Recognition of Pollen-bearing Bees from Video using Convolutional Neural Network

Iván F. Rodriguez [†]
ivan.rodriguez5@upr.edu

Rémi Mégret [*]
remi.megret@upr.edu

Edgar Acuña [‡]
edgar.acuna@upr.edu

José L. Agosto-Rivera [§]
jose.agosto1@upr.edu

Tugrul Giray [§]
tugrul.giray@upr.edu

[*] Department of Computer Science, University of Puerto Rico, Río Piedras campus
[†] Department of Mathematics, University of Puerto Rico, Río Piedras campus
[‡] Department of Mathematical Sciences, University of Puerto Rico, Mayagüez campus
[§] Department of Biology, University of Puerto Rico, Río Piedras campus

Abstract

In this paper, the recognition of pollen bearing honey bees from videos of the entrance of the hive is presented. This computer vision task is a key component for the automatic monitoring of honeybees in order to obtain large scale data of their foraging behavior and task specialization. Several approaches are considered for this task, including baseline classifiers, shallow Convolutional Neural Networks, and deeper networks from the literature. The experimental comparison is based on a new dataset of images of honeybees that was manually annotated for the presence of pollen. The proposed approach, based on Convolutional Neural Networks is shown to outperform the other approaches in terms of accuracy. Detailed analysis of the results and the influence of the architectural parameters, such as the impact of dedicated color based data augmentation, provide insights into how to apply the approach to the target application.

1. Introduction

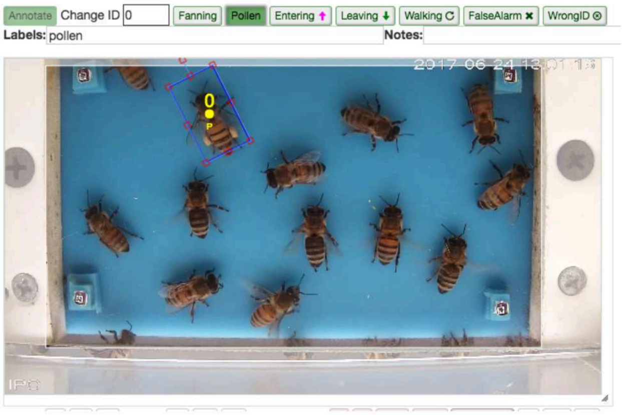
Bees play essential role in pollination, which is crucial for agriculture and ultimately for human existence. They also behave in very complex social way that includes hierarchy, roles, schedules and interactions. In order to understand these behaviors, very careful observation and registering needs to be done. With the use of recent technology developments, this observation is not only feasible, but possibly even broader, making easier to find and register detailed individual and group conducts.

The interest on observation of honey bees activities within and outside the colony began to be documented since nearly a century ago [17]. For the most part, the traditional technique remains human observation and manual annotation, as this is the only approach that enables the extraction of a wide range of behaviors and is readily available to bee specialists. It is a very time consuming and expensive task that requires long periods of observation and sometimes specific expertise in order to be meaningful. Thus, important insights may still missing to be observed or demonstrated due to lack of data. Computer vision and machine learning techniques provide the framework needed to analyze the insects behavior automatically and provide new insights [3].

The observation of Honey Bee hives is of interest for multiple applications. Bee keepers, for instance, might get better understanding to prevent sickness in the colony caused by external factors that can be recognizable in video [19]. Early detection of poisonous materials that bees are bringing as fraudulent pollen [6] or diagnosing the health of the hive [8]. Furthermore, biologists can understand better the pollen scheduling and individual roles within the hive, which can be linked to DNA individual composition.

Concept recognition from images have been a matter of very fast and growing performance in the last decade. Several methods have been proven to be effective at this task. In particular, Convolutional Neural Networks (CNN) [15] have been shown to learn both low-level and higher-level features without requiring explicit supervision.

In this work we present a study and comparison of different techniques for detection of pollen in video. Classifiers such as KNN, SVM and Naive Bayes were used as baseline. Convolutional Neural Networks were tested using different parameter configuration: shallow models of



Change ID 0

Labels: pollen

Notes:

Figure 1. Example of capture: A bee with pollen entering the colony and other bees

one or two convolutional layers, and deeper models using architectures from the literature such as VGG16, VGG19 and ResNet50. As color is a priori a relevant feature for the presence of pollen, a color feature specific to pollen was also considered.

The organization of the paper is as follows: in Section 2, related work will be presented; in Section 3, we will describe the proposed approach and the experimental setup. In Section 4, the results will be presented and discussed, before the Conclusion in Section 5.

