A MOBILE APPLICATION TO ASSIST IN REPORTING AND CLEANING SPOTS OF OCEAN LITTERS USING MACHINE LEARNING

Xiaoming Zhang[1], Tiancheng Xu[2]

[1]Robert Louis Stevenson School, 3152 Forest Lake Rd, Pebble Beach, CA, 93953
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the critical issue of ocean pollution, a growing environmental challenge exacerbated by the accumulation of plastic waste in marine ecosystems [4]. Ocean trash not only poses a significant threat to marine life but also impacts human health through the consumption of contaminated seafood. To tackle this problem, we propose a mobile application designed to mobilize community efforts towards ocean cleanup activities [5]. The app leverages cloud databases for real-time information sharing, machine learning models for predicting ocean trash accumulation, and Google Maps for location services, facilitating efficient and targeted cleanup operations [6].
Key challenges included ensuring the reliability of user-generated reports and optimizing the app for user-friendly navigation towards cleanup spots. Solutions such as implementing user report validation mechanisms and sorting cleanup locations by proximity were integrated to enhance the app's functionality. Experimentation across various scenarios demonstrated the app's potential to significantly increase community engagement in ocean conservation efforts.
The application represents a novel approach to environmental preservation, combining technology and community action. Its success in mobilizing users towards meaningful environmental impact underscores its value as a tool for global ocean conservation efforts, making it a significant asset for individuals and organizations committed to safeguarding marine ecosystems [7].

KEYWORDS

Mobile APP, Machine Learning, Database, Google Map API

1. INTRODUCTION

According to statistics, oceans have been significantly polluted since the industrial revolution. There is an estimated 75 to 199 million tons of plastic waste currently in our oceans, with a further 33 billion pounds of plastic entering the marine environment every single year. This amount of ocean trash could irreversibly damage the marine ecosystem. The crisis also posts a threat toward human health as microplastics have been found in seafood consumed by humans [8]. In this project, we develop a mobile application which aims to assist in solving ocean littering problems. As the scuba diving experiences reveal the beauty of the unseen part of the Earth, we know that preventing littering and cleaning existing trash could help put off parts of the negative effects. The app also helps to engage users in the cleanup and prevention efforts, raising awareness, and providing data and resources to help other people and groups.

David C. Wyld et al. (Eds): NBIoT, MLCL, NMCO, ARIN, CSITA, ISPR, NATAP-2024
pp. 91-100, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140508

Methodology A aims to connect children with nature using a mobile app. It successfully increases nature connectedness but relies heavily on device availability, potentially distracting from direct nature interaction. Your project builds on this by focusing on tangible environmental action, directly linking digital engagement with ocean cleanup efforts.

Methodology B integrates IoT with mobile apps to provide real-time pollution data, raising environmental awareness [9]. Its limitations include potential data accuracy issues and the need for extensive sensor deployment. Your project addresses these by enhancing community participation in cleanup, offering direct environmental benefits beyond awareness.

Methodology C combines IoT with gamification to boost user engagement. While effective in engaging users, it may face privacy concerns and complexity in gamification design. Your project mitigates these by ensuring inclusivity and accessibility in gamification strategies, focusing on real-world environmental improvements through community-driven efforts.

The app creates an online platform for people to report waste’s location and engage in cleaning activities. It allows users to report litter or plastic waste they come across with specific location and details; organize or join local cleanup events, with the help of nearby volunteers and organizations; and offers educational resources, articles and other contents to help raise awareness of the problem people face. It will be an effective solution as it embodies real-time data, global collaboration, and wider engagement. Compared to other existing methods, such as local beach cleanups, the app offers advantages in real-time reporting and a wider range of data collection.

The first experiment tested the Check Trash Function of your app, using a Kaggle dataset to simulate user queries for nearby litter types and amounts based on specific locations [10]. It aimed to verify the accuracy of the app's location-based litter reporting feature. The significant finding was the app's ability to correctly identify user locations and litter details, though it also revealed limitations due to the dataset's comprehensiveness, affecting result accuracy for some areas.

The second experiment focused on the app's reporting function, assessing its capability to accurately document user reports of litter, including location, type, and images. By simulating user reports, the experiment aimed to ensure the app correctly stored this data for community access. The results met expectations, with the app effectively capturing and storing all necessary information, demonstrating reliability in facilitating community cleanup efforts. The success of both experiments highlights the app's potential impact but also underscores the importance of data quality and system scalability.

