UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias en Ingeniería

MosCla app: An android app to classify Culicoides species

Manuel Sebastián Gutiérrez Jiménez

Ingeniería en Sistemas

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

Quito, 7 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

MosCla app: An android app to classify Culicoides species

Manuel Sebastián Gutiérrez Jiménez

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

Quito, 7 de mayo de 2020

DERECHOS DE AUTOR

Nombres y apellidos: Manuel Sebastián Gutiérrez Jiménez

Código: 00108723

Cédula de identidad: 1717526196

Lugar y fecha: Quito, 7 de mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Los mosquitos picadores *Culicoides* son conocidos por ser vectores de transmisión de diversas enfermedades causadas por arbovirus afectando a humanos y animales alrededor del mundo. Un método óptimo y rápido de clasificación para estas y otras especies ha sido un desafío y una necesidad, especialmente en áreas con recursos limitados y problemas de salud pública. En este trabajo, nosotros desarrollamos una aplicación móvil llamada *MosCla* para clasificar dos especies de *Culicoides* utilizando el análisis del patrón morfológico de sus alas. La aplicación implementa un método de clasificación automático basado en el cálculo de siete características morfológicas extraídas de las imágenes del ala y una máquina de vectores de soporte como clasificador para producir la clase final de *Pusillus* u *Obsoletus*. La aplicación propuesta fue validada usando un conjunto de datos experimental de 87 imágenes de ala con el método de validación cruzada de 10 iteraciones (10-fold cross-validation), y el rendimiento de clasificación, en términos de la media de la métrica de AUC, obtuvo un puntaje sobresaliente de 0.98. La viabilidad de la aplicación se evaluó utilizando las métricas de media de tiempo y de consumo de batería en emuladores móviles del teléfono Pixel 2 y tableta Pixel C. Los resultados obtenidos fueron 12 y 7 segundos y 0.11 y 0.03 mAh para los emuladores del teléfono y la tableta respectivamente. Estos resultados se consideran muy buenos en el desarrollo de aplicaciones móviles. Reducir el espacio de características de siete a tres características usando un método de envoltura externo nos proporcionó una mejora considerable en términos del rendimiento de clasificación basado en el AUC mejorando de 0.95 a 0.98 y disminuyendo el volumen de información para el entrenamiento del clasificador. Por lo tanto, estos resultados permiten a la aplicación propuesta ser una excelente aproximación para aquellos especialistas que necesitan una herramienta práctica para clasificar especies de *Culicoides* como *Pusillus* u *Obsoletus* en el medio ambiente silvestre.

Palabras clave: Clasificación de especies Culicoides, aplicación Android, procesamiento digital de imagen, clasificador SVM, selección de características.

ABSTRACT

Culicoides biting midges are known for being transmission vectors of various diseases caused by arboviruses affecting humans and animals around the world. An optimal and fast classification method for these and other species has been a challenge and a necessity, especially in areas with limited resources and public health problems. In this work, we developed a mobile application called *MosCla* to classify two *Culicoides* species using the morphological pattern analysis of their wings. The app implemented an automatic classification method based on the calculation of seven morphological features extracted from the wing image and a support vector machine classifier to produce the final classification of *Pusillus* or *Obsoletus* class. The proposed app was validated using an experimental dataset of 87 wing images with the 10-fold cross-validation method, and the classification performance in terms of the mean of the AUC metric obtained an outstanding score of 0.98. The app feasibility was assessed using the mean of time and battery consumption metrics on a phone Pixel 2 and tablet Pixel C mobile emulators. The obtained results were 12 and 7 seconds and 0.11 and 0.03 mAh for the phone and tablet emulators, respectively. These results are considered very good when developing mobile applications. Reducing the feature space from seven to three features using an external wrapper method provided us a considerable improvement in terms of the AUC based classification performance from 0.95 to 0.98 and decreasing the volume of information for the classifier training. Thus, these results enable the proposed app as an excellent approximation for those specialists that need a practical tool to classify *Culicoides* species like *Pussillus* or *Obsoletus* in the wildlife environment.

Key words: *Culicoides* species classification, Android application, digital image processing, SVM classifier, feature selection.

AGRADECIMIENTOS

Presento este trabajo agradeciendo a Dios, a mis padres por su apoyo y paciencia, y a mis profesores y tutor por su guía y conocimiento.

