The object of the research is the detection of explosive objects in an image, with a particular focus on the identification of anti-personnel landmines. The objective of this research is to develop effective tools for the recognition of landmines.

A mobile application for the recognition of explosive objects, trained on a deep learning model using landmine replicas, has been developed. The application was tested on images of actual landmines. The model utilized in the application exhibited a recall rate of 89% (calculated as the ratio of correctly identified landmines to the total number of landmines present in the image). The results indicated that the recall rate for a specific category of landmines was less than that observed for the others. The average time required for offline image recognition was 2.1 seconds.

This paper presents the results of the evaluation of the effectiveness of the mobile application for landmine detection and classification. Furthermore, it describes the ways in which the application allows for the improvement of the model through the collection of data from users. It also describes the architecture and interface of the application, as well as an analysis of its potential applications in landmine recognition.

The efficacy of the mobile application can be attributed to its intuitive interface, the high accuracy of the deep learning model, and the capacity to obtain user feedback promptly. The program enables not only the identification of hazardous objects but also the transmission of data for the enhancement of the model.

The mobile application has the potential to be utilized for a multitude of tasks pertaining to the detection of explosive objects, in addition to enhancing the precision of the model. Furthermore, the app can be utilized in training centers for deminers and in mine-contaminated areas. The mobile application can be employed to identify unknown explosive objects and enhance the efficacy of deep learning models. The resulting models can be leveraged in the future to automate the demining process

Keywords: landmine detection, explosive ordnance disposal, humanitarian demining, mobile demining application

UDC 004.93

DOI: 10.15587/1729-4061.2024.317103

LANDMINE DETECTION WITH A MOBILE APPLICATION

Oleksandr Kunichik
PhD Student
Department
of Mathematical Informatics
Taras Shevchenko
National University of Kyiv
Volodymyrska str., 60,
Kyiv, Ukraine, 01033
E-mail: ao669517126@gmail.com

Received 09.09.2024
Received in revised form 15.11.2024
Accepted 29.11.2024
Published 25.12.2024

***How to Cite**: Kunichik O. (2024). Landmine detection with a mobile application. Eastern-European Journal of Enterprise Technologies, 6 (2 (132)), 6–13. https://doi.org/10.15587/1729-4061.2024.317103*

1. Introduction

Landmines continue to represent a significant and ongoing threat to global security, even in the aftermath of armed conflicts. A minimum of 10,254 individuals sustained injuries from landmines and explosive remnants of war across the globe during the 2021–2022 period (3,843 fatalities, 6,370 injuries, and 41 cases with unknown outcomes) [1, 2]. In 85 % of cases where the status of the victim was known, the victims were civilians. Among civilian victims for whom the age was known, approximately half were children. The extent of the issue related to mined areas is significant. This issue is particularly salient in Ukraine. To illustrate, Ukraine's State Emergency Service (SES) has identified approximately 128,000 square kilometers of mined land and 14,000 square kilometers of mine-contaminated water. The presence of landmines has a significant negative impact on approximately 6 million Ukrainians [3]. The presented evidence highlights the imperative for the practical adoption of contemporary methodologies for demining Ukrainian territories.

Recognizing explosive ordnance is paramount in restoring normalcy in contaminated regions. A variety of techniques are employed to achieve this objective, including manual demining and the utilization of explosive-detecting animals [4]. Artificial intelligence technologies, particularly machine learning algorithms, have demonstrated the capacity to develop models capable of accurately detecting landmines. Nevertheless, the application of such models is currently confined to a limited circle of demining professionals.

The implementation of cutting-edge technology has the potential to expedite the demining process significantly. However, the demining community still employs antiquated and conventional techniques, including metal detectors and demining probes. A mobile landmine detection application can assist in identifying hazardous objects by capturing images of them and notifying the responsible services.

Moreover, it is essential to develop the ability to identify unknown explosive objects that sappers may encounter during their duties. In such instances, mobile apps can serve as a valuable tool. The feedback function enables the mobile application to function as a data source when the model cannot accurately identify explosive objects.

The widespread use of mobile devices and the development of modern machine learning algorithms make the development of mobile applications for landmine detection using computer vision models a promising avenue of research. Such research can facilitate the rapid and secure identification of landmines, in addition to training for demining personnel, enhancing model efficacy, and alerting pertinent demining services.

2. Literature review and problem statement

The study [5] comprehensively overviews contemporary demining technologies, including metal detectors, ground-penetrating radars, infrared sensors, magnetometers, and other advanced devices. The study provides an extensive overview of the latest technological developments in the field of demining. Nevertheless, the work does not propose practical solutions to facilitate the demining process. As indicated in this study, only a limited number of projects have been able to secure funding and have reached the implementation stage. However, no specific examples are provided. According to a 2013 United Nations (UN) report [6], many promising projects were not implemented due to a lack of funding or a mechanism to match donors and technological capabilities. The same report references the utilization of mobile applications for landmine safety training and outlines plans to develop a training application for improvised explosive devices. Consequently, there are initiatives employing mobile technologies in demining operations, but they do not have automatic recognition functionality.

