Pest detection and classification to reduce pesticide use in fruit crops based on deep neural networks and image processing

Agustina Suárez*, Romina Soledad Molina*†, Giovanni Ramponi† Ricardo Petrino*,
Luciana Bollati‡, Daniel Sequeiros‡
*Universidad Nacional de San Luis — LEIS, Departamento de Electrónica
Av. Ejército de los Andes 950, D5700, San Luis, Argentina
E-mail: (A.S.) aguss.sl1463@gmail.com - (R.P.) rpetrinodlv@gmail.com
†Università degli Studi di Trieste — IPL, Dipartimento di Ingegneria e Architettura
Piazzale Europa, 1, 34127 Trieste TS, Italy
E-mail: (R. M.) rominasoledad.molina@phd.units.it - (G. R.) ramponi@units.it
‡Nectras
Av Belgrano 758, Sunchales, Santa Fe, Argentina

***Abstract*—The aim for organic farming is obtaining food of the highest quality, avoiding synthetic chemicals, protecting the environment and preserving the fertility of the land. In this context, effective pest control allows to reduce yield loss and pesticides application producing pollution-free vegetables. In fruit crops, Carpocapsa is the main pest present in pear, apple, walnut and quince trees. This insect produces irreversible damage to the fruit, since the larvae feed the seeds inside the fruit. In this paper, we present automatic pest detection and classification in the context of fruit crops based on image processing and Deep Neural Networks, employing an image collection obtained from in-field traps. Due to the limited size of the data set, we perform data augmentation to increase the number of images for training, to prevent over-fitting and to improve the deep neural network learning rate. Results showed an overall accuracy of 94.8%, while precision and recall scores for the class related with the moth were around 97.2% and 93.6% respectively, demonstrating the efficacy of this type of classifier proposed for pest detection. An inference time of 40 ms per image for the deep neural network classifier has been reached.**

***Index Terms*—image processing, deep neural networks, CNN, classification, pest detection**

I. Introduction

Nectras [1] is an integral solution for real time pest detection, evaluation and control, which combines IoT traps with spraying drones through machine learning algorithms. Fruit crops such as grapes, apples and pears are affected by various diseases and pests during their growth process. If the control is not timely, it will lead to a reduction of the soil or even to the loss of the harvest. Identify accurately insect pests and effectively control them is very important to help fruit growers improve fruit yield.

Pest prevention is the best way to reduce soil loss and reduce pesticide application to produce pollution-free vegetables. In addition, the application of agrochemicals and pesticides has a poor control effect and is easy to cause environmental pollution, which leads to excessive pesticide residues on vegetables and a higher pest resistance, lower efficiency, higher cost, stronger subjectivity, lower precision and punctuality.

An approach to solve this problem is a manual method to quantify the number of insects, but this solution is time-consuming and susceptible to errors. Nevertheless, the application of information technology provides original methods and ideas for the identification of insect pests. Also, using automatic and efficient image recognition methods could reduce cost and improve fumigation accuracy.

The particular pest to be detected is Carpocapsa (Cydia Pomonella), also known as worm or moth. The difficulty to control the lepidoptera threatens regional economies. Carpocapsa is the main pest of the pear, apple, walnut and quince tree. This insect produces irreversible damage to the fruit since its larvae (juvenile state of the insect) feed inside the fruit reaching the seeds.

In the context of pest detection, several works have been presented to tackle this problem through image processing techniques, aiming to generate an automatic detection and classification framework. The work presented in [2] proposed an IoT Smart Trap to detect a pest in coffee plantations based on image binarization, morphological operations and contour searching. Ramalingam et al. [3] employed Faster RCNN (Region-based Convolutional Neural Networks) and Residual Neural Networks 50 (ResNet50) for pest detection, using images collected from different online sources. Authors in [4] implemented a Deep Convolutional Neural Network architecture with a data set composed by 200 images, using a data augmentation technique to increase the data base, obtaining accuracy rates of detection and classification in 84% and 86% respectively. They remark that Single Shot Detector (SSD) object detection algorithm was optimal for the case of study.

Another approach was presented in [5] using IP102 insect pest data set. AlexNet, GoogleNet and SqueezeNet were chosen as base networks to perform transfer learning, due to the smaller amount of layers compared to other architectures. The results exhibited that the model based on AlexNet achieved the highest testing accuracy at 89.33%.

