Deep Feature-based Classifiers for Fruit Fly Identification (Diptera: Tephritidae)

Matheus M. Leonardo*, Tiago J. Carvalho†, Edmar Rezende‡, Roberto Zucchi§, Fabio A. Faria*,
*Institute of Science and Technology - Universidade Federal de São Paulo (UNIFESP), São José dos Campos, Brazil
Email: matheus.macedo.leonardo@gmail.com and ffaria@unifesp.br
†Federal Institute of São Paulo, Campinas, Brazil
Email: tiagojc@ifsp.edu.br
‡University of Campinas, Campinas, Brazil
Email: edmar.rezende@gmail.com
§Luiz de Queiroz College of Agriculture, University of Sao Paulo, Piracicaba, Brazil
Email: rzucchi@usp.br

***Abstract*—Fruit flies has a big biological and economic importance for the farming of different tropical and subtropical countries in the World. Specifically in Brazil, third largest fruit producer in the world, the direct and indirect losses caused by fruit flies can exceed USD 120 million/year. These losses are related to production, the cost of pest control and export markets. One of the most economically important fruit flies in the America belong to the genus *Anastrepha*, which has approximately 300 known species, of which 120 are recorded in Brazil. However, less than 10 species are economically important and are considered pests of quarantine significance by regulatory agencies. The extreme similarity among the species of the genus *Anastrepha* makes its manual taxonomic classification a nontrivial task, causing onerous and very subjective results. In this work, we propose an approach based on deep learning to assist the scarce specialists, reducing the time of analysis, subjectivity of the classifications and consequently, the economic losses related to these agricultural pests. In our experiments, five deep features and nine machine learning techniques have been studied for the target task. Furthermore, the proposed approach have achieved similar effectiveness results to state-of-art approaches.**

I. Introduction

Anastrepha is the most diverse genus in the Americas, with over 300 species known in the tropics and subtropics regions. In Brazil, 120 species are recorded [1], but less than 10 species are considered as agricultural pests.

Species of *Anastrepha* are known as fruit flies, as females lay their eggs in healthy fruit, and larvae feed inside the fruit. Later on, larvae leave the fruit and pupate in the soil and, after some days, the adults emerge.

Due to the damage caused by larvae in commercialized fruits, some species of *Anastrepha* are of significant economic importance. Furthermore, some *Anastrepha* species are also economically important as quarantine pests as they hinder the international trade of fruits. Consequently, an accurate identification of quarantine pests of fruits for exportation is highly relevant, due to the strict trade quarantines to prevent their spread.

However, identification of cryptic species (closely related and morphologically similar) is problematic. Some species of fruit fly quarantine pests comprises complexes of cryptic species, and it is difficult to delimit the species boundaries within these complexes.

Identification of cryptic species of agricultural importance is a huge challenge, especially those with a quarantine status. On the other hand, misidentification are problematic for quarantine restrictions, control programs (e.g. integrated pest management) and basic studies (biology, geographical distribution, plant hosts, and natural enemies).

Anastrepha fraterculus, the South America Fruit Fly, is a complex of species in the Americas. It is a major pest only in some areas of its geographical distribution from Mexico to northern Argentina. Another complex with species economically important is the *Anastrepha* obliqua complex. Species of these two complexes are widely distributed in Brazil and the real challenge is to find out which species of these complexes are actually agricultural pests.

Several techniques (e.g. crosses, morphometric and molecular analyses [2], [3]) have been used for clarifying the identity of the species within these complexes. Recently, image analysis techniques were used to identifying three species of *Anastrepha*, two of them of quarantine importance ([4]–[6]).

In insect identification literature, more specifically in identification of fruit flies of the genus *Anastrepha*, different methods rises in the last years [2], [3]. Martineau et al. [7] compiled the most recent works, an overall forty four works, based on image processing and machine learning techniques in a survey.