Sebastián

TABLA DE CONTENIDO

Introduction **11**
Materials and Methods **13**
Automatic Culicoides species classification **13**
Image pre-processing 13
Wing’s particles detection and zones segmentation 14
Feature calculation 14
Feature selection 14
Classification 15
Development enviroment **16**
Integrated development environment (IDE) 16
External Libraries 16
Source Code optimization 16
Proposed MosCla app **17**
Experimental setup **19**
Culicoides wing image dataset 19
Experimental mobile devices 20
SVM training and test 20
SVM configuration 20
Validation metrics 20
Energy measurement protocol 20
Results and Discussion **22**
Classification performance **22**
MosCla app feasibility **23**
Conclusions and Future Work **24**
References **25**

ÍNDICE DE TABLAS

Table 1 Summary of Experimental Mobile Devices Emulators 21

ÍNDICE DE FIGURAS

Figure 1 Workflow of the proposed MosCla app 15
Figure 2 Example of the proposed MosCla app execution.. 17
Figure 3 Separation of both Culicoides classes 23
Figure 4 MosCla app performance.. 23

I. INTRODUCTION

Culicoides is a genus of biting midges implicated in the transmission of various human and animal diseases [1], [2]. For example, *Culicoides* can transmit Bluetongue viruses (BTV), Akabane virus (AKAV), bovine ephemeral fever virus (BEFV), epizootic haemorrhagic disease virus (EHDV), the Schmallenberg virus (SBV) which produce viral diseases that can affect domestic ruminants (cattle, sheep, goats) and many species of deer, and African horse sickness virus affecting horses, mules, and donkeys. They are also responsible for insect-transmitted protozoan parasites such as *Haemoproteus*, *Leucocytozoon* that can be present in birds, and *Onchocerca* which can produce blindness in humans.

More than 50 arboviruses have also been isolated from *Culicoides* including viruses from *Bunyaviridae*, *Reoviridae*, and *Rhabdoviridae* families [3]. Therefore, their influence is not only ecological but also economical, since they have a direct effect on agriculture, forestry, and even on animal and human health [4]. Thus, it is important to have fast and accurate identification methods to detect the different species of these midges in order to determine vectors and conduct entomological monitoring programs for evaluating the diversity of species in a specific area to take defensive actions against such gnats.

Morphological characteristics such as: wing pigmentation pattern, antennal segment length and shape, male genitalia characteristics, antennae sensillae distribution, and the spermathecae number and size in females have been used to identify *Culicoides* species [5], [6]. Adults Culicoides are remarkable for their wing patterns and pigmentation (distribution and color of spots). These patterns can be used in certain species as the main criteria for diagnosis. Biting midges wing geometric morphometrics is an established, inexpensive, and reliable identification technique of *Culicoides* species. [7], [8], [9], [10], [11]. Consequently, the wing appears to be a good character for species discrimination. But, the cryptic species seem to share approximatively 20% of wing skeleton.

In the recent scientific literature, other studies and methods have been applied for mosquito species classification. A convolutional neural network (CNN) model was implemented in [12] to extract features from images and identify species like the genus *Aedes* known for being transmitter of the Zika virus . In [13] the authors implemented ionization time of flight mass spectrometry with protein mass fingerprints as an alternative to identify *Culicoides* species. Furthermore, an artificial neural network (ANN), using ribosomal DNA data, was implemented in [14] for classification of the genus *Anopheles* known for being a malarial transmission vector. Moreover, in [15] the classification of *anopheline* mosquitoes using cluster analysis was carried out based on the characteristics of the habitat were specimens were collected. Finally, the authors in [16] use DNA barcodes to identify the main mosquito specie in China based on morphological characteristics using a Neighborhood-Joining tree.

Despite the recently developed methods for *Culicoides* biting midges classification, this problem remains as a challenge. Besides, there is a need for implementing portable tools associated with the developed methods. Therefore, in this work, we proposed the development of a mobile application called *MosCla* to classify *Culicoides* species based on the morphological pattern of their wings. The proposed application implements an automatic *Culicoides* species classification method using the Android development environment to produce the final app. An experimental dataset containing wing images from two BTV transmission vectors species *C. pusillus* (French Guyana), and *C. obsoletus* (France) will be used to benchmark the proposed approach. Also, a feasibility study was carried out to know how practical in terms of classification performance, execution time, and battery consumption, the implemented app could be. This study was made using two commercially available devices with limited resources, a Pixel C Tablet, and Pixel 2 phone (emulators). This app will constitute a feasible and practical tool to be used by biologists or entomologists researches on the field.

The remainder of this paper is organized as follows: the Materials and Methods section, presents the step by step automatic *Culicoides* species classification implemented as an Android mobile application, the development environment used on this project, the proposed *MosCla* app, and the experimental setup designed to validate it. The Results and Discussion section presents the classification performance based on the area under the receiver operating characteristic curve (AUC) score obtained by the classifier; and the application performance in terms of the mean execution time and battery consumption metrics. Finally, Conclusions and future work are drawn in the last section.

II. Materials and Methods

A. Automatic Culicoides species classification

The *Culicoides* species classification is obtained by considering the morphological characteristics of their wings. We used a set of computer vision functions for digital image processing to obtain morphological features from the wings images [17] and a machine learning classifier to distinguish amongst the species under analysis. Thus, the image pre-processing, wing particle detection and zones segmentation, feature calculation and selection, and the machine learning classifier (MLC) are essential aspects to describe here.

