REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

BUILDING DAMAGE ASSESSMENT FROM POST-EARTHQUAKE DRONE IMAGES USING DEEP LEARNING BASED SEGMENTATION METHODS: THE CASE OF FEBRUARY 6, 2023 TÜRKIYE EARTHQUAKES

Beyza GÜRER

MASTER OF SCIENCE THESIS

Department of Computer Engineering

Computer Engineering Program

Supervisor

Prof. Dr. M. Elif KARSLIGİL

July, 2024

REPUBLIC OF TÜRKİYE

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

BUILDING DAMAGE ASSESSMENT FROM POST-EARTHQUAKE DRONE IMAGES USING DEEP LEARNING BASED SEGMENTATION METHODS: THE CASE OF FEBRUARY 6, 2023 TÜRKIYE EARTHQUAKES

A thesis submitted by Beyza GÜRER in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE is approved by the committee on 22.07.2024 in Department of Computer Engineering, Computer Engineering Program.

Prof. Dr. M. Elif KARSLIGİL
Yıldız Technical University
Supervisor

Approved By the Examining Committee

Prof. Dr. Mine Elif KARSLIGİL, Supervisor
Yildiz Technical University _____

Prof. Dr. Gökhan BİLGİN, Member
Yildiz Technical University _____

Prof. Dr. Yusuf Sinan AKGÜL, Member
Gebze Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of "Building Damage Assessment From Post-Earthquake Drone Images Using Deep Learning Based Segmentation Methods: The Case Of February 6, 2023 Türkiye Earthquakes" supervised by my supervisor, Prof. Dr. M. Elif KARSLIGİL. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Beyza GÜRER

This work was supported by the the project no “FBA-2024-6117 of YTU Coordinatorship of Scientific Research Projects”

Dedicated to my family

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Prof. Dr. M. Elif KARSLIGİL who has been encouraging and supportive in all aspects throughout the course of this research. Your guidance, expertise and invaluable feedback have been a constant source of inspiration. I deeply appreciate your mentorship.

I extend my sincere thanks to my sister Dilara Zeynep Gürer, for always listening to my concerns both research wise and mentally.

My heartfelt thanks go to my cousins, especially Fatma Nurdem Çelen and Züleyha Gül for the constant support and advice they provided throughout this research.

To my family, thank you for your unconditional love and support. To my parents, Nihat GÜRER and Hatice GÜRER and my little sister Ecrin Mina Gürer; your support and belief in have been my greatest motivation.

I also want to acknowledge TÜBİTAK for support provided by TÜBİTAK Bilim İnsanı Destek Programları Başkanlığı (BİDEB) 2210 National MSc Scholarship Program.

This thesis is dedicated to all those who have supported me in one way or another during this academic journey. Thank you all.

Beyza GÜRER

TABLE OF CONTENTS

LIST OF ABBREVIATIONS viii
LIST OF FIGURES ix
LIST OF TABLES xi
ABSTRACT xii
ÖZET xiv
1 INTRODUCTION 1
1.1 Introduction 1
1.2 Related Work 4
2 METHOD 9
2.1 Dataset Preparation for Damaged Building Detection 10
2.2 Damaged Building Detection with Image Segmentation 15
2.2.1 Semantic Segmentation 16
2.2.2 Instance Segmentation 20
2.3 Performance Metrics 23
2.3.1 Pixel Accuracy 23
2.3.2 Intersection Over Union 24
2.3.3 Average Precision 25
2.4 Combining Semantic Segmentation and Instance Segmentation for Enhancing Detection of Damaged Buildings 26
3 EXPERIMENTS 29
3.1 Data Splitting 30
3.2 Results for Damaged Building Detection with Semantic Segmentation 31
3.2.1 Results for Damaged Building Detection with U-Net 31
3.2.2 Results for Damaged Building Detection with SegFormer 32
3.2.3 Results for Damaged Building Detection with OneFormer 33
3.2.4 Comparing Semantic Segmentation Models for Damaged Building Detection 34
3.2.5 Making The Best Semantic Segmentation Model Better 35
3.3 Results for Damaged Building Detection with Instance Segmentation 44
3.3.1 Damaged Building Detection Mask R-CNN 44

3.3.2 Damaged Building Detection OneFormer 46
3.3.3 Damaged Building Detection BoundaryFormer 47
3.3.4 Comparing Models 48
3.3.5 Making The Best Model Better 49
3.4 Comparing Semantic and Instance Segmentation Best Models For Damaged Building Detection 59
3.5 Combining Semantic and Instance Segmentation Models for Enhancing Detection of Damaged Buildings 61
3.6 Comparing Model Results with Human Inference 64
4 CONCLUSION **66**
4.1 Conclusion 66
REFERENCES **69**
PUBLICATIONS FROM THE THESIS **74**