Moth detection was performed in [6] using a sliding window-based detection pipeline with a ConvNet as classifier. Hong et al. [7] performs the moth detection in pheromone traps using deep learning techniques. Despite the good results, the images used for training were not real-time remote sensing images and the traps were photographed in the laboratory.

Most of the developments devoted to this type of applications are based on transfer learning, due to the reduced amount of images for training obtained in real climatic conditions. On the other side, it is not an easy task to access real data sets, making it difficult to verify and validate these techniques.

In this work, we present automatic pest detection on fruit crops based on image processing and Deep Neural Networks (DNN), employing an image collection obtained from in-field traps provided by Nectras, in real environment conditions. Through pre-processing techniques, we are able to detect the objects in the images obtained from the IoT traps, generating the data base for training the DNN. We use data augmentation to extend the data set, preventing over-fitting and improving the DNN performance.

The paper is organized as follows: section II presents the overall system for data acquisition, section III exposes the methodology to perform object detection and classification. Results are discussed in section IV and finally, conclusions are presented in section V.

II. Overall system

The expansion of technological developments has made possible the arrival of data processing in many areas of life. The growth of organic ecology is advancing in favor of the environment and the care of the land and crops, to produce pollution-free vegetables. In this direction, Internet of Things (IoT) traps have been developed to help pest control and detection to improve food safety. This technology connects devices and sensors, collects and stores the generated data and allows new understanding of pest trends and their characteristics.

The pest to be detected through this research is the moth called Carpocapsa, which is captured using intelligent IoT traps designed by Nectras. The sensors present in the electronic traps allow constant monitoring of the insects existing in the field. The capture surface is gummed with pheromones and its dimension is 16x18cm, which allows capturing up to 150 moths per trap. Each pheromone lure lasts approximately 5 to 6 weeks. Due to its specific pheromone-based attractants, male insects are attracted and stuck to the gummed bottom of the trap, however it has no effect on other insects. Regarding its distribution, 1 trap is placed every 3 to 5 trees approximately, which is hanging from a branch at 1.5-2m height. Fig. 1 shows the IoT trap placed in the operating environment and its inside is depicted in Fig. 2.

Fig. 1. IoT trap in the operating environment.

Fig. 2. IoT trap - Inside view.

An Arduino OV2640 2M camera (Maximum image size and sampling rate: 1600×1200@15fps, Transfer rates: SVGA@30fps and CIF@60fps, 352x240) placed at the top of the IoT trap captures the images periodically, during the morning and in the afternoon, since at these times the moth makes its flight, allowing constant control of the insects.

Digital images are sent to the cloud through a custom hardware developed by Nectras and stored in .jpg format,with a resolution of 800x600 and 1600x1200 pixels in the RGB color space. Then, the images are downloaded and processed through dedicated software. From the user side, an application allows them to be informed of the status of the crop. When the system detects a predefined number of moths, it sends an alert to the user to proceed with the fumigation by the drone in the area where the IoT trap is located. If the suggestion is accepted, the platform sends the flight plan to the fumigator drone. By means of this methodology, the process is constrained in location and time, decreasing the total amount of the applied pesticide. Moreover, this is a way to avoid strong infestation in the crops.

III. Methodology

The general flow for the pest detection and classification in fruit crops is depicted in Fig. 3. The input image obtained in the field is downloaded from the cloud and pre-processed in

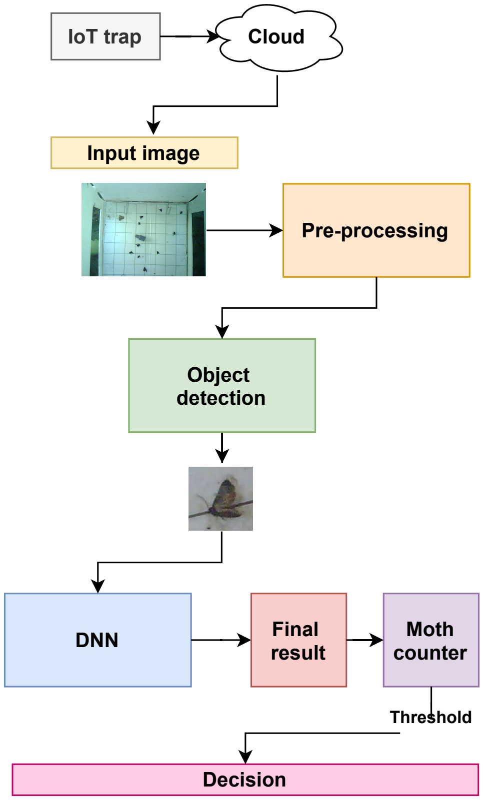

Fig. 3. General flow.