2. Related work

Several studies have discussed the use of video for monitoring bees or insects. In [4], [5], for instance, advantages of using video instead of invasive methods such as RFID tags are illustrated. They proposed the use of computer vision and machine learning algorithms for detection of foraging activities at individual level, placing individual tags to bees to track the schedules of departures and arrives within the colony. In [19] the use of machine learning is first used for detection of Varroa mite, which recognizes spots of color red in the bodies of the bees. In [22] computer vision is used for automatic behavior analysis. In [13], behavior of flies is analyzed based on their temporal trajectory.

2.1. Pollen recognition

Recognition and classification of pollen have been widely studied at a microscopic scale (i.e. by observing only the collected pollen in a microscope). For instance, [16] and [14] aim at detecting pollen on air for allergy diagnosis. In [6], a microscopic scale study was performed to detect fraudulent pollen getting in the colony.

At macroscopic scale (i.e. when observing the bees bearing the pollen), very few studies have been performed so far. In [3] a system is presented that targets the embedded Rasperry Pi device with a camera of resolution 1280×720 for recording inside the hive. A dataset composed of 121 pollen bearing bees and 770 non pollen bearing bees was used. A Mixture of Gaussians (MOG) model was used for segmenting the background color and variance and eccentricity of color was used as light-weight features for classification by a Nearest Mean Classifier. This approach was reported at 88.7% accuracy. A codebook approach using VLAD descriptors [12] computed from color MSIFT features [2] reached 92.1% accuracy on average using 200 training samples.

2.2. CNN for visual classification

During the past few years the success of Convolutional Neural Networks (CNN) for image classification have promoted this approach as the state of the art method for visual classification [15]. The use of CNNs for classification of images have even surpassed human performance on a few applications [10]. The rise in computer power has enabled its application at large scale. However there is still a very active discussion on what is the optimal architecture one should use. Although improvements were shown using very deep CNNs [21], recent experimental results suggest that much shallower architectures may achieve similar results [11].

To the best of our knowledge, no previous work has attempted to apply CNNs for pollen detection, although several points argue in favor of their fitness to the problem at hand: invariance of the convolution operation to process pollen balls at various locations in the image, learning of feature maps that take into account both color and geometry, flexible and powerful architecture that allows reusing parts of the network to adapt the models to a new experimental setup without having to retrain the models from scratch.

3. Problem statement and data collection

This work is motivated by the automatic monitoring of bees to obtain a large amount of behavior information for long-term tracking of the colony health and large-scale scientific studies. The objective of detecting the presence of pollen is of prime importance to assess the success of foraging tasks and study the division of labor amongst bee workers. One difficulty of such detection lies in the current necessity to retrain the models for different types of bees and experimental setup. Indeed different species of bee can hold the pollen balls differently, and the bee and the pollen colors can vary significantly depending on the type of flowers that are foraged, as well as depending on the illumination and viewing conditions. The amount of training data is therefore highly dependent on the investment by bee specialists in annotating a representative sample of the conditions encountered.

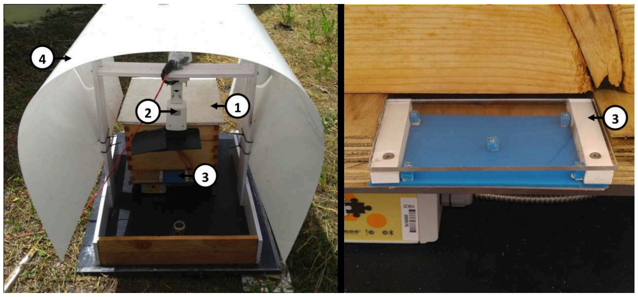


Figure 2. Video capture system used in the field: overview of the system installed at the entrance of the colony and detail on the entrance. (1) bee hive, (2) camera, (3) entrance ramp, (4) protection against direct sunlight.

3.1. Video capture system

The video capture system is designed to observe the ramp through which all foraging bees must pass to exit or enter the colony. Figure 2 shows the system used in this work. We used a 4 Mpixels GESS IP camera connected to a networked video recorder configured at 8Mbps for continuous recording. A transparent acrylic plastic cover located on top of the ramp enforces that the bees remain in the focal plane of the camera. Due to constraints to avoid interfering with the bee biological cycles, only natural light is used. A white plastic diffuses the natural light received, and a black mask is put around the camera to reduce the direct reflections that could be visible on the ramp cover. The videos where acquired in June 2017 at the UPR Agricultural Experimental Station of Gurabo, Puerto Rico. The two videos used in this work are of one hour duration and were recorded at 10 a.m. and 1 p.m. to take into account different lightings.

