ELAION: ML-Based System for Olive Classification with Edge Devices

Dimitris Theodoropoulos[1(✉)], Konstantinos Blazakis[3], Dionisios Pnevmatikatos[1,2], and Panagiotis Kalaitzis[3]

[1] Telecommunication Systems Institute, Technical University of Crete, Chania, Greece
dtheodoropoulos@tuc.gr
[2] Institute of Communication and Computation Systems, National Technical University of Athens, Athens, Greece
[3] Mediterranean Agronomic Institute of Chania, Chania, Greece

Abstract. Discrimination of morphological characteristics for olive fruits is widely used to quickly classify their cultivar. This process is usually based on visual observations, which require experience and often appear to be very subjective, inconsistent and inaccurate. Towards automating and providing an error-free procedure for olive fruit classification, this work presents ELAION, an end-to-end system for olive cultivar identification using edge devices, such as smartphones and tablets. An application utilizes the device's camera to send olive images to a back-end server for feature extraction. Results are relayed back to the application, which identifies the originally depicted olive cultivar using pre-trained machine learning models. As a result, ELAION greatly reduces the time and errors on olive fruit identification with on-site results, thus paving the way for becoming an on-site key-tool for olive growers, breeders, and scientists.

Keywords: Olive classification · Machine learning · Edge device application

1 Introduction and Motivation

Morphological markers are used for the identification and discrimination of olive germplasm. Discriminating morphological characteristics of olive fruits are commonly used for a quick cultivar identification based on appearance, but visual observations require experience and sometimes appear to be very subjective, inconsistent and inaccurate. Other approaches are based on manual techniques, such as using screw gauge or calliper and gridded paper, have been used for the morphological analysis of olives. Although there are methodologies [1,2,4,5]

This work (T2EΔK-02637) was co-financed by the Special Managing and Implementation Service in the areas of Research, Technological Development and Innovation (RTDI) - Greece, and the European Union.

C. Silvano et al. (Eds.): SAMOS 2023, LNCS 14385, pp. 459–464, 2023.
https://doi.org/10.1007/978-3-031-46077-7_31

that assist on the olive morphological analysis, automated tools for olive cultivar identification based only on the morphology are still in the very early stages.

ELAION targets to automate the procedure of on-site cultivar classification; it proposes a genuine end-to-end system that enables identification of olive varieties using everyday edge devices, such as smartphones and tablets. The ELAION front-end application, namely Eliapp, utilizes the device's camera to upload olive images to its image processing server (IPS), which extracts important features, such as nipple curvature, color, and area. Extracted features are sent back to Eliapp, which facilitates a pre-trained machine learning (ML) model to quickly classify the cultivar id. Eliapp is coupled with back-end serverless infrastructure that hosts a database, user information, and a set of ML models. As a result, ELAION has the potential to become a valuable tool for on-site olive classification for olive growers, breeders, and scientists, since it reduces the time and errors on olive fruit identification with direct and error-free results.

Overall, the main contributions are below:

- A front-end application that uses ML techniques to classify olives using the device camera (Sect. 2);
- a robust ML model that uses extracted features to classify up to 25 different olive varieties (Sect. 3);
- the ELAION infrastructure, a complete and modular system that allows users to quickly identify olives with only their smartphone (Sect. 3).

2 System Overview

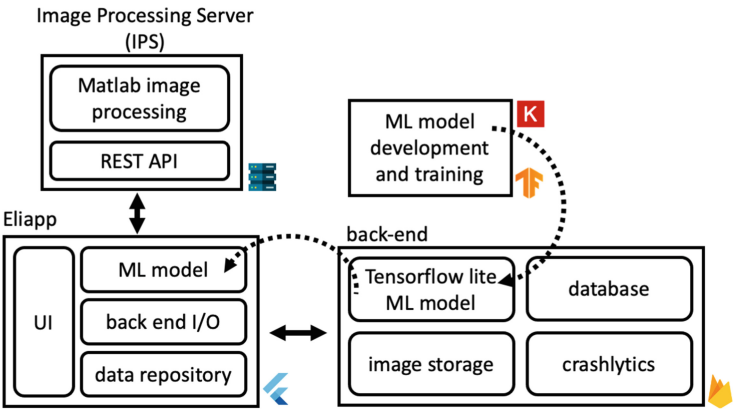

Fig. 1. Overview of the ELAION system.

