AN INTELLIGENT PLANT AND ANIMAL IDENTIFICATION MOBILE APPLICATION FOR INCREASED BIODIVERSITY AWARENESS AND SAFETY USING MACHINE LEARNING

Austin Xiao[1], Ang Li[2], Wyatt Bodle[3]

[1]Troy High School, 2200 Dorothy Ln, Fullerton, CA 92831
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768
[3]California Baptist University, Riverside, CA 92504

ABSTRACT

Dangerous animal encounters have steadily increased over time and consumption of deadly plants is an important issue [12]. Our paper introduces a new mobile application that addresses the critical need for accurate animal and plant identification and classification to help mitigate safety risks for humans.With up to five million animal attacks reported every year in the United States alone, and over 100,000 cases of toxic plant exposure there is a need and a responsibility to increase awareness of the risks associated with animal and plant ignorance.Our proposed app utilizes innovative classification technologies, offering our users a swift and simple identification of both select plant and animal species.The app will relay information about the potential dangers and general facts about the classified animal [14].This will help our users to understand the environment they live in and to best prepare themselves against it.Some challenges with this proposal are curating a broad and efficient dataset, there are estimated to be eight million eukaryotic species which is unattainable for one dataset.We then had to decide which valuable information would be best to present without providing unwanted distractions in our user interface.We utilized Google Firebase to ensure secure authentication and data storage while using TensorFlow Lite to power the image classification.We then integrated all of this into flutter to create a friendly user interface and application that can run on both iOS and Android [15].Once our app was complete, we ran two experiments, one to test the accuracy of our classifications in plants and animals and another to test the effect of lower resolution images on classification accuracy.The experiments shed light on challenges and potential improvements for the application to help improve its efficiency as a tool for users to enhance their awareness, safety and understanding of the environment they live in.

KEYWORDS

Wildlife Identification, Plant and Animal Classification, Mobile Application, Biodiversity Awareness, Safety Technology

1. INTRODUCTION

The accurate identification of animals and plants is critical for influencing safety, biodiversity, ecology, and public health.Statistics from 2017 indicate the severity of animal attacks on humans where there were up to five million incidents in the United States.There were two million bites, some of which were fatal or caused life threatening injuries[3]. Along with animal incidents,

David C. Wyld et al. (Eds): AIBD, MLSC, ACSTY, NATP, CCCIoT, SVC, SOFE, ITCSS -2024
pp. 417-427, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140433

there are over 100,000 cases of humans exposed to toxic plants annually, leading to human health risks[7]. It is crucial to address these issues relating to plants and animals which often stem from a lack of awareness or understanding.People tend to approach animals based on their appearances rather than the knowledge of their natural instinct or levels of danger.Exposure or ingestion of harmful plants is often driven by ignorance of those who undermine the dangers of some plants [1]. Children also frequently ingest poisonous plants which can turn out to be fatal. Reckless or ignorant travelers and tourists may come in risk of animal attacks from unawareness and lack of safety precautions[6]. People with a lack of knowledge of the animals they encounter are more likely to provoke animal incidents. Our app aims to solve those issues to increase awareness of the user's environment in the sense of plants and animals.

There are many methods on the techniques to create and train models to identify different species accurately. Norouzzadeh et al. devised a method to identify wild animals in camera trap images with deep learning. They attempted to help in animal identifications and save a large amount of time for biologists and volunteers on labeling images while at the same time performing at a high accuracy level. Although their model was effective at identifying animals with an accuracy over 93.8%, it had limitations through its inability to automatically handle multispecies images and the nonuniform dataset of animals. G. Komarasamy et al.'suse of Convolutional Neural Networks demonstrates very promising results in animal classification, but as is the case with most neural networks there is significant room for improvement [4]. They faced limitations with intra-class heterogeneity and human efficacy.Our app provides a very user-friendly mobile app, fostering community interaction to allow users to share discoveries and gain mutual assistance which combats the issues we addressed faced by Komarasamy et al. To address the issue of analyzing large amounts of images from camera traps, Micheal et al. trained machine learning models using neural networks with TensorFlow. While their model performed well at identifying the correct species with 97.6% accuracy for the top prediction, it performed worse for species and groups with less available training images [5]. To improve upon our project, we implemented an intuitive and navigable interface to allow users to easily identify animals or plants. We created a fairly consistent dataset of the animals and plants to ensure our model was adequately trained on all of them.