It should be pointed out that all the employed functions in this work were carefully selected to maximize performance while minimizing the battery consumption on the target devices.

1) Image pre-processing: this module aims to the correctly extraction of a wing binary mask and their corresponding bounding box. Thus, the original RGB (red, green, blue) image was first transformed to a gray-scale (pixels values up to 255) color space. Then, a filtering operation was carried out using a (3×3) median filter to remove possible noises while conserving the contours of the objects of interest in the wing's image. A morphological erosion operation was then applied to remove all the non-desired objects that were still present in the filtered image such as isolated pixels or noise. The eroded wing image was followed by a filling operation to obtain the final binary mask of the wing and, subsequently, its bounding box, which was determined by using the Otsu's method [18]. Finally, an opening and closing morphological operations were applied to conserve the shape and size of particles inside the wing image without other objects overlapping.

2) Wing's particles detection and zones segmentation: the Moore-Neighbor tracing algorithm [19] with the Jacob's stopping criteria was applied on the binary wing image to detect the particles inside the wing contour. The presence of particles was verified by tracking whether or not there are changes in the pixel intensity value of the 8-connected pixels in the vicinity of the current pixel (pixel under analysis), any intensity change was used as the stopping criteria by the algorithm. Moreover, the watershed method [20] was applied to segment the zones inside the wing. This method discovers ”basins” and ”ridges” in the image surface. The algorithm assumes that light and dark pixels represent elevations and depressions linked to the particles and zones, respectively.

3) Feature calculation: a set of seven morphological features [17] (number of particles, number of zones, elongation, solidity, circularity, hydraulic radius, and eccentricity) were computed from the binary images. All the values were normalized using the min-max method [21] to bring them into the range [0,1].

4) Feature selection: this technique is efficient when using high-dimensional data to reduce the heavy load of inputs to MLCs. This selection is applied to avoid overfitting, improve model performance and to provide faster and more cost-effective models [22]. In this work, the feature space is composed of seven numerical variables (features) regarding morphological descriptors of the wing images. Despite the small number of computed features, the feature space is still significant to be explored on equipment with limited resources like mobile devices. Thus, we used a wrapper method to reduce the feature space. These methods combine heuristical search with MLCs as a unique piece to score subsets of features according to their predictive power related to the output class [22]. The subset that provides the higher score represents the model output.

We wrapped a greedy step-wise search method (to avoid stuck in local minimums) together with a linear SVM classifier (to prevent algorithm complexity and gain processing speed) to find out the best subset of features.

5) Classification: a supervised learning problem may be viewed as a problem of data separation into different classes, i.e., the output value of a prediction function after the classifier has been trained using several input-output valid pairs [23]. The classification of *Culicoides* species is a problem that involves two discrete output classes: *Pusillus* and *Obsoletus* species. Hence, it can be modeled as a two-class classification problem. Among the vast number of available classifiers and multiply classifiers combinations that can be employed to solve problems in different areas, we decided to use a support vector machine (SVM). Since mobile devices have limited resources, it was mandatory to consider a classifier with an acceptable trade-off between successful performance and low battery consumption. The SVM is based on the definition of an optimal hyperplane, which linearly separates the training data. In comparison with other classification methods, the SVM aims to minimize the empirical risk and maximize the distances (geometric margin) of the data points from the corresponding linear decision boundary [24], [25].

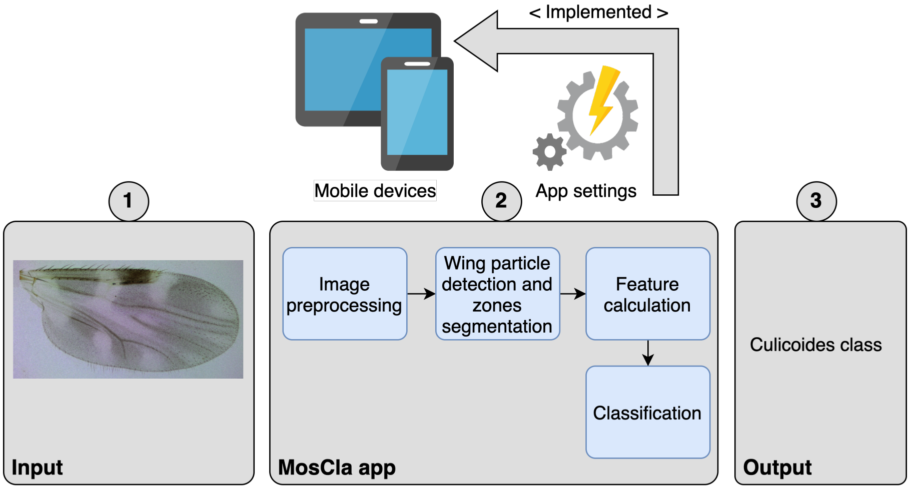

Fig. 1. Workflow of the proposed *MosCla* app implemented on mobile devices.