2. CHALLENGES

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

2.1. How to Connect the Mobile App with a Cloud Database

To implement such an application, one of the biggest challenges is how to connect the mobile app with a cloud database. Because we are attempting to build a virtual community for all ocean cleaning lovers, we need a centralized database system, so that each individual user can communicate via the application. That is, we need to have all our app users connected to a shared database. One solution is to use the firebase cloud platform developed by Google [11]. It supports flutter and is easy to set up. We can use the cloud service to build a bridge between our users.

2.2. How to Locate the User's Position

Another problem could be how to locate the user's position. In order to communicate with clarity and accuracy, people usually use latitude and longitude as the standard way to describe a location. However, latitude and longitude are usually not easily understandable to human users, mainly because they are numbers rather than names of places. It's usually impossible to ask the user to remember their current latitude and longitude. To solve this issue, we could use the Google Map service to help the user get their location in a digital way. We could sign up for a Google Map account, and use the provided APIs to locate the user.

2.3. How to Find out the Amount of Trash Given a Specific Location

Another different problem could be how to find out the amount of trash given a specific location. Sometimes our user wants to know how much trash is around. One way is to find all trash in one area, but it is usually infeasible: first, we don't have the resources to check trash in an area; secondly, the situations are changing each day. To have an estimate of how much trash is around at a specific place, we use a machine learning algorithm called k-nearest neighbors. The method will take a dataset and learn the features inside. Then it will predict based on the given data so that we will have an estimate.

3. SOLUTION

Our app starts with a splash screen page, where it shows the logo and version number. Then it is directed to the homescreen. In the homescreen, we have 4 different pages that users can go to using the bottom navigation bar. The 4 pages are namely dashboard, report, check trash and help cleaning pages.

In the dashboard page, we will have some history stats that indicate what we have cleaned so far. It also shows some ocean pollution facts.

In the report page, we display all the reports that have been submitted so far. At the bottom of the page, there is a adding report button. If the user clicks on it, the app will be directed to an adding report page, where the user can locate themselves and take pictures. A report usually consists of an image, a latitude and a longitude. Once the user has everything ready, they can click the submit button, and the report will be uploaded to the cloud database.

In the map page, we have text fields that ask the user to enter their latitude and longitude. Unfortunately, people do not know their latitude or longitude for most of the time, unless they are geography experts. Therefore, we have an auto-locating button along with the text fields, which can figure out the user's location automatically. This feature is implemented by using the Google Maps platform, also known as the Google Maps API [12]. Once the user locates themselves, the location data will be sent to a machine learning model, which is implemented in Python. The machine learning model is a k-nearest neighbor model, it will predict how much garbage is around the user's location.

In the event page, we display the upcoming ocean cleaning events. Each event has a title, date, location and website link. If the user clicks the website link, they will be redirected to the sign up page of the event. The page will be viewed in a webpage browser, which might be different given different users [14].

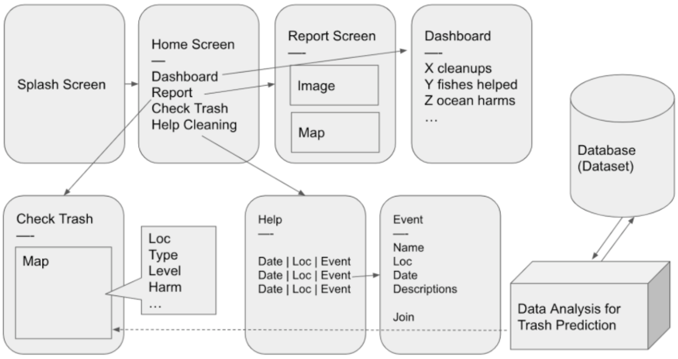

Figure 1. Overview of the solution

The first component that I want to talk about is the cloud database we are using. This feature is crucial to our ocean cleaning program since one of our proposals is to let the community share their ocean cleaning information. Our database's name is Cloud Friestore, which is developed by Google. The database allows us to upload and download structured data. In our cases, the types of the data are reports and events.

Figure 2. Ocean Debris Report

```
Future<Map<String, dynamic>?> getReport() async {
 Map<String, dynamic>? data;

 await FirebaseFirestore.instance
     .collection("ocean_cleaning")
     .doc('report')
     .get()
     .then((DocumentSnapshot documentSnapshot) {
       if (documentSnapshot.exists) {
         data = documentSnapshot.data() as Map<String, dynamic>?;
       } else {
         print('document does not exist on the database');
       }

 });
 return data;
}

Future<bool> addReport(Map data, date) async {
 FirebaseFirestore.instance
     .collection("ocean_cleaning")
     .doc('report')
     .set({date: data}, SetOptions(merge: true));
 return true;
}
```

Figure 3. Screenshot of code 1

These are the two functions that communicate with the database: namely an upload function and a download function. They both use the Cloud Firestore APIs. We use the Firebase Firestore. instance. collection("ocean_cleaning").doc('report').get() to download the report data currently in the cloud database and FirebaseFirestore.instance.collection("ocean_cleaning").doc('report').set() to upload data. It turns out it works as expected. In the UI screenshot, it displays all the reports that we added to the database from the local machine. Therefore, the user can share reports of ocean trash globally.