Different literature works associating image processing and machine learning have been proposed to solve problems that involves wild life. Digital Automated Identification SYstem (DAISY) classifies spiders, pollen grain, and butterfly through a semi-automated classification approach based on principal component analysis (PCA) [8]. SPecies IDentified Automatically (SPIDA-web) identifies Australian spiders to distinguish 121 species using Daubechies 4 wavelet function [9]. Automatic Bee Identification System (ABIS) recognizes bee species of genus *Bombus*, *Colletes*, and *Andrena* through the use of support vector machine (SVM) and kernel discriminate analysis techniques [10], [11]. DAIIS tool classifies a sample of

120 owlflies (Neuroptera: Ascalaphidae) based on their wing outlines using Elliptic Fourier coefficients and SVMs [12], [13]. In [14], an insect recognition system that combines different visual properties such as color, texture, shape, scale-invariant feature transform (SIFT) [15] and histogram of oriented gradients (HOG) [16] features has been created through the use of multiple-task sparse representation and multiple-kernel learning (MKL) techniques [17]. However, a very important fact needs to be pointed out as motivation of this work. Despite the biological and economic importance of the Tephritidae family (fruit fly), only three papers have been found in the literature. The first one adopted a successful framework of classifier selection and fusion [18], which combines many global image descriptors and machine learning techniques for a multimodal classification approach, using image of wings and aculei of three species of the *fraterculus* group: *A. fraterculus* (Wied.), *A. obliqua* (Macquart) and *A. sororcula* Zucchi [4]. In the second work, a mid-level representation based on BossaNova approach [19] has been adopted to improve the effectiveness results achieved in previous experiments with global image descriptors [6]. Finally, in the third work, the authors proposed a sparse representation using SIFT features densely sampled as input for two machine learning techniques (multi-layer Max-pooling ScSPM [20] and linear SVM [10]). It has performed experiments with three unreported genus and twenty species [21].

In this work, we take advantage of well known deep learning architectures, designed to object recognition problems, to extract relevant features for fruit fly identification. Associating these features with simple classifiers, we are able to achieve an effective accuracy on fruit fly classification without necessity of laborious hand-crafted features extraction. Furthermore, we compare the effectiveness of our results with different machine learning techniques using those hand-craft features to support the development of a real-time system for fruit fly identification of the genus *Anastrepha*. We believe that this system can be a good solution for more precise identification, allowing time reduction, costs in performing and assisting the few specialists in their tasks.

The main contributions of this work are:

- Decrease of complexity in features engineering process when compared with state-of-the-art methods;
- Proposition of a new approach for fruit fly identification task of the genus *Anastrepha* based on a deep Convolutional Neural Networks (CNN) model combined with a transfer learning approach;
- An effectiveness analysis among the best deep learning architecture and the state-of-art approaches existing in the literature;

The remainder of this paper is organized as follows. Section II describes different feature extraction techniques used in this works for insect identification task. Section III shows the experimental protocol we devised to validate the work while Section IV discusses the results. Finally, Section V concludes the paper and points out future research directions.

II. Materials and Methods

The typical image classification pipeline is composed of the three following steps: (i) local visual feature extraction, which extracts information directly from the image pixels, (ii) mid-level feature extraction, which makes the representation more general, aggregating abstraction to the model, and (iii) supervised classification, a machine learning technique allowing the extraction of a general model from the data.

Usually, stages (i) and (ii) are strongly depend from a expensive hand-craft features extraction process, where an expert looks for the best algorithm (or combination of algorithms) that better represent the problem samples.

We take advantage of robust Deep Neural Networks (DNN) architectures, to extract a set of features which are very specific for the problem, avoiding the laborious hand-craft features extraction step.

In a nutshell, our method uses a DNN architecture (without the top layer responsible by classification) to extract features, named *bottleneck features*. Bottleneck term refers to a topology of a neural network where the hidden layer has significantly lower dimensionality than the input layer, assuming that such layer — referred to as the bottleneck — compresses the information needed for mapping the neural network input to the neural network output, increasing the system robustness to noise and overfitting. Conventionally, bottleneck features are the output generated by the bottleneck layer [22]. These bottleneck features are used to train a binary classifier.

A. Deep Learning Architectures

In the proposed approach, we have used deep convolutional neural networks based on VGG (VGG16 and VGG19), GoogLeNet (Inception V3 and Xception) and ResNet (ResNet-50) architectures, shown in Figure 1, pre-trained for object detection task on the ImageNet dataset.

1) VGG Architecture: The VGG networks [23] with 16 layers (VGG16) and with 19 layers (VGG19) were the basis of the Visual Geometry Group (VGG) submission in the ImageNet Challenge 2014, where the VGG team secured the first and the second places in the localization and classification tracks respectively.