3.2. Dataset

As part of the contributions of this work an annotated dataset has been released for public access (`https://github.com/piperod/PollenDataset`). This dataset contains high resolution images of pollen-bearing and non pollen-bearing honeybees as shown in Figure 4. These images were extracted from the videos captured using the procedure described below.

Using in-house annotation system based on [20], the videos were manually annotated using a protocol defined to avoid near-duplicate samples and ensure a balanced and representative dataset. For each video of one hour, the video was visualized in chronological order and the annotator instructed to stop as soon as a pollen bee was entering the ramp. The annotation would then be performed as the bee reached the middle of the path leading to the entrance. A second bee without pollen would be annotated on the same frame to account for similar lighting conditions and ensure a balanced dataset. Since the ramp contains dozens of non pollen-bearing bee at all times, this could be done without repeating the same individual in a similar position. The pollen bee would not be annotated again in its trip toward the colony to avoid duplicates. The annotation consists in the position of the bee's thorax, its orientation angle, and the presence of pollen, as illustrated in Figure 1, where part of the annotation system is visible.

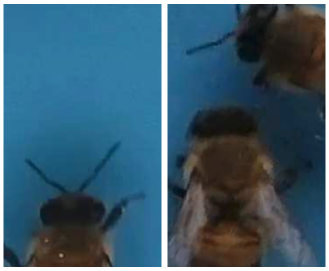


Figure 3. Misaligned samples for Pollen and Non Pollen bearing bees.

The dataset used for the recognition was created by extracting the individual images of the bees, with their respective pollen/nopollen labels. The orientation of the bees was compensated to ensure in all image samples that the bee is facing upwards. With this information, the image dataset was built fixing the size of the cropping rectangle to 180×300 pixels, such that the annotated thorax position appears centered at coordinates (90,100) and that the bee is fully visible.

It can be noted that the orientation could be inferred automatically by using bees marked with coded tags such as [4] or other automatic alignment approach. This was not done in the present study to focus on the evaluation of the intrinsic difficulty of the pollen recognition task using good quality manually annotated data.

A total of 810 bees images were sampled by the annotators, half of them labeled as Pollen bearing and the other half as Not Pollen. This raw dataset was curated by a different person, who removed a total of 100 samples that had misplaced annotations with misaligned samples (Figure 3). A few slightly misaligned samples judged non ambiguous by the curator remained (see for instance Figure 13). The resulting dataset used in this work contains a total of 710 samples (354 Pollen and 346 Not Pollen).

To our knowledge, this dataset is the first public dataset of this type and size, i.e. using natural light, good resolution imaging and manual annotation of the bee position and orientation.

4. Classification approaches

Three different approaches have been considered in this work: direct classification using baseline classifiers, shal-

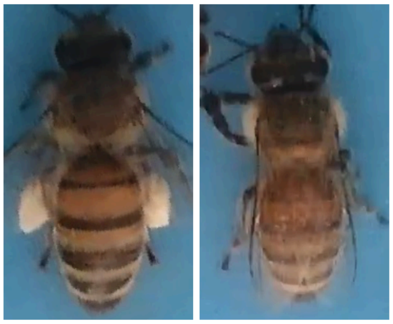


Figure 4. Pollen and Non Pollen bearing bees.

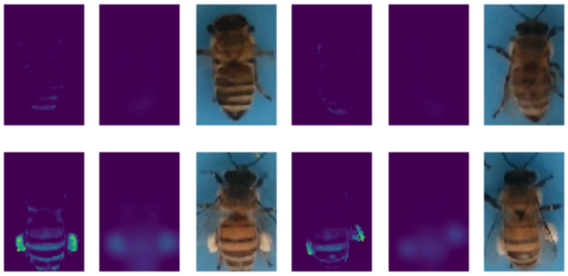


Figure 5. Non Pollen Bearing bees (top) and Pollen Bearing Bees (bottom). With their respective feature extraction. (Color Extraction left and Gaussian Blur right)

low Convolutional Neural Networks (CNN), and deep CNN from the literature. For the baseline classifiers, KNN, Naive Bayes and SVM with linear and RBF kernels were used. For the CNN, shallow networks with one and two convolutional layers were considered. Deep architectures such as VGG16, VGG19 and ResNet50, followed by a dense layer to fit the Pollen/Not Pollen classes were also tested.

