UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

BeetleID: An android solution to detect ladybird beetles

Ricardo Esteban Muriel Jaramillo

Ingeniería en Sistemas

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 16 de mayo de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

BeetleID: An android solution to detect ladybird beetles

Ricardo Esteban Muriel Jaramillo

Nombre del profesor, Título académico **Noel Pérez, Phd in Computer Science**

Quito, 16 de mayo de 2021

© DERECHOS DE AUTOR

Nombres y apellidos: Ricardo Esteban Muriel Jaramillo

Código: 00125858

Cédula de identidad: 1723663090

Lugar y fecha: Quito, 16 de mayo de 2021

ACLARACIÓN PARA PUBLICACIÓN

UNPUBLISHED DOCUMENT

RESUMEN

Las Coccinellidae, o escarabajos mariquita, se introducen con frecuencia como agentes de control biológico en ambientes agrícolas para remover plagas de insectos. Las Harmonia Axyridis, se encuentran dentro de estas especies, y debido a su comportamiento predativo y competitivo, también son consideradas como plagas que causan daños en ecosistemas, consumiendo a especies locales de flora y fauna. Una solución óptima y accesible para detectar a estas especies ha presentado ser un desafío y una necesidad. En este trabajo, se desarrolló una aplicación móvil Android llamada BeetleID para la detección de escarabajos mariquita mediante métodos de pre-procesamiento de imágenes y un modelo de profundo aprendizaje de redes neuronales convolucionales. El modulo de pre-procesamiento de imágenes consiste en tres algoritmos principales: mapas salientes, contornos activos, y segmentación por super pixeles. La red neuronal convolucional usada fue validada por un conjunto de 2611 imágenes de especies de escarabajos mariquita con un método de validación cruzada de cinco iteraciones, que consiguió un puntaje de exactitud y área bajo la curva del característico operativo receptor de 0.92 y 0.98 respectivamente. Además, la factibilidad de la aplicación fue evaluada mediante las métricas de media de tiempo de ejecución y consumo de batería en los emuladores móviles, de teléfono, Pixel 3a XL, y tablet, Pixel C, que obtuvieron como resultado 16.32 y 18.43 segundos, y 0.07 y 0.11 miliamperios hora respectivamente. Estos resultados, demuestran que la aplicación propuesta es una solución excelente, pero con ciertos problemas de optimización, para especialistas en detectar con exactitud escarabajos mariquita en medios ambientes silvestres.

Palabras clave: *Detección de especies Coccilinedae, aplicación Android, pre-procesamiento de imágenes, modelos de profundo aprendizaje.*

ABSTRACT

The Coccinellidae, or ladybird beetles, are often introduced as biological control agents in agricultural environments to remove field pests. The Harmonia Axyridis are among these species, and because of their predatory and competitive behaviour, they are considered a field pest as well, causing damages in ecosystems by consuming local flora and fauna. An optimal and accessible solution for detecting this species has been a challenge and a necessity. In this work, an Android mobile application named BeetleID was developed for the detection of ladybird beetles through image pre-processing methods and a deep learning convolutional neural network model. The image pre-processing module consists in three main algorithms: saliency map, active contour, and super pixel segmentation. The used convolutional neural network was validated with a 2611 image set of ladybird beetle species with a five fold cross validation method, and achieved an accuracy and area under the curve of the receiver operating characteristic scores of 0.92 and 0.98 respectively. Furthermore, the applications feasibility was assessed by the mean execution time and battery consumption metrics of mobile emulators, phone Pixel 3a XL and tablet Pixel C, which obtained results of 16.32 and 18.43 seconds, and 0.07 and 0.11 miliampere hour respectively. This results, prove that the proposed application is an excellent solution, with a few optimization issues, for specialists to accurately detect ladybird beetles in wildlife environments.

Key words: *Coccinelidae detection, Android application, image pre-processing, deep learning models.*

AGRADECIMIENTOS

Presento este trabajo a mis padres y hermana, agradeciendo su soporte, amor, y por siempre alentarme a seguir adelante, y a mis profesores, por todos los conocimientos que fueron impartidos en mí hasta este momento.

