UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

# Identifying the Effects of Russian Aggression on Agricultural Fields in Ukraine through Classification Approaches and Satellite Imagery

*Author:*
Sofiia MYNTIUK

*Supervisor:*
Oles DOBOSEVYCH
Rostyslav HRYNIV
Bogdan KULYNYCH

*A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science*

*in the*

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

# Declaration of Authorship

I, Sofiia MYNTIUK, declare that this thesis titled, "Identifying the Effects of Russian Aggression on Agricultural Fields in Ukraine through Classification Approaches and Satellite Imagery" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

*"Decide what is yours to hold and let the rest go."*

Dr. T.A. Swift

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

**Identifying the Effects of Russian Aggression on Agricultural Fields in Ukraine through Classification Approaches and Satellite Imagery**

by Sofiia MYNTIUK

# *Abstract*

Detection and assessment of shelling-induced damage to the agricultural fields of Ukraine are crucial for ensuring the safety of civilians, as it helps to estimate the number of unexploded ordnances in the region. Most existing approaches to damage detection solve this problem for buildings, and the crater detection task is usually solved either for historical or planetary images. In this thesis, we aim to explore the applicability of existing approaches to the task of crater detection in agricultural fields in Ukraine. We collect and annotate a dataset with satellite images of Ukrainian agricultural fields. We experiment with solutions that include classification methods and conduct the hyperparameter search to find the best model for our data. We analyze the impact of each hyperparameter on the network performance and demonstrate the network's ability to generalize to new locations.

The code used in our solution can be found in the project's GitHub repository.

# *Acknowledgements*

I am thankful to my supervisors, Oles Dobosevych, Rostyslav Hryniv, and Bohdan Kulynych, for guiding and helping me throughout the period of working on this thesis. I am eternally grateful to my teachers, who have supported and motivated me during my studies. I would like to express my gratitude to Iryna Popovych for her help with data acquisition and advice. Many thanks to all of my family, friends, and colleagues for supporting me.

I could not have worked on this thesis without all the people who helped on all the project stages: the annotation team and all the people who gave invaluable advice.

Words cannot express my gratitude to the Armed Forces of Ukraine for protecting our lives.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **ML** | **M**achine **L**earning |
| **DL** | **D**eep **L**earning |
| **CV** | **C**omputer **V**ision |
| **CNN** | **C**onvolutional **N**eural **N**etwork |
| **SVM** | **S**upport **V**ector **M**achine |
| **UXO** | **U**ne**x**ploded **O**rdnance |
| **SESU** | **S**tate **E**mergency **S**ervice (of) **U**kraine |
| **NIR** | **N**ear-**i**nfra**r**ed |
| **SWIR** | **S**hort**w**ave **i**nfra**r**ed |
| **RGB** | **R**ed, **G**reen, **B**lue |
| **MM** | **M**onochromatic **m**ode |
| **PM** | **P**anchromatic **m**ode |
| **DIC** | **D**aily **i**maging **c**overage |
| **AUROC** | **A**rea **u**nder the **r**eceiver **o**perating **c**haracteristic |

*Dedicated to the Armed Forces of Ukraine*

# Chapter 1

# Introduction

The ongoing Russo-Ukrainian war has had disastrous effects on the Ukrainian economy, including one of the country's largest and most important economic sectors – agriculture. War has devastated agriculture, jeopardizing food production chains and making it impossible to harvest crops and use the fields that have been shelled. According to Kyiv School of Economics (2022), as of June of 2022, the damage to the agricultural sector caused by Russia's invasion totals to $4.29 billion, of which $2.135 billion is attributed to farmland and unharvested winter crops, and $89.1 million is attributed to perennial crops. Numerous agricultural fields in Ukraine now can not be used due to craters and unexploded ordnances (UXO). As the frontline regions that have suffered from shelling the most are also famous for their fertile soil (chernozem), the urgency of restoring those fields to their pre-war condition will only grow with time as the stocks of harvested crops decrease.

Despite the scale of the impact of shelling on the economy, the danger that UXOs pose to both troops and the civil population of Ukraine is much more severe, as it puts human lives at risk[1,2]. To this day, UXOs from World War II can still detonate if disturbed, causing harm to people nearby[3]. Fortunately, unlike in the mid-XX century, science and technology can help us prevent such long-lasting damage and ensure that all Ukrainian lands are safe.

The goal of this thesis is to explore the usage of machine learning (ML) and deep learning (DL) approaches in combination with satellite imagery to detect and assess the damage to Ukrainian agricultural fields caused by the Russian invasion of Ukraine. We specifically target craters from the missiles under the assumption backed up by the State Emergency Service of Ukraine's (SESU) experience that there is a higher risk of UXOs around the areas with many craters from detonated missiles. SESU can not cover all the potentially dangerous fields quickly to determine which ones can be used and walked on and which require demining, as it is a thorough and laborious process. Therefore, possible applications of the results of this project include putting all the potentially dangerous locations on a map for civilians to use to ensure safer transportation and prevent processing fields that may contain UXOs. Another way to use the proposed approach is to pass the results to SESU so that they can prioritize the order of the investigation and future demining.

Accomplishing this task presents several challenges. One of them is that, according to Solovey (2022), there are several hundred types of soil in Ukraine, each possessing different characteristics, reacting differently to the humidity level, having different fertilizers used on them, etc. Differences in natural terrain elements, even

[1] https://lb.ua/society/2023/02/25/547129_hersonshchini_zaginuli_dvoie.html

[2] https://suspilne.media/422451-pidirvavsa-na-rosijskij-mini-pid-cas-polovih-robit-na-hersonsini-zaginuv-65-ricnij-colovik/

[3] https://suspilne.media/347302-u-centri-kieva-znajsli-artilerijski-snaradi-casiv-drugoi-svitovoi-vijni/

within a small area, make the damage assessment even harder, as the ground can slope and vary in elevation. Therefore, craters form differently in different places, even within relatively close proximity.

Another obstacle is the variety of missiles that are used in the Russo-Ukrainian war. They differ in gauge, blast radius, fragmentation capability, age, and other characteristics that affect the appearance of craters. Moreover, they affect the unexploded ordnance rate, meaning the accurate estimation of the number of UXOs is nearly impossible. Even the angle and the distance from which the missile was fired affect the appearance of craters (Department of the Army, 1996).

One of the most critical issues that arise in damage assessment of the fields is rapid changes in the appearance of the crater and the field overall that can be caused by season, weather (for example, water erosion after rain or falling snow), changes in the greenery around the crater, human activity (for instance, farmers covering the craters with the ground, making it harder not only to identify the crater from the air but also to extract the missile's body later in time). Some craters that appeared at the beginning of the Russian invasion of Ukraine now may be barely visible, especially on satellite imagery. Such rapid changes make it crucial to analyze historical data for the same location starting from before the invasion. In that way, we do not risk missing any craters and UXOs. However, this need eliminates the usage of aerial data (usually of higher resolution) because it is not collected continuously. The fact that the whole sky in Ukraine can be covered in clouds for months during colder seasons is an additional obstruction to collecting a large enough dataset solely from satellite imagery.

In this work, we direct our attention toward Machine Learning based approaches to understand how well such approaches can adapt to the domain of Ukrainian agricultural fields to solve the task of detecting and assessing the damage inflicted on them.

In Chapter 2, we provide a general overview of the existing approaches to the damage detection of buildings and agricultural fields. Chapter 3 covers the details of the proposed methods for data processing, dataset annotation, and damage detection of agricultural fields. In Chapter 4, we describe the conducted experiments more thoroughly, and, finally, in Chapter 5, we shortly describe the conclusions of this thesis and outline future steps.

# Chapter 2

# Related Work

This chapter provides a general overview of the war-inflicted damage assessment task and related questions. In Section 2.1, we provide an overview of several existing satellite imagery providers and their products. In Section 2.2, we outline both supervised and unsupervised approaches to the damage assessment of the buildings; Section 2.3 contains a review of the crater detection approach specifically. Section 2.4 covers synthetic crater data generation approaches.

## 2.1 Satellite imagery sources

In recent years, both the quantity and quality of satellite imagery have improved significantly, and the number of providers has increased. The data vary not only in spatial and temporal resolution but also in the type of data they produce. This section gives an overview of existing satellite imagery providers that we considered using for this project.

### 2.1.1 Sentinel-2 data

Sentinel is the name of several Earth Observing missions started by the Copernicus Programme. Sentinel-2 is the second mission launched by Copernicus. The images acquired through these satellites have a spatial resolution between 10 and 60 m and a revisit rate of 10 days. In fact, the presence of the second satellite in this constellation reduced the revisit cycle duration to 5 days. In addition to the visible red, green, and blue bands, the Sentinel-2 captures ten other frequencies from the spectrum's near-infrared (NIR) and short-wave infrared (SWIR) parts.

The Sentinel-2 data is publicly available through the Copernicus Data and Information Access Service cloud environments. Even though it covers the whole territory of Ukraine, the spatial resolution (from 10 meters) is not enough to detect smaller craters. Therefore, this type of imagery is not fully applicable to crater detection in agricultural fields.

### 2.1.2 Planet data

Planet Labs PBC is an Earth-imaging company that operates several satellite constellations. First, Dove is a constellation of small satellites that provide high-frequency imaging. The resulting spatial resolution is 3-5m, and the bands include RGB and NIR frequencies. SkySat is a constellation that takes imagery with submeter spatial resolution. The imagery is multispectral, containing RGB, Panchromatic, and NIR frequencies. Finally, the PlanetScope constellation captures high-frequency (RGB and NIR frequencies) imagery of the Earth's surface with a spatial resolution of 3-5 meters.

Even though the PlanetScope and Dove constellations provide higher spatial resolution than the Sentinel-2 imagery, it is still suboptimal for detecting some craters. The submeter resolution of the Planet SkySat constellation would be suitable for the crater detection task; however, these high-resolution Planet data are not publicly available, and we have not succeeded in obtaining them from the provider.