4.1. Data preparation

As pollen balls have an obvious color component, an ad-hoc pollen color feature map (hereafter refered to as *Color*) was considered as an addition to the RGB images. The images were converted to Hue Saturation Value colorspace (HSV) and the following Gaussian model was applied:

$$K = e^{-\frac{1}{2}\frac{(h-\mu_h)^2}{\sigma_h^2}} * e^{\frac{1}{2}\frac{-(s-\mu_s)^2}{\sigma_s^2}} * e^{\frac{1}{2}\frac{-(v-\mu_v)^2}{\sigma_v^2}} \quad (1)$$

with σ_h = 0.1, σ_s = 0.8, σ_v = 0.3 , σ_h = 0.05, σ_s = 0.05 and σ_v = 0.8 determined empirically to highlight most of the pollen balls in the dataset.

The Color feature map was also smoothed with a Gaussian blur of σ= 8 pixels to produce the *Gaussian* feature map.

The RGB image, Color and Gaussian feature maps were normalized and resized to match the classifier architectures as described below.

4.2. Baseline Classifiers

Three models of supervised classification (KNN, Naive Bayes and SVM) were applied to the three feature map images described previously (RGB images, Color and Gaussian). Before feeding the classifiers, each image was normalized to have a [0,1] range and resized to half size 90×150.

Given the high dimensionality of the input data and the dataset size, the use of Principal Components Analysis was also evaluated. We ran experiments keeping 160, 80, 40, 20, 10, 5, and 2 dimensions with respectively 98%, 95%, 87%, 77%, 65%, 53% and 39% explained variance for Gaussian features, and 88%, 78%, 56%, 31%, 20% and 15% explained variance for Color. RGB showed similar explained variance as Color.

Knn. This supervised algorithm, known as the K nearest neighbor algorithm, outputs directly class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. If $k = 1$, then the object is simply assigned to the class of that single nearest neighbors. In the experiments, $k = 3$ was used.

Naive Bayes. Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes theorem with the naive assumption of independence between every pair of features. We used the the implementation available in [23] with a Gaussian distribution model.

SVM. Support Vector Machines (SVM), with both linear and radial basis functions (RBF) were considered in this study, for the ability to separate hard instances in data. We used default parameter $C = 1$ for both and $\gamma = 1/n$ for the RBF where n is the dimensionality of the input data.

4.3. CNN Classifiers Architectures

We considered shallow CNNs constructed on the following basic module: sequence of 2d-convolution, relu-activation, max-pooling. The complete architecture is build as the sequence of 1 or 2 basic modules, followed by flattening and a dense layer with 2 outputs. Depending on the number of basic modules, we will refer to this architecture as 1 or 2 layer model. The parameters to be specified were the number of kernels, size of kernels, pooling size and step, and units of the summarizing layer.

The deep CNN architectures, VGG16, VGG19 and ResNet50 were pre-trained using the publicly available weights from their respective sources. The top layer of each architecture was replaced by a binary output dense layer before training the whole network on the pollen task. The input images were used at their initial size of 180×300 pixels to take into account the deep architecture with multiple subsampling layers.

5. Results

5.1. Experimental setup

The experiments were divided in three different approaches: baseline classifiers, shallow CNN and deep CNN.

Stratified split was used to create the training (70%) and validation (30%) datasets. Accuracy was the metric considered in all the approaches to measure performance.

All the experiment were performed using Scikit-learn [18], Scikit-image [23], OpenCV [1] and Keras [7]. They were run on a 6-core Intel Xeon E5 Core i7 with 64 GB RAM.

5.2. Baseline classifiers

The results of the classification task using KNN, Naive Bayes and SVM are summarized in Table 1. SVM RBF classifier with PCA reached the best accuracy using the Gaussian feature map at 91.16%. The table shows the best results according to the dimensions kept after PCA was performed. In general PCA showed positive impact on performance. In some cases accuracy improved up to 30%, suggesting in these cases that the high dimensionality generated overfitting. Best results were obtained with less than 80 dimensions. The running time for each classifiers was dominated by the PCA computation, therefore minor impact on time was observed when running the classifiers with different dimensions.

The lowest accuracy was obtained using the raw RGB Image. Consistent improvement was observed when using the Color features combined with the Gaussian features. This suggests: first, that the intuitive approach of performing pollen color detection actually enhances relevant information; second, that spatial filtering also has a positive effect. The CNN architecture unifies these two aspects (color selection and spatial filtering) in a form that is trainable, which we discuss next.

5.3. Shallow models

5.3.1 Influence of the parameters

To choose the best hyper parameters for these architectures, parameter exploration was performed. Figures 6, 7 and 8 summarize the effect of different choices of parameters on the performance of the network in our particular problem. The parameters tested were the number of kernels, size of kernels, pooling size, step size and units of the summarizing layer.

The results obtained show accuracies varying from 50% to 96%. The choice of larger kernel sizes reveals a better performance on both architectures tested. This suggests that a larger kernel size is necessary on such shallow network to cover enough of the pollen ball to be able to differentiate it from other parts of the bee.