LIST OF ABBREVIATIONS

AP	Average Precision
ASR	Assessment, Search and Rescue
BBox	Bounding Box
CNN	Convolutional Neural Networks
DL	Deep Learning
FCN	Fully Convolutional Networks
FN	False Negative
FP	False Positive
GT	Ground Truth
IOU	Intersection Over Union
mIOU	Mean Intersection Over Union
ML	Machine Learning
MLP	Multilayer Perceptron
NLP	Natural Language Processing
RoI	Region of Interest
Segm	Segmentation
TN	True Negative
TP	True Positive

LIST OF FIGURES

Figure 2.1 Damaged Building Detection System Block Diagram **10**
Figure 2.2 An example drone image captured from Hatay after the earthquake .. **12**
Figure 2.3 Labeled drone image from created dataset **14**
Figure 2.4 U-Net architecture taken from original paper **17**
Figure 2.5 SegFormer architecture taken from original paper **18**
Figure 2.6 OneFormer architecture from original paper **19**
Figure 2.7 Mask R-CNN framework from original paper **21**
Figure 2.8 Architecture of BoundaryFormer from original paper **22**
Figure 2.9 IOU = Intersection / Union visualized **24**
Figure 2.10 Effect of threshold on calculation of AP **25**
Figure 2.11 Flowchart for combining semantic and instance results **27**
Figure 3.1 Data split for supervised deep learning models **30**
Figure 3.2 From left to right, original image, true mask and U-Net model output **32**
Figure 3.3 From left to right, original image, true mask and SegFormer model output **33**
Figure 3.4 From left to right, original image, true mask and OneFormer model output **34**
Figure 3.5 Loss graph for SegFormer model **36**
Figure 3.6 Accuracy graph for SegFormer model **37**
Figure 3.7 Original image, ground truth labels and Segformer model output **38**
Figure 3.8 Misclassified building from SegFormer inference **39**
Figure 3.9 Minor and No Damage labels from SegFormer inference **39**
Figure 3.10 Output examples from Segformer model. From left to right; original image, true masks, SegFormer output **40**
Figure 3.11 Original image, ground truth and model output for 2-class **41**
Figure 3.12 Output examples from SegFormer model with 2 classes. From left to right; original image, true masks, SegFormer output **42**
Figure 3.13 Original image, ground truth and model output 1-class **43**
Figure 3.14 Output examples from SegFormer model with 1 class. From left to right; original image, true masks, SegFormer output **44**
Figure 3.15 From left to right: Ground truth, Mask R-CNN inference result **45**

Figure 3.16 From left to right: Ground truth, OneFormer instance segmentation inference result **47**

Figure 3.17 From left to right: Ground truth, BoundaryFormer inference result . **48**

Figure 3.18 Loss graph for instance segmentation model BoundaryFormer. **50**

Figure 3.19 Ground truth labels for instance segmentation **51**

Figure 3.20 BoundaryFormer model inference outputs on test image **52**

Figure 3.21 BoundaryFormer model missing buildings on inference **53**

Figure 3.22 Examples from model From left to right: True labels, BoundaryFormer outputs **54**

Figure 3.23 2-class ground truth labels in test image **56**

Figure 3.24 2-class BoundaryFormer predictions **56**

Figure 3.25 Examples from mode with “Stable” and “Damaged” classes. From left to right: True labels, BoundaryFormer output **57**

Figure 3.26 1-class ground truth labels and model output **58**

Figure 3.27 Examples from model with “Destroyed” class only. From left to right: True labels, BoundaryFormer outputs **59**

