The Effectiveness of Machine Learning in Construction and Demolition Waste Recognition From Satellite Images in Astana

Kazbek Iliyas, Bachelor's in Civil & Environmental Engineering

Submitted in fulfillment of the requirements

for the degree of Master of Science

in Civil & Environmental Engineering

School of Engineering and Digital Sciences Department of Civil & Environmental Engineering Nazarbayev University

53 Kabanbay Batyr Avenue, Astana, Kazakhstan, 010000

Supervisor: Mert Guney

Co-supervisor: Ferhat Karaca

06 August 2024

Declaration

I hereby declare that this manuscript, entitled “The Effectiveness of Machine Learning in Construction and Demolition Waste Recognition from Satellite Images in Astana”, is the result of my work except for quotations and citations which have been duly acknowledged. I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Kazbek Iliyas
Date: 06.08.2024

Abstract

In rapidly urbanizing areas like Astana city, identifying and managing construction and demolition waste (CDW) is becoming more and more difficult. The growing volume and complexity of CDW are excessive for traditional waste management techniques to handle, which results in operational and environmental inefficiencies. To solve the issue, this thesis assesses how well machine learning techniques work in identifying CDW in these cities from satellite images. Accurate detection can greatly help with the management of waste, which is essential for maintaining the health of urban environments. The model was trained with state-of-the-art object recognition and segmentation techniques, resulting in a mean intersection over union (IoU) of 0.380. Although this performance is below the benchmark norms (0.457 to 0.56), as reported in similar research, it still shows great potential. 200 photos were gathered and annotated as part of the process, which was then used to train and validate the model. Key findings include the effect of image quality on detection accuracy and notable differences in performance across various waste types. The model showed an accuracy of 0.80 for both training and validation; however, recall (2.12%) and precision (3.22%) still need to be improved. Some misclassifications were observed during visual inspection since CDW and non-waste materials had similar appearances. To improve detection accuracy, we suggest that future studies look into more sophisticated data augmentation methods and effective model architectures. The use of Google Earth imagery and a simplified two-class classification scheme are two of the study's limitations. These drawbacks imply that to fully reflect the complexity of CDW detection, future research should take multi-class classification into account and include a wider range of data sources. Our findings contribute to the field of environmental monitoring by demonstrating both the potential and challenges of applying machine learning to urban waste management.

Keywords: Construction and demolition waste (CDW), Machine learning, Satellite imagery, Object recognition, Semantic segmentation, Waste Management, Astana, Kazakhstan.

Acknowledgments

I would like to show my acknowledgment to my supervisor and co-supervisor, Prof. Mert Guney and Prof. Mert Karaca, whose knowledge of environmental engineering inspired me to take the thesis topic related to the issue of waste management. Moreover, they guided me through the whole process and thanks to them the year of research gave me an immeasurable experience. I want to thank them and external and internal examiners for having faith in me and this thesis project, despite it heavily connecting to computer science and machine learning domain. Thanks to them, I learned and implemented the given knowledge to finish my thesis.

I would like to recognize the effort of the research team, created by Prof. Mert Guney, who also contributed to the paper. I especially want to highlight Aiganym Kumisbek, my senior researcher, whose expertise in GIS software and general research experience were very helpful in solving different problems that occurred throughout the research process.

I also would like to acknowledge the efforts of my close friends: Damir Assylbek and Maksat Kengeskanov. They are both experts in machine learning and agreed to guide me through the basics of model training. If it wasn't for them, I wouldn't be able to finish the code and successfully run it for my thesis work.

I am also grateful for my friends and family, who supported me throughout these past two years. Mental support from them was very necessary. I would also like to thank the university for the opportunity it gave me to successfully conduct and finish my research. I can say that I had an important experience both as a Master's student and a researcher.

Table of Contents

List of Abbreviations & Symbols

CDW	Construction and Demolition Waste
IoU	Intersection Over Union
CNN	Convolutional Neural Network
AI	Artificial Intelligence
CV	Computer Vision
DNN	Deep neural networks
MSW	Municipal Solid Waste
GIS	Geographic Information System
JPG	A commonly used method of lossy compression for digital images
TIF	A computer file used to store raster graphics and image information
ArcGIS	software that allows users to create, edit, visualize, analyze, and publish geospatial information
GEP	Google Earth Pro
GB	A unit of data storage capacity
ASPP	Atrous Spatial Pyramid Pooling
GPU	Graphics Processing Unit
TP, TN, FP, FN	True Positive, True Negative, False Positive, False Negative (variables in model training)

List of Figures and Tables

Chapter 1 - Introduction

1.1 Overview

This paper represents the master's thesis. Here are the main ideas for the research, the future work plan, and the output expectations of the thesis. It covers the introduction to machine learning, the concept of convolutional neural networks, and ways to use it in construction waste management. The main purpose of the thesis is to evaluate the effectiveness of the identification of construction and demolition waste (CDW) in Astana through satellite images. This will be achieved by running the computer vision model and comparing the effectiveness with existing methods. The thesis focuses on state-of-the-art methods, object recognition and segmentation processes, and machine learning, for the identification of CDW.

For urban ecosystems to remain functional and healthy, sustainable waste management is essential. The management of waste, especially construction and demolition waste (CDW), becomes more important as cities grow. In cities in Central Asia, like Astana, the situation is particularly serious due to growing urbanization [1]. The general plan for the city (for 2035) portrays the expansion of the construction to the outskirts of the city to accommodate an estimated population of 2.3 million people in 2035 [2]. CDW management is becoming an important issue due to the country's goal to achieve zero emissions within the next 30 to 40 years [1]. The United Nations analysts predict that the number of people living in cities worldwide will continue to rise, with major development anticipated in metropolitan centers in Central Asia [3]. Rapid urbanization causes sprawling housing developments and urban sprawl, both of which raise the production of solid waste.

Specifically, the garbage generated in urban areas is largely contributed to by the construction and demolition industries. Effective CDW management is crucial for the economic and social well-being of urban populations as well as the sustainability of the environment. The unique qualities of CDW, including its quantity, diversity, and possibility for recycling, put it at the center of environmentally friendly waste management strategies.

Major Kazakhstani cities, such as Astana, are rapidly growing and becoming more urbanized. Strong management plans are needed to control the resulting increase in CDW because of this urban growth. The use of the latest technologies, such as computer vision and machine learning, presents innovative methods to enhance waste management procedures. Modern computational models and satellite

imagery can be used to improve CDW identification and management, which will help cities achieve their sustainability objectives.

1.2 Objectives and Questions

The **research objectives** are

- to compare and select a model that can distinguish construction and demolition waste from other objects with the highest accuracy and use it on construction sites in Astana
- annotate and train the selected model
- to identify whether local differences in construction sites will affect the effectiveness of the model
- to give the knowledge of creating a machine learning model through state-of-the-art methods through the comparison with existing methods

The main **research question** can be formulated as to what extent a trained image recognition/segmentation (or other general term) model can be effective/accurate in identifying and separating CDW from other objects (including buildings, vegetation, and roads) on satellite imagery.

Chapter 2 - Literature Review

2.1 Circular Economy and Construction and Demolition Waste

By emphasizing the concepts of reduce, reuse, and recycle, a circular economy attempts to minimize and eventually eliminate waste. It encourages cutting back on the consumption of raw materials, reusing materials to make new products, and recycling old ones [4].

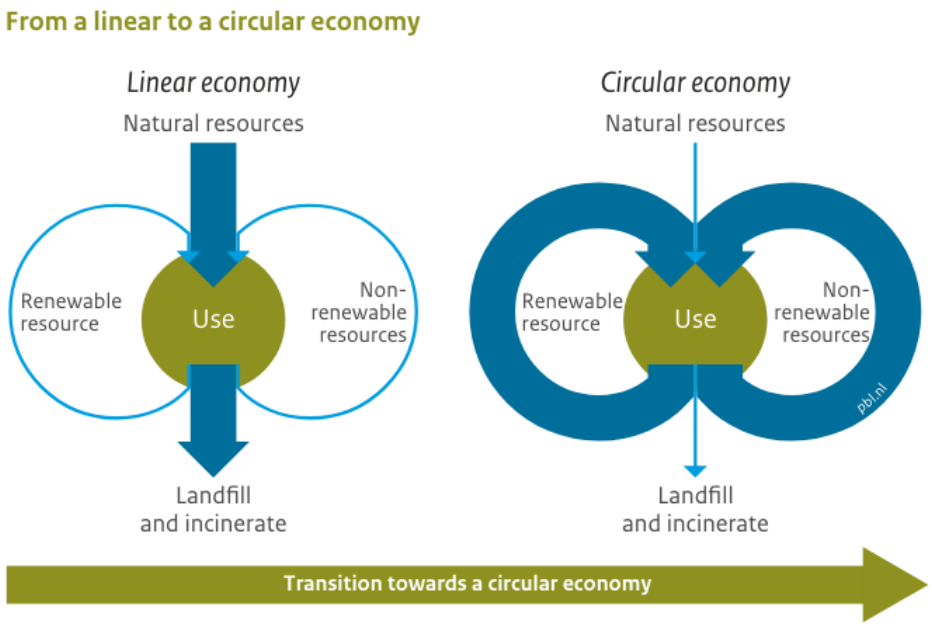

Figure 2.1. The shift to the circular economy, according to the PBL Research Institute (Netherlands) [5]

A linear economy, in contrast, adopts a "take, make, and dispose" strategy. Even though the linear model is now used by most modern economies, there is an expanding global movement to switch to the more effective circular model. In Figure 2.1, the key difference is shown to demonstrate the strategy to shift to the circular economy. Minimizing the number of resources that go directly to landfills and incineration is the key point in the strategy. Through the Horizon 2020 initiative, the European Union is pressing its member states to reach a minimum recycling rate of 70% for inert and non-hazardous building and construction and demolition waste (CDW) by 2020. This is in the interests of sustainable development and the circular economy [6]. Similarly, in the United States, the Environmental Protection Agency (EPA) has set its first-ever National Recycling Goal, aiming to increase the national recycling rate to 50% by 2030. The EPA's goal underscores the importance of shared objectives in achieving substantial progress in recycling practices [7].

The attention to CDW is also growing in Kazakhstan, which is mainly focused on the two biggest cities – Almaty and Astana. Currently, there are 35 thousand tonnes per year received in Astana landfill [8]. Residents and entrepreneurs in Astana, Kazakhstan have been fined 30 million tenge for improper waste disposal. Despite the high fines, the number of violations remains high, with the outskirts of the city and riverbanks cluttered with waste and approximately 400 illegal dumps identified. Unscrupulous entrepreneurs try to save money by disposing of garbage, mainly at night.

The capital has only one landfill that charges around 4000 tenge per ton of solid household waste, but many find it cheaper to violate environmental regulations. [9]

Bakhyt Aubakirova, PR manager of the waste disposal organization mentions that specialized separate organizations should address the construction waste. Makeshift landfills across the country are identified through satellite monitoring. This year, almost 5000 illegal points were discovered, with a significant portion resulting from the accumulation of construction waste. [9]

The issue of construction waste is currently a problem, according to the Ministry of Ecology, Geology, and Natural Resources of the Republic of Kazakhstan. In this regard, the new version of the environmental code includes requirements that construction waste not be accepted at solid waste landfills. It rather be handled by organizations that will sort, process, and reuse the waste. Only minimal volumes that cannot be processed or utilized remain at the landfill for burial. In March 2023 a special landfill for storage and processing of construction materials was launched in the capital [10]. A crushing complex will be installed on its territory to process waste such as bricks, asphalt, and concrete structures. [9] The recovery rate of CDW is indeed low (at 2.96%) compared to Europe's 90% recovery rate [8]. With the country's plan for zero emissions and the overall state of waste management, the demand for new and applicable solutions is apparent [1].

2.2 Development of Computer Vision Use in Waste Management

Artificial intelligence is a critical piece of technology that can help the circular economy transition in a revolutionary way. The subject of science and technology known as artificial intelligence (AI) is concerned with creating intelligent machines that can think, learn, acquire knowledge, communicate, perceive, plan, and manipulate objects [11]. The implementation of AI comes with several advantages. AI can, for instance, use sophisticated algorithms to learn from enormous volumes of data and apply the known information to commercial and practical applications [12]. AI provides significant potential for increasing productivity by quickly and effectively evaluating large datasets. Additionally, once trained, AI systems and technologies can quickly forecast and generalize about complicated and nonlinear problems [13]. Therefore, within the context of a circular economy, AI can support product design and optimization when effectively incorporated [14].