We propose a mobile application allowing users to quickly identify plants and animals as well as obtaining important information of the species they encounter. The development of an innovative plant and animal classification technology in the palm of a user's hand is important for allowing users to remain safe when in contact with dangerous conditions. The app is an effective solution because it enhances people's knowledge and understanding of the biodiversity they may come in contact with. The app facilitates quick classifications of plants and animals while also providing the user with a list of information on the given organism.The information given on each identification includes a description, advice when encountering it, dangers, risk level, invasiveness, their habitat, and what their diet includes. This information allows the user to make informed decisions on how to interact with their environment. Compared to alternative methods of field guides, online searches, or guessing, the app provides a quick, easy and accurate output.Using a field guide or online search leads to the high chance of human error and can be time consuming.In time-sensitive scenarios, users do not have the time to complete an online search or browse a field guide.Our solution provides a quick and easy way to identify an animal or plant, as easy as snapping a single picture.The user-friendly interface even furthers the speed at which a user can identify and inform themselves on an organism.The app is also developed for both iOS and Android making it very accessible to all who own a smartphone.

The purpose of both of the experiments we have done is to determine the limitations in accuracy and confidence with a machine learning image classifier using TensorFlow Light [8]. The first experiment dealt with the accuracy of each plant and animal with different images which showcased a wide variety of different lightings, resolutions, colors… etc. Our second experiment dealt with the problem of bad resolution or a blurred camera lens.The purpose of this was to determine how badly the accuracy would be hurt if the image passed to our algorithm was blurred.We found that animal accuracy was reliable and consistent with a good accuracy of over 70% while plants had a lower accuracy as was expected at close to 35%.This is most likely due to plants having less visual markers that differ from one another for a computer to distinguish. This is similar to how a human brain has a harder time identifying plant species than animals when only given a visual [9]. During our blurr test we had some interesting results with the animals, we actually observed that animals tended to have a higher accuracy when blurred.Plants followed the expected trend with an almost 10% average reduction in accuracy for the blurred images. We attribute this to noise reduction for plants which allows more distinctive features to show. The reason we believe this did not benefit the plants is because plants are less distinctive, where a parrot might have multiple different distinguishable colors and parts a plant might have two.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Populating Animal And Plants Data

Populating effective animal and plant data to train our dataset using machine learning was crucial for allowing the program to accurately distinguish between different organisms. Scientists have predicted around 8.7 million eukaryotic species inhabit Earth, meaning it would be infeasible and beyond the scope of our research to gather the data and provide information for each specific species. Thus, it would need to be a broad species of animals and plants. Gathering clean data would be important in ensuring the animals and plants could be correctly identified. There should be a sizable amount of training data for each organism, with the images being clear, in frame, and consisting primarily of the animal or plant. This would allow for more accurate classifications or identifications. If the training data was bad, this could lead to the dataset being improperly trained and providing incorrect identifications.

2.2. Ensuring The Provision Of Valuable And Relevant Information About Identified Animals And Plants

Our solution would provide no real world benefit if it was not for the information displayed about an identified organism.For this reason another challenge is ensuring the data we choose to display is both valuable and relevant.Too much information may lead to the same issue as using a field guide or a google search where the user spends too much time locating the information they want.If a user is spending too long reading it could potentially be fatal in a dire situation where the user needs answers immediately.Too little information may leaveusers in the exact same state of ignorance.The user would also waste time using the app if it did not include relevant information to the behavior of the organism.We would use anywhere from five to eight pieces of short relevant information allowing the user to quickly scan the data and determine their next steps.Smartphones are inherently distracting especially among younger adults which is why we could also minimize the amount of distractions on screen and only provide the information without any other distracting factors [11].

2.3. Using Our Dataset To Enable Accurate Discrimination Of Various Organisms Through Image Recognition Technology

The dataset would need to be accurately trained to allow for the image recognition technology to correctly identify different plants and animals. Increasing the number of epochs would help in making it more accurate, although it should not be more than a necessary amount, as this could lead to overfitting. The time the dataset is trained should be a considerable amount to allow for the model to be fairly accurate in its classification. The accuracy of the image recognition and classification is also reliant on the training data being accurate. It would be important to ensure there was enough high-quality data for the animals and plants so the dataset could be trained correctly.