In [7], the potential of smartphones in conjunction with artificial intelligence for applications in the medical industry is investigated. This domain is closely related to explosive ordnance detection, as erroneous judgments can have significant consequences. The paper reviews several mobile applications and discusses the potential of a smartphone camera. The authors of the study illustrate the potential for utilizing artificial intelligence to identify objects in the medical domain. Nevertheless, the question of enhancing models by incorporating feedback from the application still needs to be solved. Additionally, the authors highlight the challenges associated with on-device recognition due to the intricate nature of the models. To address these complexities, it may be beneficial to consider the implementation of server-based recognition as a potential solution. This approach is analogous to that described in reference [8]. Still, it is constrained by the necessity of utilizing a desktop computer as a server, which restricts the system's mobility and its deployment in the field, where access to a desktop computer may be limited or non-existent.

The authors of the study [7] observe that the advancement of object recognition software is hindered by the need for more data and the absence of feedback. One potential solution to these issues is the implementation of online recognition. This approach was proposed in [9], which presents the findings of a study investigating the potential of a mobile application for classifying explosive objects using virtual reality. However, the study does not utilize its own dataset and model, which limits its applicability to a large number of landmines. Instead of using its own dataset, the application employs a third-party platform designed for the development of augmented reality software. To evaluate performance, the authors employ the accuracy metric, which is not sufficiently informative to assess the performance of landmine detection models. In the context of landmine detection tasks, other metrics, such as recall and F1-measure, which consider both false positives and false negatives, are of greater importance. The application is implemented using three scripts that perform the main functions. However, the issues of feedback and annotation of incorrectly recognized landmines still need to be solved.

Regarding reporting instances of landmine detection, the SES website offers a landmine reporting option [10]. The SES website provides insight into the present state of demining operations, the extent of contaminated and occupied territories in Ukraine, and related matters. Reporting a landmine requires visiting the service's website and registering through BankID, an electronic identification system. The Bezpeka-Info website [11] has various landmine safety resources. The project aims to disseminate information in an accessible format to a target audience of children. Additionally, an online course is available. However, the resources [10, 11] need to be improved in convenient mobile applications and the capacity to operate offline. To retrieve information, users must have internet access and navigate to the relevant websites.

The paper [12] proposes using mobile devices as a substitute for metal detectors for landmine detection. The magnetic sensor built into the smartphone detects the landmine. However, the proposed system is experimental and has not been put into practical use. The use of a smartphone to recognize landmines is not economically feasible, as it is sufficient to use a conventional sensor.

Consequently, at the time of writing, no software is currently available that can detect landmines in images. The existing online resources lack the functionality to recognize landmines. Using smartphones to recognize landmines can significantly simplify and speed up the process. The mobile application can work offline, allow for sending coordinates of explosive objects, and provide information about landmines. Therefore, conducting a study on using mobile devices in landmine detection would be advisable.

3. The aim and objectives of the study

The aim of this study is to develop and evaluate the effectiveness of a mobile application for the detection of landmines. The application should provide landmine recognition both online and offline. Online mode will utilize real-time processing on a remote service. In contrast, offline mode enables users to detect landmines without internet connectivity, relying on on-device processing using a locally stored model. The widespread use of smartphones makes it possible to reach a large audience, as modern smartphones are accessible to most people. Using a mobile landmine recognition application will simplify and speed up the process of landmine identification. In addition, the mobile application will allow for additional functions such as notifying about an explosive object, sending its coordinates, and providing additional information about landmines.

To accomplish this goal, the following tasks were identified as necessary:

– implement landmine detection in online and offline modes, test the application offline on a set of test images, and measure the recall and speed of detection;

– implement the ability to correct recognition results by marking objects and selecting the type of landmine;

– provide the ability to send data about the app's operation to improve machine learning models further.

4. Materials and methods

4. 1. Object and main hypotheses of the study

The object of this study is an information system for recognizing explosive objects in an image, including anti-personnel landmines, anti-tank landmines, and shells.

The main hypotheses of the study:

1) a mobile application that uses a deep learning model is able to detect the presence of explosive objects;

2) images where the model made mistakes can be used for further training and sent from the application to the server.

The following assumptions underpin the study:

– it is assumed that the majority of images captured by users to verify the application will be of sufficient quality to facilitate accurate recognition;

– it is further assumed that users will be motivated to obtain accurate recognition results and will, therefore, take photos of an appropriate quality;

– during the testing phase, the model will be evaluated using images that have been specifically selected to assess its resilience to a range of conditions and objects that may potentially lead to false alarms.