The VGG architecture is structured starting with five blocks of convolutional layers followed by three fully-connected layers. Convolutional layers use 3×3 kernels with a stride of 1 and padding of 1 to ensure that each activation map retains the same spatial dimensions as the previous layer. A rectified linear unit (ReLU) activation is performed right after each convolution and a max pooling operation is used at the end of each block to reduce the spatial dimension. Max pooling layers use 2×2 kernels with a stride of 2 and no padding to ensure that each spatial dimension of the activation map from the previous layer is halved. Two fully-connected layers with 4096 ReLU activated units are then used before the final 1000 fully-connected softmax layer.

A downside of the VGG16 and VGG19 models is that they are more expensive to evaluate and use a lot of memory and

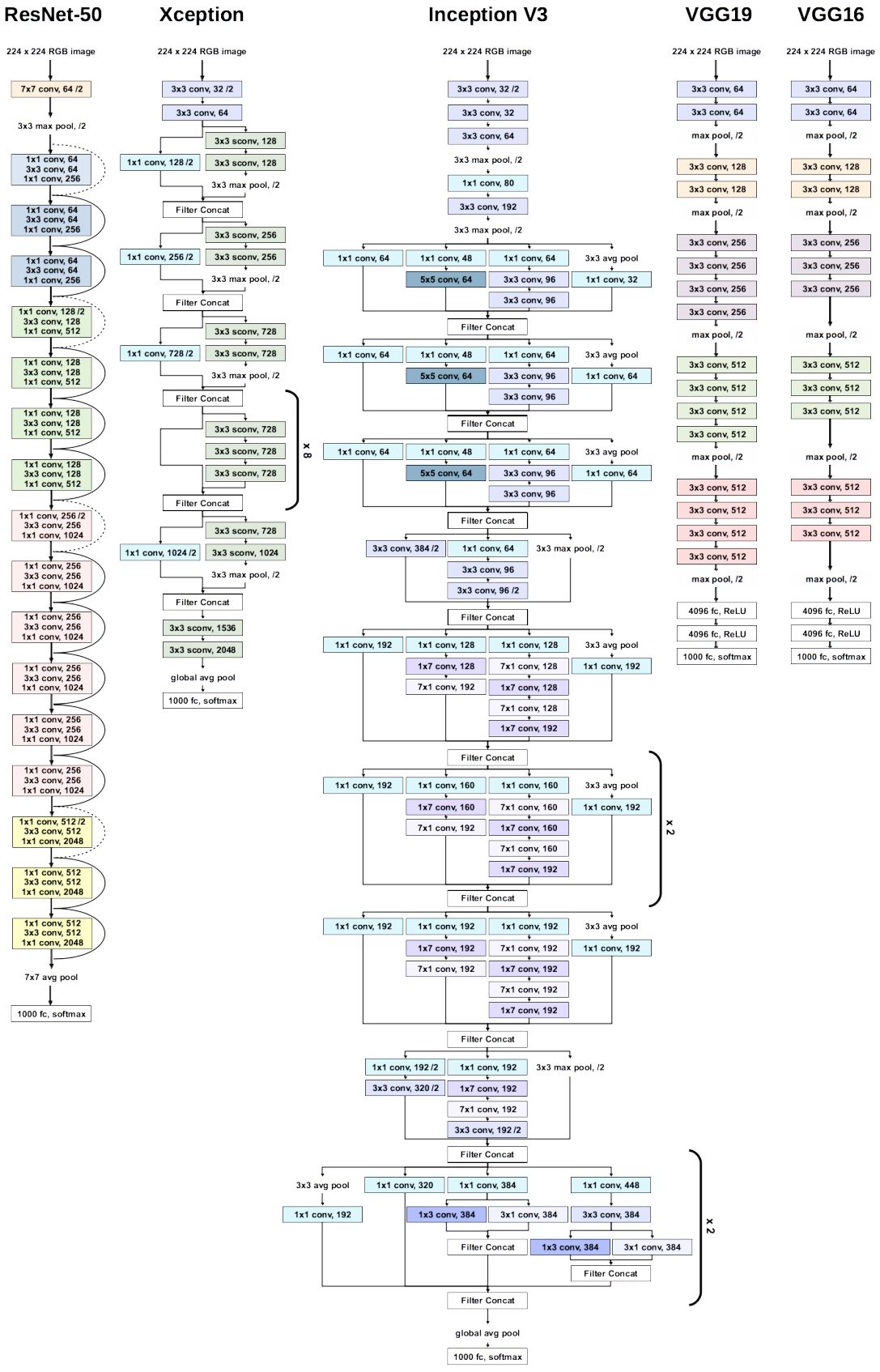

Fig. 1. VGG16, VGG19, Inception V3, Xception and ResNet-50 architectures.

parameters. VGG16 has approximately 138 million parameters and VGG19 has approximately 143 million parameters. Most of these parameters (approximately 100 million) are in the first fully-connected layer, and it was since found that these fully-connected layers could be removed with no performance downgrade, significantly reducing the number of necessary parameters.