3. SOLUTION

Our program begins with an authentication page powered by Google Firebase, a platform for app development allowing for authentication, data storage, as well as many other tools used to grow your apps.Firebase authentication easily integrates user sign-up and sign-in functionalities by securely storing a user’s credentials.Once a user has been authenticated by logging in using their email, username, and password they are brought to our home page.Our home page keeps track of and stores data on recognized plants and animals the user discovers through our app.Plant and animal tracking functionality is powered by Firebase’ Realtime Database, an easy way to read and write data in real time.The next tab is an animal classification tab where the user has the choice to take a picture or get an image from their camera roll for processing.The image is processed using TensorFlow Lite, a open-source machine learning framework developed by Google.Once the image is processed the app will pull data from a JSON file containing information on the species found. Similarly, there is a tab for plant classification that follows the same logic.We have also implemented a community page allowing users to connect and share information about the species they encounter with the ability to flag and block users for their own safety. There is a profile settings tab at the end of the navigation bar which allows the user to customize their profile picture, change their username, and keep track of their blocked users.User’s may also securely logout or delete their account data in this tab.

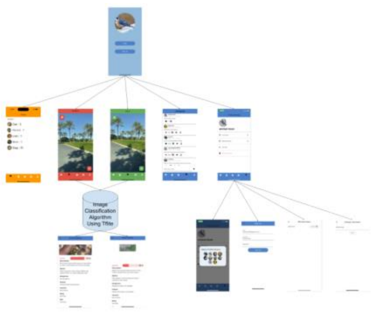
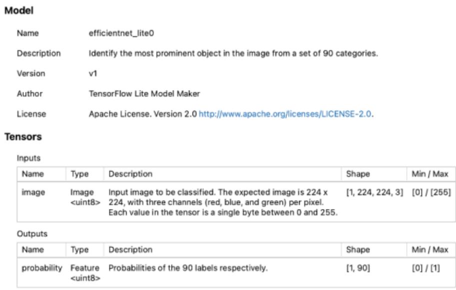


Figure 1. Overview of the solution

The image classification in our app is powered by TensorFlow Lite (Tflite) which enables efficient image classification in Flutter using a plugin.It integrates models we have trained on a vast database of plants and animals.Tflite allows for quick, real-time predictions without requiring access to the internet ensuring our users can identify species even when out of cellular range.



Model

Name	efficientnet_lite0
Description	Identify the most prominent object in the image from a set of 90 categories.
Version	v1
Author	TensorFlow Lite Model Maker
License	Apache License. Version 2.0 http://www.apache.org/licenses/LICENSE-2.0.

Tensors

Inputs

Name	Type	Description	Shape	Min / Max
image	Image <uint8>	Input image to be classified. The expected image is 224 x 224, with three channels (red, blue, and green) per pixel. Each value in the tensor is a single byte between 0 and 255.	[1, 224, 224, 3]	[0] / [255]

Outputs

Name	Type	Description	Shape	Min / Max
probability	Feature <uint8>	Probabilities of the 90 labels respectively.	[1, 90]	[0] / [1]

Figure 2. UI Screenshot

```
Future<void> _loadModel() async {
  final options = InterpreterOptions();

  // Load model from assets
  _interpreter = await Interpreter.fromAsset(_modelPath, options: options);
  // Get tensor input shape [1, 224, 224, 3]
  inputTensor = _interpreter!.getInputTensors().first;
  print(inputTensor);
  // Get tensor output shape [1, 3]
  outputTensor = _interpreter!.getOutputTensors().first;
  print(outputTensor);
}

Future<void> _loadLabels() async {
  log('Loading labels...');
  final labelsRaw = await rootBundle.loadString(_labelPath);
  _labels = labelsRaw.split('\n');
}

Future<List> analyseImage(String imagePath) async {
  log('Analysing image...');
  // Reading image bytes from file
  final imageData = File(imagePath).readAsBytesSync();

  // Decoding image
  final image = img.decodeImage(imageData);

  // Resizing image fpr model, [224, 224]
```