Deep learning has been used successfully throughout the last years, finding implementations in land use and its classification [15, 16], object detection [17], and semantic segmentation [18]. Semantic segmentation plays a significant role in image analysis [19], medical imaging [20, 21], and remote

sensing [22-24]. Studies have been ongoing to research the application of computer vision (CV) in waste management. At the beginning of the 21st century, the application of robots that segregate desired items from the stream of waste on conveyor belts was developed. The main recognition tool for such robots was optical or visual sensors, such as cameras. Initially, visual recognition relied on the hand-engineered features fed to machine learning models; therefore, they weren't as effective at dealing with the complex variety of waste that exists.

The situation has significantly improved since 2012 because of improvements in computer power and the accessibility of massive datasets. According to Krizhevsky, this development has made it possible for end-to-end deep-learning techniques to be successful [25]. Deep neural networks (DNN) are used for dataset analysis and modeling in deep learning, a subfield of machine learning. A particular kind of DNN called a convolutional neural network (CNN) is frequently used in image categorization. After training, CNN can recognize the traits of individual grains in an aggregate image and determine the make-up of the total sample shown in the image [26].

Yang and Thung used AlexNet, a deep convolutional neural network (CNN), in one application of these methods to categorize several types of municipal solid waste (MSW), including paper, glass, and cardboard [27]. The dataset was built on TrashNet source and consists of photo images of waste taken on a light background. They built further experiments on top of their TrashNet dataset, which combined resulted in greatly improving classification accuracy, which initially was under 70% [28]. However, with the help of various techniques, such as Mao et al. (2021's usage of a genetic algorithm for hyperparameter optimization), accuracy rates increased to above 90% [29]. On the same TrashNet dataset, Zhang included a residual network with a self-monitoring module for the classification of recyclable garbage [30]. Researchers also investigated how to locate and identify different types of garbage in photos. Based on the TrashNet dataset, Awe trained a Fast R-CNN model and obtained a mean average precision of 0.683 [31].

The locations of current CDW regions have been predicted by certain research using Geographic Information System (GIS) techniques [32]. There is, however, a lack of studies on CDW detection in satellite photography. Currently, human field surveys and remote sensing monitoring are the main approaches used to identify CDW. Due to the extensive distribution and various CDW storage facilities [33], performing manual field surveys requires a significant amount of material and human resources, which reduces job efficiency.

To visually evaluate the spatial distribution of solid waste and rubbish dumps, researchers have used remote sensing data sources [34, 35]. The spectral and textural features of CDW are somewhat complex [36] because it is frequently mixed with household waste and its surfaces may be hidden by plants or green coverings. The difficulty of visually interpreting CDW increases due to this complexity [37]. Consequently, a classification method that relies solely on mathematical statistics and visual interpretation falls short of meeting the efficiency and accuracy requirements for classifying CDW in remote sensing images [38].

Chapter 3 - Methodology

3.1 Data Collection

The most important and extensive part of the research is the data collection. Special datasets are used to collect high-resolution satellite images from different construction sites. Such datasets are found through GitHub, a service for coding individuals where information and necessary tools are shared. It is important to note that the imagery should be in a file format readable by Python, which is usually a TIF. Other images can be acquired through commercial satellite providers (e.g., Airbus, and Maxar Technologies) [39, 40]. Those mentioned companies specialize in producing high-quality images, which are also used in Google Earth imagery. Airbus, for instance, had Pleiades and Pleiades Neo satellites that have a quality of 0.5 m/pixel and 0.3 m/pixel respectively.

Table 3.1 Cost features for satellite sensors. [41]

	Minimum order area, ha	Price per unit, $ ha^{-1}	Minimum area price, $	Computational demand, KB ha^{-1}
Deimos-2	10000	0.060	700	50
Dove	10000	0.012	218	8
GeoEye-1	10000	0.275	2850	100
Kompsat-2	2500	0.055	237.5	20
Kompsat-3	2500	0.110	375	30
Kompsat-3A	2500	0.160	500	100

Landsat-7/8	3700000 (one scene)	0	100	0.5
Pleiades-1A/1B	10000	0.213	2225	100
Rapideye	10000	0.012	218	4
Sentinel-2	1200000 (one scene)	0	100	0.63
Spot-6/7	10000	0.045	550	8
WorldView-2/3/4	10000	0.275	2850	130

As can be seen in Table 3.1, the cost of commercial satellites has a lower limit in the form of a minimum order area. The minimum area price is found by following the formula:

$$Minimum\ area\ price = Minimum\ order\ area * Price\ per\ unit + Data\ processing\ cost \quad (1)$$

Some of the popular and very high-quality satellite types like Pleiades 1A/1B are very expensive, while some have free options like Landsat-7/8 and Sentinel-2.

Another option is to obtain Maxar, Airbus, Copernicus, or other providers' satellite imagery using a Google Earth Pro, as the company uses images from this company to update the layout of Earth Map. The second option was chosen as an affordable and commonly used option [42].

3.1.1 Sectoring in ArcGIS Pro

The ArcGIS Pro software was obtained for sectoring work. The urban area to be investigated was reduced from the official city boundaries to the developed area (lands with medium to high-density development) according to the 2023 satellite imagery. The city was divided into 8 sectors, as obtaining data for the whole city might become very heavy. The approach was to work on one sector at a time and obtain high-quality images for it step by step.

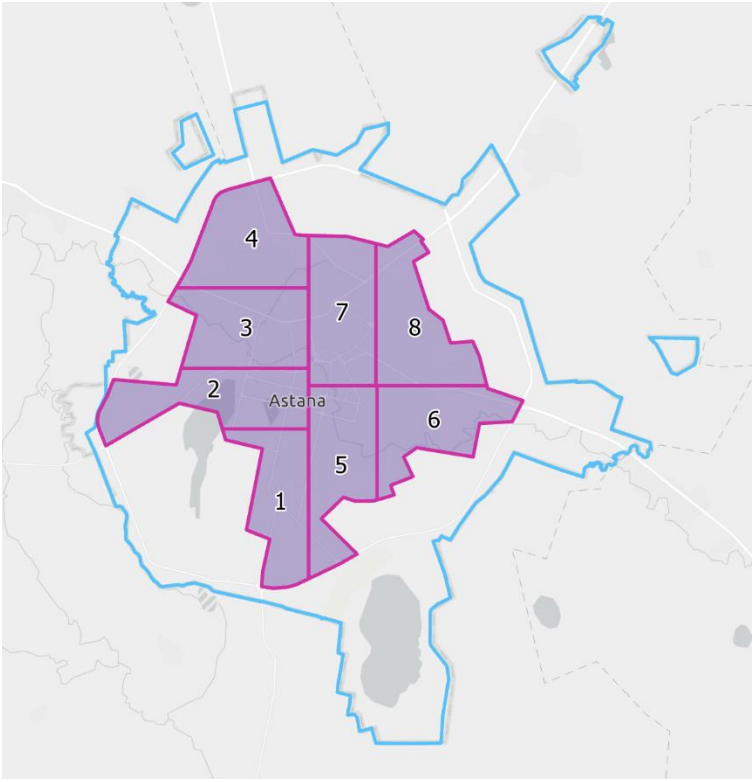

Figure 3.1.1 Administrative boundaries of Astana city (blue) and selected developed area sectors (pink) for systematic aerial image extraction using Google Earth Pro software.

The boundaries of "developed areas" of the city were first reduced to fit the major bypass ring highway around the city (marked as white lines in Figure 3.1.1). After that, the underdeveloped regions were manually removed because it was unnecessary to examine photos that contained vacant space. Using ArcGIS Pro's Subdivide Polygon function, the remaining area was divided into equal sectors in the area, setting the number to 8. The size of imagery for one sector is about 2 GB. The division of the city into 8 sectors was done for technical limitations purposes, to extract a total of 16 GB of cache information. In the case of a larger flash drive or hard drive, the process can be simplified. This was found after caching aerial imagery from Google Earth Pro (with the last updated ver.7.3.6.9796 (64-bit) of the year 2024), which justified the division and chosen number. The resultant polygon layer was exported as an ESRI Shapefile, which is a collection of files with extensions like SHP, PRJ, etc., and was saved in a folder separate from the ArcGIS Pro project's home folder. There was a concern in the training model that the number of images collected might need to be larger [43], so a substantial number of images must be gathered to ensure an effective training phase.

3.1.2 Pathing in Google Earth Pro (GEP)

The next step is to use the GEP software's Touring function to cache Google Earth imagery. This enables the user to open and view up to 2048 MB of imagery without requiring an internet connection. The software will rewrite the oldest cached tiles if the amount of imagery has grown beyond the limit. The maximum size of the cache can be checked and adjusted using Tools -> Options (see Figure 3.1.2). This tab will also be used to clean the cache every time before “recording” each sector.

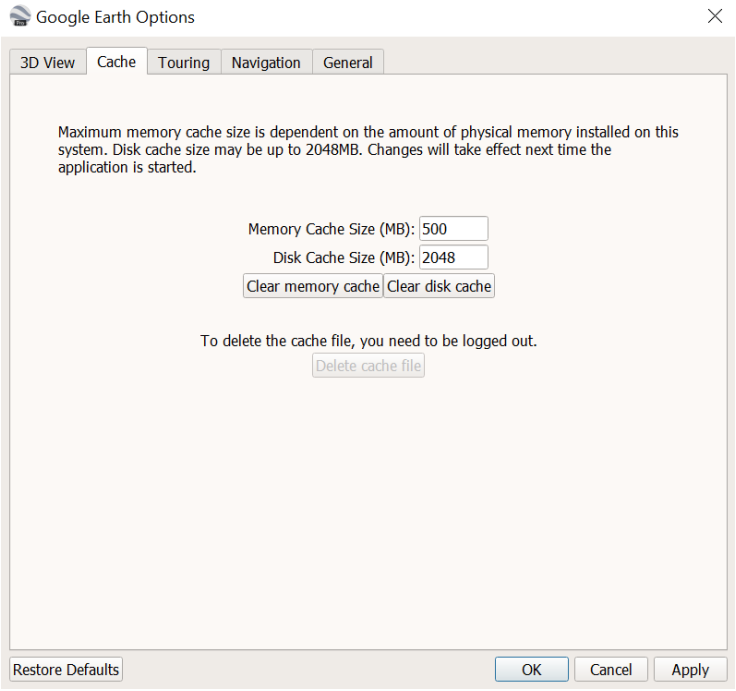

Figure 3.1.2 Cache management options.

Using File -> Open, the sectors shapefile was first loaded into GEP. Subsequently, the polygon-style properties were adjusted to achieve transparency and visible boundaries.

Second, by selecting View -> Grid, grid view was made available, and the Add Path button was used to create the paths. It is recommended to draw a path by first following the sector polygon's perimeter and then filling in the sector with a snake-like pattern (spacing between horizontal lines should be around two-thirds of the smallest vertical grid divisions). Editing the pattern is possible at any moment.

Third, by right-clicking the path in the Places panel and choosing Save Places created paths should be stored as KMZ files on the PC for later use. The sector borders should also be saved individually as KMZ files (open the SHP layer in the Places panel), ideally in the same folder if they are to be used

for future purposes as well. The distance between lines can be shortened if, during the quality check, a few middle tiles are not loaded at maximum quality (see Figure 3.1.3 for details).