2) GoogLeNet Architecture: The GoogLeNet architecture was introduced as GoogLeNet (Inception V1), later refined as Inception V2 and recently as Inception V3 [24].

While Inception modules are conceptually convolutional feature extractors, they empirically appear to be capable of learning richer representations with less parameters. A traditional convolutional layer attempts to learn filters in a 3D space, with 2 spatial dimensions (width and height) and a channel dimension. Thus, a single convolution kernel is tasked with simultaneously mapping cross-channel correlations and spatial correlations.

The idea behind the Inception module is to make this process easier and more efficient by explicitly factoring it into a series of operations that would independently look at cross-channel correlations and at spatial correlations.

The Xception architecture [25] is an extension of the Inception architecture which replaces the standard Inception modules with depthwise separable convolutions. Instead of partitioning input data into several compressed chunks, it maps the spatial correlations for each output channel separately, and then performs a 1×1 depthwise convolution to capture cross-channel correlation. This is essentially equivalent to an existing operation known as a "depthwise separable convolution", which consists of a depthwise convolution (a spatial convolution performed independently for each channel) followed by a pointwise convolution (a 1×1 convolution across channels). We can think of this as looking for correlations across a 2D space first, followed by looking for correlations across a 1D space. Intuitively, this 2D + 1D mapping is easier to learn than a full 3D mapping.

Xception slightly outperforms InceptionV3 on the ImageNet dataset, and vastly outperforms it on a larger image classification dataset with 17,000 classes. Most importantly, it has a similar number of parameters as Inception V3, implying a greater computational efficiency. Xception has 22,855,952 trainable parameters while Inception V3 has 23,626,728 trainable parameters.

3) ResNet Architecture: Residual Networks (ResNets) [26] are deep convolutional networks where the basic idea is to skip blocks of convolutional layers by using shortcut connections to form blocks named residual blocks. These stacked residual blocks greatly improve training efficiency and largely resolve the degradation problem present in deep networks.

In ResNet-50 architecture, the basic blocks follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled. The downsampling is performed directly by convolutional layers that have a stride of 2 and batch normalization is performed right after each convolution and before ReLU activation.

When the input and output are of the same dimensions, the identity shortcut is used. When the dimensions increase, the projection shortcut is used to match dimensions through 1×1 convolutions. In both cases, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2. The network ends with a 1,000 fully-connected layer with softmax activation. The total number of weighted layers is 50, with 23,534,592 trainable parameters.

B. Transfer Learning

Transfer learning consists in transferring the parameters of a neural network trained with one dataset and task to another problem with a different dataset and task [27].

Many deep neural networks trained on natural images exhibit a curious phenomenon in common: on the first layers they learn features that appear not to be specific to a particular dataset or task, but general in that they are applicable to many datasets and tasks. Features must eventually transition from general to specific by the last layers of the network. When the target dataset is significantly smaller than the base dataset, transfer learning can be a powerful tool to enable training a large target network without overfitting.

In the proposed approach, we have used VGG16, VGG19, Inception V3, Xception and ResNet-50 as the base models, pre-trained for object detection task on the ImageNet dataset. The ImageNet is a public dataset containing 1.28 million natural images of 1,000 classes.

III. Experimental Settings

We now discuss the experimental setup, including the dataset, details on the image acquisition, and machine learning techniques used in this work.

A. Dataset

In our experiments, we have used lab-based samples [7] of the specimens *A. fraterculus*, *A. obliqua* and *A. sororcula* from the collection of the Instituto Biológico of São Paulo (e.g., Figures 2(a-c)). Specimens have been collected through McPhail-type traps (Figure 2b) and reared flies from fruit as well.

(a) (b) (c)

Fig. 2. (a) A fruit fly example (drawing) [28]; (b) a McPhail-type trap; and (c) a fruit fly laying eggs. Extracted from [4].