Figure 3. Screenshot of code 1

The code shown in figure___ displays loading our Tflite model for a classification.Tflite uses machine learning concepts to do tasks such as image classification and natural language processing.It first creates an instance of the InterpreterOptions class.We then create an interpreter variable that using the options and path to our Tflite model will create an interpreter for our classification. The interpreter executes these models locally, enabling on-device inference.Then we give it the input tensor which is an image of type uint8.And then load the output type for later and echo both the input and output tensor to the console for debugging.We also have a second function called loadLabels which takes the provided labels in the form of a .txt file and gives us an output based on the image classification completion.Lastly, we have the analyzeImage function which will determine given the image what type of plant or animal it is.This is then returned to our main program to display to the user.

The purpose of the frontend for our app is to render components of the app that the user can see and interact with. The login pages. The classification page takes the plant or animal name that has been identified by the Tflite model and uses a JSON file containing data for each of the animals and plants in our database to match the name with its associated information, such as its description or habitat. An accuracy percentage is also displayed in an accuracy bar to indicate the model's confidence in its classification.

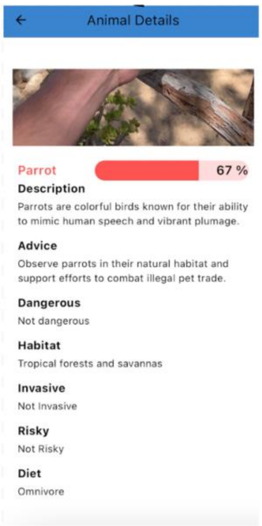


Figure 4. Animal details

```
),  // Row
buildAnimalProperty(
    'Description',
    listOfJsonMap[index]?['description'] ??
        'Description not available'),
SizedBox(height: 15),
buildAnimalProperty(
    'Advice',
    listOfJsonMap[index]?['advice'] ??
        'Advice not available'),
SizedBox(height: 15),
buildAnimalProperty(
    'Dangerous',
    listOfJsonMap[index]?['dangerous'] ??
        'Dangerous status not available'),
SizedBox(height: 15),
buildAnimalProperty(
    'Habitat',
    listOfJsonMap[index]?['habitat'] ??
        'Habitat information not available'),
SizedBox(height: 15),
buildAnimalProperty(
    'Invasive',
    listOfJsonMap[index]?['invasive'] != null
        ? (listOfJsonMap[index]!['invasive'] as bool
            ? 'Invasive'
            : 'Not Invasive')
        : 'Invasive information not available'),
```

Figure 5. Screenshot of code 2

This code showcases one of our many build methods which render information onto our users' mobile devices.This build function specifically uses a Widget of buildAnimalProperty which we designed to display one section of data from the JSON data file.Each buildAnimalProperty widget has a text that will display a specific section of data, for example, descriptions, advice, danger levels, habitats,and if they are invasive.They are then chained together with space between them to show one cohesive list of data all formatted and structured for the user to easily read and interpret the data.If data is not found for a given section, it will inform the user that that section is not currently available.We also for boolean values have a ternary operator which allows us to convert our true or false values to something more easily interpreted by the user such as "Invasive" if the animal or plant is invasive or "Not Invasive" if they are not.Each of our screens has a distinct and separate build method that will render different parts of our data to the user.

Firebase is essential to the success of this app.Firebase particularly serves as a way for us to store user-generated content such as our posts or user data.Using Firebase Realtime Database has ensured for seamless and real-time synchronization of posts across different accounts and devices.This enhances the experience of the user by providing data storage where a user can track which plants and animals they have classified as well as talk with other users on the danger of plants or animals they have discovered.Firebase database provides an easily scalable backend which simplifies data management, enlightening the user experience.