Ricardo

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

I. INTRODUCTION

The Coccinellidae, or ladybird beetles, have predatory characteristics and are often used as biological control agents for repelling insect pests in agricultural activities [1]. They have proven positive results in preventing self-perpetuation and controlling that no new environmental problem arises, however, species such as the Harmonia Axyridis, are identified as crop pests, and therefore considered dangerous when introduced into environments as they can cause negative impacts on both local flora and fauna [2].

This species of ladybird beetles are dominant in cold ecosystems, but aren't limited to them, as they are able to strive in both natural and semi-natural environments [3]. Their invasions can be attributed to their rapid adaptation, generic diet, flexible life history, fast reproduction, and competitive and predatory behaviours [4]. These reasons make important to develop tools, such as Android applications, that can be executed anywhere, to correctly detect and identify ladybird beetles so they can be removed, saving damages to both natural and agricultural environments.

The efforts to detect species in biology, as seen in recent scientific literature, have achieved uses in identifying, monitoring, and protecting wildlife flora and fauna. For this purposes, several image detection methods, and deep learning models to recognize species have been studied. In [5], saliency and edge detection methods are utilized, to monitor animal species and protect them from excessive killing and hunting. In [6], a focus is showed on detection and classification of paddy field pests insects, first using saliency maps to detect them on image, which is continued with a deep convolutional neural network, to classify them between twelve typical species. In [7], feature transfer, a convolutional neural network previously trained with general image classification, is used to identify and classify insects to their respective groups by using small data sets of a hundred or less entries, and still showing accurate results. In [8], light-trap rice pests are identified by extracting 156 features from images taken, which then are fed to a support vector machine classifier with a seven fold cross validation to increase accuracy.

Despite of the development and study of mentioned methods, this work presents a challenge, as our objective focuses on the development of an Android mobile application, developed in Android Studio, which combines image pre-processing techniques and deep learning models to detect ladybird beetles in random images. The image pre-processing methods are, saliency map, to detect important sections in an image, active contour, to find object boundaries from the saliency map's output and contain them in bounding boxes, and super pixel segmentation, to segment images with pixel clusters. The generated bounding boxes will then be classified by a detector based on the convolutional neural network architecture to correctly identify which of them contain ladybird beetles. This implementation provides an Android solution, with an accessible way for specialist to use on their working field to correctly detect ladybirds, that when introduced in ecosystems as biological control agents, can also be invading crop pests, presenting danger to local species.

II. MATERIALS AND METHODS

A. Automatic ladybird species detection

The BeetleID Android application protocol consists of two modules, image pre-processing and a deep learning model based in the convolutional neural network (CNN) architecture. In the image pre-processing module three algorithms are present, saliency map, active contour, and super pixel segmentation, followed then by a CNN. The protocol, starting with an input image with a ladybird beetle on it, begins with saliency map, which detects the most important and salient pixels. Then, a first instance of active contour is used given the output of the saliency map, that will lead to contours of objects being formed and creation of initial bounding boxes, represented by square red outlines drawn in the image. Afterwards, super pixel segmentation is used to divide the image by pixel clusters, and a second instance of active contour is utilized to obtain the final bounding boxes. Finally, the CNN predicts accuracy scores on the images within every bounding box, redrawing successfully detected ladybird bounding boxes in yellow color, while not successful ones maintain their red color. Detailed information of the modules follows down below.

1) *Image pre-processing :* The objective of this module is to correctly obtain bounding boxes of areas where a ladybird beetle might be found. For that, three main methods are used, saliency map, active contour, and super pixel segmentation. Saliency map refers to the identification of important pixels within an image. These selected pixels are considered points of interest as they hold information regarding the taxonomic characteristics, such as color, of a possible ladybird. To achieve this, noise reduction using a mean filter is used to clear out the image from possible inconveniences without losing relevant information.