	KNN	NB	SVM	SVM rbf
Without PCA				
Image (RGB)	77.92	77.18	77.31	50.66
Color	79.54	79.54	82.34	59.35
Gaussian	84.84	79.25	82.78	58.62
Training time	40-60s	5-10s	60-80s	140-160s
With PCA (Best dim)				
RGB (80 dims)	80.73	77.11	77.45	73.04
Color (20 dims)	87.43	77.79	82.79	89.85
Gaussian (80 dims)	84.60	77.69	84.79	91.16
Training time	81s	81s	81s	81s

Table 1. Accuracy of baseline classifiers with and without PCA preprocessing, using different feature map images as input. Only best results are shown based on different dimensionality reduction. For each approach the range of total computing time for the training is shown (including the PCA preprocessing when used).

Approach	Acc	Architecture (f,k,p,s,u)	Time per epoch (s)
1-Layer	96.4	(4,7,8,2,15)	10-25s
1-Layer + Color	95.2	(4,7,8,1,10)	45-60s
2-Layer	96.4	(8,1,8,1,15)	15-30s
2-Layer + Color	95.2	(4,7,8,1,15)	55-70s
VGG16	87.2	see [21]	1300-1400s
VGG19	90.2	see [21]	1650-1750s
ResNet50	61.7	see [9]	1700-1800s

Table 2. Results: Shallow Architectures Accuracy (f=filters, k=kernel size, p=pooling size, s=pooling step/stride, and u=units)

The use of large pooling size combined with small step sizes produced the best performance. More generally, larger step sizes yielded very poorer, even coupled with larger pooling sizes. The reduction of resolution linked the pooling step size had a marked detrimental effect.

The number of kernels did not show a clear impact in performance, in terms of the best performing networks, although higher numbers (8 and 16) had better average accuracy. Having more diversity in the computed features therefore seemed to help, and did not lead to marked overfitting.

5.3.2 Comparison to baseline classifiers

For this evaluation, best models were selected for each shallow architecture. The parameters and performance are reported in Table 2, as well as their ROC curve in Figure 9.

The 2-layer model showed similar performance than the 1-layer model, when using small step sizes, getting up to 96.4% in the best configuration. The 1-layer models for both RGB Image and color feature map inputs reached the same accuracy.

It is noteworthy that the Color feature map as input did

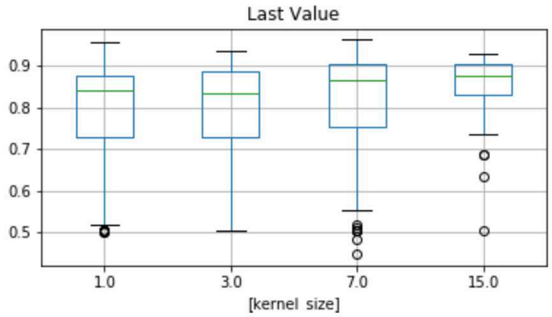


Figure 6. Shallow-CNN performance by Kernel Size

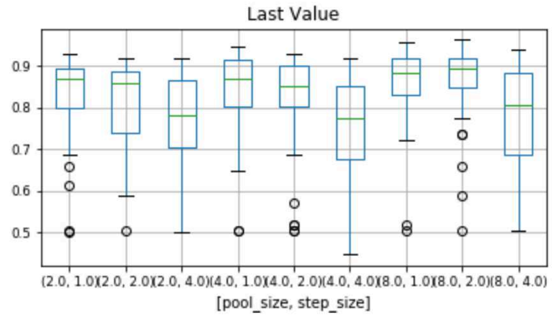


Figure 7. Shallow-CNN performance by Pooling Size and Strides

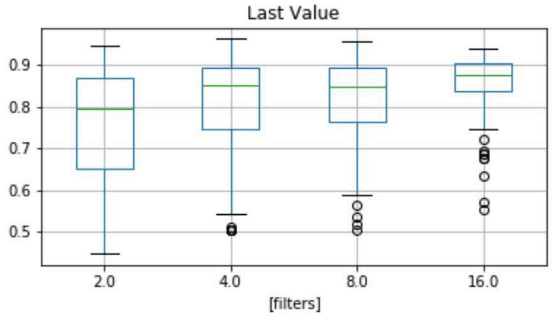


Figure 8. Shallow-CNN performance by Number of Filters used

not improved the performance as it did for the baseline classifiers. By looking at Figure 10, which represents the performance of the same configuration on the different features map, we can nevertheless confirm the relevance of the information it conveys, as we notice the learning curves of the approaches using the Color feature start faster. The networks trained only from the RGB images has also higher variability, but if given enough epochs, reaches the same performance as when using Color. This suggests the CNN architecture was able to learn how to extract this information during the training without the need of human defined ad-hoc model.