Figure 6. Flagged content

```
void getData() async{
  print("GETTING DATA FOR THE HOME PAGE");
  _auth = FirebaseAuth.instance;
  final user = _auth.currentUser;

  //Populate animal map
  final animalUsersEvent = await databaseReference.child('users').child(user!.uid).child('foundAnimals').once();
  final animalUserSnapshot = animalUsersEvent.snapshot;
  final animalUsersData = animalUserSnapshot.value;

  if(animalUsersData is Map<dynamic, dynamic>){
    animalMap = animalUsersData;
  }
  //Populate plant map
  final plantUsersEvent = await databaseReference.child('users').child(user!.uid).child('foundPlants').once();
  final plantUserSnapshot = plantUsersEvent.snapshot;
  final plantUsersData = plantUserSnapshot.value;

  if(plantUsersData is Map<dynamic, dynamic>){
    plantMap = plantUsersData;
  }
  setState(() {});
}
```

Figure 7. Screenshot of code 3

The code snippet shows a function for reading data from our database called getData.This function specifically populates the data and displays found plants and animals as well as their count on the home page.It creates an instance of the current user's authentication allowing for user specific data.The next line gets the current user from that instance of Firebase Auth.After we have gotten our user we then move on to securing this user's data and proceed to processing it.We have split this into two sections, populating data for animals and populating data for plants.It will asynchronously retrieve data from the Firebase Realtime Database under the paths we have provided.The retrieved data is in map format and is then populated into the two different map variables animalMap and plantMap. This ensures that the user-generated content is quickly and cleanly assimilated into our application. We then call the setState() method which ensures that the home page will reflect the data that we have retrieved.

4. EXPERIMENT

4.1. Experiment 1

For this experiment we will be choosing a random set of twenty animals and twenty plants.We will then feed the app classification three images of each animal or plant and determine if it did in fact accurately identify the animal and how high the confidence level was.If the classification is incorrect the confidence was scored at a 0.When selecting an image for classification we randomly picked an image off our search engine of each of the twenty plants and animals.The experiment was set up this way because in order to test the accuracy of the animal or plant we would have to feed it different images from different sources and different angles.By feeding images found on the internet we could ensure that all variables were changed in the image.For example, two distinct images on google will not have the same amounts of light, distance, or resolution if they are unrelated to each other.This would mean that the image classification algorithm would truly be tested to make sure it is reliable and accurate.

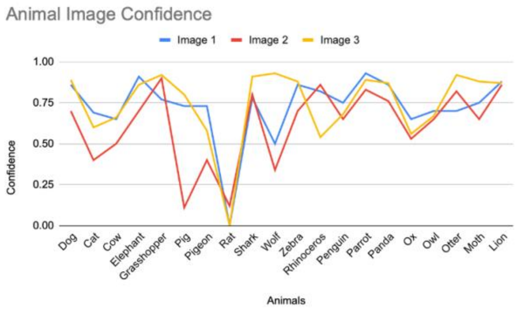


Figure 8. Animal image confidence

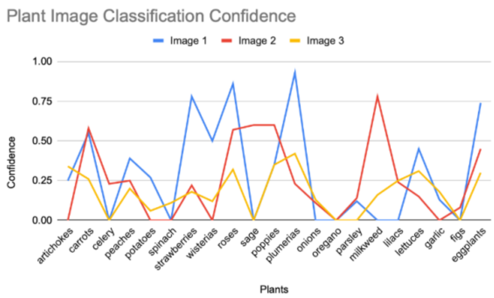


Figure 9. Plant image classification Confidence

Our data for plants and animals varied drastically which is what we would expect from an image classification algorithm. Animals are more distinct from one animal to the other than plants are which makes it harder for the machine learning algorithm to determine which animal it is. We see this in our data with the mean for our animals being 76.39% while plants were a mere 34.27%. Aside from the fact that plant accuracy is difficult to achieve, different animals or plants had different confidence and accuracy levels from each other. For example, when given three pictures of rats, the image was classified as a mouse every time. This is another issue similar to the inaccuracy of plants in that rats and mice are extremely similar and it is very difficult for a computer to determine one over the other. Distinctive animals or plants are easier for the model to determine, for example, a parrot had almost 100% accuracy and confidence. For animals a carrot was the most distinctive and recognizable plant most likely due to its vibrant colors and shape. The factors for making accuracy better and the reason some of our data was inaccurate is most likely due to similar visual features, limited training data, background noise, and other visual factors.

4.2. Experiment 2

Another possible blind spot with using our app is that different resolutions or different smudges on the lens may change the accuracy of our data. If a user has an older phone with a worse camera or if someone smudges their lens before taking a picture we want to determine if the app will still be reliable.