   Active contour, a technique in which contours of objects can be detected, is then used on the salient parts of the image indicated by saliency map´s output. In this process, outlines of objects are drawn and to represent them, bounding boxes are created in the image.

   Once the bounding boxes are successfully drawn, super pixel segmentation, a technique in which an image can be segmented by super pixels (group of pixels with similar characteristics), is used to get a more clear division. Then, active contour is used a second time to conclude the process, getting tighter contours and creating final bounding boxes. A more detailed explanation of every used algorithm follows below.

Saliency Map The saliency map algorithm indicates that specific parts of an image are of more importance than others, and by their salient characteristics, attention is naturally directed to these zones [9]. During observation, it is found normal to have fixations on specific targets, which is why, this process is important to correctly recognize visually attractive elements that a person might be focusing on when observing any environment, either if is by their color, contrast, or brightness.

Active Contour Active contour is described as an image segmentation method which creates sub-regions with defined boundaries. This makes possible to segment objects with their respective shapes in an image [10]. This algorithm works by creating boundaries presented by closed curves. In the beginning, this curves don't represent very accurately the shapes of encountered objects, which is why shrinking and expansion methods are used through various iterations to achieve a tighter and more accurate contour over an object.

Super pixel Segmentation Super pixels are defined as a group of pixels which share similar characteristics. By using them, the process of analysing an image is simplified, as the areas in it are greatly reduced [11]. Super pixels are a convenient and compact way of representation as they hold more meaning and information than independent pixels. The SLIC (Simple Linear Iterative Clustering) super pixel is the algorithm chosen for this work, as it has seen equal or better performance compared to other methods in the areas of object recognition and image segmentation. It works by generating a clustering of pixels in a five dimensional space given by the L, a, b values that represent the CIELAB color space [12].

2) *ROI calculation:* The ROI (area of interests), represented by the bounding boxes, are determined by the mentioned methods, and are dynamically created depending on the size of an interesting area. For each ROI, some padding must be included for

normalization purposes, so that, when they are trained by the CNN, no problem arises with images being not equal in characteristics.

3) *ROI Classification:* Convolutional neural network (CNN) is a deep learning algorithm that is primary used for image recognition tasks. It is composed of neurons that optimize themselves through learning. It's functionality is defined by, an input layer, in which values of pixels of an image are received and feature maps are created; a convolutional layer, that determines the output of neurons by calculating a scalar product between weights and inputs; and a pooling layer, that will operate and reduce each feature map separately, eliminating irrelevant features and keeping important ones [13]. In this work, the CNN is trained and developed on a desktop computer in the Python programming language, and then saved with a tflite file extension, using the Tensorflow library, which will then be integrated with the BeetleID Android application. The CNN will then make prediction accuracy scores on the ROI images created from the user selected image, to correctly determine which of them contain ladybird beetles, highlight them in different color bounding boxes.

B. Development environment

1) *Integrated development environment (IDE):* The environment Android Studio 4.1.3 was used for developing the BeetleID application, which allows the design and creation of the user interface for the developed activities, with their respective xml codes. The used run time environment is Java Development Kit 13.0.2 for 64 bits operating systems and the selected Android SDK version is 27, 8.1 Oreo.

2) *External Libraries:* The IDE and development kit used don't provide enough support for image pre-processing and deep learning methods, therefore it was decided to include the OpenCV (Open Source Computer Vision) library version 3.4.1, that provides the necessary tools to manage images and their properties with the provided Mat objects

[14]. Furthermore, the Tensorflow Lite library, which is supported in Android Studio since the 4.1 version releases, is utilized to import trained models saved as tflite extension files and using them for predictions [15]. With this libraries, it is possible to process images and get accurate predictions on them, which is why it is an adequate approach towards this process.

3) *Source Code optimization:* The use of C and C++ libraries to perform image pre-processing methods may influence in mobile device slowdowns when executing, which couldn't be avoided as they have limitations in resources. So, to provide a better and fluid experience when using the BeetleID application, the CNN model was trained and developed on a desktop machine using the Python programming language to relief some work of the device and reduce slowdowns and possible crashes.