The second component in our project is the location service. We are developing an application mainly for mobile phones, the location of the user is usually varying. For human beings, we tend to describe a location using names. For instance, we could use city names or street names. The downside of such a location naming system is that the name is not unique. There could be multiple cities or streets that share the same name. In addition, a street name is usually not accurate in terms of where the user is, especially given that a street could be long.

Therefore, we need an accurate system to describe locations: latitude and longitude is the best candidate since they can identify a specific position on the earth. However, human beings usually are not able to memorize the latitude and longitude of certain places, unless they are geology experts.

As a result, we need an automatic geological locating service that will take the user’s location as the input and return the latitude and longitude as the output. In this project, we use the Google Maps API to locate users. Google Maps uses GPS to measure the latitude and longitude [13].

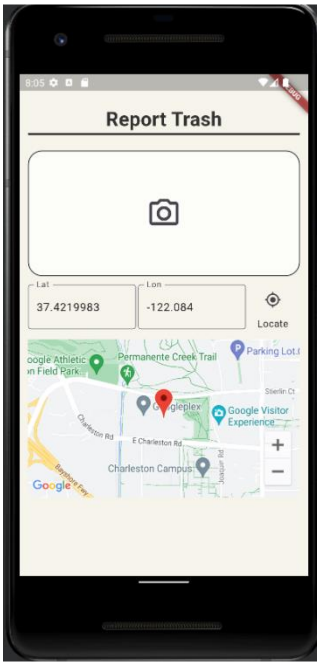

Figure 4. Report trash

```
Future<bool> _handleLocationPermission() async {
 bool serviceEnabled;
 LocationPermission permission;

 serviceEnabled = await Geolocator.isLocationServiceEnabled();
 if (!serviceEnabled) {
  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
     content: Text(
       'Location services are disabled. Please enable the services')));
  return false;
 }
 permission = await Geolocator.checkPermission();
 if (permission == LocationPermission.denied) {
  permission = await Geolocator.requestPermission();
  if (permission == LocationPermission.denied) {
   ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text('Location permissions are denied.')));
   return false;
  }
 }
 if (permission == LocationPermission.deniedForever) {
  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
     content: Text(
       'Location permissions are permanently denied, we cannot request permissions.')));
  return false;
 }
 return true;
}


Future<void> _getCurrentPosition() async {
 final hasPermission = await _handleLocationPermission();

 if (!hasPermission) return;
 await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high)
    .then((Position position) {
  setState(() {
   _latController.text = position.latitude.toString();
   _lonController.text = position.longitude.toString();
   _showMap = true;
  });
 }).catchError((e) {
  debugPrint(e.toString());
 });
}
```

Figure 5. Screenshot of code 2

There are two functions that implement this feature. The first function is for checking if we have the permission to access the user's location, since location is the user's privacy. It politely asks the user to turn on the location access service so that our geolocater can work. The second function is running the locating service via Google Maps API. It will take the returned latitude and longitude and save them into storage for later use.

The last component we want to talk about is the machine learning model for predicting the amount of trash near the user. In the app, we provide an AI service to the user to estimate how much garbage is in a certain area. In this component, we first use the Google Maps API to get a location just as previously described, and then use that location to predict the amount of trash. The predicting model is an AI model, called k-nearest neighbor. It works by taking an input location and a dataset, and output the information of the location based on the data points in the dataset. It looks at the data points that are closest to the input location, summarizing the information and output by an average or majority count.

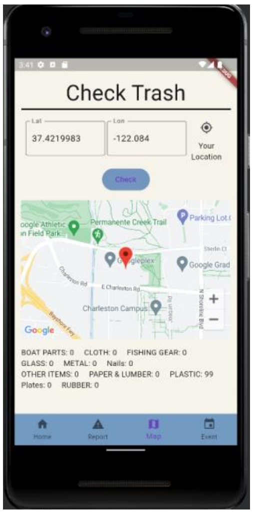

Figure 6. Check trash

```
# Input latitude and longitude for prediction
def predict_materials_quantity(input_latitude, input_longitude):
    # Create a Linear Regression model for each material
    print((input_latitude, input_longitude))
    models = {}
    for material in unique_materials:

        model = KNeighborsRegressor(n_neighbors=5)
        model.fit(final_df[['latitude', 'longitude']], final_df[material])
        models[material] = model

    # Predict quantities for each material
    predictions = {}
    for material, model in models.items():
        prediction = model.predict([[input_latitude, input_longitude]])
        predictions[material] = math.ceil(prediction[0])

    return (predictions)
```

Figure 7. Screenshot of code 3

In the code above, we use a k-nearest neighbor regression model where the k-value equals to 5. Then we let the model fit the data in final_df, which is a dataset of garbage. We used the pre-defined predict function of the model to predict the amount of ocean garbage of the given location.