Another possible blind spot with using our app is that different resolutions or different smudges on the lens may change the accuracy of our data. If a user has an older phone with a worse

camera or if someone smudges their lens before taking a picture we want to determine if the app will still be reliable.

Clear **Smudged**

Figure 10. Clear and smudged picture

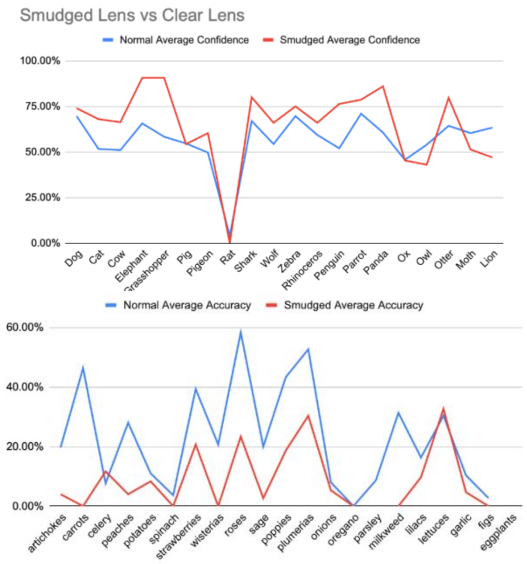


Figure 11. Smudged lens vs clear lens

Our results with the blurred vs clear images revealed that the smudged or blurred images for animals actually on average have a higher image accuracy than that of the clear lens. A clear lens for animal data provided a greater average accuracy and confidence by almost 10%. We see the opposite in our plant data, average accuracy drops from 34.27% with a clear lens to 24.2% with a smudged lens. This was extremely surprising because we expected to see the accuracy greatly decrease in both cases, not just with plants. One of the factors that could have led to smudged or blurred images having better results could be that there is a significant noise reduction when blurring an image. This would change the variation in color between pixels and allow the model to view only the distinctive features that have not been reduced. Plants when blurred, however, lose their distinctive features because most plants are green and when blurred the shapes of the leaves, stems, and other parts of the plant are lost.

5. RELATED WORK

Norouzzadeh et al. used deep neural networks to automatically identify and extract features of animals from motion-sensor camera traps. Their deep neural network could identify animals with >93.8% accuracy. They created a training set consisting of “284,000 capture events and two test sets.” In their work, they applied their method to the world’s largest dataset of wild animals called the SS dataset. They prevented the issue of their model not generalizing well to dissimilar images by putting capture events containing similar events into the training or test set. They were limited by their decision to remove automatic handling of multispecies images for simplicity. When provided images with multiple species, their network could only provide one species label. The dataset was very imbalanced and the performance of the DNNs was worse for rare classes. They admitted that machine-learning techniques become more heavily biased to classes consisting of more examples. Our project utilized a relatively uniform dataset consisting of a similar number of images for a large majority of the different animals and plants in order to help reduce the potential bias in our model.

G. Komarasamy et al used Convolutional Neural Networks to train datasets in order to classify animals. It functions by training a data set and using the trained data to classify the animal.It employs CNN for animal recognition, CNNs are known for their ability to automatically take hierarchical features of images, making CNNs good for image classification[12]. This methodology involves data preprocessing, training set creation, learning algorithm, evaluation and final output. This is extremely similar to our methodology although we chose to use Tensorflow Lite. Their solution showed promising results in the precision of animal classification with their data reaching over 90% accuracy.The animals with complex intra-class heterogeneity and inter-class similarities cannot adequately reflect variation of animal groups.Another limitation is that it is moderately contrasted with human efficacy for animal identification. This study ignores the ease of access to the general public for these image classification models. Our project provides the user with a better experience through an easily accessible mobile application. Users can pull their phone out of their pocket and take a picture quickly in order to return a classification of an animal or plant. We also allow our users to communicate with each other, sharing their findings and providing assistance to others.
H.
Micheal et al. aimed to reduce the burden of analyzing millions of images from camera traps and manually classifying them[13]. They trained machine learning models using 3,367,383 training images to classify wildlife species from camera trap images. It utilized a neural network architecture using the TensorFlow framework using Mount Moran, a computing cluster. Their model achieved 97.6% accuracy at identifying the correct species with the top guess. It was limited by the number of training images they used for each animal, as they found a general trend in deep learning where “species and groups that had more images available for training were classified more accurately.” Their solution was more limited in scope in only dealing with camera trap images, while our solution can be used by any person with a phone.We have chosen to create a broad range of users rather than just for those who choose to use it for hunting or other trap image related reasons.