C. Proposed Beetle ID app

The proposed application consists of a development in Android devices that, by combining image pre-processing methods and a deep learning CNN model, can correctly detect ladybird beetles in random images. As indicated in Fig. 1, the workflow of the application begins by receiving an input image, selected by the user from the device gallery, then, the image pre-processing module, determines and calculates various ROI, represented by bounding boxes. Afterwards, the CNN is used to correctly classify every ROI and predict in which of them a ladybird beetle is present, showcasing a final image to the user where bounding boxes in yellow, representing a found ladybird, can be seen.

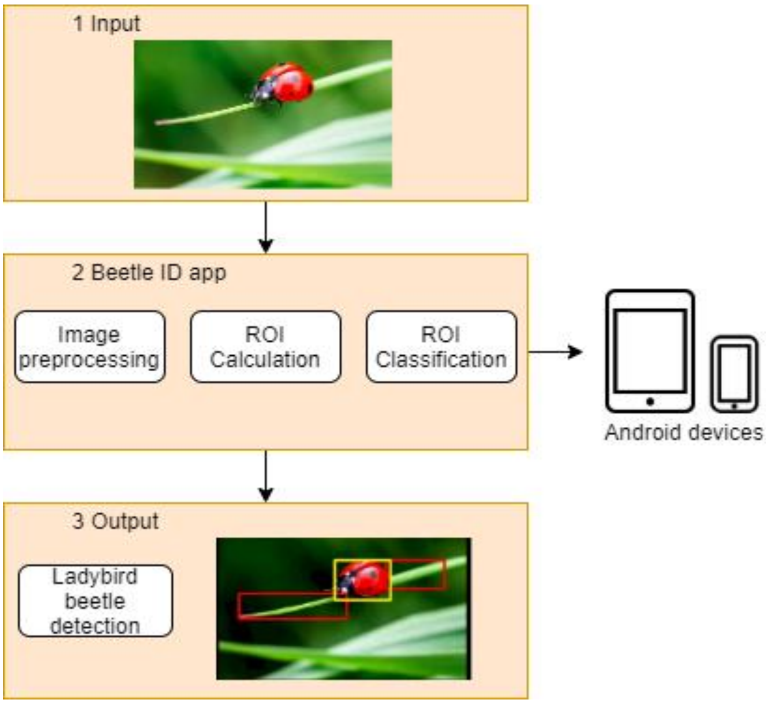


Fig. 1: Workflow for the Beetle ID application

Configuration There are configurations needed to fully explore and utilize the BeetleID application for its correct execution in Android devices. Permissions will be asked to every user on the first start up of the application, such as the access to the device camera, and the ability read and write in local memory. Such commands are included in the Android manifest xml file.

Initialization When opening the BeetleID application, the user will visualize a welcome page. In it, is possible to either close the app, or open the next activity window. In the next window, the user needs to select an image from their device gallery, and by pressing the process button, the applications workflow will initiate. The user will see his selected image in the image view component on the left side of the screen, and when the process finishes, the same image but with the created bounding boxes will be displayed on the image view in the right side. This last image will have yellow and red bounding boxes, representing areas with and without ladybird beetles respectively. Finally, at the lower portion of the screen, statistical values regarding the process are displayed.

D. Experimental setup

1) *Ladybug beetles image dataset:* The image dataset is composed of 2611 ladybug beetle images from species in Ecuador and Colombia that are registered in the iNaturalist project. The iNaturalist project is an initiative between the California Academy of Science and the Natural Geographic Society, to register species observation data provided by specialists from a diversity of countries [16]. The selected ladybird beetle species are adult only specimens, as their taxonomic characteristics, such as color and shape, have fully developed and can be better identified. From these, ROI images will be obtained to train the CNN classifier, to then import it into the BeetleID Android application.
2) *Experimental mobile devices:* Two android mobile devices were used for the development and testing of the BeetleID application, which were created through the Android Studio Device Manager. The characteristics of each device are included in Table 1.