ELAION is an end-to-end system that enables real-time olive classification using edge devices. As shown in Fig. 1, ELAION comprises three main components, an image processing server (IPS), a smartphone application, and finally

serverless back end support. The IPS exposes a REST API for communication with the Eliapp, i.e. receiving an olive image and sending back its features. The IPS also runs ELAION's image processing software [3] to extract all features. The latter are sent back to Eliapp, which utilizes the currently available ML model to classify the olive variety. Moreover, Eliapp is supported by a serverless back-end deployed on Google's Firebase infrastructure, which stores user data, past classifications metadata (e.g. location, variety, olive image), and the currently used ML model. The latter can be updated/enhanced with new training data anytime by ML developers and then uploaded to ELAION's back-end, allowing Eliapp to instantly use the latest ML model for olive classification.

Algorithm 1 Pseudo-code that leverages actual feature measurements to generate training data for a target cultivar.

```
Require: csvData, featureResolution
featuresCsv = csvData[:,0:20]
cultivar = csvData[:,21]
allMins = np.array([featuresCsv.min(axis=0)]).T
allMaxs = np.array([featuresCsv.max(axis=0)]).T
allSteps = np.array([((featuresCsv.max(axis=0) - featuresCsv.min(axis=0)) / fs)]).T
dataBundle = np.concatenate((allMins,allMaxs,allSteps), axis=1)
for row in dataBundle do
    tmpData = np.array([np.arange(start,stop,step)])
    featureData = np.concatenate((featureData,tmpData), axis=1)
end for
rows, columns = featureData.shape
cultivarCol = np.empty(shape=(rows,1))
cultivarCol.fill(cultivar[0])
finalData = np.concatenate((featureData,cultivarCol), axis=1)
return finalData
```

3 Prototype Implementation

Image Processing: As discussed earlier, the ELAION IPS exposes a RESTful API for receiving an olive image to be classified, and returns its features. The API is linked with ELAION's image processing software [3] using the MATLAB Production Server RESTful API for MATLAB functions. REST calls to the IPS require as input a base64-encoded image of an olive for classification. Extracted features are returned to Eliapp in a JSON format representation for the final classification.

ML Model: ELAION's ML models can be developed with any neural network development framework, as long as it can be converted to a TFlite representation. The current ML model supports 20 features (inputs), utilizes a hidden

layer with an activation function, and can classify 25 different olive varieties [3]. Moreover it was developed using the Keras deep learning framework, which leverages Google's TensorFlow ML platform.

Algorithm 1 lists the main steps for generating intermediate training data for a specific cultivar. Its input is (i) *csvData*, an array that contains a set of feature measurements for each supported cultivar, and (ii) the desired resolution, i.e. the number of generated values between the minimum and maximum measured value for a specific feature. In Lines 1–2, stored data from *csvData* are divided into the *featuresCsv* and *cultivar* arrays that contain all measurements from each feature and the cultivar id respectively. Lines 3–4 find the min and max values of each feature, and based on the desired *featureResolution*, line 5 calculates the feature step between consecutive values. Line 6 assembles all data into a single array. Lines 7–10 iterate *dataBundle* to generate the actual intermediate values for each cultivar feature, and save output data to array *featureData*. Line 11 applies the *featureData* dimensions to the *rows* and *column* values respectively. Finally, lines 12–14 generate the *finalData* array that contains in each column the generated intermediate values for a specific feature, whereas the last column designates the cultivar id itself.

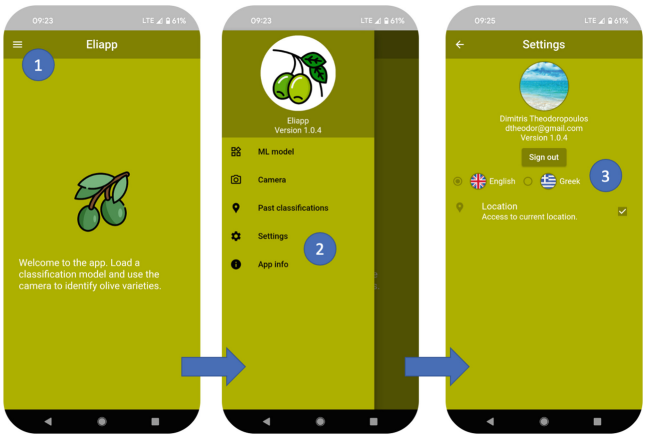

Fig. 2. From left to right: Welcome screen, navigation drawer with options, and application settings.