order to perform object detection through conventional techniques, such as filtering, color balance, noise reduction, among others. After this stage, an object detection is performed to crop the elements contained in the images. The result is sent to a classifier based on DNN, which is responsible to identify the class of the input object. If a moth was detected, a counter is increased until certain threshold. Once this value is overcome, the application sends a suggestion to proceed with the fumigation.

In real environment conditions, in-field traps are exposed to rain, wind, temperature variation and non-uniform illumination, among others. Also, leaves, dust and raindrops could be present inside the traps located in natural environments. These circumstances are a challenge when applying algorithms based on image processing, as we can observe in Fig. 4. Several other factors should be considered when processing images with digital techniques (such as natural conditions, paper colour among others), looking for the optimal combination of strategies to perform image analysis.

A. Pre-processing

The first step is to adjust the image size to a pre-defined square by cropping (or expanding) the image until the number of rows and columns become an integer multiple of the chosen window value, focusing the analysis on the region of interest (the base paper). After this operation, color space conversion from RGB to grayscale is performed and denoising is carried

Fig. 4. In-field traps under natural conditions.

out by means of a Gaussian filter. Finally, morphological operations (erosion and dilatation) are used with different kernels.

B. Object detection

This stage is aimed at detecting different elements (insects, pheromone lure and other objects) contained in the traps. Using the image obtained after the pre-processing, we obtain the contour of the objects applying edge detection based on Canny and convex hull, followed by bounding-box calculation and its area.

The starting point for image analysis is to use files containing only the pheromone lure, applying the techniques mentioned above to characterize it. Once pheromone lure's features are defined, they are used to discard this object, proceeding with insects and objects detection, which are present in small quantities, achieving their characterization and location. Examples of these distributions are shown in Fig. 5. As evidenced, from the left image the pheromone lure is present and in the right image, moth, pheromone lure and flies appear, each of one with unique characteristics in a first attempt to separate them from the background.

Objects having a maximum size larger than 250 pixels were discarded, since insects and pheromone lures have smaller size.

Nevertheless, the object area is not sufficient by itself for discrimination, since it is possible that the area of two objects be similar. In this case, a second level of filtering is required using the bounding box properties to select the elements. Fig.

Fig. 5. Object detection.

Fig. 6. Object detection.

6 shows the output of this stage, where each element is marked with a rectangle.

Once objects are detected, the next step is to realize a crop operation and save the corresponding images. The result of this stage is presented in Fig. 7, which contains separated insects and pheromone lures.

After this first attempt of detection based on traditional techniques and to decrease the misclassification rate, we introduced a DNN classifier, to obtain an efficient model for this task. The new images generated from the object detection process were used to build the data set to perform the training.

C. Classification based on DNN

For this stage, the image collection employed was obtained from in-field traps. As images were collected in real conditions and due to the limited amount of them, compared with the data sets used for training networks such as VGG16 (over 14 million high-resolution images) [8], we performed data augmentation [9]. This technique allows to increase the amount of images by creating copies of them, which will be slightly modified by operations such as: horizontal and vertical flip, rotation, scaling, cropping, translation, among others. Data augmentation contributes to prevent over-fitting when training machine learning models, acting as a regularizer. For instance, works presented in [10], [11], [12] proved the effectiveness and benefits of data augmentation when network training is done with a reduced number of input images.

Regarding the operations used in data augmentation, we selected horizontal and vertical flip, rotation, scaling, zoom and height shift.

Starting from the new image collection, which is formed by the objects detected in the previous stage, we perform a color

Fig. 7. Objects present in the trap.

Fig. 8. Objects present in the trap after Gray-world algorithm application.