Table 1. Characteristics of experimental devices.

Device	**Specification**	**Value**
Phone	Name	Pixel 3a XL
	CPU	Google API Intel Atom (x86)
	API Level	27
	RAM	1536 Mb
	Resolution	2160x1080: 400dpi
Tablet	Name	Pixel C
	CPU	Google API Intel Atom (x86)
	API Level	27
	RAM	1536 Mb
	Resolution	2560x1800: hdpi

API – application program interface; RAM – random access memory.

3) *Detector training and test:* The ROI images detected by the application, both with and without ladybird beetles, are used as inputs to train the CNN model. To create training and test sets, a ten time, five fold stratified cross validation method is used to correctly measure the models ability to generalize statistic results of the data set [17] and correctly observe both type of images. This will allow the CNN model to do accurate predictions on randomly fed images.
4) *Detector configuration:* The CNN architecture used in this work is composed of an input layer of 32 convolutional filters and a size of [144 x 144], that uses a rectified linear unit (ReLU) activation function with a [5 x 5] kernel size, who’s goal is to create a feature map representation from image pixel values. A max pooling layer of [3 x 3]

size is then applied to the feature map, reducing irrelevant features and conserving the important ones. The described convolutional and max pooling layers configurations are used twice. Afterwards, the now reduced feature map is fed to two convolutional layers of 64 convolutional filters, each with [3 x 3] kernel size, which then is followed by a max pooling layer of [3 x 3] size, that concentrates in classifying with the already most relevant features. Lastly, a fully connected layer of 256 and 1 neurons with a sigmoid activation function is used to obtain the final output. This CNN's architecture was selected as the best performing one in [18], from a series of other tested configurations.

5) *Validation metrics:* The proposed CNN is validated by the scoring of two metrics, accuracy, and area under the receiver operating characteristic curve (ROC-AUC). Accuracy, and AUC scores are used for validation because they inform a performance summary of how correct or incorrect classification is done [19], and how good in ranking the model is [20]. The performance of the BeetleID application is validated during it's execution, by measuring scores of the mean executing time and battery consumption metrics given by the Android Debug Bridge (adb) command line tool.

6) *Energy measurement protocol:* The android debug bridge service (adb command line tool) [21] facilitated us a satisfactory way to recover two logs from the emulated mobile devices. The battery stats log, containing important stats regarding the battery consumption of every single process (for our purpose, we took only the information from the process that executed the Beetle app) and the process activity log, containing % stats about the execution time (in nanoseconds) of the developed app.

III. RESULTS AND DISCUSSION

A total of 2611 images were analyzed, from which, 9925 ROI images were collected from the image pre-processing methods. This images were then used to train the deep convolutional neural network and generate a file with the tflite extension that contains the weights and architecture of the model. In the BeetleID Android application, a user will select an image from the devices gallery to obtain ROI's, represented by red bounding boxes. Then, the Tensorflow Lite library is used to import the CNN model and get prediction scores of the ROI images to determine which of them contain ladybird beetles. The criteria of selection of a correctly found ladybird within a bounding box, is done by obtaining a prediction score of 0.9 accuracy or superior. The ROI images that achieve this scores have their bounding boxes redrawn in yellow color to differentiate them from the rest. The resulting image can be visualized in the image view component at the right side of the devices screen and is also saved in the picture gallery. An example of the BeetleID app running on the phone emulator Pixel 3a XL is showcased in Fig. 2.