4. EXPERIMENT

4.1. Experiment 1

One blind spot in my program that I wanted to test out is the Check Trash Function. It is one of the major components of the app in which Google Map Platform is used to locate the position.

The experiment is set up as if a user is curious of the types and count of trash around their location. They would input their latitude and longitude of their position and the app should be able to locate the amount and types of different litter being recorded around them. The experiment is set up this way to make sure that the app provides the accurate result with the correct location. The sourcing data in the experiment came from kaggle and documents litter around beach in A with the website listed below: https://www.kaggle.com/datasets/maartenvandevelde/marine-litter-watch-19502021

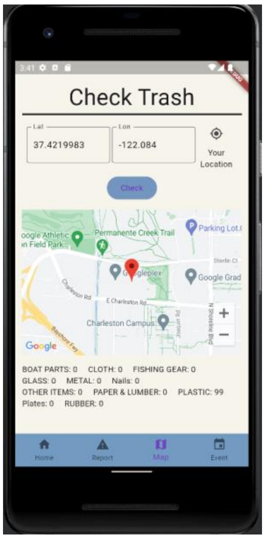

Figure 8. Figure of experiment 1

The resulting data turned out to be working perfectly. It locates the input latitude and longitude correctly with a list of potential types and amount of litter around the selected location. For example, the data shows that there might be around 99 recordings of plastics around the location. However, the result reflects the limit of the source of data in which not every existing point of litter is documented. In other words, for some locations, the result could be misleading (reflecting the wrong amount of type) and wrong.

4.2. Experiment 2

Our second experiment focuses on the reporting function of the app in which users could report latitude, longitude, types, and a picture of trash into the database. It is important to make sure that the function operates correctly.

The experiment is set up as if a user is trying to report the location, types, or picture of trash around their location that they wish for others to help to clean. They would input their latitude and longitude of their position, a picture, and the date. The app is designed so that it could record the information into its database. Therefore, when users want to help clean their community, they could quickly find the reported spots without wasting time to look for litter. The experiment is set up this way to make sure that the app could document the correct location and other information.

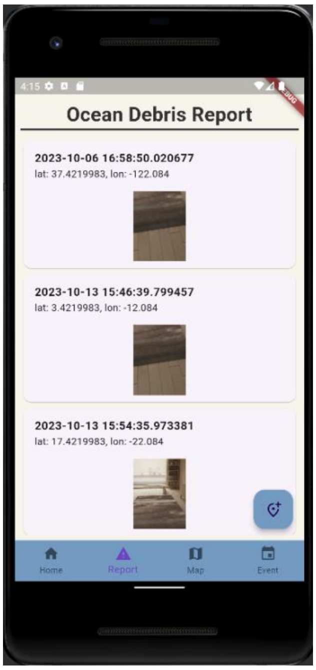

Figure 9. Figure of experiment 2

The result of the experiment meets with the expectation that the necessary data is documented by the app, including the picture, longitude, latitude, and date. The system is also able to handle multiple input by the user that is available for other users.

5. RELATED WORK

Methodology A explored the effectiveness of a mobile app in connecting children with nature, comparing it with traditional environmental education methods. It found that the app was as effective as traditional methods, adding enjoyment and engagement benefits. However, it also noted limitations such as device dependency and the potential for technological distractions. My project extends this approach by utilizing a mobile app for ocean pollution cleanup, emphasizing real-time data and community collaboration. This addresses previous limitations by linking digital engagement directly with tangible environmental action, enhancing the practical application of technology in environmental conservation efforts [1].

Methodology B introduces a mobile app prototype to enhance awareness of environmental pollution. The methodology employed combines the Scrum agile framework with Internet of Things (IoT) technologies to collect data from air pollution sensors, facilitating real-time environmental monitoring. This solution is notable for its adaptability to changes and ability to provide immediate pollution data to users, thereby raising awareness and prompting action.