B. Development environment

1) Integrated development environment (IDE): we used Android Studio version 3.6 for developing the *MosCla* app source code efficiently and neatly. This allowed us to design the Android activities UI's visually together with the XML code. Besides, the java OpenJDK (Java development kit) version 11.0.6 for 64-Bits operating systems was selected as the run-time environment and development tools, respectively. Thus, the developed application will be on the base of the Android mobile operating system with the Android SDK version 9.0 (Pie).

2) External Libraries: the employed IDEs and development tools do not provide enough support to implement the digital image processing functions needed to fulfill the wing image pre-processing as well as the classification. Thus, we added two third-party's libraries: OpenCV (Open Source Computer Vision Library) release 3.4.8 [26], which will also help to store the images as a MAT object, facilitating the heavy image processing operations and transformations on devices with limited resources, and the LIBSVM (Library for Support Vector Machines) version 3.24 [27] for the Android platform. Since both libraries were developed in native C and C++ languages, it was necessary to include the NDK (Native development KIT) version 21.0.6113669 to operate with them. Additionally, the Google API play-services-location version 17.0.0 will be used in the developed app to estimate the latitude and longitude based localization of newly acquired or existing images.

3) Source Code optimization: although using native C, C++ libraries, the performance of some modules like the digital image processing functions, the watershed transformation, the particles analysis, and the training-test process of the SVM classifier are not optimal enough when performing them sequentially. Thus, the Android asynchronous tasks will be exploited to carry out these modules on separate threads, avoiding possible application crashes, and speeding up associated processes.

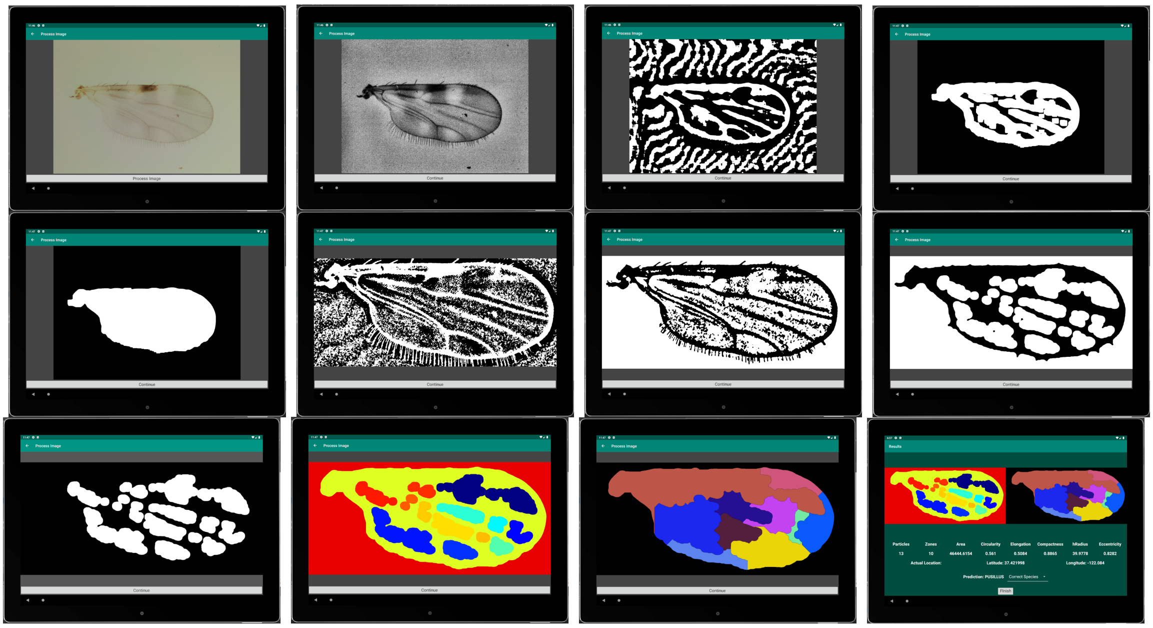

Fig. 2. Example of the proposed *MosCla* app execution on a Pixel C device with Android 9.0 (API 28) based emulator. From left to right and up to bottom: loading the image from the photo gallery, image enhancement by the CLAHE function, image filtering by a (71 × 71) median filter, background removal, binary mask creation, bounding box determination, wing region isolation, wing particles determination, particles isolation, color map application, zones segmentation and features calculation with the final classification.

C. Proposed MosCla app

The proposed application implemented an automatic *Culicoides* species classification method using the defined development environment to produce an Android application able to classify two *Culicoides* species on limited resources devices (see Fig. 1).

The app receives as input the wing images of the *Culicoides* species. Each entry transits throughout several digital image processing steps such as pre-processing, wing particle detection and zones segmentation, feature calculation to produce a feature vector linked to the entry (wing image). Then, a subset of the computed feature vector is used to feed a SVM classifier, which provides the final classification of *Pusillus* or *Obsoletus* class.