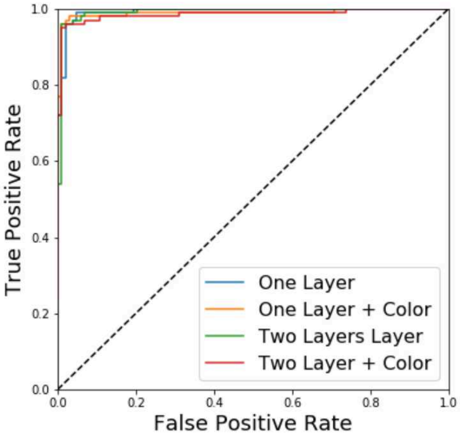


Figure 9. Roc Curves for Shallow approaches

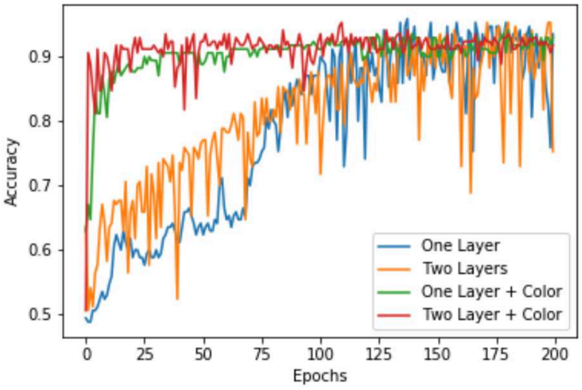


Figure 10. Learning curve of One and Two CNN layers model.

5.4. Performance on a lower resolution dataset

To evaluate the performance of the proposed approach in a lower resolution scenario, we applied the best configurations for 1-layer and 2-layer networks to the dataset used in [3]. This dataset is composed of 121 samples of pollen and 770 samples of non-pollen bearing bees. Image sizes are not fixed, but are of the order of 50x70 pixels (See Figure 11).

We trained and tested with the same amount of samples for each class. Performing 20 random splits using 80% for training and 20% for testing on each split.

The usage of as little as 50 samples for training had a detrimental effect on the 1-layer shallow network and showed poor generalization. However as the number of samples raised, the results outperformed the approach using VLAD descriptors based on fixed MSIFT features reported in [3]. The 2-layers shallow network outperformed the two

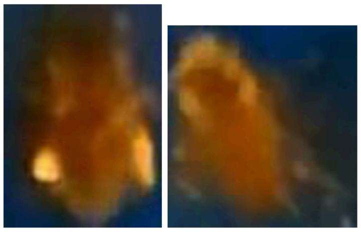


Figure 11. Pollen and Non Pollen bearing bees from dataset [3].

Training Size	1-Layer	2-Layers	VLAD [3]
50	72.4	87.9	87.42
100	91.6	92.5	90.46
200	94.6	95.9	92.14

Table 3. Performance of shallow networks compared to SVM classification of VLAD descriptors reported in [3], on the same dataset.

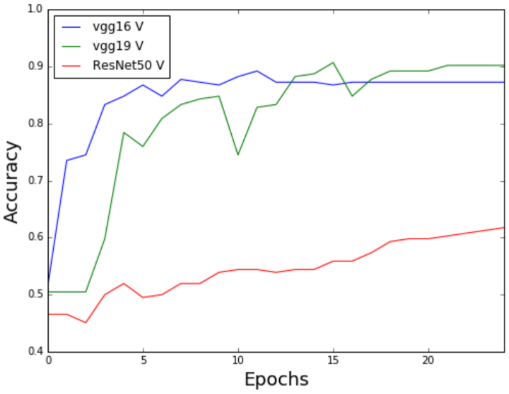


Figure 12. Learning curve of deep architectures

other approaches.

5.5. Deep models

By using the deep models based on VGG16, VGG19 and ResNet50, results were slightly lower than the best shallow architectures discussed before on our dataset. Because of to the pre-trained weights, faster learning was observed since only 5 epochs already showed a sustained accuracy over 88% , in contrast with the 70 epochs necessary to obtained the same results in a shallower network. Despite that advantage, due to the much larger number of parameters and the deeper architecture, only 25 epochs could be trained as training time already exceeded the time to train the shallow models from scratch.

The learning curves of the best architectures tested, shows that the use of the extracted feature dataset (Color) improves the performance and reduces the variability in the validation. Since very shallow models are considered, it takes up to 50 training steps to start showing results upper than 85% . However, since the number of trainable parameters are small the model require less time per training step than the other approaches tested.