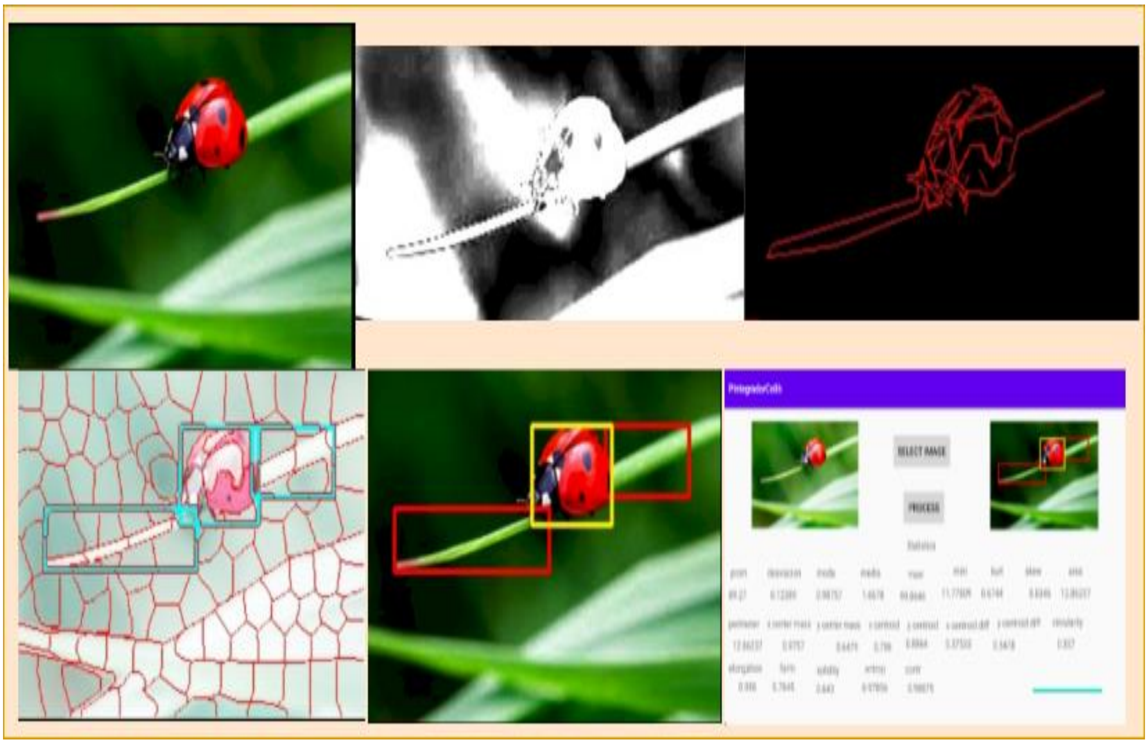


Fig. 2: Example of the BeetleID application execution on the Pixel 3a XL Android phone. From left to right, and top to bottom: images taken from the devices picture gallery, input image, saliency map, active contour, super pixel segmentation with first bounding boxes, image with final bounding boxes with detected ladybird beetle, result screen in phone emulator.

A. Beetle app performance

The results of the trained CNN gave an accuracy score of 0.92 and an area under the receiver operating characteristic curve score (ROC-AUC) of 0.98, which indicates an excellent performance by the detector at discriminating cases of positive and negative existing ladybird beetles in images. This means that the model itself will make close to perfect distinctions when faced with randomly taken images and therefore, is adequate to incorporate it with the developed BeetleID Android application.

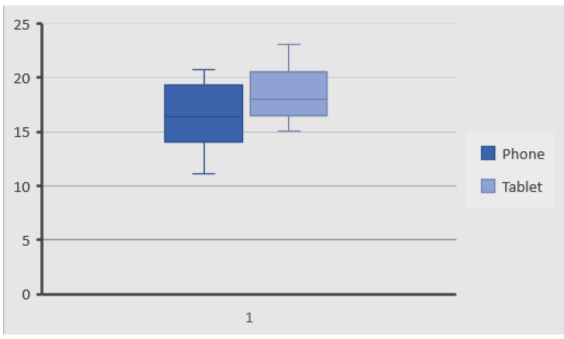
When imported into the application, the model showed excellent performance in detecting ladybird beetles from the bounding boxes within an image. This can be seen in Fig. 3, where the picture in the left is the one selected by the user; the middle picture, shows the bounding boxes created by the image pre-processing module; and the rightmost picture, shows the result after the models prediction, where bounding boxes in yellow and red color can be seen, indicating the positive and negative presence of ladybird beetles respectively. The bounding boxes on the last image, from left to right, got prediction scores of 0.10, 0.99, and 0.42. The leftmost bounding box shows no appearance of a ladybird which reflects on its low score, the rightmost one, shows a small portion of the ladybirds body on its left side, which is why it managed a higher score, finally, the middle one, which contains a ladybird, shows an almost perfect score, meaning that the model can accurately detect the presence of ladybird beetles with certainty.