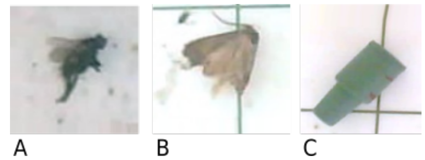

Fig. 9. Labeled objects. A. Other objects (class 0). B. Moth (class 1). C. Pheromone lure (class 2)

correction to improve the DNN performance using the Gray-world algorithm [13], a classical method of color constancy. Results are depicted in Fig. 8, showing an image enhancement compared to those presented in Fig. 7.

After this process, a total of 16500 images were obtained and three classes for the classification process were defined as follows: others objects (class 0), moth (class 1) and pheromone lure (class 2), made up of 5500 images each. An example of this distribution is shown in Fig. 9. The image labeling to define the ground-truth was performed in collaboration with a biologist. For training, the data set was split into train, validation and test.

The multi-class classification was performed using a Convolutional Neural Network (CNN) architecture, composed of a stack of 2D convolutional layers (2D Conv) and Max pooling (Max Pool) for feature extraction. At the end of the architecture, a stack of fully connected layers (FC) defines the classifier. The architecture is depicted in Fig. 10, with a total number of training parameters of 636,444. In regard to activation functions, a rectifier linear unit (ReLU) [14] was employed for each layer. Softmax was used in the output layer to calculate the final probability corresponding to each class.

IV. Results

Experimental setup: image processing was realized using OpenCv libraries, TensorFlow was employed to describe and train the CNN architecture and Python was selected as programming language.

The training was performed using k-fold cross validation [15], with 7 folds. To use a fixed input size, images were re-scaled up to 80×80 pixels in the RGB color space and normalized (each value was divided by 255). Adam [16] optimizer was chosen with a learning rate of 0.0001. As regularizer for each layer, we employed L2 norm with 0.0001. The number of epochs was configured in 32.

To perform inference tests, we employed 15% of the data set (an amount of images separated from the training and validation sets). The result is presented through the confusion matrix and it is depicted in Fig. 11. It is noticed that a high number of true positives were obtained, preventing false

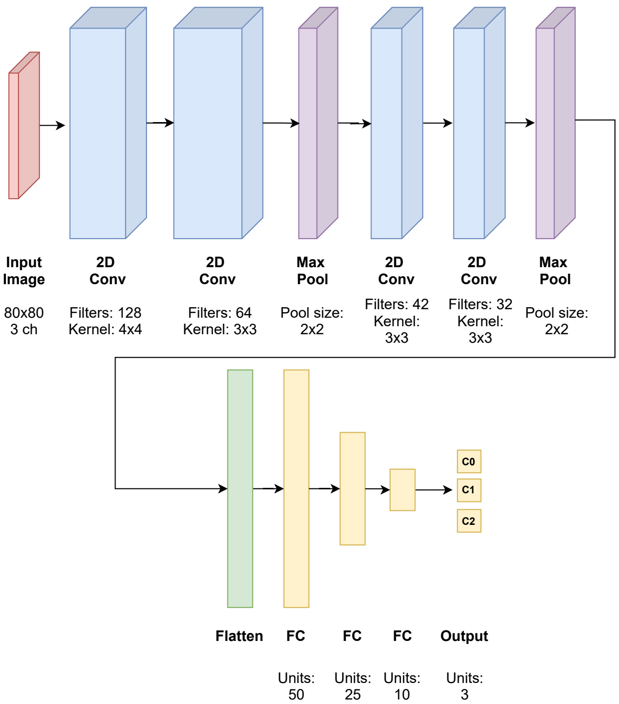

Fig. 10. Classifier based on CNN architecture.

negatives and false positives. Regarding class 0, a small percentage of images were classified as class 1.