The dataset used in this work is composed of 301 images (resolution 2560×1920) and divided into three different categories: *A. fraterculus* (100), *A. obliqua* (101), and *A. sororcula* (100). It consists of pictures of specimens reared from samples

A. fraterculus *A. obliqua* *A. sororcula*

Fig. 3. Example of wings of each specie studied. Extracted from [4].

of fruit trees in experimental and commercial orchards in the state of São Paulo, Brazil, stored in the Department of Entomology and Acarology ESALQ, Piracicaba, SP, Brazil and in the Biological Institute, Campinas, SP, Brazil.

Figure 3 shows examples of the three species used in this work.

B. Machine Learning Techniques

We have used nine different machine learning techniques: Decision Tree (DT), k-Nearest Neighbor (kNN) with $k = \{1, 3, 5, 7\}$, Multiple Layer Perceptron (MLP), Naïve Bayes (NB), Stochastic Gradient Descent (SGD) and Support Vector Machine (SVM) using linear kernel.

The proposed methods have been implemented using Python 3.5, Keras 2.0.3[1], TensorFlow 1.0.1[2] and Scikit-Learn[3]. All performed tests have been executed in a machine with an Intel(R) Xeon(R) CPU E5-2620 2.00GHz processor with 96GB of RAM and two Nvidia Titan Xp GPUs.

IV. Results and Discussion

We have performed two experiments with objective to support a system for fruit fly identification. Firstly, a comparative study among five different deep features and nine machine learning techniques has been performed. Next, a comparison among the best tuple (feature + learning technique) against two of the best literature [4], [6] approaches. For all effectiveness experiments, the average accuracies in the 5-fold cross-validation protocol have been computed (three partitions for training set, one for validation set, and one for test set).

A. Effectiveness Analysis

In this section, we have performed a comparative study among five deep features based on deep learning architectures (Inception, ResNet, VGG16, VGG19, and Xception).

Table I shows effectiveness results for all deep features and machine learning techniques. We can observe that bottleneck features extracted using VGG16 architecture has achieved eight of the best effectiveness results among nine released learning techniques (in blue). Furthermore, we can observe that Support Vector Machines (SVM) technique using linear kernel has achieved the best effectiveness results for all of the five deep learning architectures applied for features extraction (in gray cell) released in this work. In addition, SVM technique using VGG16 feature was the best tuple (feature + learning technique) performed in this work with 95.68% of average accuracy (in blue text and gray cell).

B. The Best Approaches of the literature

The second experiment performed compares the best tuples (deep feature + learning technique) of the previous experiment (Inception+SVM, ResNet+SVM, VGG16+SVM, VGG19+SVM, and Xception+SVM) against state-of-the-art methods (LCH+SVM [4] and F-SIFT+MLP [6]). LCH+SVM is a support vector machine technique with polynomial kernel using a generic color image descriptor called Local Color Histogram [29]. F-SIFT+MLP is a multiple layer perceptron technique using a Bow-of-Words (BoW) representation based on BossaNova [19] with keypoint detector FAST [30] and local feature detector SIFT [15]

Figure 4 shows the effectiveness results among the best tuples (feature + learning technique) and the best baseline existing in the literature. Although VGG16+SVM (in blue) has achieved the best mean accuracy (95.68%), when we compute the confidence interval with significance level of 0.05, it is possible to observe that there is no statistically significant difference among five deep learning approaches and the two baselines (in red) from the literature LCH+SVM [4] (93.50%) and F-SIFT+MLP [6] (94.67%). However, it is very important to note that LCH+SVM approach has achieved good effectiveness results by extracting color properties from enhanced image (e.g., segmentation and morphological operations). Therefore, this color-based histogram approach may not be able to be used in real-time systems, unlike the other two compared approaches (F-SIFT+MLP and VGG16+SVM).

Moreover, in [6], we could see that the F-SIFT+MLP approach need to find the best tuple (keypoint detector and feature extractor), as well as, to perform the BoW process for finally achieving good results in the fruit fly identification task. This fact does not occur in our proposed approach based on deep learning techniques, since we only need to find the architecture configuration that best describes the data of the target application.