The following simplifications are employed during this study:

– it is assumed that the model may be incapable of recognizing any explosive objects not included in the training dataset. In such an event, the data can be transmitted to the server;

– it is assumed that the accuracy of recognition may be diminished for images that are of insufficient clarity or those captured from an unconventional angle that were not included in the training dataset;

– it is postulated that a relatively modest number of images (approximately 50–100) is sufficient to introduce a novel object category into the model.

4. 2. Hardware and software

To test the application's functionality under various conditions, the study employed a variety of Android devices, including versions 6 through 14. The principal test models were the Samsung Galaxy A53, which features an Octa-core processor (comprising two Cortex-A78 cores at 2.4 *GHz* and six Cortex-A55 cores at 2.0 *GHz*), 128 *GB* of storage, and 6 *GB* of random-access memory (RAM), 64 megapixel camera; and Realme 6: Octa-core processor (2×2.05 *GHz* Cortex-A76 & 6×2.0 *GHz* Cortex-A55), 128GB, 8GB RAM, 64 megapixel camera.

Furthermore, additional testing was conducted on the following devices: Redmi Note 9/11/13 Pro, Google Pixel 4a 5G, Samsung Galaxy A23, Samsung Galaxy Tab S7 FE, Samsung Galaxy M11, Xiaomi Mi Play, Xiaomi 11/12 Pro, Motorola One Zoom, IIIF150 B2, and others.

Google Cloud Functions were employed to process requests to Google Cloud Platform (GCP), which has a RAM capacity of 128 megabytes (MB) to 256 MB, the number of virtual processors ranging from 1/6 to 4, and a 60-second execution timeout. GCP provides horizontal scaling, which enables the expansion of computing power in response to an increase in the number of requests.

The application was developed using the following tools: the Qt Creator 13.0.2 integrated development environment (IDE) and the Android Qt 6.7.2 Clang compilers (arm64-v8a and armeabi-v7a). The package includes the build results of both compilers to ensure compatibility with Android versions 6 to 14.

4. 3. Information system architecture

The system has a client-server architecture and can be built for most operating systems, including Windows, Linux, MacOS, IOS, and Android. This study is primarily concerned with the Android operating system, which is estimated to be used by over 70 % of smartphone users in 2023, as reported by [13].

The primary framework utilized for the system's development is Qt [14], version 6.7.2. The framework allows software creation for many operating systems within a single project. Given the inherent changes to operating systems and their accompanying peculiarities, it is imperative to conduct meticulous testing and implementation of operating system-dependent code to ensure seamless support for multiplatform code. Nevertheless, supporting the project described in this paper requires a significantly smaller investment of resources than developing unique code for each operating system.

The user interface is constructed using QML, a declarative language designed for creating dynamic and visually rich user interfaces. JavaScript heavily influences QML and allows for seamless integration of JavaScript code for scripting and application logic within the Qt framework. QML enables the creation of user interfaces that are native to the operating system, thereby facilitating user interaction and significantly accelerating the development process. This is feasible because QML enables the user to concentrate on customizing the appearance of interface elements rather than creating these elements from scratch. Furthermore, this component of the system also implements the ability to select an object by making a rectangle or a contour around it. The resulting coordinates of the object are then transferred to the subsequent level, namely the C++ part of the mobile application, where the business logic is written (image preprocessing, validation of user input, communication with the server-side modules, management of user data).

The user interface interacts with the C++ component. The C++ paradigms of object-oriented programming, inheritance, encapsulation, and multithreading are employed. Execution of lengthy processes in separate threads avoids interface hangs during their execution. The Qt Concurrent module facilitates multithreading at a high level of abstraction, obviating the necessity for manipulating low-level synchronization primitives. This module automatically determines the optimal number of threads to run, considering the number of processor cores.

The SQLite database stores the app's state, encompassing information regarding open files, change history, and annotations. SQLite enables the application to execute SQL queries and store data efficiently while minimizing resource consumption.

The system is equipped with the capability to transmit images of landmines for subsequent analysis and model training. This functionality is implemented via a module deployed on the Google Cloud Platform (GCP). Images are stored in Google Cloud Storage, and queries are processed using the Google Cloud Run Functions service. The NoSQL document-oriented database Google Firestore is employed to regulate access to the system. This database stores information about users, files, annotations, settings, and other relevant data. General settings, landmine lists, and other general information are stored in Google Cloud Storage. The Cloud Run Functions service facilitates downloading new landmine lists and general settings and transmitting logs and database data for analysis. All functions in this module have been written using the Python programming language.