VGG16, VGG19 and ResNet, were tested using 20 epochs on each one. Results shows that VGG19 have slightly better performance than VGG16 after 25 epochs, although it had a slower start. The maximum accuracy reached by VGG16 was 90%. As for ResNet50 results shows very poor performance in this application problem with a maximum of 60% accuracy reached. A possible issue may be the relatively small size of the images (180×300) of the dataset compared to the typical applications of these networks. Resolution reduction due to the pooling layers may cause oversimplification of the spatial information in the upper layers of the network, as was already noticed in the shallow network with the effect of the pooling factor.

5.6. Qualitative analysis

Figure 13 shows a selection of the classification results for the best shallow architecture (1-Layer): the samples that were correctly classified with the most confident prediction for both classes, as well as all the 7 samples incorrectly classified.

From this figure, it seems that an imprecise localization of the bee is an important factor that may lead to an incorrect classification, as 3 out of 7 miss predictions have a clearly bad centering or incomplete angle compensation. Since manual dataset annotation is a long and tedious task, such approximations are to be expected when building a large dataset, and are representative of issues that end users may face when refining the models for a particular experimental setup. Such misalignments are also expected from automatic bee detection algorithms. As a result of this study, this suggests that in future works, special attention should be paid to either refine alignment before classification, or improve the robustness to this issue, for instance by using data augmentation based on shifted images.

Another interesting observation is that the false negative samples have smaller pollen balls than the true positive samples, which confirms the intuition that the reduced size increases the difficulty of the classification.

6. Conclusion

We conclude from the results of this study that simple CNN architecture performed better than pre-trained recent CNN models and baseline classifiers for the task of recognizing pollen bearing bees. We also observed that feeding the CNN with task specific predefined features had an impact on the learning curve but without a strong impact on the final performance.

This study also uncovered that most incorrect predictions using the best CNN network had actual issues in the align-

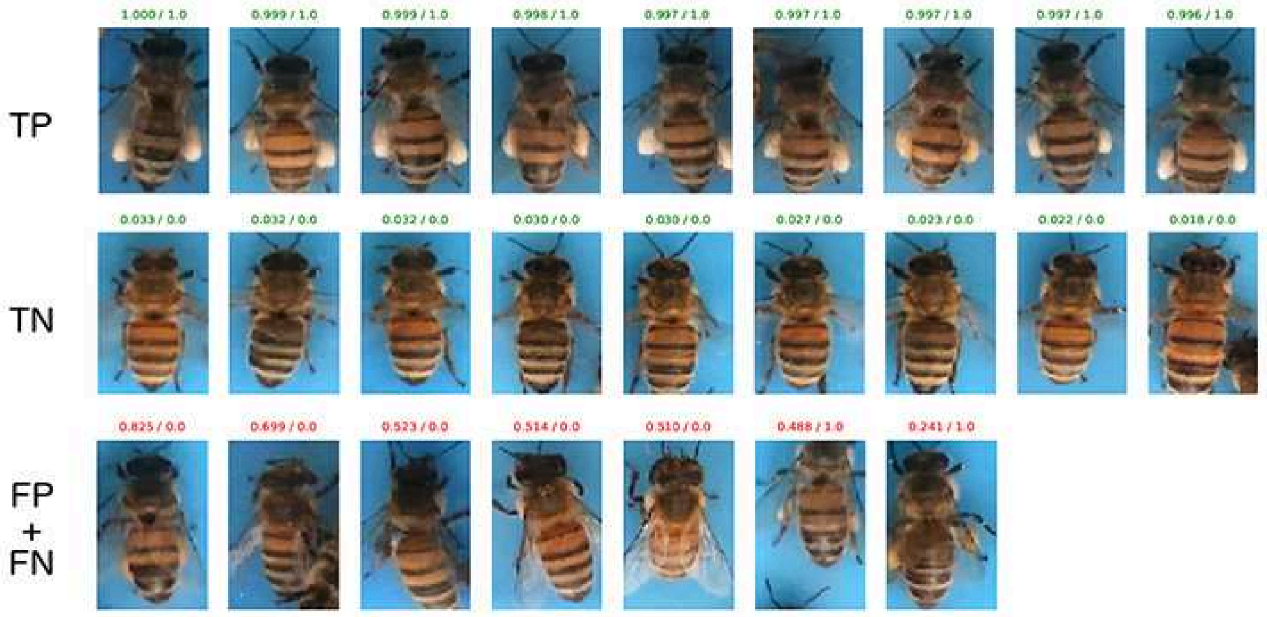


Figure 13. Selection of predicted results using the best 1-layer CNN. First row: most confident true positives. Second row: most confident true negatives. Third row: all 10 misclassified samples. The title of each image in of the form PredictedScore/TrueClass, with 0=NoPollen, 1=Pollen and a cut-off at 0.5.

ment of the bees, or presences of perturbations in the field of view. Such imprecisions in the detection of the bees are inevitable in the context of manually annotated training sets. For practical application of the pollen recognition on the field, it therefore appears important to integrate automatized management of misalignments to the annotation and recognition processes in order to reduce this source of errors.

