MushDex: A Pathway to Real-Time Detection and Classification of Mushrooms Using Raspberry Pi Zero 2W

Chiara D'Amato
MSc in Engineering Physics
Politecnico di Milano
Vittuone, Italy
chiara.damato@mail.polimi.it

Edoardo Torre
MSc in Engineering Physics
Politecnico di Milano
Merate, Italy
edoardo.torre@mail.polimi.it

Lorenzo Vergata
MSc in Engineering Physics
Politecnico di Milano
Milan, Italy
lorenzo.vergata@mail.polimi.it

***Abstract*—This research aims to find a solution that helps hikers and encourages mushroom enthusiasts to take walks in the woods in order to improve their health. The main goal of the project is to provide smart experimental support for hikers, allowing them to focus solely on the risks of their journey while leaving the real-time detection and classification of mushrooms in the surrounding area to MushDex.**
Thanks to optimal illumination and camera rotation provided by powerful LED and MEMS systems and the access to the largest available mushroom image libraries, a neural network model is able to perform real-time inference to detect and classify the mushrooms present in the scene. The device alerts the user only when a mushroom species appears, allowing them to enjoy their walk without distractions.
This work is to be understood as an investigation of the potential of a portable and inexpensive device in the field of mushroom object detection, and as such, it does not want to replace in any way the work of a mycologist. Thanks to the multi-output functionality and the vast amount of classes present in the problem addressed, the device can be a support for a further search for the genera and species of fungi, possibly with a DNA analysis.

***Index Terms*—classification, dataset, detection, mushroom, raspberry.**

I. Introduction

In the year 2023, the data provided by the Alpine and Speleological Rescue Activities show 12.365 people were rescued because of hiking-related problems; of these people, 7.622 were injured and 491 were found deceased [**?**]. These numbers increase when adding the data from the Department of Health Activities, which recorded 10.000 Italians per year experiencing a range of medical cases from mild to more severe, with symptoms including gastrointestinal issues, neurological complications, and even death.

Unfortunately, the events of COVID-19 have made people more comfortable staying at home, causing deterioration of the general mental well-being, especially for those living in large concrete cities. Taking a walk in the forest is highly beneficial for health and helps in reducing stress. The idea behind MushDex is to leverage people's interest in mushrooms and to stimulate them to take forest walks in a safer way. While we cannot directly solve the issues linked with the environmental dangers—that responsibility falls to the foresters of each country—we can help individuals focus more on their journey by allowing them to know where they could find mushrooms. In this way, they are more aware of their surroundings and concentrated on avoiding potential forest dangers. Using MushDex, a person can walk free of negative thoughts.

In order to achieve our goal, we trained the state-of-the-art object detection network YOLOv10 [**?**] on reliable custom datasets to generate boxes on a set of photos obtained by combining datasets. Due to the nature of this combined collection, we first needed to train different classification models to label the mushrooms of unknown species but known genera. This was essential to prevent data loss and to organise the photos into the right folders, named after the probable species. As if that were not enough, we used the same model to replace the "mushroom" label with the most confident prediction for each box in the dataset. By combining all the images, now containing boxes around mushrooms and labels with their species' names, we created the largest collection of mushroom pictures for the training of detection neural networks, with the largest number of classes for a problem of this kind.

Since our challenge is to implement the trained detection model on a pocket-sized device, specifically the Raspberry Pi Zero 2W [**?**], a more lightweight model was trained on our dataset to enable a fast enough inference on this device. The system takes as input the real-time images from a detection camera, accompanied by an LED light illuminating the area when the ambient light is not sufficient. In this way, the hardware can be self-consistent with the environmental conditions. An inertial system communicates with the camera, applying the right transformation to analyse the frames in an ideal condition. Finally, the device communicates the genus and species of the identified mushrooms to the user through a speaker.

After an overview of the related works regarding the problem, we present the guidelines for the creation of the dataset used in

this study, explaining our approach when dealing with large-scale downloading of photos on the web. The problems of data scarcity, data quantity, and unbalanced data are also addressed. Next, we talk more in detail about the neural network models used, justifying the preferred choices for the different trainings. After that, we describe the image capture capability of our device, the sensor used for lighting the scene, and the audio output.
Finally, the section dedicated to experimental results considers the measures to be implemented and problems to be solved, opening the doors for future ideas and work.