The system is equipped with the capability to identify images through the Roboflow computer vision service. Roboflow offers an Application Programming Interface (API) for working with datasets, model training, and online recognition. In online mode, the system can leverage the Internet for online recognition. For the majority of smartphones with high-speed internet connections, online recognition is a more suitable process than recognition on the device itself. However, in the absence of internet access or in regions with inadequate network coverage, recognition on the mobile device is the primary or even the sole option.

The core component of the system is the image object recognition module. The Java module is employed to perform landmine recognition. The Android operating system employs code written in Java to identify objects within an image. The code uses the ONNX runtime module for recognition purposes. The Open Neural Network Exchange (ONNX) [15] is an open standard for representing machine learning models widely utilized in artificial intelligence. ONNX is an open standard for model exchange supported by numerous companies, including Microsoft, IBM, Intel, AMD, Facebook, and others. ONNX Runtime [16] is a high-performance runtime environment for machine learning models. Interaction with Java code is carried out using the Qt JNI (Java Native Interface) module.

The application is multilingual, with English and Ukrainian localizations currently available. New languages can be easily added by utilizing the Qt Linguist module, a robust tool for translating the interface of Qt applications. To incorporate a new language, one only needs to download an XML file containing the translated interface from English into the desired language and rebuild the application.

In consequence, the system architecture is constituted of multiple modules and levels. This approach guarantees the modularity and flexibility of the system, enabling the alteration of individual components without affecting the remainder of the system. To illustrate, the C++ backend, where the business logic is written, can be separated into a distinct microservice. Functions performed on the GCP can be transferred to alternative cloud platforms, such as Azure or Amazon Web Services. Another online recognition service can be employed or used in parallel with Roboflow. This approach ensures the system is modular and flexible, aligning with contemporary software development requirements.

4. 4. Deep learning model for recognizing explosive objects

The system employs the YOLOv8 deep learning model [17] to identify explosive objects. The model was trained on a dataset comprising 1,438 images of 3D printed landmine replicas. The dataset was subsequently expanded to 3,452 images through the utilization of augmentation techniques, as outlined in [18]. The images were then divided into three sets: training (3021 images, representing 88 % of the total), validation (287 images, representing 8 % of the total), and test (144 images, representing 4 % of the total). The images were reduced in size to 640×640 pixels to accelerate the training and recognition process. The following hyperparameters were employed for model training: batch size, 32; initial learning rate, 0.1 (during training, the algorithm automatically reduces this to a range of 0.0001 to 0.001); number of epochs, 300. The model development and training process is described in greater detail in [19]. YOLOv8 was selected due to its high accuracy, speed, and compact size, making it well-suited for mobile applications. However, it should be noted that the recognition accuracy may be reduced in low light or unconventional shooting angles.

4. 5. The main algorithm of the application

The application's algorithm is comprised of the following steps:

1. The user may select one of the available operating modes:
– online recognition;
– offline recognition;
– mode without recognition.

2. Open image: the user can open an image from the device storage or take a photo with the camera.

3. Landmine detection (if landmine detection mode is selected):

– online mode: the application transmits the images to the server, where they are analyzed using a deep learning model. The results of this analysis, including the object boundaries and the identified landmine types, are then returned to the user's device;

– offline mode: the recognition process is conducted on the device itself, utilizing a local deep learning model.

4. Editing the results (optional): If the model fails to detect a landmine, incorrectly identifies its type, or requires clarification of the object's boundaries, the user can edit the recognition results.

5. Sending data to the server (optional): The user may choose to transmit data to the server for analysis and training purposes. The data may include:
– the application's operational logs;
– database of the application;
– images of landmines may be provided in either a reduced or original format.

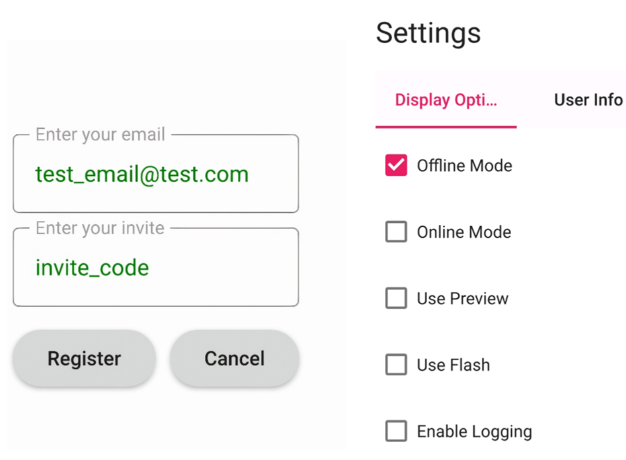
6. Data processing on the server:
– Google Cloud Functions processes requests from a mobile application;
– images are stored in Google Cloud Storage;
– the Google Firestore database stores information about user, files, annotations, and settings;
– the data is subjected to analysis and subsequently employed for the purpose of further model training.

