary light conditions or artificial light, or in the case of atypical white balance setting in a digital camera. The Hue component from the HSV color model allows for better results than the black & white color model, which leads to another conclusion from this experiment, namely color information is very important in for bee detection, as opposed to other tasks such as face recognition.

The results presented in Fig. 4a were obtained for a video-based partition of training and test data. Each video was obtained from a different camera view and most often displays a unique beehive. We used 10 videos - 44970 examples for training and 3 videos for test - 10308 examples. Half of the examples are positive. The structure for the RGB color model is shown in Fig. 2. For the other models, the structures were slightly different.

Figure 4b displays the results for a bee-based partition into training and test data. It can be treated as the same camera view for the train and test subsets, but different individual bees were used for training and for testing. This is almost equivalent to training and testing on data from the same camera position and orientation but at different times in the video. We used images from 3/4 bees and 3/4 background images - 41913 examples for training and 13435 examples for testing. The network structure remains the same for all models.

The results for a bee-based partition are significantly better than for a video-based partition, which was expected due to a greater exposure to the environments than in the video-based partition, in which each video contains either a unique hive or a unique view. In Fig. 5 the best results for 3-channel color models are magnified.

Taking into account the results from the first experiment in the second experiment, we tested if the color information alone can be enough for a computationally cheap ROI indicator. The 24×24 RGB color images were transformed into HSV model and next were transformed into 9-point histograms of the hue (H) component. The results were unsatisfactory, so 9-component feature vectors were magnified by adding similar 9-point histograms of central 12×12 subimages and 9 values of difference between both histograms. The results are also poor,