Figure 3.28 Comparison of semantic and instance segmentation outputs **60**

Figure 3.29 The resulting image for enhancement method **62**

Figure 3.30 Successful building example for proposed method **62**

Figure 3.31 Failed building example for proposed method **63**

LIST OF TABLES

Table 1.1 Earthquake Magnitude Scale **2**
Table 1.2 ASR Levels **3**
Table 2.1 Classes in dataset with examples **13**
Table 2.2 Building type distribution **15**
Table 2.3 Semantic segmentation model outputs **20**
Table 2.4 Instance segmentation model outputs **23**
Table 2.5 Confusion matrix **24**
Table 3.1 Building count distribution per class **31**
Table 3.2 U-Net pixel accuracy and IOU results **31**
Table 3.3 SegFormer pixel accuracy and IOU results **32**
Table 3.4 OneFormer pixel accuracy and IOU results **33**
Table 3.5 Comparison of semantic segmentation models **34**
Table 3.6 SegFormer best pixel accuracy and IOU results **38**
Table 3.7 SegFormer pixel accuracy and IOU results for 2 classes **41**
Table 3.8 SegFormer pixel accuracy and IOU results for 1 class **43**
Table 3.9 Mask R-CNN bounding box and segmentation results **45**
Table 3.10 Mask R-CNN bounding box and segmentation results per class **45**
Table 3.11 OneFormer bounding box and segmentation results **46**
Table 3.12 OneFormer bounding box and segmentation results per class. **46**
Table 3.13 BoundaryFormer bounding box and segmentation results. **47**
Table 3.14 BoundaryFormer bounding box and segmentation results per class **48**
Table 3.15 Models' AP results **49**
Table 3.16 Best BoundaryFormer bounding box and segmentation test results **50**
Table 3.17 Best BoundaryFormer bbox and segmentation test results per class **51**
Table 3.18 BoundaryFormer bounding box and segmentation results for 2 class labels **55**
Table 3.19 BoundaryFormer model results per class for 2 class labels **55**
Table 3.20 BoundaryFormer bounding box and segmentation results for "Damaged" class only **58**
Table 3.21 Enhanced method accuracy and IOU results **63**
Table 3.22 Survey Examples for Human – Model Inference Comparison **64**

ABSTRACT

Building Damage Assessment From Post-Earthquake Drone Images Using Deep Learning Based Segmentation Methods: The Case Of February 6, 2023 Türkiye Earthquakes

Beyza GÜRER

Department of Computer Engineering

Master of Science Thesis

Supervisor: Prof. Dr. M. Elif KARSLIGİL

The disastrous nature of earthquakes necessitates rapid assessment of damage for effective disaster response. With this goal, detection of damaged buildings can help assess the damage magnitude of large areas. This thesis presents deep learning based detection of damaged buildings after earthquakes using drone images, leveraging advanced techniques: semantic and instance segmentation.

To perform damage detection, firstly a dataset is created comprising drone images captured from Hatay after 6th February Earthquake. The buildings in high quality drone images are labeled into 4 levels of damage: “No Damage”, “Minor Damage”, “Major Damage”, “Destroyed”. The dataset contains segmentation masks and bounding boxes for each building.

To detect damaged buildings, image segmentation is applied. Firstly, three different models: U-Net, SegFormer, OneFormer are applied for semantic segmentation.

SegFormer performed the best and obtained the best result of mIOU with a value of 0.51 and a PA of 0.72. To improve results, classes are reduced to two, representing the "stable" and the "damaged" buildings. SegFormer model achieved 0.68 mIOU and 0.80 PA. When only the damaged buildings were segmented, the model achieved 0.73 mIOU and 0.84 PA.

The images are then segmented using the Mask R-CNN, OneFormer, BoundaryFormer instance segmentation. BoundaryFormer model achieved the best result with segmentation and bounding box AP of 0.17. For two class case, same model got 0.25 segmentation AP and 0.24 bounding box AP. Lastly for only damaged class, the model got segmentation and bounding box AP of 0.22.

After model training and evaluation, the results for semantic and instance segmentation are compared. Finally, using comparison results, a method to enhance inference results using both semantic and instance outputs is developed. This method combines the inference outputs of both semantic and instance segmentation and makes use of the advantages of each model.

Keywords: Damaged building detection, drone images, earthquake, deep learning, image segmentation.

ÖZET

Derin Öğrenme Tabanlı Bölütleme Yöntemleri Kullanılarak Deprem Sonrası Drone Görüntülerinden Bina Hasar Tespiti: 6 Şubat 2023 Türkiye Depremleri Örneği

Beyza GÜRER

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. M. Elif KARSLIGİL

Depremlerin yıkıcı etkisi, efektif afet müdahalesi için hasarın hızlı bir şekilde değerlendirilmesini gerektirir. Bu amaçla, hasarlı binaların tespiti, geniş alanların hasar büyüklüğünün değerlendirilmesine yardımcı olabilir. Bu tez, drone görüntüleri üzerinde deprem sonrası hasarlı bina tespiti için derin öğrenme yöntemleri kullanarak gelişmiş teknikler olan, semantik ve örnek segmentasyonundan yararlanmaktadır.

Hasar tespitini gerçekleştirmek için öncelikle 6 Şubat Depremi sonrasında Hatay'dan çekilen drone görüntülerinden oluşan bir veri kümesi oluşturulmuştur. Yüksek kaliteli drone görüntülerindeki binalar 4 hasar seviyesine göre etiketlenmiştir: “Hasarsız”, ‘Az Hasar’, ‘Çok Hasar’, ‘Yıkılmış’. Veri kümesi, her bina için segmentasyon maskeleri ve sınırlayıcı kutular içermektedir.