4. 6. Description of the main features of the application

The client component of the system is a mobile application developed for the Android operating system. The cross-platform Qt framework allows the application to be constructed and tailored for alternative operating systems. Following the installation of the application, a referral link must be entered (Fig. 1, *a*). This enables tracking of the application's sources and prevents unauthorized use. Once the initial setup is complete, the application is capable of functioning autonomously (offline) – the only instance requiring online connectivity is synchronizing the lists of new landmines recognized by the application.

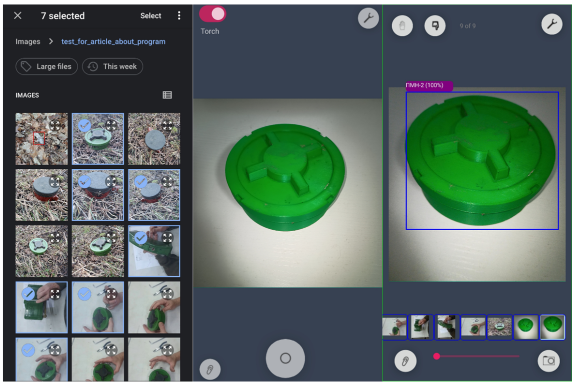
The application's operating modes may be switched in the settings menu, with the following options: online, offline, or no selection (Fig. 1, *b*).

The user can open an image from the device's storage (Fig. 2, *a*) or take a photo using the camera (Fig. 2, *b*). In each mode, annotations can be added to the image, and the data can be sent for analysis. The application allows for the simultaneous processing of multiple images (Fig. 2, *c*). Upon reopening an image, the landmine data is retrieved from the database and not re-recognized.



a *b*

Fig. 1. The application interface: *a* — dialog box for new user registration; *b* —settings menu offering a choice of operating modes



a *b* *c*

Fig. 2. The application interface: *a* – file opening dialog; *b* – shooting mode with the flashlight on; *c* – editing mode with several open files

The OpenSSL library is employed to safeguard data during its transmission between the client and the server [20]. Launching cloud functions on the server is contingent upon acquiring prior authorization. The system is compatible with Android versions six and above. Cloud functions are horizontally scalable and demonstrate resilience to high loads, irrespective of the number of clients.

5. Results of a study on the use of a mobile application for landmine detection in images

5. 1. Development and testing of a landmine detection module

The mobile application employs a deep learning model [19] to identify potential landmine-like objects in images. The user can choose between the application's operation modes: online or offline. Additionally, the user can turn off the application's recognition functionality, allowing the system to be trained on a new type of landmine before reactivating the mode. The user can select from several operational modes:

– online recognition using a deep learning model [19] deployed on the Roboflow service [21];

– offline recognition on the user's device, which typically requires 2–4 seconds, depending on the device's characteristics and current load;

– mode without recognition: images opened in the application will be automatically marked in the database for further analysis.

To evaluate the application's efficacy, 125 images of five distinct types of landmines were selected for analysis: MON-50, PMN, PMN-2, OZM-72, and PFM-1. These images were provided by professional demining experts and utilized with their consent. The average processing time for a single image, including opening and recognition, was 2.1 seconds. The results of the recognition process for each landmine are presented in Table 1.

In order to evaluate the performance of the model presented in this paper, the recall metric serves as the primary evaluation metric. The recall value (1) is the proportion of correctly identified landmines out of the total number of landmines present within the image:

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}, \tag{1}$$

where True Positives are the number of landmines that the model correctly identified, and False Negatives denote the cases where a landmine is present in an image, but the model fails to recognize it correctly, either missing it entirely or misclassifying it as a different type. This is a crucial metric in landmine detection because failing to identify a landmine (a False Negative) has far more severe consequences than incorrectly classifying something as a landmine when it is not (a False Positive).

Table 1

The results of checking the program operation

Type of landmine	Quantity	Time[1] (sec.)	Recognized			Recall (%)
			Correct	Another landmine	Incorrect	
MON-50	25	2.2	23	1 (TM-62-M)	1	92
PMN	25	2.0	23	–	2	92
PMN-2	25	2.1	23	1 (MON-50)	1	92
OZM-72	25	2.2	20	–	5	80
PFM-1	25	2.0	22	3 (MON-50)	0	88

Note:[1] is the average processing time for one image in seconds.

The 89 % recall rate suggests that the app has the potential to be an effective tool for landmine detection. The recall rate for different types of landmines exhibits considerable variation, with values ranging from 80 % for OZM-72 to 92 % for other types. This may indicate the necessity of augmenting the quantity and diversity of OZM-72 images within the training dataset. However, examining a subset of the images reveals that the model occasionally classifies visually analogous images disparately (Fig. 3).