II. Related Works

The project takes inspiration from the "2018 FGCVx Fungi Classification Challenge," in which the main goal was to identify 1,394 fungi species given around 90,000 images for model training []. Instead of closely following the aim of the challenge, we are going to classify the genera of the mushrooms present in the picture input that is given by the user, basing the analysis on a dataset built starting from a variety of sources.
There exist accessible datasets for projects about fungi, for example in [**?**] and [**?**], and these are fundamental instruments for building robust models.
Considering the 2019 work of J. Preechasuk et al. [], the aim was to classify 45 mushrooms given images of size 96 × 96 pixels, reaching an accuracy of 74%. In our work, we scan larger pictures of 640 x 640 pixels to train the neural network. However, while we have higher resolution, our device performs a real-time inference instead of working on fixed images, so it is affected by motion noise.
In the 2018 work of Kang et al. [], a small dataset of 1,478 images was used to classify 38 species of mushrooms, reaching an accuracy of 82%. On the other hand, we use a bigger dataset, looking to obtain an accuracy score close to the one above.
The 2019 study by M. A. Ottom et al. [] also accounted for the real-world sizes of mushrooms, achieving a peak accuracy of 94%. When relying only on visual features extracted from images, the accuracy dropped to 86%. While this suggests that our real-time detection may not surpass these results, a future implementation could be considered: users might be able to connect their Raspberry Pi to a personal computer and input additional information, such as precise measurements of a mushroom flagged as potentially interesting by MushDex, to obtain a more accurate classification.
[**?**] is a similar study, although the classes it considers are entirely different, and its dataset contains fewer images than the number of classes we are using in our project. Nevertheless, it serves as an example of real-time mushroom detection on an embedded system, specifically our test device: if something cannot run on a Raspberry Pi 400, it is also unable to run on a Raspberry Pi Zero 2W.
[**?**] concerns a different kingdom but provides a solid example of object detection and classification similar to what we are aiming for.
Speaking of available applications for the identification of mushrooms, most of them are not open source and fall short in both the number of species covered and the level of accuracy required for real-time inference.
An example of a paid mushroom identification app is Picture Mushroom, which uses multi-input photos to produce an output. Unfortunately, some edibility labels are incorrect, and because this classification is different across the nationalities, we have no interest in it, limiting ourselves only to the species recognition [**?**].
Mushroomizer is a good real-time app for Android only, but it is not open source, and detection carried out using a smartphone while walking can be risky.
Shroomify is an encyclopaedia of mushrooms and not an unsupervised detection app.
Riconoscere Funghi-Identific is a valid free application, but it performs poorly during real-time scanning, as it primarily focuses on photo-based classification rather than live detection.
ShroomID is a solid classification app, and notably, FungID is also open source. However, both require the user to manually detect mushrooms, get close for a photo, and sometimes even pick the mushroom, which can disturb nature and expose the user to potential mycotoxin poisoning. Moreover, the user is distracted from the journey in the forest because their attention is redirected to mushroom detection, increasing the risks connected to hiking.
To conclude, our dataset is possibly the hardest benchmark for the fungi kingdom available, and the methods and the scale of the problem are truly unique. This underscores the difficulty of detecting and classifying all known mushroom species using only photos; biological analysis is required to reach a confidence level close to 100%. Currently, the most accurate technique for mushroom recognition is DNA analysis from a small sample, but this process requires a specialised machine, which is typically bulky. In contrast, we are going to use a lightweight, portable smart device for the recognition task.

III. Methodologies

A. Hardware approach

In order to solve our problem, we identify the needed devices to connect to the Raspberry Pi Zero 2W. This is a device based on a 1 GHz Quad-core 64-bit Arm Cortex-A53 CPU, with 512 MB of SDRAM necessary for the real-time inference and a Bluetooth antenna required for the connection with the external speaker. The power supply is provided by a user's power bank equipped with a proper connector.
To carry out the real-time image detection, we used the Raspberry Pi Camera Module 3, provided by Raspberry Pi Ltd. and based on Sony technology. The camera has as its main features for our purpose a focus range from 10 cm to "enough", a 12 Megapixel sensor, and a field of view of 70 degrees, allowing us to scan the surrounding area. It is noted the sensitivity needed for our goal is much lower.
In order to lighten the area in case of low environmental illumination detected by a photodetector, we use an external LED

light connected to our device. Because we could not obtain a sufficiently powerful light from already available devices, we used a store-bought LED; we dismantled it and welded it to our device. The photodetector was already available to buy online.
We exploited the Bluetooth feature to connect a recovered external speaker, which allowed the device to spell the detected mushroom's name.

B. Dataset creation

A bad dataset is directly associated to a bad neural network model. In order to prevent this and to all reproducible, we report the steps[1]:

1) download all the images through [**?**].
2) download [**?**], and combined it with the first one.
3) do the same with [**?**].
4) download [**?**]. Some edibility classifications are wrong but we need only the images: move all the subfolders in the same directory, and rename the folders replacing "_" with a blank space. It is possible to find the script we used in **github**. Keep in mind some species names are common American ones.
5) download [**?**] and after replacing "_" with " ", combine it with the dataset of the previous step.
6) now it is the turn of [**?**]: combine it with the dataset created in the previous step.
7) do the same with [**?**].
8) [**?**] has entered the chat in the previous dataset.
9) [**?**] is to integrate with the dataset too.
10) the same for [**?**].
11) now [**?**].
12) [**?**] is a bit tricky: there is the python code used to create the directories consistently with the others, paying attention to not overwrite the files with the same name (but in this case totally different photos, not repeated by other photos with other names). Then, we hard-moved to the dataset that will be merged with the one created in the step 2).
13) now download and combine the genera datasets [**?**], [**?**], [**?**] and [**?**].
14) finally, download on a different folder all the datasets in [**?**]. We downloaded the 300px max side size due to space reasons.
15) using the metadata file, reorganize the photos in such a way

REFERENCES

[1]It is not relevant which image is to maintain in case of namesakes images during the transferring process: almost all of these photos are present in previous datasets