Edge Application: Eliapp is ELAION's edge device application for olive classification developed with Google's Flutter framework. The application comprises five screens, two screens are used for navigation and settings configuration (Fig. 2), and three screens, as shown in Fig. 3, are used as follows: The left screen allows users to type the ML model name to be used for olive classification, and download it from the back-end (1), and test it against a predefined input set (2). The middle screen turns the camera on and shows its preview in area (3). When

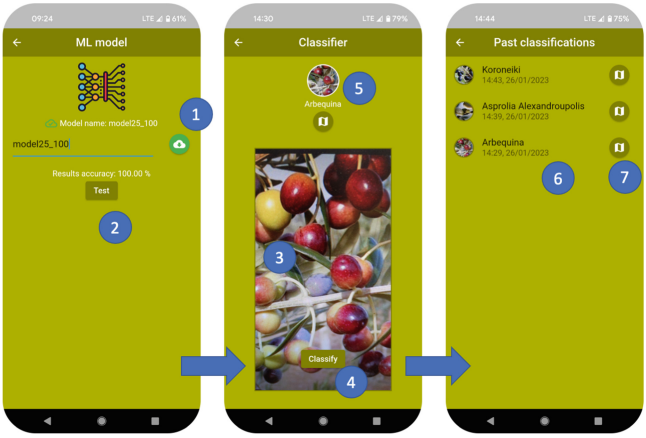

Fig. 3. From left to right: the ML model download, camera classifier, and past classifications screens of Eliapp.

tapped, the "classify" button takes a snapshot, which is uploaded to the IPS. When feature extraction is complete, results are sent back, and Eliapp classifies the olive using the configured ML model from the previous screen, and shows its name and image in area (5). Finally, all classification metadata are logged to the back-end. Past classifications metadata can be accessed as a list as shown in the right screen (6). Eliapp provides also a convenient button (7) that designates the exact location on Google Maps for a particular olive classification.

4 Conclusions and Future Work

As shown, ELAION is a full system that allows identifying olive varieties using everyday edge devices, such as smartphones and tablets. Eliapp (its front-end application) utilizes the device's camera to capture and upload olive images to the image processing server for extracting important morphological features, such as nipple curvature, color, and area. All extracted features are transmitted back to Eliapp, which utilizes a pre-trained model to quickly classify the capture olive variety. Experimental results showed that the is more efficient in terms of accuracy to utilize a simple LU activation function in the hidden layer, increasing though in most cases training time. Finally, Eliapp is coupled with Google's Firebase back-end serverless infrastructure that hosts a database, user information, and a set of ML models.

Our future work focuses on publishing Eliapp to the Play Store and App Store, where users can download it to publish their on-site results on an open database. Overall, Eliapp's has been designed with simplicity in mind, so users can easily and quickly upload olive images to our back end, and instantly receive the identified cultivar back. As such, ELAION can be a valuable on-site asset for olive growers, breeders, and scientists.

References

1. Belaj, A., et al.: Developing a core collection of olive (Olea europaea l.) based on molecular markers (DArTs, SSRs, SNPs) and agronomic traits. Tree Genet. Genomes **8**, 365–378 (2012)
2. Rugini, E., Baldoni, L., Muleo, R., Sebastiani, L.: The Olive Tree Genome. Compendium of Plant Genomes, Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48887-5
3. Blazakis, K.N., et al.: Description of olive morphological parameters by using open access software. Plant Methods **13**, 1–15 (2017)
4. Baldoni, L., et al.: A consensus list of microsatellite markers for olive genotyping. Mol. Breed. **24**, 213–231 (2009)
5. Mousavi, S., et al.: Molecular and morphological characterization of Golestan (Iran) olive ecotypes provides evidence for the presence of promising genotypes. Genet. Res. Crop Evol. **61**, 775–785 (2014)