Fig. 3.Two frames from a video displaying an OZM-72 landmine, processed through the application, are nearly indistinguishable. However, the landmine on the left remains unidentified
Source: Telegram channel of an Armed Forces officer with the military call sign Forester

The difference in results for practically identical images suggests that the model requires further enhancement, including incorporating data from disparate perspectives, to enhance recognition accuracy.

A random selection of images, although not of the highest quality, was intentionally chosen for testing purposes. For the mobile application to work as intended, users must take pictures to minimize the likelihood of recognition errors. While this assumption is not critical, users who seek precise identification of the landmine type are likely to obtain high-quality images. However, this does not imply that the model cannot recognize landmines in low-quality images. Conversely, the app's usage will facilitate model enhancement, in particular by acquiring data on its errors, thus allowing for the identification of areas requiring improvement.

5. 2. Implementation of the function of editing recognition results

In the edit mode, users can modify the recognition results if the model fails to detect a landmine or identifies it as a different type than what was actually present. This feature is crucial for enhancing model accuracy, enabling users to rectify errors and furnish additional data for training purposes.

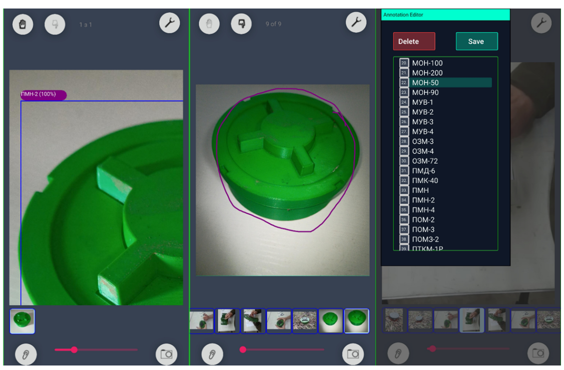
To edit the results, the user has the option to:

– zoom in/out using zoom gestures (Fig. 4, *a*);

– move the visible part of the image by dragging the user's finger across the screen (Fig. 4, *a*);

– select an area on the image containing the object of interest with a finger gesture (Fig. 4, *b*);

– select or modify the classification of the selected area by utilizing the list of available landmine types (Fig. 4, *c*).



a *b* *c*

Fig. 4. Editing: *a* — zooming in and moving around the image; *b* — selecting a landmine with a finger (after the tracing is complete, it turns into a frame); *c* — choosing the type of landmine from the menu

To illustrate, if the model fails to identify a landmine within an image, the user can use edit mode to highlight the relevant item and specify its classification. All modifications made by the user are recorded in the database, ensuring that the edited results are preserved when the image is reopened or the data is transmitted.

5. 3. Sending data to the server

To enhance the precision of the model and augment its functionality, the application offers the capability to transmit data to the server. This feature enables the collection of data regarding recognition errors, the addition of new types of landmines to the model, and improvements to its accuracy. With internet access, the user can send data to the server anytime. To send data, the user must open the settings menu and select either the "Sync with server" item or the "Send images" item (Fig. 5). The user can send the following data:

– application operation logs to analyze and detect errors;

– a copy of the application database for analysis and error detection;

– reduced images (default size is 640×640) that were open in the application at the time of transmission;

– original images that were open in the application at the time of sending;

– all images that have been modified since the last data submission.

Following submission, the data is transferred to cloud storage, where it is subjected to a verification process to ensure its accuracy and adherence to formatting standards. Once validated, the data is utilized to train deep learning models. Additionally, the application offers a synchronization feature that enables the client to synchronize their settings and landmine lists with the server via the "Sync with server" menu item. This unidirectional synchronization allows the client to download new data and update their settings. If a new setting is introduced on the server, the client can download its value without affecting other settings.

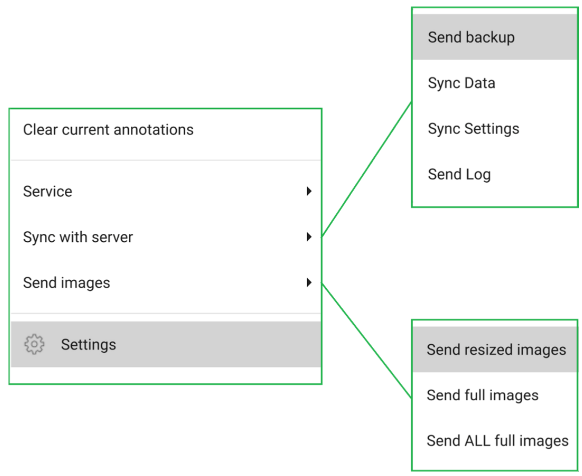


Fig. 5. Screenshots of the menu for sending data to the server: on the left — the main settings menu, on the right — the submenu for sending/receiving data

6. Discussion of the results of the study on the use of a mobile application for landmine detection in images

Modern machine learning models and the latest mobile technologies make detecting landmines possible. A multiplatform system was developed that supports several modes of operation. As part of the study, the system was tested on the Android operating system; however, the use of the Qt framework together with cloud services allows the system to be deployed on other operating systems.