### 2.1.3 Maxar data

Maxar Technologies Inc. is one of the leading space technology companies. They offer several satellite imagery platforms: WorldView-1, WorldView-2, WorldView-3, WorldView-4, and GeoEye-1. WorldView-1 provides imagery with a spatial resolution of 0.5 meters and has only a panchromatic mode (PM) (meaning that the images are in black and white). Its revisit rate is 1.7 days, and it can capture up to $750,000km^2$ per day. WorldView-2 has a resolution of 46 centimeters for both the panchromatic images and RGB images taken in multispectral mode (MM). With a revisit rate of 1.1 days, its daily imaging coverage (DIC) is up to one million $km^2$ per day. WorldView-3 has a resolution of 31 centimeters for panchromatic images and 1.24 meters for multispectral ones. Additionally, it can capture short-wave infrared (SWIR) data. With a revisit time of 1.1 days, it can cover up to $680,000km^2$ per day. WorldView-4 provides a resolution of 31 centimeters for both panchromatic and multispectral images and can cover $680,000km^2$ per day. Its revisit time is 1.1 days. Finally, GeoEye-1 has a resolution of 41 centimeters and provides both panchromatic and multispectral imagery. It has a revisit time of 3 days and can capture up to $700,000km^2$ per day.

Table 2.1 summarizes the difference between Maxar imagery platforms. In this work, we utilize RGB images, so all technologies except WorldView-1 can be applied. WorldView-2 and WorldView-4 have the best spatial resolutions and a short revisit time, so that these platforms would suit our needs the best. However, the imagery produced by the Maxar platforms is not public, which limits its usage.

| Platform name | PM resolution (m) | MM resolution (m) | Revisit time (days) | DIM ($km^2$) | SWIR mode |
|---|---|---|---|---|---|
| WorldView-1 | 0.5 | – | 1.7 | 750,000 | ✗ |
| WorldView-2 | 0.46 | 0.46 | 1.1 | 1,000,000 | ✗ |
| WorldView-3 | 0.31 | 1.24 | 1.1 | 680,000 | ✓ |
| WorldView-4 | 0.31 | 0.31 | 1.1 | 680,000 | ✗ |
| GeoEye-1 | 0.41 | 0.41 | 3 | 700,000 | ✗ |

TABLE 2.1: Summary of Maxar imagery platforms.

## 2.2 Damage assessment of buildings

Assessing the damage caused to buildings is used for loss estimation[1] and overall understanding of the scale of damage caused by war or natural disasters, especially if direct access is impossible (for example, due to shelling, occupation, or the danger of other buildings collapsing).

[1] https://damaged.in.ua/

During the past few years, growth in spatial resolution and popularization of high spatial resolution (HSR) images unlocked many applications and possibilities for research. Contrary to the approaches that used moderate-resolution remote sensing, primarily evaluating pixel-wise changes (Yusuf, Matsuoka, and Yamazaki, 2001), newer HSR data allow the extraction of information about damage for each building separately. For example, in Aimaiti et al. (2022), similarly to Sandhini Putri, Widyatmanti, and Umarhadi (2022), Tong et al. (2012), and Brunner, Lemoine, and Bruzzone (2010), where either pure HSR images or their combination with additional data sources are utilized, the authors explore the capabilities of Sentinel-1 and Sentinel-2 images for building damage assessment using change detection techniques. Sentinel-1 is a Synthetic-aperture radar (SAR) satellite, meaning that the satellite emits its own energy and measures the amount of energy reflected from the surface. Sentinel-2 data, on the contrary, is obtained from a Multispectral imager (MSI) and consists of visible, near-infrared (NIR), and short-wave infrared (SWIR) bands. Using the SAR imagery of Syrian regions, the authors compared pre-war data with the newest data, where some buildings were already ruined or partially destroyed. For the ground-truth labels, they used the damage assessment data published by UNOSAT. The authors used two standard algorithms for the change detection block: log ratio of intensity (Jung and Yun, 2020) on the SAR data and the Gray Level Co-occurrence Matrix on the Surface Reflection data from Sentinel-2. Even though buildings undergo fewer changes than the ground, some areas still had to be excluded from the analysis because of natural phenomena. The finding only confirms our belief that the described approach, although showing decent results for buildings, is hardly appropriate for the fields suffering from much more rapid changes. Moreover, the abovementioned approaches require constructing hand-crafted image features, usually specific to the site and disaster. This limitation makes the approach less generalizable and applicable to rapid disaster response.

Deep learning (DL) approaches generalize better to other types of buildings and disasters, as they usually do not require hand-crafted features. Typically, the pipeline of the DL method for the building damage assessment can be separated into two main steps (Valentijn et al., 2020; Ge, Gokon, and Meguro, 2020):

1. Building localization, the primary outcome of which is to group all the pixels belonging to the same building on the pre-disaster or pre-war images. For example, Gupta et al. (2019) utilize a segmentation approach based on the UNet architecture (Ronneberger, Fischer, and Brox, 2015).

2. Classification of the building images, which is frequently achieved by training a Convolutional Neural Network (CNN). This step aims to create correspondences between the pictures of buildings extracted in Step 1 and labels indicating whether they have been damaged (or, sometimes, mapping them to the level of damage, such as light, medium, and severe destruction). In Gupta et al. (2019), the authors utilize the ResNet-50 (He et al., 2015) for this purpose.

Despite the popularity and effectiveness of the cascade approach described above, there are alternative approaches. For example, in Mueller et al. (2021), the authors classify the images of Syrian cities with a CNN and exploit the temporal and spatial clustering of the damaged buildings by training a RandomForest algorithm (Ho, 1995) on the CNN output. To do this, they divide the whole dataset into smaller patches and assign each a label (bombed/not bombed) depending on whether, according to UNOSAT, there are destroyed buildings on this patch or not. This way, they considered historical data throughout the war, not just images before and after, and utilized the temporal consistency and spatial closeness of the destruction. Even though this method generalizes well across Syrian cities, it does not perform well on highly imbalanced datasets, where only a small percentage of buildings have been ruined.

Another approach that addresses the poor generalization across different disasters and proposes a DL-based solution is the one described in Ismail and Awad (2022). By utilizing a Graph Convolutional Network, the authors aim the algorithm to learn aggregation functions and be generalizable for rapid disaster response. Despite promising results in generalizing across different disasters, highly imbalanced datasets still exhibit an issue similar to the one described by Mueller et al. (2021).

An alternative method (Gupta et al., 2019) utilizes a Siamese-UNet architecture, meaning that there are two identical UNet networks with the same weights (Durnov, 2020). The networks are applied to the original and post-disaster images, and then the algorithm performs the pixel-level classification of the concatenated feature maps. The approach underperforms on the datasets with partially destroyed buildings. Even though the algorithm correctly recognizes the ruined area, sometimes it fails at recognizing the whole damaged building instance, incorrectly assigning several labels to the same building, resulting in semantic inconsistency.

To solve the issue, the algorithm has to assess the damage not on the pixel level but on the object level. There is a considerable amount of research on the application of object-based image analysis (OBIA) and patch-based CNNs that aims to solve the abovementioned inconsistency (see, e.g., Zhang et al. (2018), Zhang et al. (2019)). However, most buildings are of geometric shape, which differs from the problem that conventional OBIA approaches aim to solve. As a step-up from the regular OBIA-integrated CNN, the authors in Zheng et al. (2021) utilize a Siamese FCN for accurate building localization and damage assessment. The method has achieved SOTA results and has shown the ability to generalize across types of disasters, including man-made ones.

While there is extensive research on building damage assessment, there is very little analysis of how well those methods generalize to the case of war-inflicted damage cases. Despite the growing number of methods for accessing the damage caused by natural disasters, there remains a gap in understanding the extent of applicability of those solutions to war-zone buildings. Hence, further research is needed to either extend the existing damage assessment pipelines' generalization to damage resulting from warfare or develop different methods. Difficulties in data collection, labeling, and validation are one of the many factors contributing to the slow pace of research in this field.

## 2.3 Crater detection

Contrary to the building damage assessment, detecting and estimating damage in the agricultural fields has spurred much less interest in the research community. The obstacles presented by the assessment of the war-inflicted damage to the buildings are compounded when assessing and estimating the damage caused to the agricultural fields. While there exists labeled data by UNOSAT for ruined buildings, the ground-truth data is practically absent in the domain of agricultural fields suffered from shelling. The task of data search, acquisition, and labeling is time- and labor-consuming and can require expertise to distinguish between terrain elements and craters. Moreover, due to the variety of soil types and missiles, developing an algorithm that generalizes well even throughout one country is challenging.

As crater detection is an important research topic in planetary science and planetary data are more accessible, there are many methods specifically for detecting the craters on space bodies. Due to the tasks being related, we provide an overview of the approaches to crater detection on planetary images.

In subsections 2.3.1 and 2.3.2, we give a review of methods used for assessing the war-induced damage to agricultural fields and crater detection on planetary data.

### 2.3.1 Bomb crater detection

This section summarizes the approaches to crater detection and the number of UXOs estimations, as the latter commonly utilizes the solutions to the former. All the described approaches use historical data from WWII. In subsections 2.3.1.1 and 2.3.1.2, we provide the overview of the methods of bomb-induced crater detection and the number of unexploded missiles estimation, respectively.

#### 2.3.1.1 Bomb-induced crater detection

Most of the developed solutions utilize image classification in one way or another. The main difference between those approaches is the feature extraction method.

For example, Jensen, Drauschke, and Förstner (2010) perform the candidate selection by adopting a cross-correlation pattern-matching approach for the patterns derived from the labeled data. For each candidate, the feature vector is computed by applying the scale-invariant feature transform (SIFT) algorithm to the selected area of the image. After that, the classification is performed by the Linear Discriminant Analysis (LDA) algorithm. Furlanello et al. (2003) and Merler, Furlanello, and Jurman (2005) utilize a sliding window approach for candidate selection. As opposed to LDA, they applied principal component analysis (PCA) for the dimensionality reduction and then classified the eigencraters (eigenvectors of the crater images) using the AdaBoost algorithm.

Lacroix and Vanhuysse (2015) have developed a crater detection method using only the circle detection based on the intensity gradient of the image. However, this approach can not be applied to real-world data without any additional verification of the outputs, as there may exist circle-shaped terrain elements or texture present. Nevertheless, this solution may be a step up from the window-sliding method for the candidate selection step.