For a multi-class classifier, average accuracy (Acc) is defined through Eq.1, with TP: true positive samples, TN: true negative samples, FP: false positive samples, FN: false negative samples and k: number of classes.

$$Acc = \frac{\sum_{n=1}^{k} \frac{TN_n + TP_n}{TN_n + TP_n + FN_n + FP_n}}{k} \tag{1}$$

Overall accuracy gives a preliminary information in this context, but this metric makes no distinction between classes, which is needed in this particular application. Assuming a classifier one-vs-all, to analyze the behaviour of the trained model through performance indicators we selected accuracy,

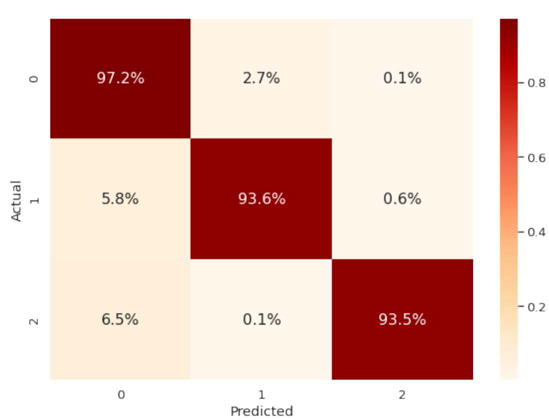

Fig. 11. Confusion matrix corresponding to the inference testing

TABLE I
METRICS PER CLASS: ACCURACY, PRECISION, RECALL AND ERROR RATE.

Metrics	Classes 0	Classes 1	Classes 2
Accuracy	**97.2%**	93.6%	93.5%
Precision	89%	97.2%	**99.2%**
Recall	**97.2%**	93.6%	93.5%

TABLE II
OVERALL METRICS: ACCURACY, MICRO-AVERAGING AND MACRO-AVERAGING FOR RECALL AND PRECISION.

Overall	
Accuracy	94.8%
Micro-averaging	
Precision	94.76%
Recall	94.76%
Macro-averaging	
Precision	95.09%
Recall	93.9%

precision and recall (sensitivity) [17]. Micro and macro averaging versions were considered for precision (P_μ and P_m) and recall (R_μ and R_m), presented from Eq. 2 to Eq. 5 and defined in [18]. To obtain these scores, we employed sklearn [19], a tool for predictive data analysis.

$$P_\mu = \frac{\sum_{n=1}^{k} TP_n}{\sum_{n=1}^{k} (TP_n + FP_n)} \tag{2}$$

$$P_m = \frac{\sum_{n=1}^{k} \frac{TP_n}{TP_n + FP_n}}{k} \tag{3}$$

$$R_\mu = \frac{\sum_{n=1}^{k} TP_n}{\sum_{n=1}^{k} (TP_n + FN_n)} \tag{4}$$

$$R_m = \frac{\sum_{n=1}^{k} \frac{TP_n}{TP_n + FN_n}}{k} \tag{5}$$

Precision score is defined as the proportion between correct results and the number of all returned results. Recall represents the percentage of correct images classified and the number of images classified that should have been retrieved. For a given n-class, precision at micro level is obtained from the individual TP, TN, FP, and FN. As regards macro level, precision is defined as the average of the performances of each class. From Table I, we conclude that precision is 89% for class 0, while for classes 1 and 2 is higher than 90%, highlighting a good percentage for class 1 (moth). Recall for class 1 presented a score larger than 90%, which implies a large ratio of correctly predicted moths to the all observations in the positive class.

Table II presents the overall behaviour of the system for the 3 classes (k=3 in the previous equations), being the average accuracy=94.8%, using 4314 test images. This value gives a sense of how effective the classifier is at the per-class level. Regarding precision and recall, the values are near 95%.

Due to this specific pest and its behaviour, the system has a threshold for the amount of moth fixed in 10. Up to this value, an alert should be send recommending the fumigation. From

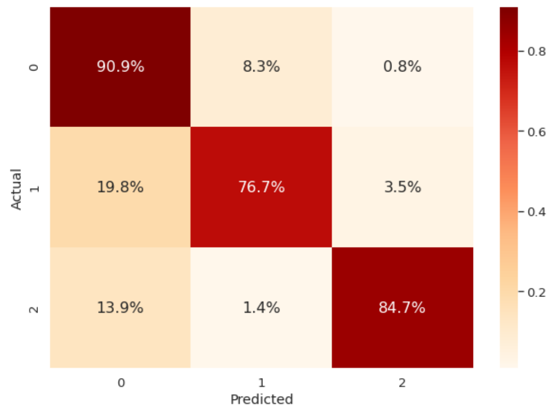

Fig. 12. Confusion matrix corresponding to the inference testing without applying Gray-World algorithm.

the metrics analyzed, we observe that the classifier presents a good behaviour at assigning positive classifications correctly and at classifying positive objects properly.