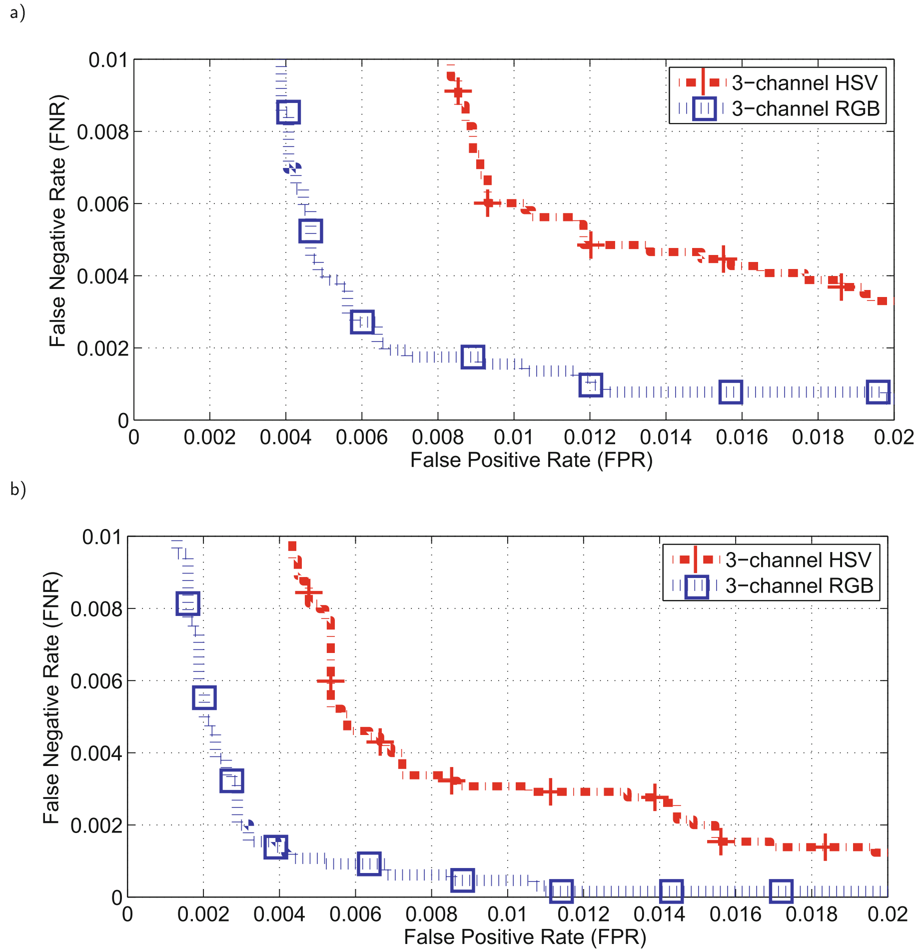

Fig. 5. ROC curves in greater scale (a) test example subset taken from videos not contained in the training example subset, (b) test example subset with bee individuals not contained int the training example subset. (Color figure online)

which may indicate as to the importance of local dependencies between pixels, which are omitted in the color histograms.

7 Discussion and Future Works

In this paper we presented our system for the acquisition of data from images containing bees, based on a deep convolutional neural network classifier used to detect bees in 48×48 pixel images. We made a comparison of color models and proposed a tournament-based algorithm for the aggregation of classification results. We also provided publicly provided our data which allows for further research to extend on our results achieved in the domain of bee detection.

The results shown in Sect. 4 indicate the significant advantage 3-channel color models have over 1-channel models, as well as the small advantage that the RGB model has over the HSV model in the individual-bee data partition. In the future we plan to test how the influence of local or global color normalization affects the results.

We also have ideas to improve the results of the binary classifier accuracy. As for now, our model classifies single images. It is our contention that the usage of frame sequences could provide information that would yield better results. For instance, a previous and next frame can be added to the CNN's input images as additional channels or recurrent variants of the CNN can be used based on the whole sequence.

Manually creating the dataset is very laborious, but having a larger number of examples for the model should work better. We are planning on extending the dataset by building a semi-automatic data acquisition system which can find ROI areas based on existing classifiers.

We plan to automate the preparation of negative images from videos during parallel binary classifier training. The usage of this approach for detection tasks has been described in a cascade classifier training process in [20]. In such a setting, the negative images set will be periodically supplemented by images generated from false positive windows. Also usage SOM for visualizing ROI shell be considered [28].

The detector described in Sect. 5. can be additionally equipped with several binary classifiers to differentiate image areas. We are planning the following work in this field:

- as we mention at the end of the Sect. 5 aggregation algorithms other than tournament-based should be tested,
- a committee of binary classifiers trained on different color models and different resolutions,
- tracing algorithms and bee direction recognition to accelerate the detection process in video frames,
- a cascade of deep learned binary classifiers from simplest, fast, to complex, and more accurate,
- the pixel-wise detection approach verification [29],
- the genetic optimization of aggregation parameters with two criteria: speed and accuracy.

We plan to deploy our algorithms for bee detection on the Android mobile environment. This should allow us to collect more data as well as extend the research on bee image-processing. In the future we plan to conduct work on the identification of bees infected by Varroa mites[6]. The automatic detection of

[6] http://www.releasewire.com/press-releases/the-beescanning-app-is-saving-bees-worldwide-through-deep-learning-technology-808184.htm.

these parasites on bees should allow one to estimate the degree of an infection and suggest plan of treatment.

We also plan to perform an analysis on the videos with bees for the detection of abnormalities at a hive's entrance. This should allow us to detect a swarming moment [30] and immediately report it to a beekeeper.

Acknowledgements. This work was partially supported by funds from the Faculty of Electronics Telecommunications and Informatics, Gdansk University of Technology and Cost Action CA 15118 FoodMC "Mathematical and Computer Science Methods for Food Science and Industry".