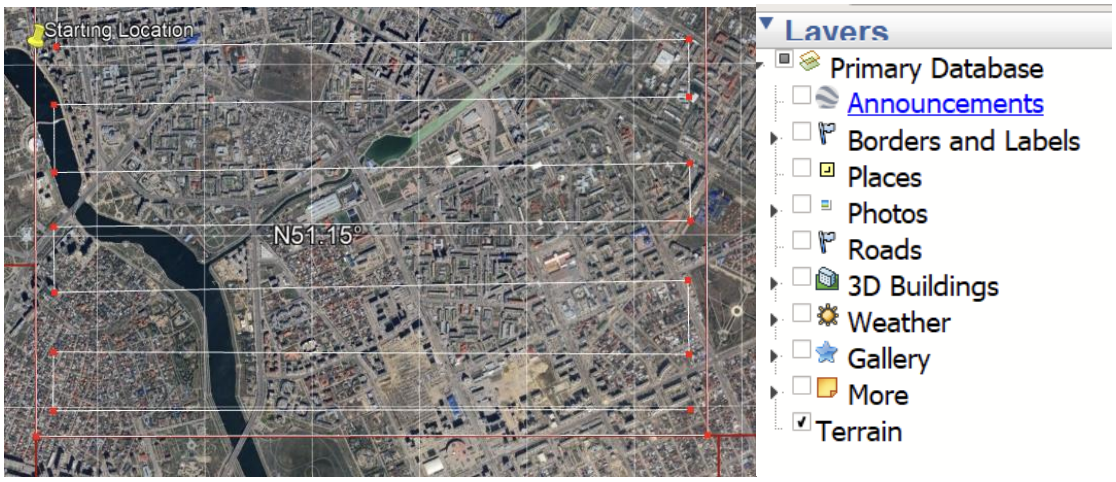

Figure 3.1.3. Adding paths and adjusting loaded layers.

3.1.3 Caching & storing GE tiles in Google Earth Pro

To avoid importing unnecessary data, it is recommended to hide all layers in the Layers panel except Terrain (particularly 3D Buildings, see Figure 3.1.3). It could be essential to avoid adding extra data to the cache limit.

The parameters for GEP touring were modified for tile caching. As seen in Figure 3.1.4, the most important parameters are the camera tilt angle (which should always be 0 to help with drawing paths), camera range (about 250 meters for the highest resolution tiles on a display of 3840 x 2160 pixel resolution), and speed (values between 90 and 100 are slow enough for low internet speed periods, one can up the speed to 150 if the connection is good). Map tile caching has little significance on the remaining parameters.

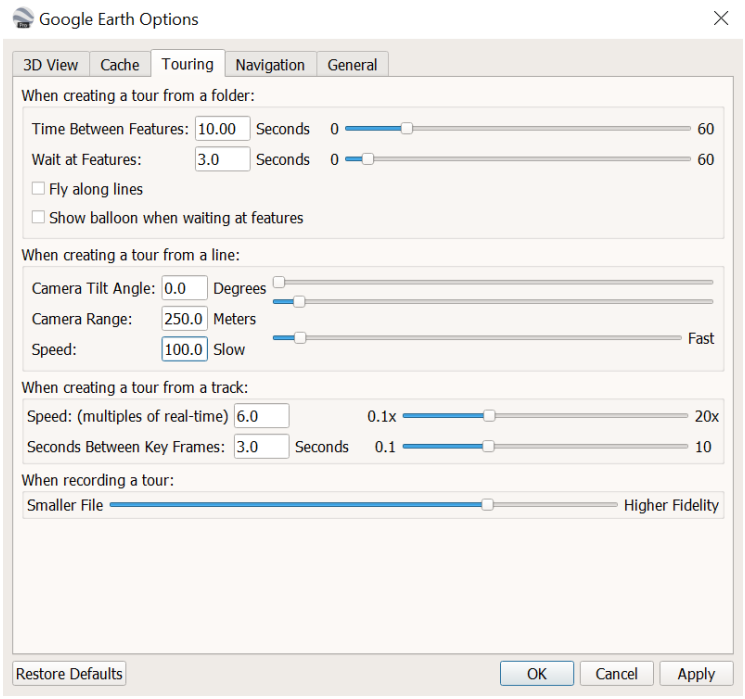

Figure 3.1.4. Touring Options

Finally, the tour was started by clicking the Play Tour button (located above the Layers panel's upper right corner) and choosing the corresponding sector path from the Places panel. It is recommended to pause touring at this point and clear the cache by using the Clear Disc Cache button on the Tools -> Options -> Cache page. The camera will first zoom in to the first location on the trail. At 90–100 mph, a single sector usually takes 90 minutes or less. After the tour, the application was shut down and the C:\Users\YOURUSERNAME\AppData\LocalLow\Google\GoogleEarth cache folder was moved to the storage device (an individual must be able to open a hidden folder to access these files).

3.2 Image preprocessing

Images collected sometimes might be subject to preprocessing, such as correcting geometric distortions, removing atmospheric effects, adjusting colors, and aligning images to a common coordinate system. The process of image preprocessing becomes much easier with some commercial imagery suggestions that have built-in filtering based on cloudiness. In such situations, the preprocessing is minimized due to the selection of images that will have cloud coverage of less than 10% or even less as shown in Figure 1. This is due to satellite imagery uses, as research or urban actions, which do not favor the presence of atmospheric effects.

The image preprocessing was avoided thanks to the usage of Google Earth Pro. The service obtains and updates images with lesser cloud coverage, so it's very user-friendly in terms of working with imagery. The images were manually stitched together and divided into equal parts. For the northwest sector of Astana, which is referred to further as Sector 4, the cache file was extracted to create sample images.

3.3 Data Annotation

In machine learning, data annotation is the task of labeling data to indicate the desired outcomes for prediction by your machine learning model. This involves marking—labeling, tagging, transcribing, or processing—a dataset with the specific features you want your machine learning system to learn and identify. Using specialized robust architecture, designed for object detection (like U-square Net), the boundaries of objects (such as structures, roads, and vegetation) for segmentation tasks are labeled and defined manually, by highlighting and filling the boundaries of target objects. In contrast to models built on existing backbones for the network, U-square Net is exclusively constructed using the suggested ReSidual U-blocks (RSU) and is planned to be used. This unique feature allows it to be trained from the ground up and adapted to various model sizes to accommodate specific environmental requirements. It can identify objects inside the image; therefore, with the right coding, it should help with labeling the boundaries for annotation. Items or classes inside the photos should be identified to do image recognition.

Classification of the objects on imagery was also an important issue to resolve for good model training. One study separated construction waste into two primary groups in the waste management system utilized in Hong Kong: inert and noninert materials [44]. These groups can be divided into more specialized types. Debris, concrete, and dirt are examples of inert materials, whereas noninert materials include things like wood, packing, and vegetation. Annotations are reviewed and checked to ensure accuracy by doing spot checks daily. A small sample is taken for error checking, and if it is found, the sample of images with the same attributes is corrected. As a result of data selection, the division of the 7:1:2 ratio was chosen as optimal for training, validation, and testing respectively [42, 43]. Once annotation preparation is done, etc., The AI model gains the ability to identify patterns and forecast outcomes using the available satellite imagery and accompanying labels during the training phase.

As the project deals with image segmentation, a good annotation tool that specializes in creating a segmentation mask was scoped. SamsungLabs has developed RITM (Reviving Iterative Training with Mask guidance) for Interactive Segmentation [45]. The repository was taken from GitHub and it is a click-based interactive segmentation. The simple feedforward model was provided that not only allows for segmenting an entirely new object but also starting with an external mask and correcting it. Its framework is built on Python 3.6 and relies on PyTorch 1.4.0+, the year 2022. The Demo of the annotation tool is shown in Figure 3.3.1.

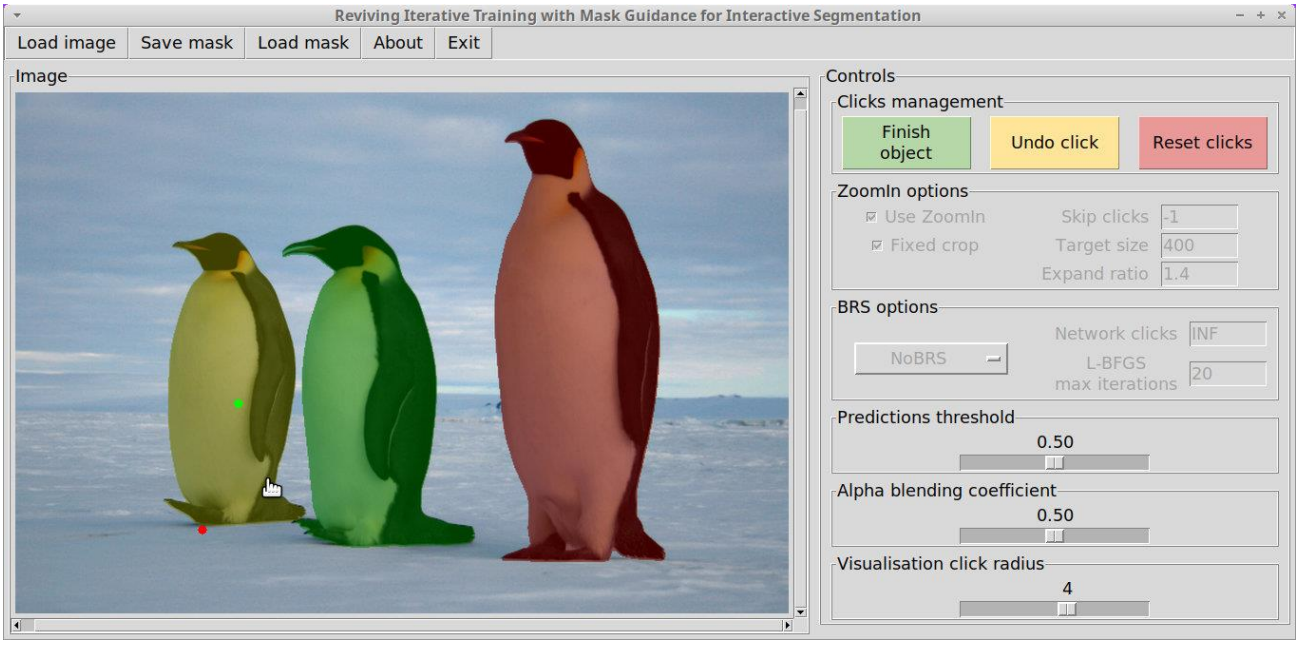

Figure 3.3.1. Demo of the annotation tool with masks

The repository was downloaded and implemented through the Visual Studio Code.

Figure 3.3.2. An example of an image with a corresponding mask

In Figure 3.3.2 the result is shown. The image was initially loaded in the tool and the area where CDW is located was clicked, creating a mask contour. The click can be undone if generated excessive area, tuned by clicking other parts that were missed and through “Prediction Threshold”, “Alpha Blending Coefficient” and “Visualisation click radius” scrolls. For the research, the prediction threshold has proven itself as a good tuning method. 100 images were annotated through the tool, resulting in the creation of grayscale segmentation masks with white color standing for CDW object and black as the background. It is important to note that different satellites have different imagery, which differs in color, saturation, and other factors. Therefore, to ensure that the area that was clicked on is indeed the CDW it was compared to the imagery produced by KazAeroSpace for the WasteOpen Space. The imagery there is more saturated and therefore easily distinguishable compared to the commercial imagery that Google Earth Pro uses. The problem with using that some key elements of imagery (especially masks for the waste) are not obtainable without access to government sources. Therefore, the publicly available map was used only to reference the location of CDW. It is important to note that the study didn’t implement pre-trained models, so classification was simplified. Therefore, some of the construction site materials were also classified as potential CDW.

3.4 Selecting a DeepLabv3+ Segmentation Model

The next step is to use a suitable deep-learning architecture to build a CW semantic segmentation model using reference models. There are many pre-built deep semantic segmentation architectures available, including Mask R-CNN and U-Net. DeepLabv3+ showed promising results from the

previous study, which proved itself as a relevant source [43]. As a result, the main architecture for our CW segmentation model in this study is DeepLabv3+, with the decision to find a suitable code and modify it for current research purposes.

The DeepLab series models, initially proposed by Chen et al., have been instrumental in advancing CNN architectures [46]. The Deeplabv1 model introduced dilated convolutions within the backbone network to address the challenge of information loss during convolution operations. This innovation effectively integrates more feature information without significantly increasing the model's parameters and computational burden. Subsequently, Chen L C et al. proposed the Deeplabv3 network, which enhanced the ASPP (Atrous Spatial Pyramid Pooling) module by adding a global average pooling branch and incorporating batch normalization after ASPP. This modification improved the network's ability to capture global information efficiently.

Recognizing the limitations of the DeepLabv3 model, particularly in terms of potential loss of object boundary details and high computational costs associated with dilated convolutions, the team introduced the DeepLabv3+ network in 2018. This model builds upon the DeepLabv3 architecture by incorporating an encoding-decoding module. DeepLabv3 serves as the encoder, while the decoder module is added to restore detailed boundary information of the target objects.