Additionally, the initial feature vector together with the final classification of each processed entry was saved in a database file, such files were used for a feature selection process using an external wrapper. The wrapper method, which was implemented outside the app (on the WEKA toolkit version 3.8.3 [28]), selected three out of seven computed features. Thus, the training - test feature vectors sets were formed with a total of three features and the output class. Besides, these sets are employed for re-training the implemented SVM classifier by the user at any time. The later option of reinforcement learning guarantees the progressive model specialization. An example of execution of the implemented *MosCla* app is shown in Fig.2.

For better user experience, the proposed *MosCla* app requires special configurations regarding permissions and initialization. These settings are defined in the *AndroidManifest.xml* file and help to avoid run-time errors. Therefore, the application always checks if the user has granted the following permissions:

- android.permission.INTERNET
- android.permission.CAMERA
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.ACCESS_COARSE_LOCATION
- android.permission.ACCESS_FINE_LOCATION

Otherwise, it prompts the user to accept missing permissions. This permission check is done at some control points during run-time because the camera, location, and storage are vital settings to exploit the maximum benefits of the app.

Regarding the initialization, there are two actions that the *MosCla* app offers when it starts: either choose an image from the device's photo gallery (*ChooseTakePhoto*) or take a new photo (*TakeNewPhoto*).

Choosing a storage image starts an activity through an ACTION.PICK intent, which allows the user to select and load the image into memory. On the other hand, the action of taking a new photo starts an implemented activity that checks the camera's existence before displaying the real-time picture preview. This activity also defines a real-time region of interest (ROI) with a rectangle shape to enclose the wing image under analysis. It should be noted that the defined ROI should constitute the maximum area containing the wing image. Enclosing the wing's image inside the ROI is essential for the successful performance of some of the digital image processing functions like the watershed transform, which overflows when desired objects are very close to the image border.

D. Experimental setup

This section is aimed to evaluate the proposed *MosCla app*. Therefore, the *Culicoides* wing images dataset, experimental mobile devices, classifier training and test, validation metrics, and energy measurement protocol are important aspects to be described next.

1) Culicoides wing image dataset: pusillus (n=45) samples were captured at French Guyana (3°59'56"N, 53°00'00" W) and *obsoletus* samples (n=42) at France (49°59'69"N, 4°01'45"E; 43°58'55"N, 3°42'58"E) by using ultraviolet traps. Samples were stored in 70% ethanol before any morphological analysis. The specimens were separated from other insects according to their wing characteristics using a stereomicroscope [29]. The digital wing images were obtained using an Olympus BX53 microscope equipped with an Olympus SC100 camera magnified at x10 with stream motion software (Olympus). For our study, left and right wings were used indistinctly because the systematic selection of one side may bias the results in case of differential directional asymmetry between species. The comparison of wings from catalogs and original descriptions with status (left or right) are mostly unknown and the distribution and color of spots on both wings are similar [17]. Thus, we formed an experimental dataset with a total of 45 and 42 wing images of *Culicoides Pusillus* and *Culicoides Obsoletus*, respectively.

2) Experimental mobile devices: two mobile device emulators were considered for installing and testing the proposed *MosCla* app. They were selected based on the minimum device specifications required to host the proposed app. It should be noted that some digital image processing functions and the classifier itself demand a certain amount of resources to fulfill the assigned tasks. The specifications of the selected devices are summarized in Table I.

3) SVM training and test: the 10-fold cross-validation method [30] was applied before the classification stage to form separate training and test partitions. Therefore, the SVM classifier was trained on different training sets and learned from different input space representations. Testing on these different sets leads to variability in the classification results and allows us to avoid overfitting.

4) SVM configuration: a SVM with a linear kernel was used to speed up the processing time, the regularization parameter *C* (cost) was set to 10 units (empirical selection), the *gamma* was tuned to 0, which means an impartial influence on new features, the bounded regularization parameter (*nu*) was set to 0.5 units, the *cache_size* used was set to 20(MB) for data exchange size and the epsilon margin of tolerance (*eps*) was set to 0.001 units.

5) Validation metrics: the proposed app was validated using a two-step procedure involving (1) the individual assessment of the employed SVM classifier using the area under the curve (AUC) score of the receiving operating characteristic and (2) the feasibility of the implementation using the mean of execution time (mET) and battery consumption (mBC) metrics gathered while running the app. Both validation steps were made on the experimental *Culicoides* wing's image dataset.

6) Energy measurement protocol: the android debug bridge service (adb command line tool) [31] 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

TABLE I

SUMMARY OF EXPERIMENTAL MOBILE DEVICES EMULATORS.