Aside from purely statistical methods, there are existing applications of DL algorithms to bomb crater detection. While some of them (Brenner, Zambanini, and Sablatnig, 2018) still make use of the sliding window approach, Clermont et al. (2019) reduced the number of candidates by performing the candidate selection using blob detection algorithm, similar to the abovementioned work by Lacroix and Vanhuysse (2015). Further classification was performed with the Inception ResNet (Szegedy et al., 2017). Their solution considered the class imbalance of the dataset by setting higher weights for false detections, making it more generalizable and robust. In contrast, the solution proposed by Brenner, Zambanini, and Sablatnig (2018) was trained under the assumption that the dataset is balanced, therefore, underperformed on real-world data, where the number of images with craters is significantly lower than the ones without any bomb-inflicted damage.

#### 2.3.1.2 UXO risk estimation

As unexploded ordnances (UXOs), also called duds, are the main risk presented by the fields with bomb-induced damage, we decided to expand on the possible approaches to estimating the number of duds. The difficulty in detecting them, even using HSR imagery, lies not only in the relatively small size of the missile body but also in the fact that the UXO itself might be buried. Furthermore, the differences between examples of the craters from the exploded and unexploded ordnance should be established with the help of the professionals for each type of missile, location, and angle from which it has been fired. This task is highly labor- and time-consuming. Therefore, most existing approaches assume that the estimated number of UXOs can be derived from the number of craters. For example, Juhász, Neuberger, et al. (2018) and Sabour, Agarius, and Sadidi (2014) suggest estimating the risk of UXOs per patch under the hypothesis that, according to previous research (Brenner, Zambanini, and Sablatnig, 2018; Byrnes, 2008), approximately 10-15% of the bombs from World War II are still not exploded. The problem with those approaches is that they all assume the constant quota of unexploded bombs among the fired or exploded ones. Such estimations are derived from practice and years of research, which is not applicable in our case due to the variety of missiles used in the Russo-Ukrainian war. It is nearly impossible to detect which ones were used to produce the specific crater based on satellite imagery alone.

It is worth mentioning that most of the described approaches utilize historical aerial imagery, which is of higher resolution compared to satellites. Additionally, it covers a smaller area, meaning there is little to no variation in soil type and weather conditions. Generally, all the research in the direction of bomb crater detection could benefit from collecting the structured and publicly available dataset and developing a benchmark for this task. Nevertheless, most of the HSR satellite and aerial imagery is not available publicly, which is a considerable obstacle to research in this field.

### 2.3.2 Planetary data crater detection

Craters on space bodies are often their most defining factor, which explains the high interest in this topic among the scholars of astrophysics. As the quality of space imagery progresses and allows wider applications, the research community's interest in automatic crater detection is growing.

Overall, we can roughly divide the solutions to automatic planetary crater detection into unsupervised and supervised approaches. While the former is usually purely statistical or utilizes classical CV algorithms, the latter includes traditional learning algorithms and DL approaches.

Most statistical approaches exploit the fact that the majority of craters are round-shaped or elliptical. Thus, many of them utilize either circle or edge detection algorithms. One of the most straightforward statistical approaches to planetary crater detection is the utilization of Hough transform (Salamuniccar and Loncaric, 2010). Some of the methods (Pedrosa et al., 2017) adopt template-matching algorithms. Both of these approaches function under the abovementioned assumption that the craters are round or elliptical, which results in underperformance on irregular-shaped or overlapping craters.

Supervised methods do not adopt this assumption, which makes them more robust and generalizable. For example, in (Kang et al., 2018), authors adopt a HOG-SVM algorithm to the crater detection task. However, even with a spatial resolution of $1.4m$, the smallest craters detected with the algorithm are $20m$ in size. Even though such craters are considered small for the lunar pictures, such results are not sufficient for bomb craters. The CNN-based approaches, such as (Silburt et al., 2019; Wang, Jiang, and Zhang, 2018), show improvement in the size of the detected lunar craters and generalization across the old and new craters.

There is still room for research on how well the solutions for the planetary data generalize to the bomb craters. In Geiger, Martin, and Kühl (2022), authors claim that the model trained on lunar images performs poorly on the existing aerial images of bomb craters. Despite having similar origins (either bomb or meteoroid), the physical properties of craters, space bodies, and their top layers vary, impacting the crater's visual characteristics. For example, due to less rapid weather changes and differences in the atmosphere structure, the craters undergo different types of changes: those on Earth may change their appearance because of water erosion due to rain or vegetation development, while the planetary craters face other weather phenomena. Additionally, the craters on Earth may vary due to the soil type, type of plant life present in the area, temperature, and so on. In contrast, the craters on the other space bodies usually do not differ with respect to such factors.

## 2.4 Synthetic data

As mentioned before in this work, the problem of finding relevant and sufficient data is crucial for automatic crater detection. The larger area and time domain we aim to cover, the more variety in plant life, weather, used missiles, and soil type it brings. Additionally, exploring the satellite datasets to find enough examples of each type of crater is highly time- and labor-consuming, and the annotation of such data requires many hours under the supervision of field experts. Thus, the question of generating the synthetic dataset arises and becomes particularly urgent.

We are aware of only one approach to synthetic crater image generation. In Geiger, Martin, and Kühl (2022), the authors utilize Generative Adversarial Network (GAN) to perform image-to-image (I2I) translation of craters from the lunar images to the Earth pictures. Later, they apply the YOLO (Redmon et al., 2016) object detection system for other crater detection. Training on the combined dataset of real and artificial images showed marginal gains compared to using solely real images. Overall, the problem, once again, lies in the limited dataset, as even I2I translation is performed on a limited area of the target domain.

Aside from the abovementioned underperformance, another problem with this approach is the lack of variety. As mentioned before, the bomb craters differ in color, the intensity of plant life, and more. With I2I translation from the lunar images domain, it is difficult to achieve the color changes in the background and surrounding areas of the craters because such changes usually do not occur in planetary data. However, generating synthetic data that could be applicable to modern bomb craters is an essential concern for the research community.

# Chapter 3

# Approach

This thesis aims to test whether the classification approaches common in building damage assessment and bomb crater detection on historical aerial photos generalize to our domain of interest: satellite imagery of Ukrainian agricultural fields.

We collect and annotate a dataset of natural-color satellite images and study the applicability of different neural network architectures and the impact of their hyperparameters on the metrics. For evaluation, we implement metrics that can be applied to rate the models' performance according to various objectives.

As described in Chapter 2, existing approaches to the damage detection task include segmentation and classification. However, segmentation was only widely applied to the damage assessment of buildings (see Section 2.2), whereas classification is the preferred algorithm for the crater detection and war-induced damage assessment of agricultural fields (see Section 2.3.1). Therefore, the algorithm we adopted for our solution is classification.

Section 3.1 describes the dataset preprocessing and annotation. In Section 3.2, we review the metrics developed for this thesis, and in Section 3.3 – algorithm used for the classification of the images, including augmentation and model selection methods used.

## 3.1 Dataset

The dataset for classification consists of the images separated into classes according to the assigned labels. As we utilize large satellite images, our task requires splitting the whole image into smaller patches, similar to Mueller et al. (2021), and classifying each patch into a category. Due to the small amount of data, we only identify two classes of damage: bombed and not bombed. The region is considered bombed if at least one crater is present and not bombed if there are no signs of bomb-induced damage.

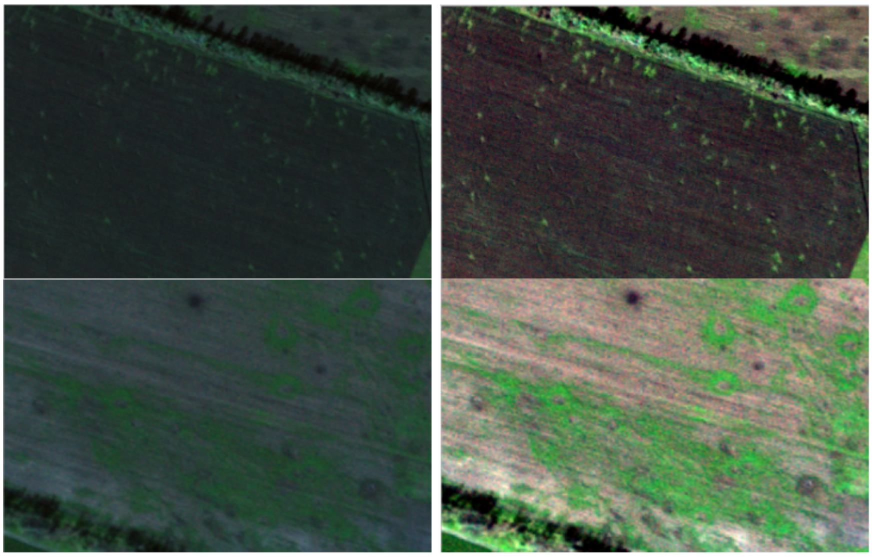
### 3.1.1 Preprocessing

For better visibility of the craters during the annotation process and overall understanding of the image context, we have performed a min-max stretch between the $2^{nd}$ and the $98^{th}$ percentiles of each channel, as it is a standard image preprocessing technique used by geographic information software (QGIS Development Team, 2009).

Let $I_{input}$ be the input image of shape $(h, w, c)$, where $h$ is the height, $w$ is the width, and $c$ is the number of channels. Then, for each channel $I^i_{input}$ we get the following output:

$$I^i_{output} = \frac{I^i_{input} - P(I^i_{input}, 2)}{P(I^i_{input}, 98) - P(I^i_{input}, 2)} \cdot 255, \tag{3.1}$$

where $P(I^i_{input}, j)$ is the $j^{th}$ percentile of $I^i_{input}$. After performing these manipulations for each channel, we stack the channels back and get the resulting image. Figure 3.1 illustrates the results of such preprocessing: on Figure 3.1a, some of the craters are barely visible, and it may be more difficult to differentiate between a crater and a terrain element in comparison to Figure 3.1b.



(A) Original images.

(B) Images after min-max stretch.

FIGURE 3.1: An example of the preprocessing result.

The satellite imagery providers typically save and distribute images according to how the Earth is positioned, which means they are sometimes rotated north-up. This results in patches with no information at the edges (see Figure 3.2) and may lead to overfitting of the model because the parts near the edge contain less image information, leading to the majority of them being assigned the 'not bombed' label.