The mobile application allows the user to open an image from the device gallery (Fig. 2, *a*) or take a photo (Fig. 2, *b*). The application enables the user to preview the image, after which the user may choose to retake it or continue with the annotation process. In the online mode, the application transmits a request to a cloud service, where recognition is conducted using a deep learning model [19]. This mode may prove sufficient for recognition when internet speeds are limited. However, the Internet is typically very slow in areas where landmines are likely to be found. Therefore, the offline mode is a significant advantage of the system. In this mode, the recognition speed depends on the configuration of the mobile device and its current load but usually takes 1 to 5 seconds. The offline mode allows the user to collect data to improve the model by opening new or existing images in the application for further sending to the server.

The user can edit the results if the application cannot recognize a landmine, its type, or its boundaries. To do so, the user must first select the object with a frame and choose its type from the drop-down list. It should be noted that this option is also available in the non-detection mode, although it is not a prerequisite.

All images opened in the app will be designated for transmission to the server for analysis and potential incorporation into the model. The application does not collect any personal data or geolocation information and employs modern encryption algorithms to safeguard data [20]. Transmission of data is exclusively at the user's discretion.

The findings of this study corroborate the efficacy of utilizing mobile applications to detect landmines. In contrast with the study [5], which merely considers the theoretical possibility, this paper presents a mobile application's practical implementation and testing results. Although the study [5] acknowledges the typical constraints associated with funding for novel initiatives, this paper demonstrates the practical applications of a mobile application. In contrast to the mobile applications referenced in [6], which are no longer supported, the application developed in this study is compatible with the majority of Android devices. Although [7] identifies challenges associated with offline mode implementation in mobile applications, this study has developed an application that can operate in both online and offline modes. In contrast to the approach taken in [8], which restricts the functionality of a mobile application to the vicinity of a desktop computer, the application developed in this study is not constrained by such limitations. In comparison to the methodology employed in [9], which utilized a third-party virtual reality engine and lacked its own dataset, this study employs a comprehensive deep learning model capable of recognizing landmines under diverse conditions. Moreover, in contrast to the online resources referenced in [10, 11], the developed application is equipped with an offline mode. It is capable of functioning in locations with limited or no internet connectivity.

The developed application successfully performs mobile landmine detection. It is suitable for use in real-world conditions due to its high accuracy, ability to work offline, and the availability of a function for editing results and feedback.

It is important to note that the online testing of the app is contingent upon the speed of the Internet, which is often inadequate or non-existent in areas where landmines are located. In light of these considerations, this paper focuses on the results of offline testing. It should be noted that the system is subject to certain limitations, including the potential impact of low internet speed on online operation and the performance constraints of specific mobile devices. Currently, the application is tested on the Android operating system, but plans include expanding its support to other operating systems.

Further development plans include adding new types of landmines, improving the model, and considering user feedback and suggestions. One development area is creating a training module where users will be asked to recognize objects. Additionally, functions for informing and displaying additional information about recognized objects are planned for inclusion.

7. Conclusions

1. This research has successfully developed and evaluated a mobile application for landmine detection, implemented in both online and offline modes. In both cases, deep learning models trained on 3D printed landmine replicas are employed. The application was tested offline on the Android operating system but can be deployed on other platforms. During its development, the app was designed to be used without internet access and to be as user-friendly as possible. The use of deep learning models enables the system to successfully recognize explosive objects, demonstrating 89 % recall and an average speed of 2.1 seconds.

2. The function of editing recognition results plays a pivotal role in the enhancement of deep learning models. It enables the selection of an area within an image through a finger touch or a rectangle gesture, as well as the identification of the specific type of landmine. It facilitates the incorporation of user feedback, which can rectify model errors and furnish supplementary data for training. With this feature, the model can receive additional data on the location and type of landmines, which allows it to adapt more effectively to real-world scenarios and enhance recognition accuracy.

3. The capacity to transmit recognition outcomes renders the application a valuable repository of data, particularly considering the dearth of information about the subject matter. The proposed approach enables the systematic organization of data collection and analysis, thereby facilitating the enhancement of models. The implementation of this approach represents a crucial phase in developing a productive mobile application for landmine detection. The capacity to transmit various data has been incorporated, including application logs, a copy of the database, reduced-quality images, the original images, and those modified since the previous transmission. The lack of a unified system for data collection on explosive devices poses a significant challenge in detecting and defusing such objects. The proposed application can streamline the gathering and examination of data from various contributors within the demining sector. In an area where information access is constrained, this could prove invaluable in advancing recognition models and consequently accelerating demining efforts within Ukraine.