References

1. Wario, F., Wild, B., Rojas, R., Landgraf, T.: Automatic detection and decoding of honey bee waggle dances. PloS one **12**, e0188626 (2017)
2. Othman, M.F., Shazali, K.: Wireless sensor network applications: a study in environment monitoring system. Proc. Eng. **41**, 1204–1210 (2012)
3. Tu, G.J., Hansen, M.K., Kryger, P., Ahrendt, P.: Automatic behaviour analysis system for honeybees using computer vision. Comput. Electron. Agric. **122**, 10–18 (2016)
4. Zacepins, A., Stalidzans, E., Meitalovs, J.: Application of information technologies in precision apiculture. In: Proceedings of the 13th International Conference on Precision Agriculture, ICPA 2012 (2012)
5. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
6. Bradbury, J.: Linear Predictive Coding. Mc G. Hill, New York (2000)
7. Cejrowski, T., Szymański, J., Mora, H., Gil, D.: Detection of the bee queen presence using sound analysis. In: Nguyen, N.T., Hoang, D.H., Hong, T.-P., Pham, H., Trawiński, B. (eds.) ACIIDS 2018. LNCS (LNAI), vol. 10752, pp. 297–306. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75420-8_28
8. Campbell, J., Mummert, L., Sukthankar, R.: Video monitoring of honey bee colonies at the hive entrance. Vis. Obs. Anal. Anim. Insect Behav. ICPR **8**, 1–4 (2008)
9. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of Bipartite graph matching. Image Vis. Comput. **27**, 950–959 (2009)
10. Chiron, G., Gomez-Krämer, P., Ménard, M.: Detecting and tracking honeybees in 3D at the beehive entrance using stereo vision. EURASIP J. Image Video Process. **2013**, 59 (2013)
11. Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer, Heidelberg (2009)
12. Tiwari, A.: A deep learning approach to recognizing bees in video analysis of bee traffic (2018)
13. Rodríguez, I., Branson, K., Acuña, E., Agosto-Rivera, J., Giray, T., Mégret, R.: Honeybee detection and pose estimation using convolutional neural networks. Technical report, RFIAP (2018)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

15. Duan, L., Shen, M., Gao, W., Cui, S., Deussen, O.: Bee pose estimation from single images with convolutional neural network. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 2836–2840. IEEE (2017)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
17. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**, 119–139 (1997)
18. Porwik, P., Lisowska, A.: The Haar-wavelet transform in digital image processing: its status and achievements. Mach. Graph. Vis. **13**, 79–98 (2004)
19. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**, 137–154 (2004)
20. Dembski, J., Smiatacz, M.: Modular machine learning system for training object detection algorithms on a supercomputer. In: Advances in System Science, pp. 353–361 (2010)
21. Hoo-Chang, S., et al.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**, 1285 (2016)
22. Girshick, R.B.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, 7–13 December 2015, Santiago, Chile, pp. 1440–1448 (2015)
23. Pinheiro, P.H.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks, pp. 1713–1721 (2015)
24. Zarit, B.D., Super, B.J., Quek, F.K.: Comparison of five color models in skin pixel classification. In: Proceedings of International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 58–63. IEEE (1999)
25. Blickle, T., Thiele, L.: A mathematical analysis of tournament selection. In: ICGA, pp. 9–16. Citeseer (1995)
26. Erdos, P., Jacobson, M., Lehel, J.: Graphs realizing the same degree sequences and their respective clique numbers. Graph Theory Comb. Appl. **1**, 439–449 (1991)
27. Cannon, R.L., Dave, J.V., Bezdek, J.C.: Efficient implementation of the fuzzy c-means clustering algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 248–255 (1986)
28. Szymański, J., Duch, W.: Self organizing maps for visualization of categories. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012. LNCS, vol. 7663, pp. 160–167. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34475-6_20
29. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**, e0130140 (2015)
30. Ferrari, S., Silva, M., Guarino, M., Berckmans, D.: Monitoring of swarming sounds in bee hives for early detection of the swarming period. Comput. Electron. Agric. **64**, 72–77 (2008)

Author Queries

Chapter 25

Query Refs.	Details Required	Author's response
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	