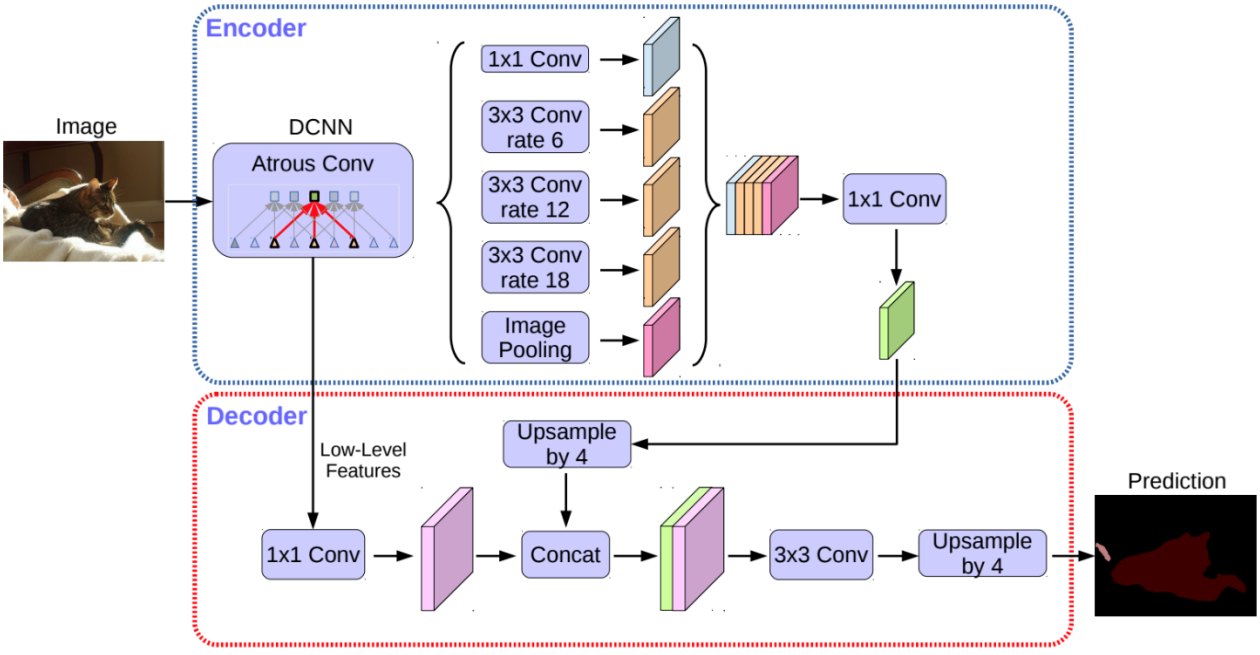

Figure 3.4.1. The outline of the DeepLabV3+ model [47]

The model consists of an encoder and a decoder, as shown in Fig. 3.4.1. The backbone network and the atrous spatial pyramid pooling (ASPP) module make up the encoder's two modules. First, the backbone extracts features using popular CNN structures such as ResNet [48] and Xception [49]. By using the feature maps that the backbone provides, the ASPP module uses atrous convolution at varying rates (ri, where i = 1, 2, 3), such as r1=6, r2=12, and r3=18, to collect features at different scales. To guarantee scale invariance, these features are then combined with the result of a 1×1 convolution operation and the global average pooling feature [50]. In the decoder, the ASPP output and low-level features retrieved from the backbone are concatenated. Through convolution and upsampling processes, the output gradually increases in resolution to match that of the original input image, finally producing the anticipated segmentation result.

In the implementation of DeeplabV3+ with MobileNetV2, several optimizations have been incorporated to enhance model performance and efficiency. The original code is taken from the CARLA Semantic Segmentation notebook in Kaggle that was done on self-driving car images [47].

3.5 Model Training and Validation

Pairs of input photos and the labels that go with them (such as segmentation masks or class labels) make up the training data. In this procedure, training data is fed into the model, the loss (a metric for the prediction error) is computed, and backpropagation is used to update the parameters. For the training phase, a neural network with a backbone in the form of MobileNetV2 is planned to be used now. The outline for the model is shown in Figure 3.4.1.

The model's performance during training is evaluated during the validation process. This is accomplished by using a distinct validation dataset from the training data. Feeding the validation data through the model and calculating metrics like accuracy, precision, or recall provides for an evaluation of the model's performance. As part of the validation, a confusion matrix will be created, and ground verification will be performed. As a result of the testing phase, the data are analyzed for two main features: segmentation accuracy and time consumption. For consumption, the mean time taken to process one image can be taken as an evaluation parameter. For segmentation accuracy, several metrics are used: accuracy, precision, recall, and F1-score across all classes. All of them have a formula based on the total number of true positive (TP), false positive (FP), and false negative (FN) pixels for class *i* over all images in the dataset. In the case of this research true positive stands for correctly predicted number of positives when the image is a CDW. True negative - correctly predicted

negatives when the image is not a CDW. False positive stands for the number of incorrectly predicted positives when the image is not a CDW and false negative is the number of incorrectly predicted negatives when the image is a CDW.

- Accuracy: defined as the ratio of the number of correctly recognized construction materials to the total number of testing samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (2)$$

- Precision: defined as the ratio of correctly recognized construction images to the total images recognized as construction material.

$$\text{Precision} = \frac{\sum_{i=1}^{N} \text{TP}_i}{\sum_{i=1}^{N} \text{TP}_i + \text{FP}_i} \quad (3)$$

- Recall (Sensitivity): defined as the ratio of correctly recognized construction images to the total number of sampling images.

$$\text{Recall} = \frac{\sum_{i=1}^{N} \text{TP}_i}{\sum_{i=1}^{L} \text{TP}_i + \text{FN}_i} \quad (4)$$

- F1-score: The F1-score is the weighted average of the precision and recall. Therefore, this metric considers both the FP and FN.

$$\text{F1} - \text{score} = \frac{\sum_{i=1}^{N} 2\text{TP}_i}{\sum_{i=1}^{L} 2\text{TP}_i + \text{FP}_i + \text{FN}_i} \quad (5)$$

In general, the higher the metrics are, the better is segmentation, with the upper limit being 1, representing a complete alignment of the predicted composition with the ground truth (annotated mask) [43]. Semantic segmentation makes use of IoU (intersection over union), a popular annotation metric [51]. Pixel-by-pixel comparison is performed between the genuine areas manually designated and the CDW zones identified by the model. Equation (6) can be used to calculate the IoU, which represents the accuracy of the recognition results.

$$\text{IoU} = \frac{\sum_{i=1}^{N} \text{TP}_i}{\sum_{i=1}^{L} \text{TP}_i + \text{FP}_i + \text{FN}_i} \quad (6)$$

The testing procedure is performed to gauge how well the model performs on untested data. To prevent bias or overfitting, a testing dataset that is distinct from the training and validation data is used. The model is applied to the testing dataset, and its predictions are contrasted with the ground truth labels.

The model's hyperparameters (such as the backbone network, inputs, outputs, image resolution, and dataset) are established through the validation procedure. The testing outcomes give a fair assessment of the model's performance in actual situations and its capacity to generalize to fresh, untested satellite imagery. The efficiency by IoU is complex to identify, however, some papers have reached 0.56 like Chen's results, which he compares to other studies that reached a mean IoU of 0.878 on Pascal VOC 2012, its performance on ADE20K is 0.457 in comparison [43]. However, these are already pre-trained results and given the nature of the self-build dataset in the current study, reaching those values would be considered as very satisfactory results. IoU values are analyzed in the context of other evaluation metrics as well.

Chapter 4 – Results and Discussion

4.1 Problem Solving Report

During the image obtaining section, the main problem for the project was the cost of the imagery as a lot of imageries were commercially distributed to various companies and they have high prices for high-quality pictures for some pictures of cities like Astana may cost up to 10000 euros as shown in Figure 4.1 [52]. The cost of 100000 credits in the given picture is equal to 10000 euros.

Figure 4.1. The cost of the Pleiades image on available sources.

The cost was the problem because for the project the imagery had to be a high-quality project focused on commercial satellite imagery for too long, so finding a good substitute method was done later. Later it was deduced to take imagery from a cache file of Google Earth Pro [53].

Obtaining images from Google Earth Pro so screenshots from Google Earth Pro were made by firstly touring through the application and then recovering the cache file and using it to form a screenshot for the imagery. Different environments were considered to work with the model [54]. The first choice

was TensorFlow and its extension of Google Colab. It was considered the best option at the beginning as it was free, had cloud storage, and was very easily accessible. However, during test runs it was estimated that the general mean time of compilation is too long, and for good running capabilities, there is a GPU needed to accelerate the process. Two methods of solving this issue were looked for:

1. to obtain a good computer with GPU
2. to find another environment that has a cloud GPU service.

For the second option Kaggle environment was chosen as the Kaggle has the accelerator function built in which uses GPU T4 x2 to accelerate the process and the runtime is much faster than in Google Colab. Projects that may contain the codes with DeepLabV3+ and its backbone networks were researched within this environment and successfully found.

4.2 Annotation Results

The Roboflow tool was found for annotation, but it was later discovered that this tool is mostly focused on object detection and research focuses on image segmentation [55-57]. Though there was an option for semantic segmentation as well it was later found out that the masks that they produce are not correct in a way that they don't produce proper masks. Initially, five classes were made for Roboflow:

- Buildings
- CDW
- Road
- Water objects
- Vegetation

You can see a sample image in Figure 4.2.1, which helps to explain the justification for the previous 5 classes.

Fig 4.2.1. The visual inspection sample for annotations.

However, some problems have occurred during the use of Roboflow. One of the problems that was faced with annotation was the buildings' classification. As the project focuses on CDW, buildings are initially annotated in models to differentiate them from the waste. The problem starts with garages, as they are relatively small and harder to confirm as a building. Annotators might consider such small structures as a building, but then container roofs were found on the city map, which is also quite fluid in its description and belonging to a building class. This might interfere with the learning process, especially if CDW is located right in containers.

The other problem is sandy areas. As the visual inspection shows, the sand is abundant in landfills and construction site areas. Aside from this fact, there is also a presence of sandy roads and some small blending of sand area with vegetation. This is a rather problematic feature to label as it poses questions, such as the number of classes to correctly identify the sand and differentiate it from the CDW that might be located on it.

Another experimental demo for the segmentation was found in the form of a repository made by the Samsung Labs team. Masks were loaded there and through the clicking activity, the masks were created in PNG form successfully. Initially, the project was much more complicated with annotation stuff as there were five classes, but due to time complications and the incompatibility of the Roboflow annotation tool, it was simplified down to two classes. The background was colored black, and the

area that contains the construction demolition waste was colored white. The images and masks were collected in separate folders after finishing the task, see Figure 4.2.2.

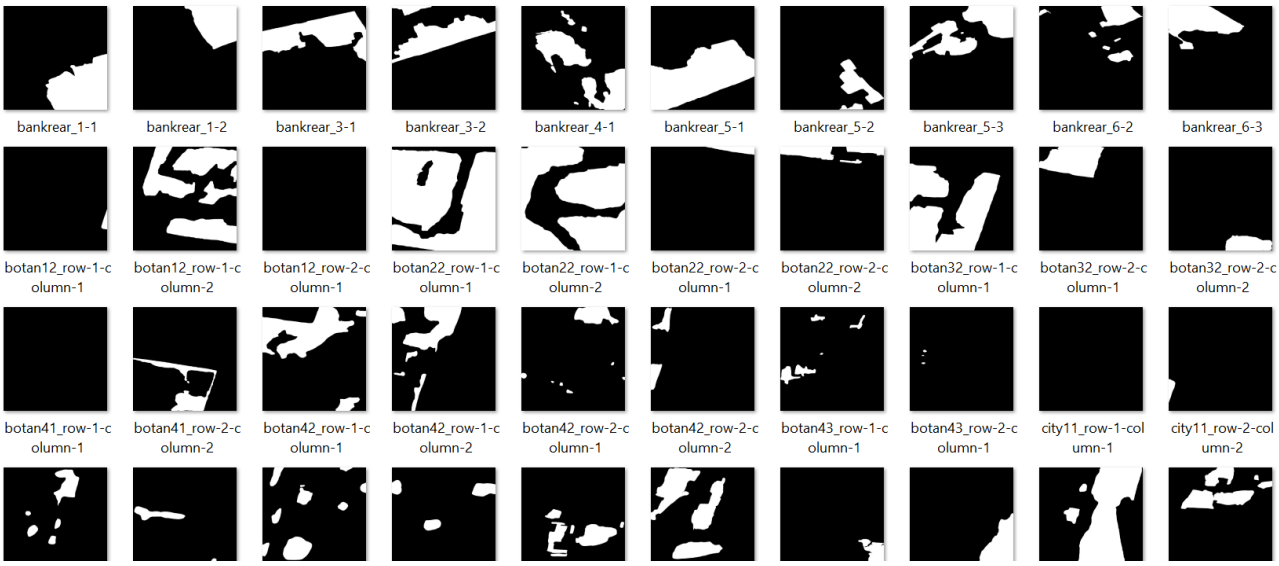

Fig 4.2.2 Masks, collected in one folder.