Although deeper architectures may have the potential for improved performance, they did not actually perform better than shallower architectures on this dataset and involved longer computations. Indeed, by involving a large number of parameters, they typically require much larger datasets. In this respect, we point out that the size of the dataset used represents an upper limit to the investment that could be requested from an end user in terms of fully supervised annotation to refine the models for a specific system on the field.

In order to evaluate how to improve the performance and applicability in the field, it is therefore an interesting question for future work, how larger-scale datasets with good quality annotation could be created by leveraging the classifiers proposed in this study and automatized collection and validation of bee images.

7. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1707355 and 1633184. T. Giray acknowledges support from Puerto Rico Science and Technology Trust (PRSTRT; 2016-00161, 2017-00164). The authors would like to thank Stephanie Feliciano and Janpierre Aleman who helped with the acquisition at the UPR Agricultural Station of Gurabo of the videos used in the generation of the dataset and Christian Esteves, Emmanuel Nieves and Grace Rodriguez who helped with the annotation of the videos.

Simulations were run at the High-Performance Computing facility of the University of Puerto Rico, sponsored by the Puerto Rico INBRE grant P20 GM103475 from the National Institute for General Medical Sciences (NIGMS), a component of the National Institutes of Health (NIH) and awards 1010094 and 1002410 from the Experimental Program to Stimulate Competitive Research (EPSCoR) program of the National Science Foundation (NSF).

References

[1] OpenCV, open source computer vision library. https://opencv.org/.

[2] A. Avramović and V. Risojević. Block-based semantic classification of high-resolution multispectral aerial images. *Signal, Image and Video Processing*, 10(1):75–84, Jan 2016.

[3] Z. Babic, R. Pilipovic, V. Risojevic, and G. Mirjanic. Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-7:51–57, 2016.

[4] J. Campbell, L. Mummert, and R. Sukthankar. Video monitoring of honey bee colonies at the hive entrance. *Workshop*

Visual observation & analysis of animal & insect behavior, International Conference on Pattern Recognition, 2008.

[5] C. Chen, E.-C. Yang, J.-A. Jiang, and T.-T. Lin. An imaging system for monitoring the in-and-out activity of honey bees. *Comput. Electron. Agric.*, 89:100–109, Nov. 2012.

[6] M. Chica and P. C. Cervera. Standard methods for inexpensive pollen loads authentication by means of computer vision and machine learning. *CoRR*, abs/1511.04320, 2015.

[7] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[8] B. E. D. Frias, C. D. Barbosa, and A. P. Lourenço. Pollen nutrition in honey bees (apis mellifera): impact on adult health. *Apidologie*, 47(1):15–25, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[12] H. Jgou, F. Perronnin, M. Douze, J. Snchez, P. Prez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, Sept 2012.

[13] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods*, 10(1):64–67, Dec. 2012.

[14] E.-L. Kalman, F. Winquist, and I. Lundström. A new pollen detection method based on an electronic nose. *Atmospheric Environment*, 31(11):1715 – 1719, 1997.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[16] S. H. Landsmeer, E. A. Hendriks, L. A. de Weger, J. H. Reiber, and B. C. Stoel. Detection of pollen grains in multifocal optical microscopy images of air samples. *Microscopy Research and Technique*, 72(6):424–430, 2009.

[17] A. E. Lundie. The flight activities of the honeybee. 1925.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] S. Schurischuster, S. Zambanini, M. Kampel, and B. Lamp. Sensor study for monitoring varroa mites on honey bees (apis mellifera). *Proceedings of the Visual observation and analysis of Vertebrate And Insect Behavior (VAIB) Workshop*, 2016.

[20] S. Serralles and R. Mégret. Computer assisted annotation system for study of insect behavior. In *Poster presented at Caribbean Celebration of Women in Computing at Puerto Rico (CCWiC)*, Mayagüez, Puerto Rico, apr 2016.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] G. J. Tu, M. K. Hansen, and P. A. Per Kryger. Automatic behaviour analysis system for honeybees using computer vision. *Computer and Electronics in Agriculture*, 122, 2016.

[23] S. van der Walt, J. L. Schónberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.