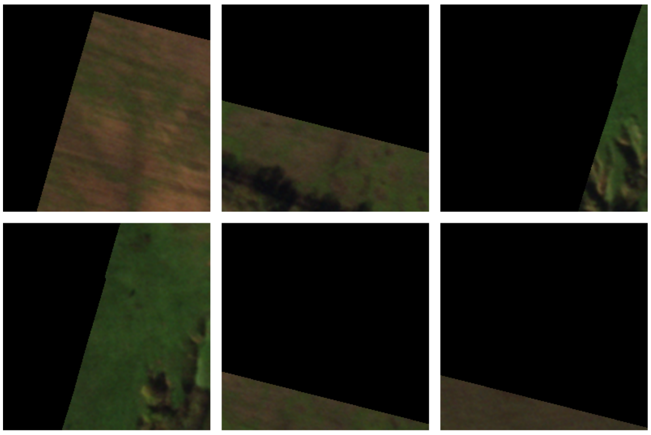


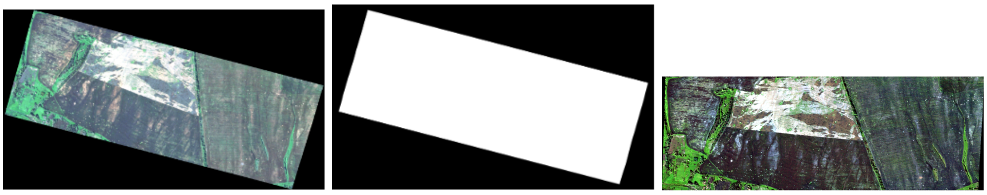
FIGURE 3.2: Examples of patches with no information at the edges.

To avoid further overfitting of the models on such dark corners without jeopardizing the size of the dataset, we rotate and crop the image to minimize the black borders.

First, we extend the image's borders to avoid artifacts at the corners. After that, the image is thresholded so that all the pixels containing information are white. We perform the morphological closing operation with an elliptical kernel to reduce possible noise from black pixels that can occur inside the picture content (see Figure 3.3b). After extracting the contours, we find the one with the largest area and its minimum bounding rectangle. After that, the transformation matrix is constructed and applied so the resulting image is rotated and cropped to the content (see Figure 3.3c). Algorithm 1 shortly describes the process of the rotation of the image.

**Algorithm 1:** Rotate and crop satellite image

```
Input: image $I$
Output: transformed image $I'$, transformation matrix $M$, output shape $S$
// extend borders to avoid artifacts at the corners after the
   morphological operations
$I_{extended} \leftarrow$ extended $I$ with borders of $I$ by 100 pixels in all directions with
  black pixels;
// resize to reduce noisy contours
$I_{thresholded} \leftarrow$ binary image of $I_{extended}$;
$I_{morphed} \leftarrow$ result of morphological closing to $I_{thresholded}$ with an elliptical
  kernel;
$Contours \leftarrow$ contours in $I_{morphed}$;
$ContourMax \leftarrow$ the contour with the largest area;
$Rectangle \leftarrow$ minimum bounding rectangle for $ContourMax$;
$\theta, (cx, cy), (w, h) \leftarrow$ angle, center, and size from $Rectangle$;
// define destination points for affine transformation
$Dest \leftarrow ((0, w), (0, 0), (h, 0), (h, w))$;
$M \leftarrow$ affine transformation from $Rectangle$ to $Dest$;
$I' \leftarrow$ warp $I_{extended}$ using $M$ to size $(w, h)$;
```



(A) Input image.

(B) Image after border expansion, thresholding, and morphological closing.

(C) Warped image.

FIGURE 3.3: An example of image rotation and crop.

### 3.1.2 Labeling

Typically, the annotation process for the classification datasets consists of assigning labels to every image. However, in our case, the pictures are patches of large images, and the size of such patches is predefined at the splitting step. Therefore, in case of changing the patch size, it would be required to reannotate the whole dataset. Additionally, for the annotation process, the general context is crucial to understand whether some details on the image are bomb-induced craters or terrain elements, and by cutting images into patches and presenting them to the annotation team separately, we remove the patch out-of-context, possibly reducing the quality of the labeling.

Therefore, the labeling process, performed by the team of volunteers, consisted of creating a segmentation mask covering all the craters on the image. Then, the annotations were validated by the authors of the work and exported. To produce a dataset suitable for the classification, all the images were post-processed so that if a patch contained a labeled crater, it was assigned the ′bombed′ label and ′not bombed′ otherwise. Algorithm 2 describes the process of assigning the labels to the patches.

This approach sometimes produces patches where only a few pixels of the mask are present. Such patches are assigned the ′bombed′ label, even though no crater is visible. To resolve the issue, we empirically established a threshold for the minimal number of mask pixels on a patch needed to declare it ′bombed′, which minimizes the number of incorrect labels in the train/validation dataset.

**Algorithm 2:** Assign labels to the image patches according to the mask

```
Input: image $I$, mask $M$, patch size $S$, output paths $bombedPath$,
       $notBombedPath$
Output: None
for $i \leftarrow 0$ to $\lfloor \frac{Height(I)}{S} \rfloor$ do
    for $j \leftarrow 0$ to $\lfloor \frac{Width(I)}{S} \rfloor$ do
        // get the beginning and the end of the current patch
        $x \leftarrow i \cdot S; y \leftarrow j \cdot S; x_{\text{end}} \leftarrow x + S; y_{\text{end}} \leftarrow y + S;$
         $patch \leftarrow I[x{:}x_{\text{end}}, y{:}y_{\text{end}}]; patch_{\text{mask}} \leftarrow M[x{:}x_{\text{end}}, y{:}y_{\text{end}}];$ if
         $\sum patch_{mask} \leq threshold$ then
            Label patch as bombed;
        else
            Label patch as not bombed;
        end
    end
end
```

To summarize, the modifications described in this chapter allow us to get the dataset applicable for the classification task. Moreover, they make our approach adaptable to the patch size, meaning we can change it and produce a dataset with an arbitrary patch size. Additionally, the labeled data can be used for other approaches, such as segmentation or (after some modifications) object detection.

## 3.2 Metrics

We have utilized several metrics to evaluate and select the best classification algorithm. Most of the agricultural fields in Ukraine, especially those away from the frontlines, are not bombed, resulting in the class imbalance in the test dataset. In contrast, the training and validation datasets are relatively balanced or imbalanced the other way (the bombed class is better represented). Such inequality in the number of class elements risks running into the base-rate fallacy.

The base-rate fallacy is a tendency to neglect information about the base rate – in our case, the percentage of the agricultural fields that have war-induced damage

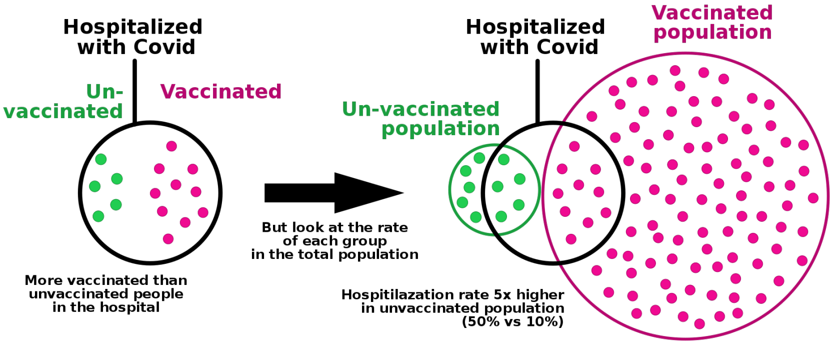

Hospitalized with Covid
Un-vaccinated
Vaccinated
More vaccinated than unvaccinated people in the hospital
But look at the rate of each group in the total population
Hospitalized with Covid
Vaccinated population
Un-vaccinated population
Hospitilazation rate 5x higher in unvaccinated population (50% vs 10%)

FIGURE 3.4: Base-rate fallacy example. Taken from Washington State Department of Health (2022)

present. Therefore, tracking both true and false predictions rate is crucial for understanding the model's learning. Figure 3.4 illustrates an example of the base-rate fallacy in the case of vaccinated and unvaccinated patients.

In our case, the "bombed" class is considered the positive class, and vice versa, "not bombed" - is the negative one; therefore, the positive class in our dataset is represented better than in the real-world data. Hence, the False Positive Rate (FPR) is crucial to understanding the model's actual performance, as it is essential to reduce the number of false alarms.

In this case, tracking precision and recall, which have the number of true positive predictions (TP) in the numerator, is not the best choice of the metrics. Those metrics would still be high even when a classifier yields no true negative (TN) predictions.

$$Precision = \frac{TP}{(TP+FP)} \quad Recall = \frac{TP}{(TP+FN)} \tag{3.2}$$

In this thesis, one of the metrics that we have used for the evaluation of the classification methods is the area under the receiver operating characteristic (AUROC), as it measures the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR), and taking into account FPR, as mentioned above, is critical in our case.

Another metric that we have implemented is a True Positive Rate at a fixed False Positive Rate (TPR@FPR). It measures the classifier's performance at a fixed FPR threshold. For example, if we allow FPR=0.4 (40% of the not bombed patches being classified as bombed), then we can find the TPR that corresponds to it, obtaining the value of TPR@FPR=0.4. Figure 3.5 illustrates an example of the ROC curve and how it is related to the TPR@FPR.

This metric is applicable in scenarios where the cost of FP and FN predictions are different. For example, in one application, one may need to minimize FPR to avoid false alarms, while the other task may require FNR minimization. By fixing the FPR threshold, we can measure the performance of the classifier at a specific point, which can be changed based on the application and desired sensitivity to the negative samples. In our case, reducing false positives is crucial; therefore, we measure TPR@FPR=0.2, TPR@FPR=0.1, and TPR@FPR=0.05.

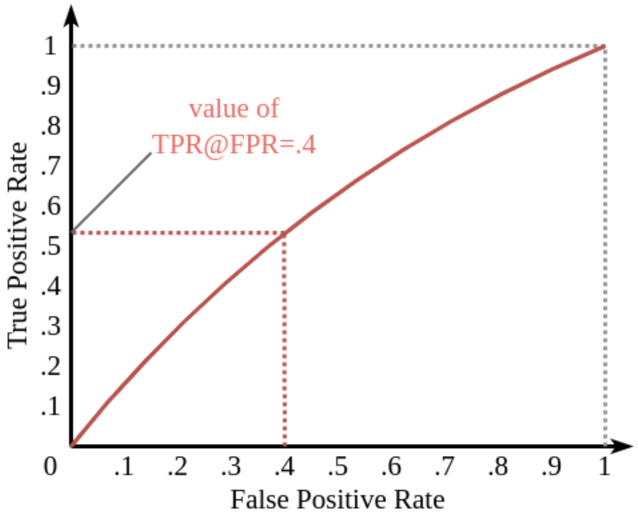


FIGURE 3.5: Example of the ROC curve and TPR@FPR values.