4.3 Model Training Results

4.3.1 Troubleshooting the Original Code

Most research papers that are focused on semantic segmentation and related to CDW mentioned the DeepLabV3+ showing the most promising results so the code was found and later developed further down to create a good running model [58, 59]. Initially, it was image semantic segmentation with DeepLabV3+ that was used on the self-driving car simulators to find the difference in the background of the road. The dataset of CDW was uploaded to this and initially, it was run with the original model used in the code which was DeepLabV3+ and Resnet50 as the backbone network [60, 61], but the training was not successful. It happened due to Resnet50 being a very complicated model that divides images down into small parts and at some point, with the dataset that research had it produced blank pixels and couldn't train further. This led to a decision to change a backbone network to something lightweight, which was a MobileNetV2 [62]. The backbone network, instead of utilizing the ResNet50 network, has opted for MobileNetv2 as shown in Figure 4.3.1.

```python
def DeeplabV3Small(image_size, num_classes):
    model_input = keras.Input(shape=(image_size, image_size, 3))
    mobilenetv2 = keras.applications.MobileNetV2(
        weights="imagenet", include_top=False, input_tensor=model_input
    )

    # Freeze the layers
    for layer in mobilenetv2.layers:
        layer.trainable = False

    x = mobilenetv2.get_layer("block_13_expand_relu").output
    x = DilatedSpatialPyramidPooling(x)

    input_a = layers.UpSampling2D(
        size=(image_size // 4 // x.shape[1], image_size // 4 // x.shape[2]),
        interpolation="bilinear",
    )(x)
    input_b = mobilenetv2.get_layer("block_3_expand_relu").output
    input_b = convolution_block(input_b, num_filters=32, kernel_size=1)

    x = layers.Concatenate(axis=-1)([input_a, input_b])
    x = convolution_block(x, num_filters=128)
    x = convolution_block(x, num_filters=128)
    x = layers.UpSampling2D(
        size=(image_size // x.shape[1], image_size // x.shape[2]),
        interpolation="bilinear",
    )(x)
    model_output = layers.Conv2D(num_classes, kernel_size=(1, 1), padding="same")(x)
    model = tf.keras.Model(inputs=model_input, outputs=model_output)

    return model
```

Figure 4.3.1. The excerpt of the code with the implementation of MobileNetv2

MobileNetv2's lightweight architecture and efficient feature extraction capabilities were particularly advantageous for the task of semantic segmentation [63]. The test runs with ResNet50 have shown that the backbone network is too complicated for the small-scale dataset that the research had, so it was a logical step to refer to lightweight architecture like MobileNetv2, which proved itself to be more efficient. This was also mentioned in several papers, and it was chosen as it can work with smaller datasets. The code was changed for it to work and finally, the training process started. In summary, the implementation of DeeplabV3+ with MobileNetV2 incorporates various optimizations tailored to enhance segmentation accuracy, reduce model complexity, and improve efficiency. These optimizations have collectively contributed to a more effective and efficient semantic segmentation model for the specific task or application.

Regarding the convolution blocks, due to its connection to Resnet50, which divided it into smaller sections, the necessary regularizer was added for it to stop at some point, where it starts showing blank blocks. They were regularized in that way to produce effective training as shown in Figure 4.3.2.

```
def convolution_block(block_input, num_filters=128, kernel_size=3, dilation_rate=1, padding="same",
    x = layers.Conv2D(
        num_filters,
        kernel_size=kernel_size,
        dilation_rate=dilation_rate,
        padding=padding,
        use_bias=use_bias,
        kernel_initializer=keras.initializers.HeNormal(),
        kernel_regularizer=keras.regularizers.l2(l2_reg),
        )(block_input)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    return x
```

Figure 4.3.2. Keras Rugularizer implementation in the code

The form of the final images and masks were in JPG format, so decoding was changed to fit the format and for masks, as they are not in PNG as the original dataset, they were normalized and converted to uint8 for proper work. The images were divided according to the 7:1:2 ratio, as it was mentioned before, and during the coding process, some hyperparameters were chosen for example batch size, number of channels, image height, width, filters, learning rate, and optimizers.

4.3.2 Accuracy and Loss Graph Assessment

The epoch number was initially 30, but during checks on the training and validation accuracy and training and validation loss graphs, it was estimated that the process wasn't complete. It is noticeable from Figure 4.3.3 as well. 50 epochs were run with this one to ensure that there were no significant changes for the training further down.

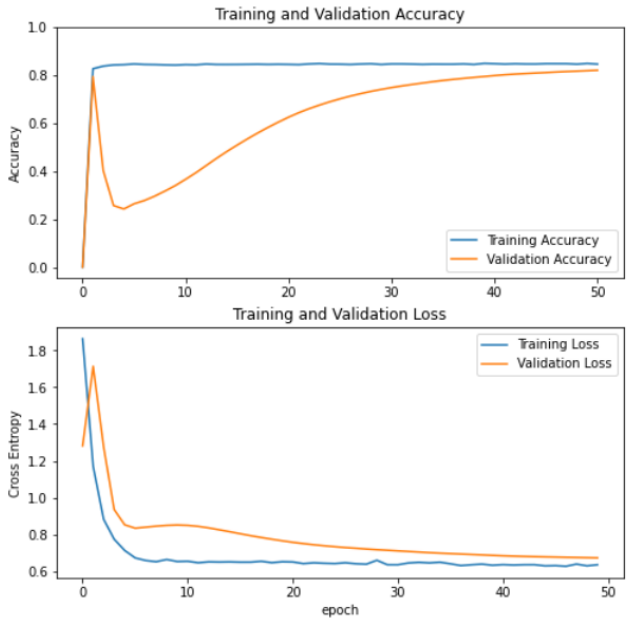

Figure 4.3.3. The graph for loss and accuracy (training and validation)

Later, hyperparameter tuning was done to define the best ones, and those hyperparameters were used to run the training, and training variables were adjusted according to those results.

As a result of the model training, the best hyperparameters were chosen. The learning rate, the best results were produced with:

- Learning rate = 10E-04
- 'Atom' optimizer,
- Number of filters = 32
- The height and width of the image = 512x512 pixels,
- batch size = 25

The training accuracy quickly converges to a high value, stabilizing around 0.85 at about 2-3 epochs. In contrast, the validation accuracy starts low and improves steadily, reaching approximately 0.80 by the 50th epoch. This indicates good generalization with minimal overfitting. The training and validation loss curves display rapid initial convergence, with the training loss approaching 0.65 and the validation loss stabilizing around 0.7, suggesting effective learning and model stability throughout the training process.

4.3.3 Visual Inspection

As can be seen in Figure 4.3.4, the masks produced during training, validation, and test datasets have some kind of resemblance to the masks that were drawn by the annotation tool.

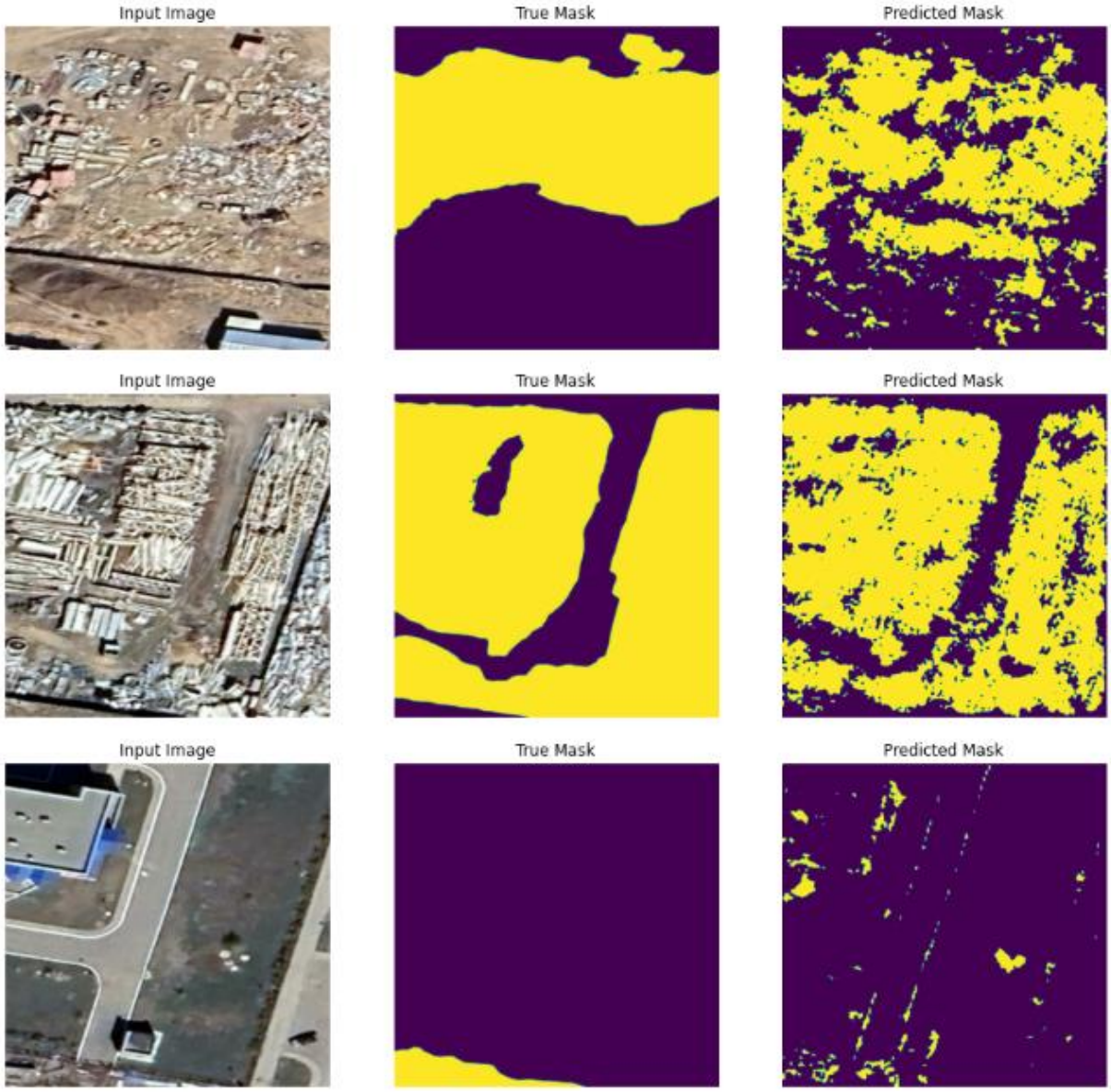

Figure 4.3.4 The comparison between the image, true mask, and predicted mask

So, it can be said that artificial intelligence can be trained by machine learning to be able to recognize construction demolition waste to an extent. The results shown in Figure 4.3.4 are promising for a small dataset of 200 images. It is evident from the visual inspection that the predicted mask resembles the true one that was drawn by the annotator. There is an advantage in masks drawn from the learning as they are better at distinguishing sand from the waste and leaving it out of the mask for CDW. However,

in the third picture, the results show that the parts of a building can be also considered as CDW by the model due to their resemblance in color and texture. It is more evident in Figure 4.3.5, where a sample image from validation results is shown.