Device	Specification	Value
Tablet	Model	Pixel C.
	CPU/ABI	Google Play intel Atom (x86)
	API	28 (Android 9.0 Pie)
	Target	google_apis [Google APIs] (API level 28)
	Skin	pixel_c
	Resolution	2560x1800 xhdpi
	Camera	Host Camera (Acer Nitro 7 Laptop)
	Internal Storage	2048 MB
	RAM	2048 MB
	VM heap	127 MB
Phone	Model	Pixel 2.
	CPU/ABI	Googel Play Intel Atom (x86)
	API	29 (Android 10.0 Q)
	Target	google_apis_playstore[Google Play] (API level 29)
	Skin	pixel_2
	Resolution	1080x1920 xxhdpi
	Camera	Host Camera (Acer Nitro 7 Laptop)
	Internal Storage	6144 MB
	RAM	1536 MB
	VM heap	256 MB

API - application program interface; RAM - random-access memory; VM - virtual machine of every single process (for our purpose, we took only the information from the process that executed the *MosCla* 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 87 wing images from both classes were analyzed. The classification performance of the SVM classifier was outstanding using the 10-fold cross-validation method.

A. Classification performance

The AUC performance score improved while reducing the feature space from seven to three features. Using the whole space of seven features, the SVM classifier achieved a mean of AUC and absolute error scores of 0.95 and 0.04, respectively. On the other hand, using the reduced space of three features, the SVM classifier obtained a mean of AUC and absolute error values of 0.98 and 0.02, respectively. The AUC score difference of 0.03 could be interpreted as a small variation in terms of classification performance. But, considering the initial constrains of classifying on limited resources devices, it is possible to state that this variation is significant and beneficial to the proposed app.

The separation of both *Culicoides* species using the reduced feature space could be seen in Fig. 3. From this figure is possible to notice that by using only three features it is visually simple to find out a decision boundary (see Fig. 3, left plot). This assumption was also made by the SVM classifier, which wrongly classified only two samples in the reduced feature space versus the four miss-classified in the original space, respectively (Fig. 3, green samples in the right plot). Therefore, the use of the external feature selection wrapper method is an important addition to take into consideration for future mobile app developments.

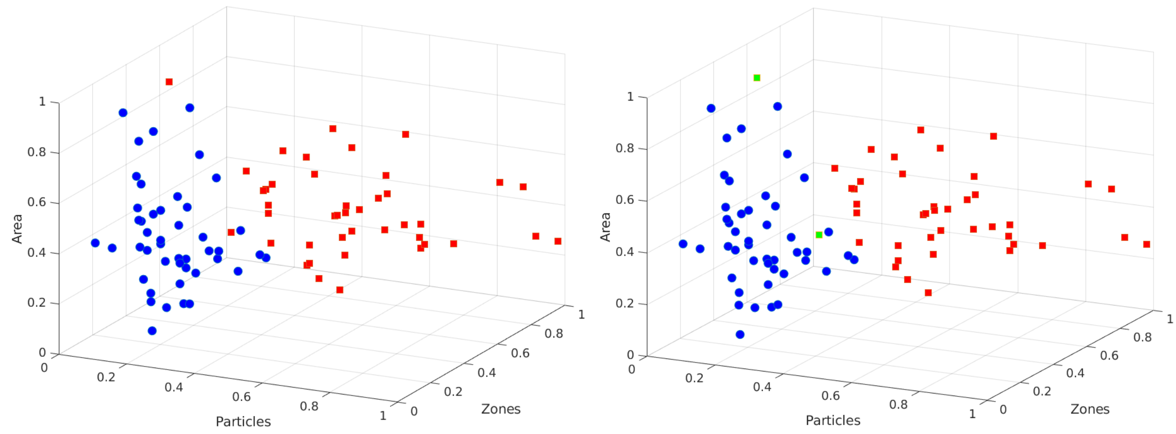

Fig. 3. Separation of both *Culicoides* classes using the three selected features space (left) and the SVM classifier (right).

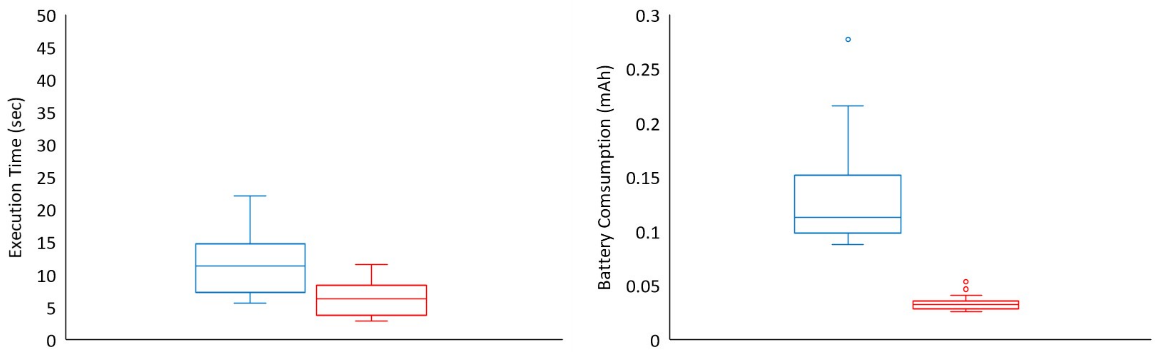

Fig. 4. *MosCla* app performance according to the mean of execution time (left) and battery consumption (right) on a phone Pixel 2 (blue box) and tablet Pixel C (red box) emulators, respectively.