In relation to the properties extracted by compared approaches, we can also verify differences between them. LCH deals with color properties, F-SIFT works with keypoint detector that are gradient dependent (shapes, edges, and corners) and deep learning works with the combination of different

[1]https://keras.io

[2]https://www.tensorflow.org

[3]http:http://scikit-learn.org/stable/ (As of January, 2018)

TABLE I

EFFECTIVENESS RESULTS (IN %) AMONG FIVE DEEP LEARNING ARCHITECTURES AND NINE MACHINE LEARNING TECHNIQUES FOR A 5-FOLD CROSS-VALIDATION PROTOCOL. IN BLUE ARE THE BEST IMAGE DEEP FEATURES FOR EACH MACHINE LEARNING TECHNIQUE. IN GRAY CELL ARE THE BEST MACHINE LEARNING TECHNIQUES FOR EACH DEEP LEARNING ARCHITECTURE USED FOR BOTTLENECK FEATURES EXTRACTION.

Deep Features	**Machine Learning Techniques**								
	DT	**KNN1**	**KNN3**	**KNN5**	**KNN7**	**MLP**	**NB**	**SGD**	**SVM**
Inception	57.83	71.09	70.09	70.09	66.77	87.7	54.13	88.03	**88.37**
ResNet	65.43	75.08	77.08	77.74	80.06	89.03	73.40	89.70	**90.36**
VGG16	**75.75**	84.39	**89.02**	**87.71**	**87.73**	**95.02**	**75.75**	**93.36**	**95.68**
VGG19	72.1	**84.72**	82.08	80.09	81.39	92.68	67.13	91.35	**94.34**
Xception	51.48	60.79	56.12	55.77	55.44	68.33	51.82	78.41	**78.74**

properties (e.g., corners, edge/color conjunctions, and texture [31]). In a real system, if some visual property is deficient, it is believed that only the deep learning approach might be able to extract features of the available data and to achieve some good result.

Finally, we can use all of three approaches (LCH, F-SIFT, and VGG16) working together and complementary so that the final effectiveness results become more robust. Another idea is to mix concepts BoW and feature deep like in paper [32].

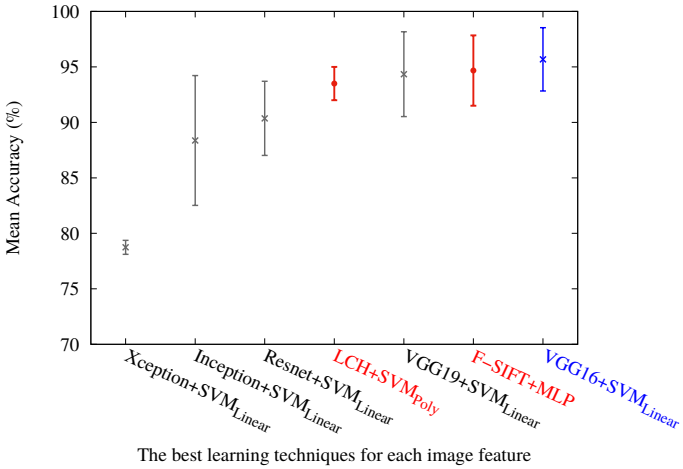

Fig. 4. Effectiveness results for each image feature with 95% confidence interval (CI), i.e, a significance level of 0.05. In blue is SVM_{Linear} using VGG16 feature that has achieved the best mean accuracy.

V. CONCLUSION

In this work, we proposed a method that take advantage of deep learning architectures associated with transfer learning approach for fruit fly identification task. Extracting bottleneck features, we characterize images of three species of the genus Anastrepha. Then we use a shallow classifier to detect the correct class of the image.

Five well known deep architectures and nine machine learning techniques have been compared, achieving a better accuracy of 95.68% when associating VGG16+SVM.

When compared against state-of-art approaches, our method perform better in evaluated images, without any additional image processing enhancement operation. This fact is very important for constructing a real-time system that helps fighting back pests in agriculture.

Unlike the mid-level representation approach (F-SIFT) that need to choose the best combination keypoint detector and feature extractor, adding the BoW process.

The use of deep learning techniques applied to fruit fly identification with transfer learning approach is quite new. Mainly because of the lack of samples to train a deep architecture from scratch, this approach brings a feasible way to use CNNs in the target task, eliminating the necessity of generate hand crafted features.

As future work, we intend to perform experiments with species and learning techniques as classifier ensemble. Other work might be the development of a mobile system to assist the few experts from the biology area on their field works.

VI. ACKNOWLEDGMENT

The author thanks the support of scientific funding agency CNPq through the Universal Project (grant #408919/2016-7) and the support of NVIDIA Corporation with the donation of the GPUs used for this research.