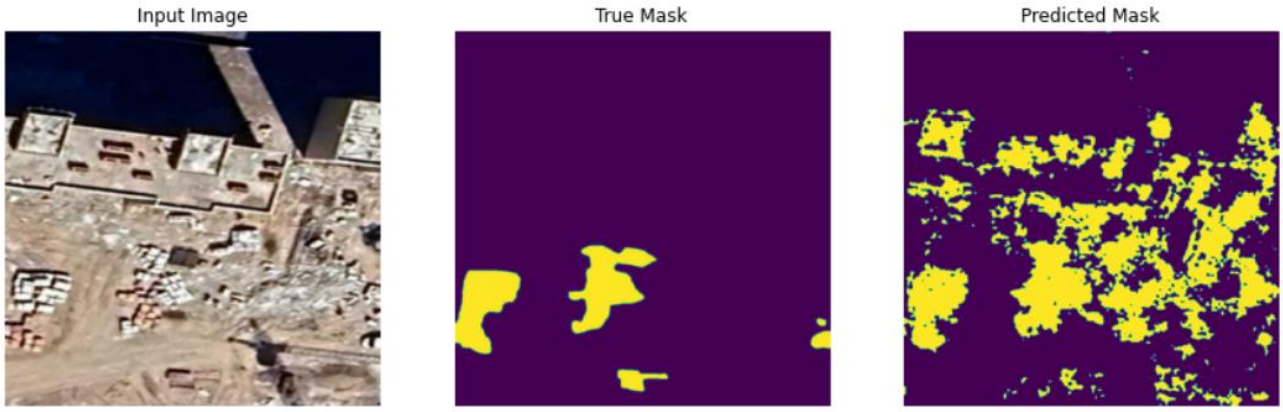

Figure 4.3.5. The validation sample image with masks

Here, the model tried to cover more of the CDW, especially the area on the right side, but it also considered the parts of the roof of a building during the construction phase as CDW. This occurred often with the imagery of CDW near construction sites, as the waste is taken from those sites and consists of the same materials. Some cars on the roads, especially those that have white or grey roofs (concrete color) were taken by the model as part of the CDW. Containers that are metallic and red were considered in the same way, though they are most likely to resemble metal and brick waste.

4.3.4 Evaluation Metrics Analysis

4.3.4.1 Accuracy, Precision, Recall and F1-Score

From the training and validation accuracy, both validation and training accuracy show stable graphs during later epochs and nearing an 80% percentage accuracy. The training and validation loss curves look very stable.

Table 4.1. Evaluation metrics for the project

Metric	**Value**
Test Accuracy	0.828
Test Precision	0.032

Test Recall	0.021
Test F1-score	0.068

When compared to ground truth labels, the test accuracy of 82.85% indicates that the model's predictions are generally accurate as shown in Table 4.1. On the other hand, recall (2.12%) and precision (3.22%) show difficulties in precisely detecting CDW pixels and reducing false positives and false negatives. The 6.85% F1-score indicates that the model's classification capabilities need to be improved by striking a balance between recall and precision.

4.3.4.2 Confusion Matrix

The confusion matrices in Figure 4.3.6 for the training, validation, and test datasets highlight key trends in the model’s performance. The model performed well in recognizing non-waste and waste instances in the training set, achieving a high True Negative (TN) rate of 28594864 and a high True Positive (TP) rate of 2897353. False Positive (FP) and False Negative (FN) numbers were notable however, at 3285671 and 1922272, respectively, suggesting some misclassifications.

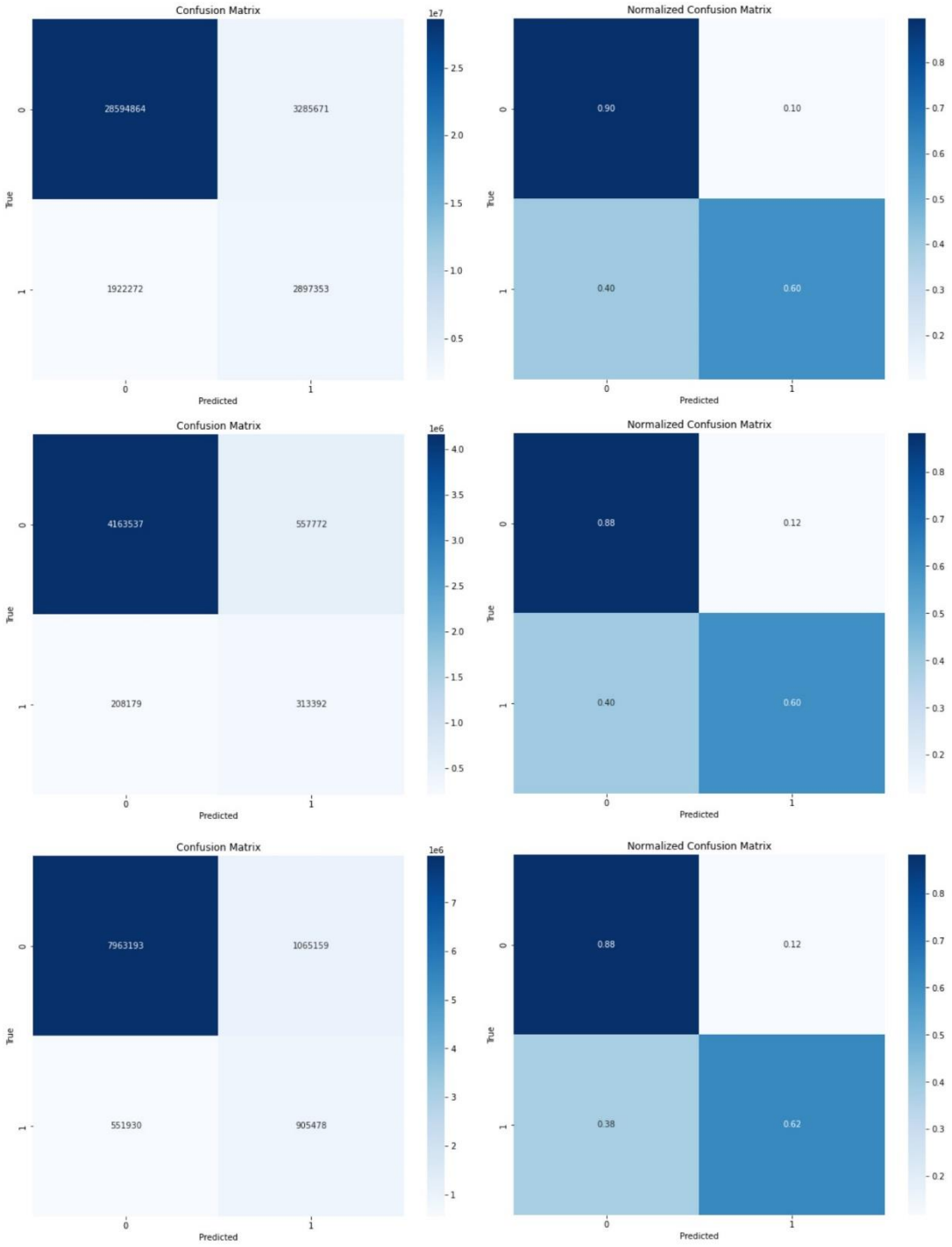

Figure 4.3.6. Confusion Matrices for Training, Validation, and Test from top to bottom

The model performed consistently in the validation set, yielding TN and TP values of 4163537 and 313392, respectively. The FP and FN counts, which were 557772 and 208179 respectively, demonstrated the consistency of the model over various data splits. The model performed steadily for the test set, with TN and TP counts of 905478 and 7963193, respectively. The counts for FP and FN

were 1065159 and 551930, respectively. This suggests a minor increase in misclassifications, which is normal in real-world scenarios.

Normalized confusion matrices provide more insight into such trends. The model accurately recognized 60% of waste occurrences (TP) and 90% of non-waste instances (TN) in the training set. Similar results were seen in the validation set, which had TP rates of 60% and TN of 88%. The test set demonstrated a slight improvement in TP rate at 62%, with 88% TN, suggesting the model generalizes well to unseen data but still has room for improvement in correctly identifying waste instances.

Overall, the confusion matrices and their normalized counterparts show where waste detection still needs work but also demonstrate how reliable the model is at identifying non-waste situations. To further increase performance and accuracy, future improvements might concentrate on lowering false negatives and improving model architectures, data augmentation, and hyperparameters.

4.3.4.3 Mean IoU score

However, for segmentation, it might be more helpful to use mean IoU values [64]. Those results are depicted in Table 4.2.

Table 4.2. Mean IoU values for each split of the dataset

Metric	**Mean IoU**
Training	0.373
Validation	0.313
Test	0.380

The accuracy of the model in defining CDW regions can be more fully understood by examining the mean Intersection over Union (IoU) values. The IoU values show an adequate capacity to identify CDW areas, with roughly 0.373 on the training dataset, 0.313 on the validation dataset, and 0.380 on the test dataset. Compared to the results of other studies, it lacks in numbers of benchmark (0.457 to 0.56) given by other studies [43]. However, due to the origin of our dataset (not pretrained and collected manually), and field of satellite imagery, and the complexity of its features, the result is considered satisfactory. Still, there is space for improvement, especially in terms of improving the

model's ability to generalize to new data, as evidenced by the difference in IoU scores between training and validation/test.

The complexity of the CDW segmentation task is highlighted by the disparity between segmentation and classification measures. Lower IoU values result from the model's potential accuracy in classifying pixels, but it still has difficulty precisely defining the borders of CDW zones. Strategies like data augmentation, further regularization approaches, or investigating more resilient architectures could increase performance on unknown data and improve generalization to solve this. The lower result might be attributed to the previously mentioned faults of the model assuming other objects as CDW.

The other possible reason for an inaccurate result can be the quality of the image and this is related to the imagery that was extracted from Google Earth and its comparison to the KazAeroSpace one.

Figure 4.3.7. The comparison between the imagery of KazAeroSpace (left) vs. Google Earth (right)

As seen in Figure 4.3.7, the saturation of the image might have influenced the results and in the future, either imagery of KazAeroSpace can be used as the dataset or code can be modified to preprocess images to make them more saturated, so the CDW can be highlighted on the imagery. It is possible that with a bigger dataset and a much clearer depiction through the masks, the model will be able to identify and distinguish the waste much more clearly. The performance can be further increased by employing pre-trained models, which is otherwise known as transfer learning [65-68]. Similarly, contrastive learning, which uses unlabeled data to learn important image features can be implemented in the future [69].

4.3.5 Numerical Results

After checking the results by visual inspection and evaluating metrics, the testing phase results were chosen as a sample for numerical comparison. Given the knowledge of the original image size and correspondence of the distance per pixel, the decision was to modify the code even further, to give a rough estimate of the percentage of area that is considered as CDW. This was presented in Figure 4.3.8 form, where the initial area for the true and predicted mask is given. Two important notes should be made before the analysis of the results. Firstly, the true masks are round shaped and cover whole sections, which is due to the nature of the RITM segmentation tool, therefore, might be larger in area than the actual waste. The predicted mask, which is more detail-oriented, is less in the area and has a rough shape. This leads to a second point, which was that the overlay comparison was created to check how much of the area that was annotated the model will assume as CDW. This was done to compare the approach as was mentioned before, but also to eliminate the problem of incorrectly assumed objects, like building roofs, containers, and cars. This overlay can give information about the difference between the segmentation tool and model training.

```
Image 1:
True mask percentage area: 27.30%
Predicted mask percentage area: 32.30%
Accuracy within true mask: 71.89%
```

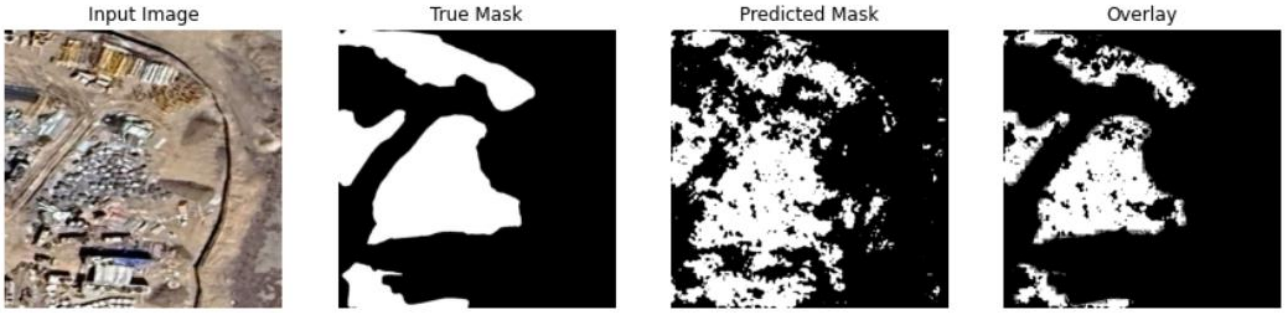

Figure 4.3.8. Numerical Comparison of True Mask vs. Predicted Mask

In Table 4.3 the numerical analysis is shown to deduce the area of the waste. The calculations for the last graphs are taken from the code excerpts that show the percentage of the total image in Figure 4.3.5. The original image is 1056 x 1002 pixels, with 0.075 m per pixel resolution, which results in a total area of 5952 m^2.