## 3.3 Algorithm

We utilize the classification approach for the bomb-induced damage assessment of the task. Similarly to Mueller et al. (2021), Furlanello et al. (2003), and Merler, Furlanello, and Jurman (2005), for the candidate selection step, we separate the whole image into patches and run each one through a classifier.

### 3.3.1 Augmentation

To avoid overfitting and expand the diversity of the dataset, we perform several data augmentations, similar to those applied by Mueller et al. (2021): first of all, the image is rotated randomly by an angle in the range $(-45°, 45°)$, randomly flipped in horizontal and vertical directions. As the craters may result from the missile being fired from different locations, their shape and form may change, and the rotation and flip augmentations prevent the overfitting. The image can also be normalized to the training set mean and standard deviation depending on the hyperparameters set. Also, depending on the abovementioned hyperparameters, we perform the upsampling of the underrepresented class.

### 3.3.2 Classification approaches

For the classification algorithm, we have tested two architectures: the ResNet-50 (He et al., 2015) and the Vision Transformer (ViT) (Lee, Lee, and Song, 2021). For the training, we experimented with pretrained weights and evaluated the models pretrained on the ImageNet (Russakovsky et al., 2015) dataset and a dataset with Sentinel-2 images.

#### 3.3.2.1 ResNet

ResNet (short for **Res**idual **Net**work) is a family of neural networks designed to mitigate the risk of vanishing gradients that comes with increasing the depth of the network. The main idea introduced with ResNet is the usage of skip connections that enable the flow of information between earlier and later layers, allowing the network to learn more effectively by allowing the gradient values to reach the early layers of the network.

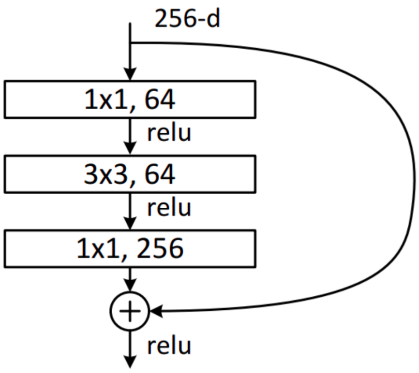
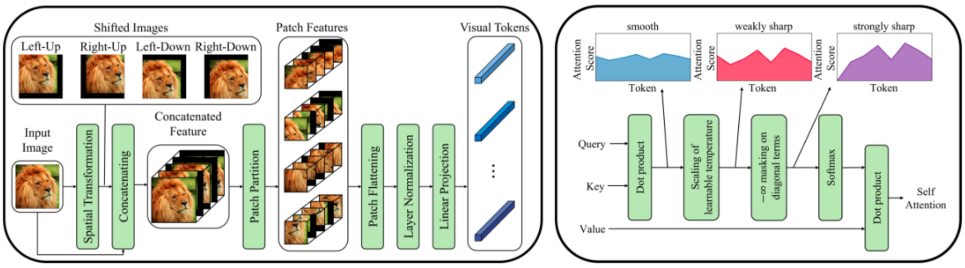


FIGURE 3.6: ResNet-50 building block. Taken from He et al. (2015).

The ResNet-50 model is a member of the ResNet family that consists of 50 layers. It is widely used for image classification tasks, as well as other tasks in computer vision. Building blocks of ResNet-50 consist of three convolutional layers with batch normalization and ReLU activation. The block is followed by a skip connection that adds its input to the output (see Figure 3.6). These blocks are also called ’bottleneck blocks.’

#### 3.3.2.2 Vision Transformers (ViT)



(A) Shifted Patch Tokenization.

(B) Locality Self-Attention.

FIGURE 3.7: ViT architectures of the Shifted Patch Tokenization and Locality Self-Attention. Taken from Lee, Lee, and Song (2021)

Vision Transformer (ViT) is a family of architectures applied to computer vision tasks that, instead of standard computer vision blocks (such as convolution), utilize transformer blocks, which are widely used for natural language processing (Vaswani et al., 2017). ViT models take an image as an input and flatten it into a sequence of patches later fed through a transformer block to learn a representation of the image (see Figure 3.7). It showed a significant improvement in the image classification tasks when pretrained on large datasets.

ViT Small Patch is an architecture from the ViT family that passes smaller patches (for example, $16 \times 16$) to the transformer block. It makes ViT Small Patch more effective for tasks that require high-resolution images and increases its ability to recognize more fine-grained details. On the negative side, it increases the training time and requires more computational resources.

### 3.3.3 Model selection

The architectural details of a neural network affect its ability to learn and generalize. Deeper networks can learn more complex features but are more prone to overfitting. Models that use smaller patches (e.g., ViT Small Patch) are generally better at recognizing fine-grained details but require more computational time and resources. Hyperparameter choice also affects the model's performance. In this thesis, we have searched the following hyperparameters: base size, batch size, minority class upsampling presence, dropout rate, learning rate, learning rate schedule patience, normalization presence, optimizer, weight decay, and the usage of pretrained weights (see Table 4.3).

The base size hyperparameter is the size to which all the input patches are resized. It is only used for ResNet models, as the ViT Small Patch architecture requires fixed-size input images. A larger base size can benefit the model's ability to recognize smaller details on the image. A smaller base size decreases computational resources and time required for the model training but may make the model less sensitive to smaller features in the input image.

The batch size hyperparameter determines the number of input samples fed into the model between two subsequent model training updates. For ResNet models, a larger batch size may increase memory usage, but it reduces training time. ViT models are based on self-attention, which requires processing all input images simultaneously. Therefore, the batch size may not significantly impact training time, but a larger batch size may result in better generalization.

Minority class upsampling is performed during training by altering the probability of the underrepresented class (in our case, 'not bombed' one) occurring in the batch. Generally, the upsampling may improve the model performance, especially on imbalanced datasets, by allowing it to learn better representations of the minority class. However, the upsampling may increase the time and memory required for the training process.

Dropout is a regularization technique that prevents overfitting in neural networks. It randomly sets some neurons' weights to zero during training. As a result, the model learns more robust features, as it can not rely too much on any neuron. The dropout hyperparameter determines the probability that the neuron will be dropped during the training forward pass. A lower dropout rate can generally lead to better performance on the training set but worsen generalization abilities. A higher dropout rate results in worse metrics while training but a great generalization ability. However, as mentioned in Steiner et al. (2021), the peak drop probability for which the ViT models show improvement in terms of regularization is 0.1. Therefore, higher dropout values may lead to a decrease in performance.

The learning rate hyperparameter is crucial for the model's training and performance. It determines the step size of the optimization algorithm, directly affecting the model's weights and minimizing the loss function. The model may take longer to converge if the learning rate is too low. On the other hand, with a large learning rate, the optimization algorithm may overshoot the optimal value and fail to converge.

Learning rate schedule patience is a hyperparameter that determines how often the learning rate is reduced by a factor (known as learning rate decay). Smaller patience means the learning rate will be reduced more often, sometimes leading to longer training times but better performance. A larger learning rate schedule patience results in faster training and, sometimes, suboptimal performance.

Input image normalization is a popular preprocessing technique. It is usually performed by calculating each image channel's statistical mean and standard deviation on the training dataset and then normalizing images using these values. This technique ensures that all the neural network inputs are on the same scale, which helps with faster convergence.

The optimizer used for the training may affect the computational resources needed for the training and the final performance. AdamW and Stochastic Gradient Descent (SGD) are among the most popular optimizers. AdamW is an adaptation of Adam Optimizer. It introduces weight decay directly into the weight update, combining it with adaptive learning rates. Such an approach causes the model weights to stay small and the optimizer robust to noisy gradients. Typically, AdamW converges faster than the other optimizers. SGD is a classic optimization approach that updates the weights based solely on the gradients of the loss function and a learning rate. Usually, it requires more tuning in order to find the optimal learning rate and other hyperparameters and may take longer to converge. However, it requires less memory than AdamW. AdamW is an excellent choice for smaller datasets, whereas SGD is used more for larger datasets and models with fewer parameters.

Weight decay is a hyperparameter that affects the regularization ability of the neural network. It adds the magnitude of the parameters to the loss function, encouraging them to stay reasonably small. Usually, this technique is used to prevent overfitting, but in case of high weight decay values, the model may underfit and perform sub-optimally.

Using pretrained weights significantly affects the model performance and training. The weights are usually learned from a large dataset (such as ImageNet). Due to the usage of such weights, the model can already be capable of capturing important features and patterns needed for the task, as it already has some knowledge about the domain, contrary to the model with randomly initialized weights. Usage of the pretrained weights can speed up the training process and increase performance, especially when the training set size is small.

It is important to note that each hyperparameter depends on the other hyperparameters, model architecture, and the dataset. To find the best set of hyperparameters, it is best to perform a hyperparameter search.

# Chapter 4

# Experiments

## 4.1 Implementation and training details

The whole pipeline for our solution was implemented using PyTorch library (Paszke et al., 2019) and PyTorch Lightning framework (Falcon and The PyTorch Lightning team, 2019). The model architectures and pretrained weights were taken from the Torch Image Models (timm) Python library. Other libraries, such as NumPy (Harris et al., 2020), Scikit-learn (Pedregosa et al., 2011), and OpenCV (Bradski, 2000), were used for data processing and visualizations. As an annotation tool, we used the CVAT (Sekachev et al., 2020). We used Weights and Biases software (Biewald, 2020) for experiment tracking, hyperparameters search, hyperparameters importance estimation, and model comparison. Code snippets for the model's fine-tuning, as well as model pretrained weights, were taken from the TorchGeo (Stewart et al., 2022) documentation. All the training was executed on a Google Colab's NVIDIA Tesla T4 GPU.