However, the approach has limitations, such as potential data accuracy concerns and the need for widespread sensor deployment to ensure comprehensive coverage. It may overlook the broader socio-economic factors influencing pollution and the engagement of stakeholders beyond app users. My project builds on this by emphasizing community participation in ocean cleanup, offering a more direct approach to environmental action. By integrating real-time data collection with community-driven cleanup efforts, your app not only raises awareness but also facilitates tangible contributions to pollution reduction, enhancing the practical impact on environmental conservation [2].

Methodology C examines the integration of Internet of Things (IoT) with gamification to enhance user engagement across various application domains. This methodology combines advanced technology (IoT) with innovative design (gamification) to increase engagement by making interactions more interactive and fun. The review indicates that IoT-enabled Gamification (IeG) leads to higher engagement levels, demonstrating its effectiveness in encouraging user participation and interaction.However, the approach has limitations, such as potential privacy concerns with IoT devices and the complexity of designing effective gamification strategies that appeal to a wide audience. It might overlook the need for inclusivity in engagement strategies, ensuring that gamified elements are accessible and enjoyable for all users, regardless of their familiarity with technology. My project enhances this by focusing on community-driven environmental efforts, ensuring the app is user-friendly and inclusive. By leveraging real-world impact through gamification, your app not only increases engagement but also contributes to tangible environmental improvements, thus addressing some limitations of IeG by ensuring that engagement leads to real-world action [3].

6. CONCLUSIONS

There are multiple limitations within the app. To begin with, the app cannot differentiate the validity of user's uploaded reports. All reports are shared with the users and it could be troublesome and time-wasting if someone intentionally uploads multiple useless reports. However, it could be solved if there's a limit towards user's daily upload or a report function to ban users from uploading unwanted data points. Another potential limitation is that users cannot locate clean up spots that are closest to them; instead, they are only available to the uploads sorted by time. This limitation could be solved if the data set could sort the data based on distance with users' shared location.

In this project, we make an ocean cleaning application on mobile phones. The main purpose of the application is to provide the ocean cleaning community with an efficient and user-friendly platform to protect the oceans. We support real-time sharing of ocean pollution information using cloud databases, ocean trash prediction using machine learning models, and location service using Google Maps [15].

REFERENCES

[1] Crawford, Maxine R., Mark D. Holder, and Brian P. O'Connor. "Using mobile technology to engage children with nature." Environment and Behavior 49.9 (2017): 959-984.

[2] Ramos-Romero, Anthony, Brighitt Garcia-Yataco, and Laberiano Andrade-Arenas. "Mobile application design with iot for environmental pollution awareness." International Journal of Advanced Computer Science and Applications 12.1 (2021).

[3] Xiao, Ruowei, Zhanwei Wu, and Juho Hamari. "Internet-of-gamification: A review of literature on IoT-enabled gamification for user engagement." International Journal of Human–Computer Interaction 38.12 (2022): 1113-1137.

[4] Monteiro, Raqueline CP, Juliana A. Ivar do Sul, and Monica F. Costa. "Plastic pollution in islands of the Atlantic Ocean." Environmental Pollution 238 (2018): 103-110.

[5] Kangas, Eeva, and Timo Kinnunen. "Applying user-centered design to mobile application development." Communications of the ACM 48.7 (2005): 55-59.

[6] Mehta, Heeket, Pratik Kanani, and Priya Lande. "Google maps." International Journal of Computer Applications 178.8 (2019): 41-46.

[7] Jacobides, Michael G., Carmelo Cennamo, and Annabelle Gawer. "Towards a theory of ecosystems." Strategic management journal 39.8 (2018): 2255-2276.

[8] Andrady, Anthony L. "Microplastics in the marine environment." Marine pollution bulletin 62.8 (2011): 1596-1605.

[9] Madakam, Somayya, et al. "Internet of Things (IoT): A literature review." Journal of Computer and Communications 3.05 (2015): 164.

[10] Quaranta, Luigi, Fabio Calefato, and Filippo Lanubile. "Kgtorrent: A dataset of python jupyter notebooks from kaggle." 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). IEEE, 2021.

[11] Moroney, Laurence, and Laurence Moroney. "Cloud functions for firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 139-161.

[12] Hu, Shunfu, and Ting Dai. "Online map application development using Google Maps API, SQL database, and ASP .NET." International Journal of Information and Communication Technology Research 3.3 (2013).

[13] Aughey, Robert J. "Applications of GPS technologies to field sports." International journal of sports physiology and performance 6.3 (2011): 295-310.

[14] Wang, Haoyu, et al. "Similarity-based web browser optimization." Proceedings of the 23rd international conference on World wide web. 2014.

[15] Song, Congzheng, Thomas Ristenpart, and Vitaly Shmatikov. "Machine learning models that remember too much." Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security. 2017.