Table 4.3. The numeral analysis of the sample image

Area resolution	0.075 m per pixel
Original Image Size (in pixels)	1056 x 1002
Total Area	5 951.88 m^2
True Mask Waste Area	1 624.86 m^2
Prediction Mask Waste Area	1 922.46 m^2
Correctly Predicted Area (Overlay)	1 168.11 m^2

Through annotation about 1 625 square meters were marked as CDW. The model, however, picks up more area that was annotated. In this case, it gives a positive result, as from the visual inspection the predicted mask is closer to the true depiction than the true mask, making the area of 1 922.46 square meters a more believable result. However, as mentioned before, in other cases the non-CDW objects are also taken as a CDW, which highlights the flaw of the annotation tool as well as tuning parameters, enlarging the dataset, and further improving the model.

Table 4.4. The waste area analysis for the test dataset

#	Image	True Mask Waste Area	Prediction Mask Waste Area	Correctly Predicted Area (Overlay)

1		1624.86 m^2	1922.46 m^2	1168.11 m^2
2		832.14 m^2	1168.06 m^2	510.85 m^2
3		255.58 m^2	774.44 m^2	92.47 m^2
4		929.02 m^2	1482.73 m^2	567.72 m^2

5		872.13 m^2	531.01 m^2	362.80 m^2
6		863.44 m^2	265.05 m^2	182.62 m^2
7		272.99 m^2	429.41 m^2	121.97 m^2
8		485.28 m^2	641.41 m^2	273.94 m^2

9		736 m^2	531.67 m^2	216.46 m^2
10		252.62 m^2	1101.35 m^2	208.86 m^2

In Table 4.4, it is shown that construction images such as Image #3 and #4 yield a larger area in predicted masks than true masks, because of mistakenly taking the building as CDW. For the rest of the images, the predicted mask fills some area left by the annotator but is also much more detailed and complex in the figure. Therefore, the true result should be somewhere between the values of the CDW area of true and predicted masks. For such cases, an estimated range with upper and lower boundaries can be made from the two values provided. An example from Image #7: an estimated CDW area in the image lies in a region of 273 to 429 square meters, with an average estimate of 351 square meters. The same approach can be applied to other images.

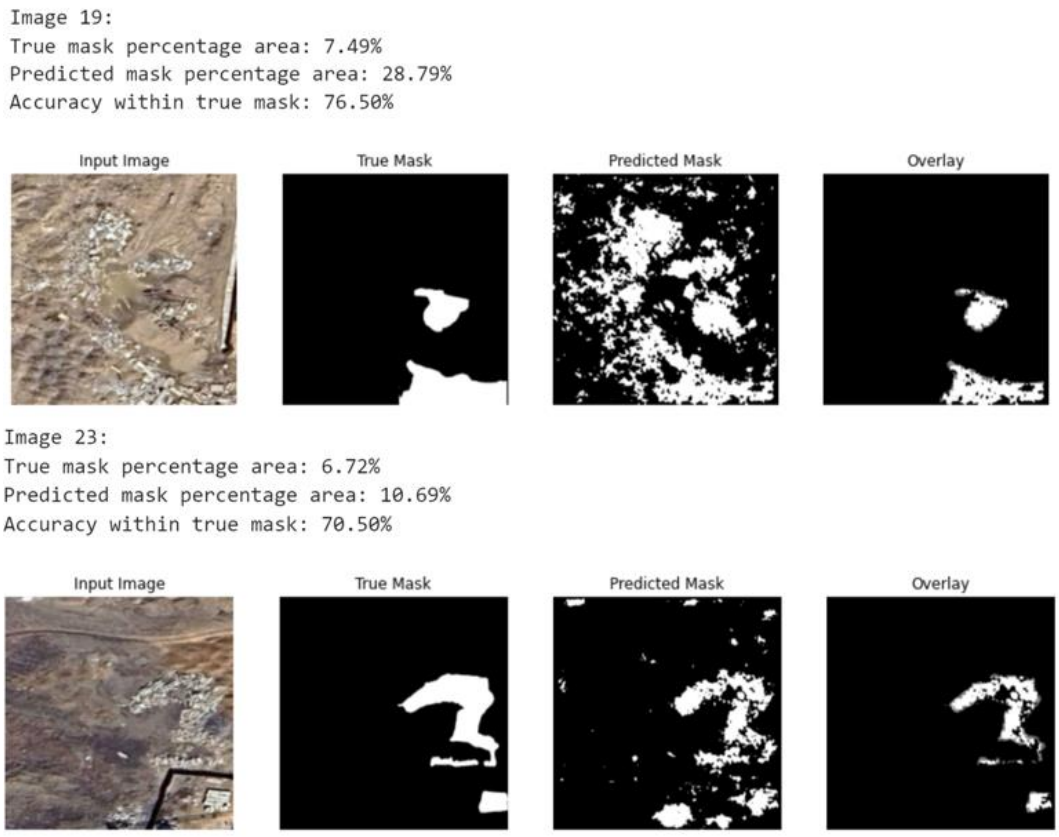

Figure 4.3.9. Examples of superior prediction by the model

There are, however, implications that the annotation process needs better accuracy, as there are examples of images, where the model was better at identifying CDW than the drawn true mask, as in Figure 4.3.9. This indicated that the model training process needs to be attuned to match the effectiveness. With the small dataset, it outperformed the annotator in several cases, which shows promising results for future works.

Chapter 5 - Conclusions

5.1 Conclusions

This study's main goal was to figure out if a convolutional neural network (CNN) could effectively distinguish construction and demolition waste (CDW) in urban areas from satellite images. The main findings and conclusions of the study include:

- The convolutional neural network can distinguish CDW from the surroundings, albeit to a certain extent with a limited dataset. The predicted masks generated show some similarity to

the true masks that were done by an annotator and visual inspection shows its effectiveness. The model achieved a mean IoU of 0.380 on the test dataset, indicating adequate capacity to identify CDW areas. However, it is still below the benchmark values cited in other studies (0.457 to 0.56), suggesting room for improvement. The research didn't use transfer learning and other methods to enlarge the dataset, therefore future studies have the advantage of further developing the field of study.

- Accuracy in training and validation settled at 0.80, suggesting low overfitting and strong generalization. Evaluation metrics, however, such as recall (2.12%) and precision (3.22%) show how challenging it is to accurately identify CDW pixels and lower the number of false positives and negatives. The confusion matrices clearly show this, indicating that although the model correctly detects non-waste cases, it still needs refinement in terms of effectively recognizing waste.
- Visual inspection revealed that the model incorrectly identified certain car and building elements as CDW, which impacted the model's performance. The main reason is the similarity of colors and textures. This suggests that additional model optimization is required to increase accuracy.
- The study revealed that it is possible to train a model through machine learning to identify CDW. Such methodology can be implemented further with the collaboration of KazAeroSpace and enhanced for use in waste management. Several articles show that there is a growing demand for simplified management tools and machine learning is showing as one of the promising solutions to waste handling in Kazakhstan.
- The model can be applied to the management of waste resources around the city. If focused only on the bigger pieces of masks, it can be effective at predicting huge dumps of waste or piles left from construction sites. The model also consists of two classes, so making a multiclass model might be even more helpful to differentiate the materials.

The research was successful at dealing with and adapting to challenges that were raised during the span of the work. The study faced several limitations: availability and pricing of high-quality satellite imagery, reliance on Google Earth imagery, annotation tool, and simplification of classification. The high price of commercial satellite imagery was one of the project's major obstacles. This was addressed by using photos from Google Earth Pro, proving to be a practical and affordable way to get good-quality images. Simplifying the classification to two classes—CDW and non-CDW—resolved the initial issues with annotation tools, especially Roboflow, which generated inaccurate masks. Due

to its effective GPU acceleration, which greatly improved the training process's runtime, the switch to the Kaggle environment for model training was critical. The architecture of the model was optimized by moving from the heavyweight ResNet50 to the lightweight MobileNetV2, which worked better with the smaller dataset.

5.2 Limitations and Future Improvements

Future improvements should focus on data augmentation, further regularization approaches, and investigating more resilient architectures. The annotation tool might be changed to more detailed and fine-grained, given the complexity of CDW depiction on satellite imagery. Building multi-class models, which will separate waste of different origins (concrete, steel, etc.) will also improve the quality of the study. As the number of studies regarding CDW recognition through machine learning is growing, it enables future studies to use this and other models for transfer learning and implement contrastive learning. Collaboration with local companies like KazAeroSpace might help future studies avoid some of the limitations that our study faced. Overall, the model shows promising results, which can be further developed with mentioned methods.

References

[1] B. Torgautov, A. Zhanabayev, A. Tleuken, A. Turkyilmaz, C. Borucki, and F. Karaca, "Performance assessment of construction economy: A balanced scorecard approach," Sustainable Production and Consumption, vol. 33, pp. 991-1004, Sep. 2022. https://doi.org/10.1016/j.spc.2022.08.021

[2] "Проблема утилизации строительных отходов в Казахстане," Министерство экологии, геологии и природных ресурсов Республики Казахстан, 25-Aug-2021. [Online]. Available: https://www.gov.kz/memleket/entities/ecogeo/documents/details/487827?lang=ru

[3] "UN Population Prospects: Case of Central Asia," Eurasian Research Institute, Jun. 27, 2023. [Online]. Available: https://www.eurasian-research.org/publication/un-population-prospects-case-of-central-asia/. [Accessed: Sep. 3, 2024].

[4] Jouni Korhonen, Antero Honkasalo, Jyri Seppälä, Circular Economy: The Concept and its Limitations, Ecological Economics, Volume 143, 2018, Pages 37-46, ISSN 0921-8009, https://doi.org/10.1016/j.ecolecon.2017.06.041.

[5] PBL Netherlands Environmental Assessment Agency. “Circular economy: Measuring innovation in product chains.” PBL Netherlands Environmental Assessment Agency, https://www.pbl.nl/en/publications/circular-economy-measuring-innovation-in-product-chains (accessed Jun. 19, 2024).

[6] Official Journal of the European Union, Directive 2008/98/EC of the European Parliament and of the Council on waste, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32008L0098&from=EN, (2008).

[7] U.S. Environmental Protection Agency. “Fact Sheet about the National Recycling Goal: 50 percent by 2030.” EPA, https://www.epa.gov/circulareconomy/fact-sheet-about-national-recycling-goal-50-percent-2030 (accessed Jun. 19, 2024).

[8] F. Karaca and A. Tleuken, "Reforming Construction Waste Management for Circular Economy in Kazakhstan: A Cost–Benefit Analysis of Upgrading Construction and Demolition Waste Recycling Centres," Recycling, vol. 9, no. 1, pp. 1-11, Dec. 2023. https://doi.org/10.3390/recycling9010002

[9] Sabina Balashkina. "Полигон для строительных отходов запустят в столице " 24kz, August, 2021. Retrieved from https://www.youtube.com/watch?v=A2U6lIXwnKc

[10] Площадка по приему строительных отходов запущена в Астане, Akimat of Astana, 2023. Available: https://www.gov.kz/memleket/entities/astana/press/news/details/526466?lang=ru

[11] Darko, A., Chan, A.P.C., Adabre, M.A., Edwards, D.J., Hosseini, M.R., Ameyaw, E.E., 2020. Artificial intelligence in the AEC industry: scientometric analysis and visualization of research activities. Autom. Constr. 112, 103081. https://doi.org/10.1016/j.autcon.2020.103081.