B. MosCla app feasibility

Regarding the maximization of resources on mobile devices, we have not yet implemented the proposed app on the real tablet or phone device, and this is the practical limitation of this work. Instead of that, we approximated the time and battery consumption by using the established energy measurement protocol on two device emulators, in which we implemented the proposed app. Then, the experimental *Culicoides* wing images dataset was analyzed by the proposed app installed on each device emulator.

Results in terms of the mean of execution time and battery consumption for both emulators are shown in Fig. 4. As expected, the tablet Pixel C (red box plot) emulator was faster and has better energy savings than the phone Pixel 2 (blue box plot) emulator. However, both devices were able to reach mean scores under the 15 seconds and consuming only 0.15 mAh per wing image, which are considered a successful performance on both endpoints. These results can be generalized and provide a good approximation to those specialists that need a practical tool to classify *Culicoides* species in the wildlife environment.

IV. Conclusions and Future Work

In this work, we developed an Android-based mobile application called *MosCla* to classify two *Culicoides* species. The proposed app implemented an automatic classification method based on the calculation of morphological features extracted from wing's images and a SVM classifier to produce the final classification between the *Pusillus* or *Obsoletus* species. The application of an external wrapper method to reduce the features space from seven to three features improved the SVM classifier performance from an AUC score of 0.95 to 0.98, using a 10-fold cross-validation method on the experimental wing's images dataset.

Regarding the feasibility, the *MosCla* app reached satisfactory mean execution values of 12 and 7 seconds and a mean battery consumption values of 0.11 and 0.03 mAh for the Pixel 2 phone and the Pixel C tablet emulators, respectively. These results are can be generalized, and with further enhancements, the proposed app could be a valuable tool for those specialists that need a practical tool to classify *Culicoides* species in the wildlife environment. As future work, we plan to include other *Culicoides* species such as *Foxi* and *Insignis* to the experimental dataset. We also plan to implement other classifiers to explore their performance on limited-resource devices. Finally, we are also interested in testing the proposed app on real mobile devices to compare the time and battery consumption metrics against the results obtained by the emulated environments.

REFERENCES

[1] A. Borkent, "The subgeneric classification of species of culicoides," 2014.

[2] R. Szadziewski, S. Filatov, and P. Dominiak, "A redescription of culicoides griseidorsum kieffer, 1918, with comments on subgeneric position of some european taxa (diptera: Ceratopogonidae)," *Zootaxa*, vol. 4107, no. 3, pp. 413–422, 2016.

[3] R. Meiswinkel, T. Baldet, R. De Deken, W. Takken, J.-C. Delécolle, and P. S. Mellor, "The 2006 outbreak of bluetongue in northern europe—the entomological perspective," *Preventive veterinary medicine*, vol. 87, no. 1-2, pp. 55–63, 2008.

[4] D. R. Benalcázar, D. S. Benítez, S. Zapata, and D. Augot, "Geometrical characterization of dipterans' wings towards automatic mosquito identification," in *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Nov 2018, pp. 1–6.

[5] P. Rawlings, "A key, based on wing patterns of biting midges (genus culicoides latreille-diptera: Ceratopogonidae) in the iberian peninsula, for use in epidemiological studies," *Graellsia*, vol. 52, no. 11, pp. 57–71, 1996.

[6] J. Boorman and D. Hagan, "A name list of world culicoides (diptera: Ceratopogonidae)," *Int J Dipt Res*, vol. 7, pp. 161–92, 1996.

[7] F. Muñoz-Muñoz, S. Talavera, and N. Pagès, "Geometric morphometrics of the wing in the subgenus culicoides (diptera: Ceratopogonidae): from practical implications to evolutionary interpretations," *Journal of Medical Entomology*, vol. 48, no. 2, pp. 129–139, 2011.

[8] F. Muñoz-Muñoz, S. Talavera, S. Carpenter, S. A. Nielsen, D. Werner, and N. Pagès, "Phenotypic differentiation and phylogenetic signal of wing shape in western european biting midges, culicoides spp., of the subgenus avaritia," *Medical and veterinary entomology*, vol. 28, no. 3, pp. 319–329, 2014.

[9] F. Muñoz-Muñoz, J. Ramoneda, N. Pagès, N. Pujol, and S. Talavera, "Is the morphology of culicoides intersexes parasitized by mermithid nematodes a parasite adaptation? a morphometric approach to culicoides circumscriptus (diptera: Ceratopogonidae)," *Journal of invertebrate pathology*, vol. 135, pp. 1–9, 2016.