## 4.2 Dataset

The dataset used in this thesis was collected with the help of a representative of an NGO that works on humanitarian projects aimed at supporting Ukraine. As of the time of writing, the dataset consists of satellite imagery data collected from Planet SkySat satellites (see Section 2.1.2). The spatial resolution of images is $0.5 \times 0.5m$. The imagery covers a geographical area in the Bakhmut region, a frontline city in Ukraine.

| Patch size, pixels | 'bombed' class count | 'not bombed' class count | Total dataset count |
|---|---|---|---|
| 64 | 3130 | 443 | 3573 |
| 128 | 1009 | 886 | 1895 |
| 256 | 576 | 29 | 605 |

TABLE 4.1: Class distribution in the train/validation dataset depending on the patch size.

As described in Section 3.1.2, the large satellite images can be split into patches of various sizes and are assigned labels depending on the craters' presence on them. For future experiments, we have used patches of various sizes, resulting in datasets of different sizes and class distributions. Table 4.1 shows how the patch size affects the dataset size and class distribution. For size of patch 64 × 64 pixels, finding the optimal threshold for minimum mask pixels number is difficult. The main reason behind this is that the masks are imperfect, and while inaccuracies are easier to filter out with larger patches, they are more significant with smaller ones. The dataset is highly imbalanced if the patches are 256 × 256. This is caused by the fact that the patch is so large it almost always contains at least one crater, so it is classified as bombed. We used 128 × 128 patches for our further experiments, as such size produced the most balanced dataset.

| Class label | Class count in train set | Class count in validation set | Class count in test set |
|---|---|---|---|
| ’bombed’ | 909 | 100 | 182 |
| ’not bombed’ | 798 | 88 | 433 |
| **total** | **1707** | **188** | **615** |

TABLE 4.2: Class distribution and size of subsets.

Table 4.2 describes the subsets’ size and class distribution in them. The train and validation sets were created by randomly splitting all the image patches. The training set accounts for 90% of the total elements, and the validation consists of the remaining 10%. All the data from the train/validation dataset comes from one large agricultural field in the Bakhmut region. The test set consists of images that also come from this region but from a different field. They were captured by the same satellite and have the same spatial resolution and other characteristics. Even though the images from the test set come from another field, they are still from the same region and are close to each other, so it is fair to assume that the soil type does not differ significantly from the one in the train/validation sets.

However, the class distribution in the test set significantly differs from that of the train/validation sets. The ’not bombed’ class is much more represented and dominates in terms of class count compared to the ’bombed’ class, while the class distribution in the train and validation sets is relatively balanced, with a slight bias towards the ’bombed’ class. Moreover, the test set shows noticeable distinctions from train/validation sets in terms of the inclusion of various terrain elements (such as trees and hills) and textures that are underrepresented or absent in the train/validation sets. Because of such differences, the test set exhibits a potential distribution shift, leading to variations in the performance and generalization of the model.

## 4.3 Hyperparameters selection

For the best model selection, a hyperparameters search has been performed. Table 4.3 summarizes the hyperparameters we searched. For a detailed explanation of how each parameter can affect the training process and performance, see Section 3.3.3.

## 4.4 Results

The main metric that we used for ranking the models by performance is the area under the receiver operating characteristic (AUROC). We also used TPR@FPR, which is described in detail in Section 3.2.

| Hyperparameter name | Values |
|---|---|
| Base size | 64, 128 |
| Batch size | 32, 64 |
| Minority class upsampling | True, False |
| Dropout | 0, 0.2 |
| Learning rate | 1e-04, 1e-05 |
| Learning rate schedule patience | 3, 5 |
| Normalization | True, False |
| Optimizer | AdamW, SGD |
| Weight decay | 0, 5e-5, 5e-4 |
| Pretrained weights | ImageNet, Sentinel-2 |

TABLE 4.3: Hyperparameters search details.

### 4.4.1 Hyperparameters importance analysis

For evaluation of how each hyperparameter affects the performance, we utilized the approach used by Weights and Biases (Biewald, 2020). It provides correlation and importance scores for each hyperparameter. As explained in the lecture by Howard (2018), the correlation is not enough for the correct selection of the most important hyperparameters, as it does not consider the fact that some of them have different scales. Therefore, the introduced importance metric is delivered from a random forest algorithm trained on the hyperparameters as inputs and the target metric as an output. Later in this thesis, we only use the importance score and the sign of the correlation score, as the absolute value of the correlation does not depict the true relations between the metric and the hyperparameter value, as explained above. As we calculated the scores with respect to the AUROC metric for the validation set, the higher the importance score, the more influence the hyperparameter has on the AUROC. If the correlation sign is positive (+), the higher hyperparameter values correspond to the higher AUROC (therefore, better performance), and vice versa.

#### 4.4.1.1 ResNet-50

| Hyperparameter name | Importance score | Correlation sign |
|---|---|---|
| Batch size | 0.220 | + |
| Weight decay | 0.216 | − |
| Normalization | 0.195 | + |
| Learning rate | 0.175 | + |
| Base size | 0.146 | + |

TABLE 4.4: Importance score and correlation sign with respect to AUROC metric calculated on the validation set for the ResNet-50 model.

Table 4.4 contains five hyperparameters with the highest importance scores and the corresponding correlation sign values for the ResNet-50 model. The batch size value correlates positively with the AUROC, which is surprising, as larger batches usually lead to poor generalization. Another unexpected positive correlation in this table is the learning rate because, typically, high learning rates may make an optimization process difficult. Weight decay is the second most important hyperparameter and it negatively correlates with the AUROC – as mentioned in Section 3.3.3, higher weight decay values may lead to underfitting. Normalization is also present in the list, and it, as expected, positively affects the performance. Similarly, the base size value also positively impacts the AUROC, which is also anticipated, as by resizing the image to a smaller size, we lose some information.

#### 4.4.1.2 ViT Small Patch

| Hyperparameter name | Importance score | Correlation sign |
|---|---|---|
| Dropout | 0.296 | − |
| Weight decay | 0.248 | + |
| Learning rate | 0.152 | + |
| Batch size | 0.096 | + |
| Normalization | 0.061 | − |

TABLE 4.5: Importance score and correlation sign with respect to AUROC metric calculated on the validation set for the ViT Small Patch model.

Table 4.5 contains the importance score and the correlation sign of the five most important hyperparameters for the ViT Small Patch model. As expected for the ViT models, high dropout values lead to a decrease in performance. Contrarily to the ResNet-50 model, here we observe a positive correlation between the weight decay value and the AUROC. Similarly to ResNet, the learning rate and batch size correlate positively with performance. However, the positive impact of the batch size in ViT models is more anticipated, as described in Section 3.3.3. Surprisingly, the normalization of the input images negatively affected the AUROC value in this case.

### 4.4.2 Models Comparison

To compare and demonstrate the hyperparameters' impact on performance, we select the best model for each architecture and the model that differs in the hyperparameters with the highest importance score. We select the best-performing model among the models that differ from the best in the most important hyperparameters.

From tables 4.6 and 4.7, we can see that diverging from the automatically selected in Section 4.4.1 values of the hyperparametrs does indeed change the model performance the way that it was expected during the importance score and correlation sign calculation.

### 4.4.3 Examples of model outputs

For the demonstration, we have selected the best-performing ViT Small Patch model (see Table 4.7), as it has shown the best metric scores. All its hyperparameters are given in Table 4.8.

Figure 4.1 illustrates examples of the model's output on the images that were present in the train and validation sets. As can be seen, the model performs well and correctly classifies both bombed and not bombed regions in Figure 4.1b. However, it

| Hyperparameter/metric name | Value in the best-performing model | Value in the model that differs in the most important hyperparameters |
|---|---|---|
| Batch size | 64 | 32 |
| Weight decay | 0 | 5e-4 |
| Normalization | True | False |
| Learning rate | 1e-4 | 1e-5 |
| Base size | 128 | 64 |
| **AUROC** | **0.998** | **0.55** |
| **TPR@FPR=0.2** | **0.99** | **0.24** |
| **TPR@FPR=0.1** | **0.99** | **0.05** |
| **TPR@FPR=0.05** | **0.98** | **0.01** |

TABLE 4.6: Comparison of the best-performing ResNet-50 model and the model that differs in the most important hyperparameters. Metrics are calculated on the validation set.

| Hyperparameter/metric name | Value in the best-performing model | Value in the model that differs in the most important hyperparameters |
|---|---|---|
| Dropout | 0 | 0.2 |
| Weight decay | 5e-4 | 5e-5 |
| Learning rate | 1e-4 | 1e-5 |
| Batch size | 64 | 32 |
| Normalization | False | True |
| **AUROC** | **0.999** | **0.87** |
| **TPR@FPR=0.2** | **0.99** | **0.73** |
| **TPR@FPR=0.1** | **0.99** | **0.71** |
| **TPR@FPR=0.05** | **0.99** | **0.69** |

TABLE 4.7: Comparison of the best-performing ViT Small Patch model and the model that differs in the most important hyperparameters. Metrics are calculated on the validation set.

sometimes mislabels the complicated and textured terrain elements, such as regions with trees, present in the top-left corner of the Figure 4.1a.

To test out the generalizability of the approach, we performed testing on the images from the test set.

Figure 4.2 shows examples of the images that were separated into patches and classified by our best-performing neural network. The input images underwent the min-max stretch in the same way the images from the training and validation sets did. In both examples, the network typically incorrectly labels more textured areas as bombed ones. For example, the right-hand side of the Figure 4.2b is occupied not by field but by some terrain element that was not represented in the training set; most of this region is assigned the 'bombed' label as it is more textured. Meanwhile, most patches from the part with the agricultural field are labeled correctly. Similarly, in Figure 4.2a, most patches that were misclassified as bombed contain texture (white lines or spots).