[12] Pan, Y., Zhang, L., 2021. Roles of artificial intelligence in construction engineering and management: a critical review and future trends. Autom. Constr. 122, 103517. https://doi.org/10.1016/j.autcon.2020.103517.

[13] Schlüter, M., Lickert, H., Schweitzer, K., Bilge, P., Briese, C., Dietrich, F., Krüger, J., 2021. AI-enhanced identification, inspection and sorting for reverse logistics in remanufacturing. Procedia CIRP 98, 300–305.

[14] EMF, 2019. Artificial Intelligence and the circular economy: AI as a tool to accelerate the

Transition.

[15] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C. H. Davis, "Training deep convolutional neural networks for land-cover classification of high-resolution imagery," IEEE Geosci. Remote Sens. Lett., vol. 14, no. 4, pp. 549–553, 2017.

[16] A. Alzu'bi, A. Amira, and N. Ramzan, "Learning transfer using deep convolutional features for remote sensing image retrieval," IAENG Int. J. Comput. Sci., vol. 46, no. 4, pp. 1–8, 2019.

[17] F. Chen, R. Ren, T. Van de Voorde, W. Xu, G. Zhou, and Y. Zhou, "Fast automatic airport detection in remote sensing images using convolutional neural networks," Remote Sens., vol. 10, no. 3, p. 443, 2018.

[18] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," ISPRS J. Photogramm. Remote Sens., vol. 145, pp. 60–77, 2018.

[19] J. Morel, A. Bac, and T. Kanai, "Segmentation of unbalanced and in-homogeneous point clouds and its application to 3D scanned trees," Vis. Comput., vol. 36, no. 10, pp. 2419–2431, Oct. 2020.

[20] A. Ouahabi and A. Taleb-Ahmed, "Deep learning for real-time semantic segmentation: application in ultrasound imaging," Pattern Recogn. Lett., vol. 144, pp. 27–34, 2021.

[21] Z. Cheng, A. Qu, and X. He, "Contour-aware semantic segmentation network with spatial attention mechanism for medical image," Vis. Comput., vol. 22, pp. 1–4, Feb. 2021.

[22] L. Bragagnolo, R. V. da Silva, and J. M. V. Grzybowski, "Amazon forest cover change mapping based on semantic segmentation by U-nets," Ecol. Inform., vol. 62, p. 101279, 2021.

[23] N. Flood, F. Watson, and L. Collett, "Using a U-net convolutional neural network to map woody vegetation extent from high resolution satellite imagery across Queensland, Australia," Int. J. Appl. Earth Obs. Geoinf., vol. 82, p. 101897, 2019.

[24] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, and A. Rubtsov, "Comparison of different convolutional neural network architectures for satellite image segmentation," in Proc. 2018 23rd IEEE Conf. Open Innov. Assoc. (FRUCT), pp. 172–179, Nov. 2018.

[25] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet Classification with Deep Convolutional Neural Networks. ADV NEUR IN, pp. 1097–1105.

[26] Jean David Lau Hiu Hoong, Jérôme Lux, Pierre-Yves Mahieux, Philippe Turcry, Abdelkarim Aït-Mokhtar, Determination of the composition of recycled aggregates using a deep learning-based image analysis, Automation in Construction, Volume 116, 2020, 103204, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2020.103204.

[27] Yang, M., Thung, G., 2016. CS229 Project Report.

[28] Thung, G., Yang, M., 2019. Trashnet dataset. https://github.com/garythung/trashnet

[29] Mao, W., Chen, W., Wang, C., Lin, Y., 2021. Recycling waste classification using optimized convolutional neural network. resources, conservation and recycling 164, 105132. https://doi.org/10.1016/j.resconrec.2020.105132.

[30] Zhang, Q., Zhang, X., Mu, X., Wang, Z., Tian, R., Wang, X., Liu, X., 2021. Recyclable waste image recognition based on deep learning. resources, conservation and recycling 171, 105636. https://doi.org/10.1016/j.resconrec.2021.105636.

[31] Awe, O., Mengistu, R., Sreedhar, V., 2017. Smart trash net: waste localization and classification. arXiv preprint.

[32] Nissim, S.; Portnov, A.B. Identifying areas under potential risk of illegal construction and demolition waste dumping using GIS tools. Waste Manag. 2018, 75, 22–29.

[33] Zhang, F.; Ju, Y.; Dong, P.; Gonzalez, E.D.S. A fuzzy evaluation and selection of construction and demolition waste utilization modes in Xi'an, China. Waste Manag Res. 2020, 38, 792–801. [CrossRef]

[34] Yalan, L.; Yuhuan, R.; Chengjie, W.; Gaihua, W.; Huizhen, Z.; Yaobin, C. Study on monitoring of informal open-air solid waste dumps based on Beijing-1 images. J. Remote Sens. 2009, 13, 320–326.

[35] Wu, W.W.; Liu, J. The Application of Remote Sensing Technology on the Distribution Investigation of the Solid Waste in Beijing. Environ. Sanit. Eng. 2000, 8, 76–78.

[36] Kuritcyn, P.; Anding, K.; Linß, E.; Latyev, S.M. Increasing the Safety in Recycling of Construction and Demolition Waste by Using Supervised Machine Learning. J. Phys. Conf. Ser. 2015, 588, 012035. [CrossRef]

[37] Ku, Y.; Yang, J.; Fang, H. Researchers at Huaqiao University Release New Data on Robotics (Deep Learning of Grasping Detection for a Robot Used In Sorting Construction and Demolition Waste). J. Robot. Mach. Learn. 2020, 23, 84–95.

[38] Gu, X.; Gao, X.; Ma, H.; Shi, F.; Liu, X.; Cao, X. Comparison of Machine Learning Methods for Land Use/Land Cover Classification in the Complicated Terrain Regions. Remote Sens. Technol. Appl. 2019, 34, 57–67.

[39] Satellite imagery, 2023. https://www.airbus.com/en/space/earth-observation/satellite-imagery.

[40] MAXAR, 2023. https://www.maxar.com/products/satellite-imagery

[41] Sozzi, M. *et al.* (2021) 'Economic comparison of satellite, plane and UAV-acquired NDVI images for site-specific nitrogen application: Observations from Italy', *Agronomy*, 11(11), p. 2098. doi:10.3390/agronomy11112098.

[42] Zhou, L.; Luo, T.; Du, M.; Chen, Q.; Liu, Y.; Zhu, Y.; He, C.; Wang, S.; Yang, K. Machine Learning Comparison and Parameter Setting Methods for the Detection of Dump Sites for Construction and Demolition Waste Using the Google Earth Engine. *Remote Sens.* **2021**, *13*, 787. https://doi.org/10.3390/rs13040787

[43] Weisheng Lu, Junjie Chen, Fan Xue, Using computer vision to recognize composition of construction waste mixtures: A semantic segmentation approach, Resources, Conservation and Recycling, Volume 178, 2022, 106022, ISSN 0921-3449, https://doi.org/10.1016/j.resconrec.2021.106022.

[44] HKEPD, 2011. Construction Waste Statistics. https://www.epd.gov.hk/epd/misc/cdm/trip.htm

[45] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving Iterative Training with Mask Guidance for Interactive Segmentation," arXiv.org, Feb. 12, 2021. https://arxiv.org/abs/2102.06583

[46] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017).Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848.

[47] Ojekanmi, O., Idris S., October 2021. CARLA Image Semantic Segmentation with DeepLabV3+ (Version 3) [Source Code]. Retrieved from https://www.kaggle.com/code/oluwatobiojekanmi/carla-image-semantic-segmentation-with-deeplabv3/notebook

[48] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[49] Chollet, F., 2017. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.

[50] Chen, L.-.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation. Springer International Publishing, Cham, pp. 833–851.

[51] C. Zhang, L. Zhou, M. Du, K. Yang, and T. Luo, "A cross-channel multi-scale gated fusion network for recognizing construction and demolition waste from high-resolution remote sensing images," International Journal of remote sensing (Print), vol. 43, no. 12, pp. 4541–4568, Jun. 2022, doi: https://doi.org/10.1080/01431161.2022.2115864.

[52] Catalog, UP42, 2023. https://console.up42.com/catalog/new-order

[53] K. Malarvizhi, S. Vasantha Kumar, P. Porchelvan, Use of High Resolution Google Earth Satellite Imagery in Landuse Map Preparation for Urban Related Applications, Procedia Technology, Volume 24, 2016, Pages 1835-1842, ISSN 2212-0173, https://doi.org/10.1016/j.protcy.2016.05.231.

[54] Ehsan Haghighat, Ruben Juanes, SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, Computer Methods in Applied Mechanics and Engineering, Volume 373, 2021, 113552, ISSN 0045-7825, https://doi.org/10.1016/j.cma.2020.113552.

[55] S. Alexandrova, Z. Tatlock, and M. Cakmak, "RoboFlow: A flow-based visual programming language for mobile manipulation tasks," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 5537-5544, doi: 10.1109/ICRA.2015.7139973.

[56] S. G. E. Brucal, L. C. M. de Jesus, S. R. Peruda, L. A. Samaniego and E. D. Yong, "Development of Tomato Leaf Disease Detection using YoloV8 Model via RoboFlow 2.0," 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), Nara, Japan, 2023, pp. 692-694, doi: 10.1109/GCCE59613.2023.10315251.

[57] G. Vimalambikaipakan, C. Amarasinghe, T. Rajapaksha, T. Thayananthan and J. Mariyathas, "Traffic Sign Recognition and Auditory Alert System for Sri Lankan Drivers Using Deep-Learning," 2024 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka, 2024, pp. 1-5, doi: 10.1109/SCSE61872.2024.10550615.

[58] Chunshan Wang, Pengfei Du, Huarui Wu, Jiuxi Li, Chunjiang Zhao, Huaji Zhu, A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net, Computers and Electronics in Agriculture, Volume 189, 2021, 106373, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2021.106373.

[59] Yanchao Wang, Mengyuan Chu, Xi Kang, Gang Liu, A deep learning approach combining DeepLabV3+ and improved YOLOv5 to detect dairy cow mastitis, Computers and Electronics in Agriculture, Volume 216, 2024, 108507, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2023.108507.

[60] Francisco López de la Rosa, José L. Gómez-Sirvent, Roberto Sánchez-Reolid, Rafael Morales, Antonio Fernández-Caballero, Geometric transformation-based data augmentation on defect classification of segmented images of semiconductor materials using a ResNet50 convolutional neural network, Expert Systems with Applications, Volume 206, 2022, 117731, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.117731.

[61] Elpeltagy, M.; Sallam, H. Automatic prediction of COVID-19 from chest images using modified ResNet50. Multimed. Tools Appl. 2021, 80, 26451−26463.

[62] Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. (2018) "MobileNetV2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition: 4510–4520, doi: 10.1109/CVPR.2018.00474.

[63] Eyiokur, F. I., H. K. Ekenel, and A. Waibel. (2021) "A computer vision system to help prevent the transmission of COVID-19." arXiv Prepr. arXiv2103.08773 [Online]. Available: http://arxiv.org/abs/2103.08773.

[64] Van Beers, F.; Lindström, A.; Okafor, E.; Wiering, M. Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods; SCITEPRESS- Science and Technology Publications, 2019; pp 438−445.

[65] Schneider, S.; Taylor, G. W.; Kremer, S. C. Deep Learning Object Detection Methods for Ecological Camera Trap Data. arXiv 2018, DOI: 10.48550/arXiv.1803.10842.

[66] Prajapati, S. A.; Nagaraj, R.; Mitra, S. Classification of dental diseases using CNN and transfer learning. 2017 5th International Symposium on Computational and Business Intelligence (ISCBI) 2017, 70−74.

[67] Wurm, M.; Stark, T.; Zhu, X. X.; Weigand, M.; Taubenböck, H. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 2019, 150, 59−69.

[68] Lu, H.; et al. Landslides Information Extraction Using ObjectOriented Image Analysis Paradigm Based on Deep Learning and Transfer Learning. Remote Sens. 2020, 12, 752.

[69] Jiang, Z.; Zheng, T.; Bergin, M.; Carlson, D. Improving spatial variation of ground-level PM2.5 prediction with contrastive learning from satellite imagery. Science of Remote Sensing 2022, 5, 100052.