[10] L. H. Henni, F. Sauvage, C. Ninio, J. Depaquit, and D. Augot, "Wing geometry as a tool for discrimination of obsoletus group (diptera: Ceratopogonidae: Culicoides) in france," *Infection, genetics and evolution*, vol. 21, pp. 110–117, 2014.

[11] L. Hadj-Henni, T. De Meulemeester, B. Mathieu, J. Depaquit, and D. Augot, "Taxonomic assessment of culicoides brunnicans, c. santonicus and c. vexans (diptera: Ceratopogonidae) in france: Implications in systematics," *Infection, Genetics and Evolution*, vol. 33, pp. 324–331, 2015.

[12] D. Motta, A. Á. B. Santos, I. Winkler, B. A. S. Machado, D. A. D. I. Pereira, A. M. Cavalcanti, E. O. L. Fonseca, F. Kirchner, and R. Badaro, "Application of convolutional neural networks for classification of adult mosquitoes in the field," *PloS one*, vol. 14, no. 1, 2019.

[13] C. Kaufmann, F. Schaffner, D. Ziegler, V. Pflueger, and A. Mathis, "Identification of field-caught culicoides biting

midges using matrix-assisted laser desorption/ionization time of flight mass spectrometry," *Parasitology*, vol. 139, no. 2, pp. 248–258, 2012.

[14] A. K. Banerjee, K. Kiran, U. Murty, and C. Venkateswarlu, "Classification and identification of mosquito species using artificial neural networks," *Computational Biology and Chemistry*, vol. 32, no. 6, pp. 442–447, 2008.

[15] P. G. Ramirez, M. Stein, E. G. Etchepare, and W. R. Almiron, "Diversity of anopheline mosquitoes (diptera: Culicidae) and classification based on the characteristics of the habitats where they were collected in puerto iguazú, misiones, argentina," *Journal of Vector Ecology*, vol. 41, no. 2, pp. 215–223, 2016.

[16] G. Wang, C. Li, X. Guo, D. Xing, Y. Dong, Z. Wang, Y. Zhang, M. Liu, Z. Zheng, H. Zhang *et al.*, "Identifying the main mosquito species in china based on dna barcoding," *PLoS One*, vol. 7, no. 10, 2012.

[17] P. Venegas, D. Benítez, N. Pérez, S. Zapata, D. Augot and J. Rojo-Álvarez, "Automatic classification of culicoides species by learning the wings morphology," *PloS one*, 2020, Submitted for publication.

[18] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[19] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Pearson Education India, 2004.

[20] F. Meyer, "Topographic distance and watershed lines," *Signal processing*, vol. 38, no. 1, pp. 113–125, 1994.

[21] Y. K. Jain and S. K. Bhandare, "Min max normalization based data perturbation method for privacy protection," *International Journal of Computer & Communication Technology*, vol. 2, no. 8, pp. 45–50, 2011.

[22] N. P. Pérez, M. A. G. López, A. Silva, and I. Ramos, "Improving the mann–whitney statistical test for feature selection: An approach in breast cancer diagnosis on mammography," *Artificial intelligence in medicine*, vol. 63, no. 1, pp. 19–31, 2015.

[23] P. Venegas, N. Pèrez, D. S. Benítez, R. Lara-Cueva, and M. Ruiz, "Building machine learning models for long-period and volcano-tectonic event classification," in *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, 2019, pp. 1–6.

[24] A. Papadopoulos, D. I. Fotiadis, and A. Likas, "Characterization of clustered microcalcifications in digitized mammograms using neural networks and support vector machines," *Artificial intelligence in medicine*, vol. 34, no. 2, pp. 141–150, 2005.

[25] N. Pérez, P. Venegas, D. Benítez, R. Lara-Cueva, and M. Ruiz, "A new volcanic seismic signal descriptor and its application to a data set from the cotopaxi volcano," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11, 2020, accepted for publication.

[26] OpenCV Team, "Open Source Computer Vision Documentation," https://docs.opencv.org/3.4.8/d1/dfb/intro.html, accessed: 2020-04-08.

[27] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems*

and technology (TIST), vol. 2, no. 3, pp. 1–27, 2011.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[29] D. Augot, F. Sauvage, D. Jouet, E. Simphal, M. Veuille, A. Couloux, M. Kaltenbach, and J. Depaquit, "Discrimination of culicoides obsoletus and culicoides scoticus, potential bluetongue vectors, by morphometrical and mitochondrial cytochrome oxidase subunit i analysis," *Infection, genetics and evolution*, vol. 10, no. 5, pp. 629–637, 2010.

[30] F. G. López, M. G. Torres, B. M. Batista, J. A. M. Pérez, and J. M. Moreno-Vega, "Solving feature subset selection problem by a parallel scatter search," *European Journal of Operational Research*, vol. 169, no. 2, pp. 477–489, 2006.

[31] Developers Android, "Android Debug Bridge (adb)," https://developer.android.com/studio/command-line/adb.html, accessed: 2020-03-25.