Table 4.9 contains the metrics measured on the test images from Figure 4.2. It can

| Hyperparameter name | Values |
|---|---|
| Base size | – |
| Batch size | 64 |
| Minority class upsampling | False |
| Dropout | 0 |
| Learning rate | 1e-04 |
| Learning rate schedule patience | 5 |
| Normalization | False |
| Optimizer | AdamW |
| Weight decay | 5e-4 |
| Pretrained weights | ImageNet |
| **Architecture** | **ViT Small Patch** |

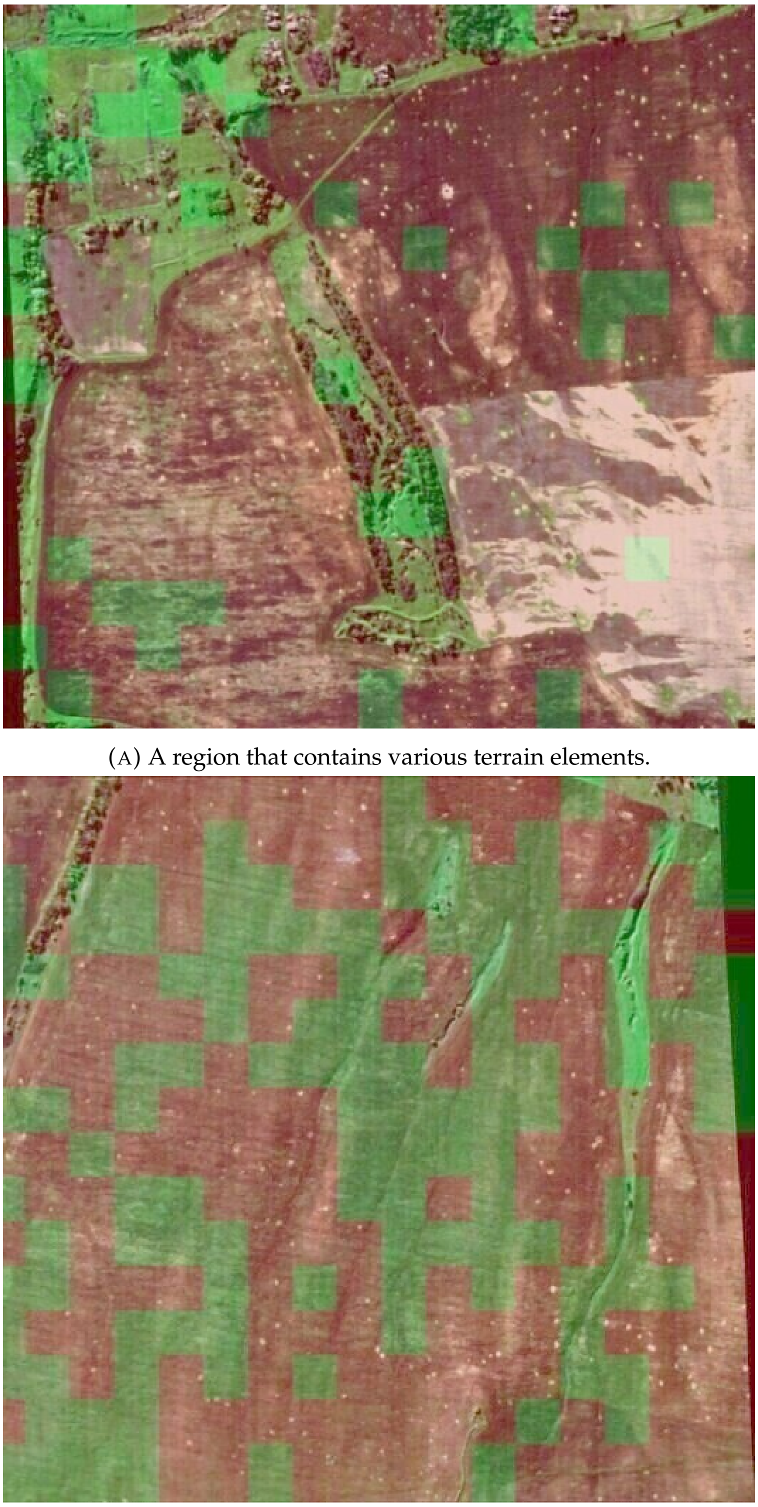
TABLE 4.8: Hyperparameters and the architecture of the best-performing model.

be observed that for the image in Figure 4.2a, the metrics indicate great model performance as opposed to the image in Figure 4.2b. This supports our claim that textured elements that are not parts of the agricultural field are associated with a decrement in the model performance, as the former figure does not contain this many regions with textures that were not represented in the training dataset compared to the latter one.

| Metric | Value for image on Figure 4.2a | Value for image on Figure 4.2b | Value for both images |
|---|---|---|---|
| AUROC | 0.88 | 0.56 | 0.61 |
| TPR@FPR=0.2 | 0.70 | 0.20 | 0.24 |
| TPR@FPR=0.1 | 0.60 | 0.09 | 0.11 |
| TPR@FPR=0.05 | 0.40 | 0.03 | 0.05 |

TABLE 4.9: Values of the metrics measured on the test dataset.

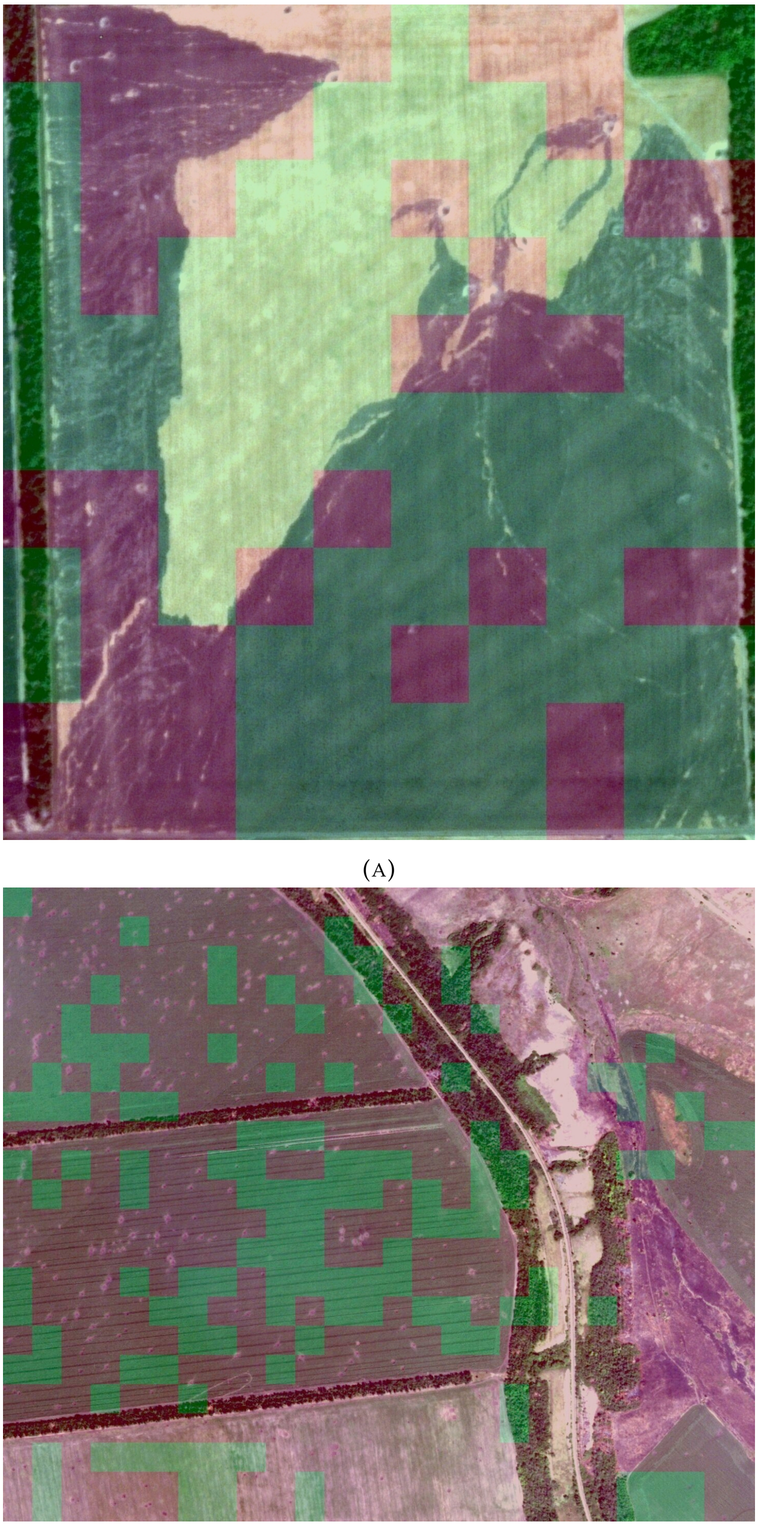
These results demonstrate, first of all, that the textures and terrain elements that are underrepresented in the training dataset are the main causes of the incorrect outputs of the model. Despite some mistakes in these cases, the network shows a great ability to generalize to new locations. However, the locations from the training set and test images are within close proximity. Due to the lack of data, we can not make any conclusions about the network's ability to generalize to agricultural fields from different regions with different soil types, plant life, etc. Similarly, judging its generalizability across images with different weather conditions is difficult. To address the aforementioned issues and increase the model's generalizability and performance, one potential avenue is to collect a larger and more diverse dataset. Additionally, finding and utilizing the data about exact agricultural fields' locations and bounds to exclude the terrain elements that are not parts of the agricultural fields from the analyzed images is the way to tackle the problem of underperformance in such areas.



(A) A region that contains various terrain elements.

(B) A region that does not contain highly textured terrain elements.

FIGURE 4.1: An example of the model output on the data from the train and validation datasets. Patches that were assigned the ’not bombed’ label are colored in green, and the ones that were assigned the ’bombed’ label are in red.



(A)

(B)

FIGURE 4.2: An example of the model output on the data from the test set. Patches that were assigned the 'not bombed' label are colored in green, and the ones that were assigned the 'bombed' label are in red.

# Chapter 5

# Conclusions and Future work

## 5.1 Conclusions

In this thesis, we explored the application of classification methods for detecting and assessing shelling-inflicted damage on agricultural fields. As classification is the most popular approach to damage assessment of buildings and crater detection, we explored how well such methods generalize to the domain of Ukrainian agricultural fields and war-inflicted damage.

We showed that the classifier trained on our dataset shows great metrics on the train and validation sets and can generalize to the new agricultural fields in the same region. However, it tends to misclassify images that contain textures and terrain elements that are either complex or underrepresented in the training set (e.g., snow and sand). This fact implies that the underperformance on the new data may be fixed by expanding the variety in the training data in both spatial and temporal domains.