6. CONCLUSIONS

There were some limitations that were present in our project. Our model was limited by the number of plants and animals in our dataset that could be trained and classified.We are also limited by the lack of machine learning or image classification plugins available to flutter.The first thing that would need to be fixed is a larger data set for a greater number of animals and a greater volume of photos for each animal.In order to implement these, we would have to write scripts or take the time to pull images of all the animals we have off of the internet in order to be trained.We would also have to recreate a new much larger list of animals and species of plants in order to extend our dataset to more organisms.These could be easily implemented if we were able to spend a large amount of time on comprising a dataset.There are predicted to be 8.7 million different species of eukaryotic organisms which would be impossible given the technology and time we have to comprise a large dataset on that many animals or plants[10]. For this reason, we would need to have an algorithm to decide which plants and animals should be showcased in our new extended database.

Through our research of animal and plant classification using TensorFlow Light we have determined through experimentation the limitations of classification through machine learning.Computers are far from perfect and so are humans, meaning we expect to see this kind of error in the early stages of AI and Machine Learning [13]. That being said, we also do believe there will be significant improvements in this field in the coming years which should bring on a new level of accuracy and confidence when using image classification on plants and animals.

REFERENCES

[1] Barrett, H. Clark, and James Broesch. "Prepared social learning about dangerous animals in children." Evolution and Human Behavior 33.5 (2012): 499-508.
[2] Wäldchen, Jana, et al. "Automated plant species identification—Trends and future directions." PLoS computational biology 14.4 (2018): e1005993.
[3] Battu, Thirupathi, and D. Sreenivasa Reddy Lakshmi. "Animal image identification and classification using deep neural networks techniques." Measurement: Sensors 25 (2023): 100611.
[4] Pärtel, Jaak, MeelisPärtel, and Jana Wäldchen. "Plant image identification application demonstrates high accuracy in Northern Europe." AoB Plants 13.4 (2021): plab050.
[5] Langley, Ricky L. "Animal-related fatalities in the United States—an update." Wilderness & Environmental Medicine 16.2 (2005): 67-74.
[6] Krenzelok, Edward P., and Rita Mrvos. "Friends and foes in the plant world: a profile of plant ingestions and fatalities." Clinical Toxicology 49.3 (2011): 142-149.
[7] Froberg, Blake, Danyal Ibrahim, and R. Brent Furbee. "Plant poisoning." Emergency medicine clinics of North America 25.2 (2007): 375-433.
[8] Eddleston, Michael, and Hans Persson. "Acute plant poisoning and antitoxin antibodies: Antivenoms." Journal of Toxicology: Clinical Toxicology 41.3 (2003): 309-315.
[9] Mora, Camilo, et al. "How many species are there on Earth and in the ocean?." PLoS biology 9.8 (2011): e1001127.
[10] Liu, Ziming. "Reading in the age of digital distraction." Journal of Documentation 78.6 (2022): 1201-1212.
[11] Komarasamy, G., et al. "Automation of Animal Classification Using Deep Learning." Integrated Emerging Methods of Artificial Intelligence & Cloud Computing. Cham: Springer International Publishing, 2021. 419-427.
[12] Wilson, Helen F. "Animal encounters: A genre of contact." Animal Encounters: Kontakt, Interaktion und Relationalität (2019): 25-41.
[13] Khanzode, Ku Chhaya A., and Ravindra D. Sarode. "Advantages and disadvantages of artificial intelligence and machine learning: A literature review." International Journal of Library & Information Science (IJLIS) 9.1 (2020): 3.
[14] Chyleńska, Zofia Anna, and Eliza Rybska. "Understanding students ideas about animal classification." EURASIA Journal of Mathematics, Science and Technology Education 14.6 (2018): 2145-2155.
[15] Goadrich, Mark H., and Michael P. Rogers. "Smart smartphone development: iOS versus Android." Proceedings of the 42nd ACM technical symposium on Computer science education. 2011.