Conflict of interest

The author declares that they have no conflict of interest about this study, including financial, personal, authorship, or other, that could affect the research and its results presented in this paper.

Financing

The study was conducted without financial support.

Data availability

The manuscript has data associated with the data warehouse (public datasets on printed and real landmines – https://universe.roboflow.com/oleksandr-kunichik-sugbr).

The data will be provided upon reasonable request.

Use of artificial intelligence tools

The author confirms that no artificial intelligence technologies were used to create this work.

Acknowledgments

The author gratefully acknowledges the assistance of members of the Ukrainian Armed Forces with the military call signs Lisnyk (Forester) and S.T.A.L.K.E.R. in obtaining real landmine photographs and sharing their expertise on landmine identification.

References

1. Landmine Monitor 2022. Available at: https://backend.icblcmc.org/assets/reports/Landmine-Monitors/LMM2022/Chapter-Images/Downloads/2022_Landmine_Monitor_web.pdf
2. Landmine Monitor 2023. Available at: https://backend.icblcmc.org/assets/reports/Landmine-Monitors/LMM2023/Downloads/Landmine-Monitor-2023_web.pdf
3. In Ukraine, 128,000 km2 of land and 14,000 km2 of water area are contaminated with explosives. Ministry of Defence of Ukraine. Available at: https://www.mil.gov.ua/news/2024/10/05/128-000-kv-km-suhodolu-ta-14-000-kv-km-akvatorii-ukraini-zabrudneno-vibuhonebezpechnimi-predmetami
4. Dog works faster than person with metal detector. Rescue operations by SES in Mykolaiv. Hromadske. Available at: https://www.youtube.com/watch?v=HDz17-1yeIk
5. Dorn, A. W. (2019). Eliminating Hidden Killers: How Can Technology Help Humanitarian Demining? Stability: International Journal of Security and Development, 8 (1). https://doi.org/10.5334/sta.743
6. Annual Report 2013. United Nations Mine Action Service. Available at: https://www.unmas.org/sites/default/files/unmas_2013_annual_report_digital_presentation_0.pdf
7. Susanto, A. P., Winarto, H., Fahira, A., Abdurrohman, H., Muharram, A. P., Widitha, U. R. et al. (2022). Building an artificial intelligence-powered medical image recognition smartphone application: What medical practitioners need to know. Informatics in Medicine Unlocked, 32, 101017. https://doi.org/10.1016/j.imu.2022.101017
8. Mori, R., Okawa, M., Tokumaru, Y., Niwa, Y., Matsuhashi, N., Futamura, M. (2024). Application of an artificial intelligence-based system in the diagnosis of breast ultrasound images obtained using a smartphone. World Journal of Surgical Oncology, 22 (1). https://doi.org/10.1186/s12957-023-03286-1
9. Hameed, Q. A., Hussein, H. A., Ahmed, M. A., Salih, M. M., Ismael, R. D., Omar, M. B. (2022). UXO-AID: A New UXO Classification Application Based on Augmented Reality to Assist Deminers. Computers, 11 (8), 124. https://doi.org/10.3390/computers11080124
10. Interaktyvna mapa terytoriy, yaki potentsiyno mozhut buty zabrudneni vybukhonebezpechnymy predmetamy. State Emergency Service of Ukraine. Available at: https://mine.dsns.gov.ua/
11. Bezpeka Info. United Nations Children's Fund (UNICEF). Available at: https://courses.bezpeka.info/home
12. Kalifa, I., Youssif, A., Adel, A. (2014). The Use of Mobile Technology for Detecting Landmines. International Journal of Computer Applications, 92 (5), 42–45. https://doi.org/10.5120/16008-5034
13. Mobile Operating System Market Share Worldwide for 2023 year. Statcounter Global Stats. Available at: https://gs.statcounter.com/os-market-share/mobile/worldwide/2023
14. C++ Framework. Qt. Available at: https://www.qt.io
15. Open Neural Network Exchange. Available at: https://onnx.ai
16. ONNX Runtime. Available at: https://onnxruntime.ai
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://doi.org/10.1109/cvpr.2016.91
18. Kunichik, O., Tereshchenko, V. (2023). Improving the accuracy of landmine detection using data augmentation: a comprehensive study. Artificial Intelligence, 28 (2), 42–54. https://doi.org/10.15407/jai2023.02.042
19. Kunichik, O., Tereshchenko, V. (2024). Determining the effectiveness of using three-dimensional printing to train computer vision systems for landmine detection. Eastern-European Journal of Enterprise Technologies, 5 (1 (131)), 17–29. https://doi.org/10.15587/1729-4061.2024.311602
20. Secure Sockets Layer (SSL). Available at: https://openssl.org
21. Dwyer, B., Nelson, J., Solawetz, J. et al. (2022). Roboflow (Version 1.0) [Software]. Available at: https://roboflow.com