Fig. 3: Example of a ladybird beetle detection

B. *Beetle app feasibility*

Regarding the use of maximum resources of Android devices, the BeetleID application has yet to be tested on real devices. Estimated execution time and battery consumption was calculated by the established energy measurement protocol of the adb command line tool. Two mobile emulators were used to test the BeetleID application (see Table 1). In terms of execution time and battery consumption metrics, the adb tool was used to get results during the applications execution in both the Pixel 3a XL phone and the Pixel C Tablet. Graphical representations of the results, in box and whiskers plots, can be visualized of the execution time (Fig. 4, left side) and battery consumption (Fig. 4, right side). In execution time, the Pixel 3a XL obtained a mean score of 16.32 seconds, while the Pixel C obtained a mean of 18.43 seconds, showing a more efficient result in the Pixel 3a XL as it overall shows less time results. In battery consumption, the pixel 3a XL shows a mean of 0.07 mAh, while the Pixel C shows a mean of 0.11 mAh, which again shows a better performance in the phone emulator.

The execution time results in both emulators are slow and undesired as the application can take long before finishing, this is suspected to be an issue with occupying C and C++ libraries and resource consuming methods of the OpenCV library during the image pre-processing module, that Android devices, which already present limited computing power, might have problems with. However, battery consumption results are optimal as they show low values, which also indicates that the application isn't going to impact greatly on battery life. With this results, the BeetleID application proves to perform well in both, the image pre-processing module and the CNN predictions, but in terms of feasibility, it suffers from slow execution times, while presenting good battery consumption scores, from which it´s concluded that the application achieves it's objective, but presents a few optimization issues.



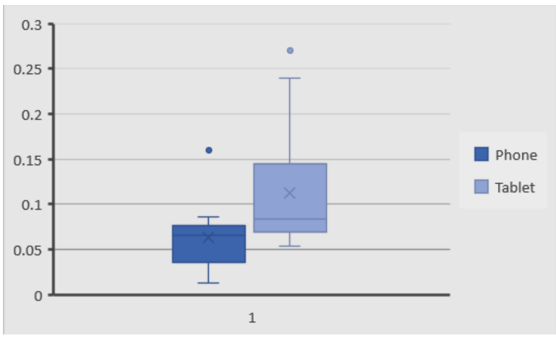


Fig. 4: Beetle ID app performance in Time Execution (left) in seconds and Battery Consumption (right) in mAh in the phone Pixel 3a XL (left) and the tablet Pixel C (right).

IV. CONCLUSIONS AND FUTURE WORK

In this work, an Android mobile application was developed to correctly detect the presence of ladybird beetles in random images. The proposed application integrates an image pre-processing module, to get bounding boxes of likely ladybird beetle locations, and a deep convolutional neural network, to correctly predict which of them truly presented one. The proposed CNN achieved an AUC and accuracy score of 0.98 and 0.92 respectively, which proves that it's an excellent model in discriminating the presence of ladybird beetles. In the applications feasibility, BeetleID obtained mean execution time scores of 16.32 and 18.43 seconds and battery consumption mean scores of 0.07 and 0.11 miliamper hour in both phone Pixel 3a XL and tablet Pixel C emulators respectively. The results of the battery consumption metric are considered good for mobile devices, but poor execution times, stipulate optimization issues in the application, that are thought to be caused by the used C and C++ libraries, as they can consume a great amount of resources in already limited in computing power Android devices. For future work, experimenting with different methods from the OpenCV library or a different library altoguether, may show better performance results in execution time for Android devices. Also, the application was only tested on emulator devices, so it is planned to implement and test it in real devices. The Beetle ID application is considered to be a good tool, with some optimization issues, for specialist in biological areas to detect ladybird beetles with excellent accuracy on their work field, as they are crop pests that can damage natural and agricultural environments.