REFERENCES

[1] Zucchi, R. A., "Fruit flies in Brazil: *Anastrepha* species and their host plants and parasitoids," http://www.lea.esalq.usp.br/anastrepha/, 2008.

[2] Z. Bomfim, K. Lima, J. Silva, M. Costa, and R. Zucchi, "A morphometric and molecular study of *Anastrepha pickeli* Lima (Diptera: Tephritidae)," *Neotropical Entomology*, vol. 40, pp. 587–594, 2011.

[3] ——, "Morphometric and Molecular Characterization of *Anastrepha* species in the *spatulata* Group (Diptera, Tephritidae)," *Annals of the Entomological Society of America*, vol. 5, pp. 893–901, 2014.

[4] F. Faria, P. Perre, R. Zucchi, L. Jorge, T. Lewinsohn, A. Rocha, and R. da S. Torres, "Automatic identification of fruit flies (diptera: Tephritidae)," *Journal of Visual Communication and Image Representation*, vol. 25, no. 7, pp. 1516–1527, 2014.

[5] P. Perre, F. A. Faria, L. R. Jorge, A. Rocha, R. S. Torres, T. Lewinsohn, and R. A. Zucchi, "Toward an automated identification of anastrepha fruit flies in the fraterculus group (diptera, tephritidae)," *Neotropical Entomology*, vol. 0, pp. 1–5, 2016.

[6] M. M. Leonardo, S. Avila, R. A. Zucchi, and F. A. Faria, "Mid-level image representation for fruit fly identification (diptera: Tephritidae)," in *2017 IEEE 13th International Conference on e-Science (e-Science)*, Oct 2017, pp. 202–209.

[7] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognition*, vol. 65, no. C, pp. 273–284, 2017.

[8] A. T. Watson, M. A. O'Neill, and I. J. Kitching, "A qualitative study investigating automated identification of living macrolepidoptera using the digital automated identification system (daisy)," *Systematics & Biodiversity*, vol. 1, pp. 287–300, 2003.

[9] K. Russell, M. Do, J. Huv, and N. Platnick, "Introducing spida-web: wavelets, neural networks and internet accessibility in an image-based automated identification system," *Systematics Association*, vol. 74, pp. 131–152, 2007.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[11] T. Arbuckle, S. Schrder, V. Steinhage, and D. Wittmann, "Biodiverssity informatics in action: identification and monitoring of bee species using abis." *International Symposium for Environmental Protection*, pp. 425–430, 2001.

[12] H. P. Yang, C. S. Ma, H. Wen, Q. B. Zhan, and X. L. Wang, "A tool for developing an automatic insect identification system based on wing outlines," in *Scientific Reports*, vol. 5, 2015, p. vol. 1.

[13] S. Chen, P. Lestrel, W. Kerr, and J. McColl, "Describing shape changes in the human mandible using elliptical fourier functions," *European Journal of Orthodontics*, vol. 22, no. 3, p. 205, 2000.

[14] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, J. Xia, and P. Chen, "Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning," *Computers and Electronics in Agriculture*, vol. 119, no. C, pp. 123–132, 2015.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[17] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[18] F. A. Faria, J. A. dos Santos, A. Rocha, and R. da S. Torres, "A framework for selection and fusion of pattern classifiers in multimedia recognition," *Pattern Recognition Letters*, vol. 39, pp. 52–64, 2014.

[19] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo, "BOSSA: Extended BoW formalism for image classification," in *IEEE International Conference on Image Processing*, 2011, pp. 2909–2912.

[20] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1794–1801.

[21] A. Lu, X. Hou, C. Lin, and C.-L. Liu, "Insect species recognition using sparse representation," in *Proceedings of the British Machine Vision Conference*, 2010, pp. 1–10.

[22] B. Zhang, L. Xie, Y. Yuan, H. Ming, D. Huang, and M. Song, "Deep neural network derived bottleneck features for accurate audio classification," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–6.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint arXiv:1610.02357*, 2016.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[28] Plantwise, "Empowering farmers, powering research — delivering improved food security," http://www.plantwise.org/, 2013.

[29] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[30] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006, pp. 430–443.

[31] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks*. Cham: Springer International Publishing, 2014, pp. 818–833.

[32] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto, "Bags of local convolutional features for scalable instance search," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ser. ICMR '16, New York, NY, USA, 2016, pp. 327–331.