Moreover, to observe the effects of color correction in DNN for this particular application, an architecture was designed using the data set without applying the Gray-world algorithm. The resulting confusion matrix is depicted in Fig. 12. We observed that, using the same architecture exposed in Fig. 10, the overall accuracy of the system was 79.2%, with a value 58% for recall and precision for class 1 (moth). After this result, we extended the DNN with two more 2D-Conv layers, getting 855,268 parameters. This modification yielded an overall accuracy of 84%, with 89% and 76.7% for precision and recall respectively for class 1.

Inference time reported by the DNN classifier was 40 ms/image, exhibiting a good compromise between execution time and the metrics analyzed, which are higher than 90% in all classes.

V. Conclusions

The focus of this research was to develop an automatic pest detection and classification algorithm combining traditional image processing techniques (for object detection) and DNN (for classification), using a CNN architecture as classifier. This development finds a direct application to reduce pesticide use in fruit crops to control Carpocapsa, the main pest presented in pear, apple, walnut and quince tree. Based on this methodology, the fumigation process can be constrained in location and time, decreasing the frequency of pesticide application.

The combination of pre-processing techniques and DNN allowed us to develop an effective classifier. Results exposed the feasibility for this type of development using images obtained in-field, where traps are exposed to natural conditions. We improve the performance of DNN by means of color correction and we conclude that data augmentation for training DNN is a good choice when having a reduced data set.

Further studies will include inference times reduction for on line generation of a map of the infested areas, classification of multiple types of insects, and implementation of the multi-class discrimination for pest detection using other machine learning architectures. In addition, compression techniques for models based on machine learning will be addressed, and detection by machine learning algorithms will be developed.

Acknowledgment

The authors would like to thank NECTRAS for providing insight, expertise, resources and image data that greatly assisted the research.

References

[1] NECTRAS. Nectras. [Online]. Available: https://www.biodrone.com.ar/

[2] V. A. C. Figueiredo, S. Mafra, and J. Rodrigues, “A proposed iot smart trap using computer vision for sustainable pest control in coffee culture,” 2020.

[3] B. Ramalingam, R. E. Mohan, S. Pookkuttath, B. F. Gómez, C. S. C. Sairam Borusu, T. Wee Teng, and Y. K. Tamilselvam, “Remote insects trap monitoring system using deep learning framework and iot,” *Sensors*, vol. 20, no. 18, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5280

[4] N. T. Nam and P. D. Hung, “Pest detection on traps using deep convolutional neural networks,” in *Proceedings of the 2018 International Conference on Control and Computer Vision*, ser. ICCCV ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 33–38.

[5] N. E. M. Khalifa, M. Loey, and M. Taha, “Insect pests recognition based on deep transfer learning models,” 2020.

[6] W. Ding and G. Taylor, “Automatic moth detection from trap images for pest management,” *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169916300266

[7] S.-J. Hong, S.-Y. Kim, E. Kim, C.-H. Lee, J.-S. Lee, D.-S. Lee, J. Bang, and G. Kim, “Moth detection from pheromone trap images using deep learning object detectors,” *Agriculture*, vol. 10, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2077-0472/10/5/170

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, pp. 1–48, 2019.

[10] R. Poojary, R. Raina, and A. K. Mondal, “Effect of data-augmentation on fine-tuned cnn model performance,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, 12 2020.

[11] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122.

[12] E. Ayan and H. M. Ünver, “Data augmentation importance for classification of skin lesions via deep learning,” in *2018 Electric Electronics, Computer Science, Biomedical Engineerings’ Meeting (EBBT)*, 2018, pp. 1–4.

[13] V. Agarwal, B. R. Abidi, A. F. Koschan, and M. A. Abidi, “An overview of color constancy algorithms,” 2006.

[14] A. F. Agarap, “Deep learning using rectified linear units (relu),” *CoRR*, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[15] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” *Encyclopedia of Database Systems*, vol. 532–538, pp. 532–538, 01 2009.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[17] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” 2020.

[18] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing Management*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457309000259

[19] Scikit-learn. Scikit-learn. [Online]. Available: https://scikit-learn.org/stable/