REFERENCES

[1] M. Halim, A. Aman-Zuki, M. A. Mohammed, and S. Yaakop, "Dna barcoding and relationships of eight ladybugs species (coleoptera: Coccinellidae) that infesting several crops from peninsular malaysia". Journal of Asia-Pacific Entomology, vol. 20, no. 3, pp. 814–820, 2017.

[2] D. W. Goetz, "Harmonia axyridis ladybug invasion and allergy". in Allergy & Asthma Proceedings, vol. 29, no. 2, 2008.

[3] P. M. Brown, C. E. Thomas, E. Lombaert, D. L. Jeffries, A. Estoup, and L. J. L. Handley, "The global spread of harmonia axyridis (coleoptera: Coccinellidae): distribution, dispersal and routes of invasion". BioControl, vol. 56, no. 4, pp. 623–641, 2011.

[4] E. W. Evans, A. O. Soares, and H. Yasuda, "Invasions by ladybugs, ladybirds, and other predatory beetles". BioControl, vol. 56, no. 4, pp. 597–611, 2011.

[5] W. Feng, J. Zhang, C. Hu, Y. Wang, Q. Xiang, and H. Yan, "A novel saliency detection method for wild animal monitoring images with wmsn". Journal of Sensors, vol. 2018, 2018.

[6] Z. Liu, J. Gao, G. Yang, H. Zhang, and Y. He, "Localization and classification of paddy field pests using a saliency map and deep convolutional neural network". Scientific reports, vol. 6, no. 1, pp. 1–12, 2016.

[7] M. Valan, K. Makonyi, A. Maki, D. Vondrácek, and F. Ronquist, "Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks". Systematic Biology, vol. 68, no. 6, pp. 876–895, 2019.

[8] Y. Qing, L. Jun, Q.-j. Liu, G.-q. Diao, B.-j. Yang, H.-m. Chen, and T. Jian, "An insect imaging system to automate rice light-trap pest identification". Journal of Integrative Agriculture, vol. 11, no. 6, pp. 978–985, 2012.

[9] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics". Journal of vision, vol. 8, no. 7, pp. 32–32, 2008.

[10] D. Baswaraj, A. Govardhan, and P. Premchand, "Active contours and image segmentation: The current state of the art". Global Journal of Computer Science and Technology, 2012.

[11] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13 964–13 973.

[12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels". Tech. Rep., 2010.

[13] K. O'Shea and R. Nash, "An introduction to convolutional neural networks", arXiv preprint arXiv:1511.08458, 2015.

[14] G. Bradski, “The OpenCV Library”. Dr. Dobb’s Journal of Software Tools, 2000.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems”. 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[16] iNaturalist, https://www.inaturalist.org/, accessed: 2021-03-19.

[17] R. Kohavi et al., “A study of cross-validation and bootstrap for accuracy estimation and model selection”, in Ijcai, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[18] N. Pérez, P. Venegas, D. Benítez, D. Riofrío, F. Calderon, G. Ramon, D. Cisneros-Heredia, j. L. R. Álvarez, and M. Coimbra, “Automatic ladybird beetle detection using deep-learning models”, Mar 2021. [Online]. Available: osf.io/x6cv9

[19] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation”, in Australasian joint conference on artificial intelligence. Springer, 2006, pp. 1015–1021.

[20] J. Huang and C. X. Ling, “Using auc and accuracy in evaluating learning algorithms”. IEEE Transactions on knowledge and Data Engineering, vol. 17, no. 3, pp. 299–310, 2005.

[21] Developers Android. “Android Debug Bridge (adb)”, https://developer.android.com/studio/command-line/adb.html, accessed: 2021-03-20.