## 5.2 Future work

The first step in our planned future work is dataset expansion. We consider collecting data from different locations and dates to enhance our dataset's diversity, including images from Ukraine and other countries that have suffered from shelling. We believe that the growth in dataset size will have a significant positive influence on the models' performance.

Next, the future work includes two main directions. First, we would like to experiment more with classification approaches, increasing their performance and generalizability. Another direction includes exploring the applicability of the segmentation approaches to our problem.

# Bibliography

Aimaiti, Yusupujiang et al. (2022). "War Related Building Damage Assessment in Kyiv, Ukraine, Using Sentinel-1 Radar and Sentinel-2 Optical Images". In: *Remote Sensing* 14.24, p. 6239. ISSN: 2072-4292. DOI: 10.3390/rs14246239. URL: http://dx.doi.org/10.3390/rs14246239.

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. Software available from wandb.com. URL: https://www.wandb.com/.

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Brenner, Simon, Sebastian Zambanini, and Robert Sablatnig (2018). "Detection of bomb craters in WWII aerial images". In: *Proceedings of the OAGM Workshop*. Vol. 2018. Verlag der Technischen Universität Graz Graz, Austria, pp. 94–97.

Brunner, Dominik, Guido Lemoine, and Lorenzo Bruzzone (2010). "Earthquake Damage Assessment of Buildings Using VHR Optical and SAR Imagery". In: *IEEE Transactions on Geoscience and Remote Sensing* 48.5, pp. 2403–2420. DOI: 10.1109/TGRS.2009.2038274.

Byrnes, James (2008). *Unexploded ordnance detection and mitigation*. Springer Science & Business Media.

Clermont, D et al. (2019). "Supervised detection of bomb craters in historical aerial images using convolutional neural networks". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences; 42-2/W16* 42.2/W16, pp. 67–74.

Department of the Army (1996). *FM 6-50 Tactics, Techniques, and Procedures for The Field Artillery Cannon Battery*. Washington, D.C.: Headquarters, Department of the Army. URL: https://www.globalsecurity.org/military/library/policy/army/fm/6-50/#VERSION.

Durnov (2020). *XView2 First Place Solution*. https://github.com/DIUx-xView/xView2_first_place.

Falcon, William and The PyTorch Lightning team (2019). *PyTorch Lightning*. Version 1.4. DOI: 10.5281/zenodo.3828935. URL: https://github.com/Lightning-AI/lightning.

Furlanello, Cesare et al. (2003). "Mapping the risk of unexploded bombs from World War Two". In: *TC-IRST: Trento, Italy*.

Ge, Pinglan, Hideomi Gokon, and Kimiro Meguro (2020). "A review on synthetic aperture radar-based building damage assessment in disasters". In: *Remote Sensing of Environment* 240, p. 111693. DOI: 10.1016/j.rse.2020.111693. URL: https://doi.org/10.1016/j.rse.2020.111693.

Geiger, Marco, Dominik Martin, and Niklas Kühl (2022). *Deep Domain Adaptation for Detecting Bomb Craters in Aerial Images*. arXiv: 2209.11299 [cs.CV].

Gupta, Ritwik et al. (2019). *xBD: A Dataset for Assessing Building Damage from Satellite Imagery*. arXiv: 1911.09296 [cs.CV].

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs.CV].

Ho, Tin Kam (1995). "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282.

Howard, Jeremy (2018). *Feature importance, tree interprener*. URL: https://course18.fast.ai/lessonsml1/lesson4.html.

Ismail, Ali and Mariette Awad (2022). "Towards Cross-Disaster Building Damage Detection with Graph Convolutional Networks". In: *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. DOI: 10.1109/igarss46834.2022.9883832. URL: https://doi.org/10.1109\%2Figarss46834.2022.9883832.

Jensen, LAURA, MARTIN Drauschke, and WOLFGANG Förstner (2010). "Automatic detection of bomb craters in digitized aerial photographs of WWII". In: *DGPF conference volume 19* 2010, pp. 228–235.

Juhász, Attila, Hajnalka Neuberger, et al. (2018). "Automatic identification of bomb craters and their potential location". In.

Jung, Jungkyo and Sang-Ho Yun (2020). "Evaluation of Coherent and Incoherent Landslide Detection Methods Based on Synthetic Aperture Radar for Rapid Response: A Case Study for the 2018 Hokkaido Landslides". In: *Remote Sensing* 12.2. ISSN: 2072-4292. DOI: 10.3390/rs12020265. URL: https://www.mdpi.com/2072-4292/12/2/265.

Kang, Zhizhong et al. (2018). "Coarse-to-fine extraction of small-scale lunar impact craters from the CCD images of the Chang'E lunar orbiters". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.1, pp. 181–193.

Kyiv School of Economics (2022). *Agricultural War Damages Review Ukraine*. https://kse.ua/wp-content/uploads/2022/06/Damages_report_issue1-1.pdf.

Lacroix, Vinciane and Sabine Vanhuysse (2015). "Crater Detection using CGC". In: *Proceedings of the International Conference on Pattern Recognition Applications and Methods-Volume 1*, pp. 320–327.

Lee, Seung Hoon, Seunghyun Lee, and Byung Cheol Song (2021). *Vision Transformer for Small-Size Datasets*. arXiv: 2112.13492 [cs.CV].

Merler, Stefano, Cesare Furlanello, and Giuseppe Jurman (2005). "Machine learning on historic air photographs for mapping risk of unexploded bombs". In: *Image Analysis and Processing–ICIAP 2005: 13th International Conference, Cagliari, Italy, September 6-8, 2005. Proceedings 13*. Springer, pp. 735–742.

Mueller, Hannes et al. (2021). "Monitoring war destruction from space using machine learning". In: *Proceedings of the National Academy of Sciences* 118.23. DOI: 10.1073/pnas.2025400118. URL: https://doi.org/10.1073\%2Fpnas.2025400118.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.

Pedrosa, Miriam Maria et al. (2017). "Improved automatic impact crater detection on Mars based on morphological image processing and template matching". In: *Geomatics, Natural Hazards and Risk* 8.2, pp. 1306–1319.

QGIS Development Team (2009). *QGIS Geographic Information System*. Open Source Geospatial Foundation. URL: http://qgis.org.

Redmon, Joseph et al. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. arXiv: 1506.02640 [cs.CV].

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs.CV].

Russakovsky, Olga et al. (2015). *ImageNet Large Scale Visual Recognition Challenge*. arXiv: 1409.0575 [cs.CV].

Sabour, SM Tavakkoli, Jürgen Agarius, and Javad Sadidi (2014). "Calculation of Per Parcel Probability for Dud Bombs in Germany". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.2, p. 261.

Salamuniccar, Goran and Sven Loncaric (2010). "Method for crater detection from martian digital topography data using gradient value/orientation, morphometry, vote analysis, slip tuning, and calibration". In: *IEEE transactions on Geoscience and Remote Sensing* 48.5, pp. 2317–2329.

Sandhini Putri, Ade Febri, Wirastuti Widyatmanti, and Deha Agus Umarhadi (2022). "Sentinel-1 and Sentinel-2 data fusion to distinguish building damage level of the 2018 Lombok Earthquake". In: *Remote Sensing Applications: Society and Environment* 26, p. 100724. ISSN: 2352-9385. DOI: https://doi.org/10.1016/j.rsase.2022.100724. URL: https://www.sciencedirect.com/science/article/pii/S2352938522000325.

Sekachev, Boris et al. (Aug. 2020). *opencv/cvat: v1.1.0*. Version v1.1.0. DOI: 10.5281/zenodo.4009388. URL: https://doi.org/10.5281/zenodo.4009388.

Silburt, Ari et al. (2019). "Lunar crater identification via deep learning". In: *Icarus* 317, pp. 27–38.

Solovey, Vadym (2022). *Paramtric soil classification*. URL: https://agrotimes.ua/article/parametrychna-klasyfikacziya-gruntiv/.

Steiner, Andreas et al. (2021). "How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers". In: eprint: arXiv:2106.10270.

Stewart, Adam J. et al. (2022). "TorchGeo: Deep Learning with Geospatial Data". In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '22. Seattle, Washington: Association for Computing Machinery. ISBN: 9781450395298. DOI: 10.1145/3557915.3560953. URL: https://doi.org/10.1145/3557915.3560953.

Szegedy, Christian et al. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1.

Tong, Xiaohua et al. (2012). "Building-damage detection using pre- and post-seismic high-resolution satellite stereo imagery: A case study of the May 2008 Wenchuan earthquake". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 68, pp. 13–27. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2011.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0924271611001584.

Valentijn, Tinka et al. (2020). "Multi-Hazard and Spatial Transferability of a CNN for Automated Building Damage Assessment". In: *Remote Sensing* 12.17, p. 2839. DOI: 10.3390/rs12172839. URL: https://doi.org/10.3390/rs12172839.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Wang, Hao, Jie Jiang, and Guangjun Zhang (2018). "CraterIDNet: An end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images". In: *Remote sensing* 10.7, p. 1067.

Washington State Department of Health (2022). *COVID-19 Hospitalizations and Deaths by Vaccination Status in Washington State*. https://doh.wa.gov/sites/default/files/2022-02/421-010-CasesInNotFullyVaccinated.pdf.

Yusuf, Yalkun, Masashi Matsuoka, and Fumio Yamazaki (2001). "Damage assessment after 2001 Gujarat earthquake using Landsat-7 satellite images". In: *Journal of the Indian Society of Remote Sensing* 29.1-2, pp. 17–22. DOI: 10.1007/bf02989909. URL: https://doi.org/10.1007/bf02989909.

Zhang, Ce et al. (2018). "An object-based convolutional neural network (OCNN) for urban land use classification". In: *Remote Sensing of Environment* 216, pp. 57–70. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2018.06.034. URL: https://www.sciencedirect.com/science/article/pii/S0034425718303122.

— (2019). "Joint Deep Learning for land cover and land use classification". In: *Remote Sensing of Environment* 221, pp. 173–187. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2018.11.014. URL: https://www.sciencedirect.com/science/article/pii/S0034425718305236.

Zheng, Zhuo et al. (2021). "Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters". In: *Remote Sensing of Environment* 265, p. 112636. DOI: 10.1016/j.rse.2021.112636.