İkinci olarak, oluşturulan veri kümesinden yararlanarak, görüntülerdeki hasarlı binaları tespit etmek için denetimli derin öğrenme yöntemleri gerçekleştirilmiştir. Bunun için hasarlı bölgeleri çıkarmak için görüntü segmentasyonu seçilmiş, öncelikle U-Net, SegFormer ve OneFormer olmak üzere üç farklı model kullanılarak semantik segmentasyon yöntemi uygulanmıştır. Bunlar arasında SegFormer en iyi performansı göstermiş ve 0.51 ile mIOU ve 0.72 piksel doğruluğu ile en iyi sonucu almıştır. Sonuçları iyileştirmek için, hasar derecelerinin sınıfları "Stabil" ve "Hasarlı" sınıfları temsil eden ikiye indirilmiştir. SegFormer modeli 0,68 mIOU ve 0,80 piksel doğruluğu elde etmiştir. Sınıf sayısı sadece hasarlı binalar için bire düşürüldüğünde, model 0,73 mIOU ve 0,84 piksel doğruluğu elde etmiştir.

Örnek segmentasyonu için Mask R-CNN, OneFormer ve BoundaryFormer modelleri hasarlı bina tespiti yapmak üzere eğitilmiştir. BoundaryFormer modeli 0.17 segmentasyon ve sınırlayıcı kutu AP'si ile en iyi sonucu almıştır. İki sınıflı durum için aynı model 0.25 segmentasyon AP'si ve 0.24 sınırlayıcı kutu AP'si elde etmiştir. Son olarak, bir sınıf için model 0.22 segmentasyon ve sınırlayıcı kutu AP'si elde etmiştir.

Model eğitimi ve değerlendirmesi gerçekleştirildikten sonra anlamsal ve örnek bölütleme için sonuçlar çıktılar incelenerek karşılaştırılmıştır. Son olarak, çıkarım sonuçlarını iyileştirmek için karşılaştırma sonuçları kullanılarak hem anlamsal hem de örnek bölütleme çıktılarını birleştiren bir yöntem geliştirilmiştir.

Anahtar Kelimeler: Hasarlı bina tespiti, drone görüntüleri, deprem, derin öğrenme, görüntü bölütleme.

1
INTRODUCTION

1.1 Introduction

Natural disasters are events that cause major damage to society. These events include earthquakes, floods, tsunamis, tornados and so on. Common denominators of these disasters are that they are mostly unpredictable and destructive. They have a significantly detrimental impact on various aspects of life, including social, economic, psychological, and ecological factors. In the worst-case scenario, natural disaster can even take the lives of numerous people. Not only that these events have instant effect but also, they have devastating and permanent damage in the long term. To exemplify, in 2011 a tsunami in Japan disabled power supplies of nuclear cooling systems causing evacuation of nearby residents and disruption of daily life [1].

Earthquakes are one of the most harming disasters as they affect large areas, cause structural damage and loss of life [2]. According to National Centers for Environmental Information of the National Oceanic and Atmospheric Administration, between dates 2000 and 2018; around 570000 people lost their lives and around 1 million people were injured worldwide due to earthquakes [3]. Earthquakes and their damage degrees are given in Table 1, according to Michigan Technical University [4]. It can be seen from the Table 1.1 that earthquakes with a magnitude of 5.5 or greater cause damage to buildings. The damage caused by earthquakes increases with their magnitude. Even though the estimated number each year for larger magnitude earthquakes is rather small, the same cannot be said for the harm they cause.

Table 1.1 Earthquake Magnitude Scale [4]

Magnitude	Earthquake Effects	Estimated Number Each Year
2.5 or less	**Usually not felt but can be recorded by seismograph.**	**Millions**
2.5 to 5.4	**Often felt, but only causes minor damage.**	**500.000**
5.5 to 6.0	**Slight damage to buildings and other structures.**	**350**
6.1 to 6.9	**May cause a lot of damage in very populated areas.**	**100**
7.0 to 7.9	**Major earthquake. Serious damage.**	**10 - 15**
8.0 or to greater	**Great earthquake. Can totally destroy communities near the epicenter.**	**One every year or two**

As seen from the Table 1.1, earthquakes cause structural damage to buildings. It is crucial to get detailed information about the extent and level of damage of the buildings for search and rescue operations [5]. Especially in countries like Türkiye, in which the 90% of the buildings are subject to the risk of earthquake [6], detection and assessment of building damage is an essential part of the resource management.

Considering the fact that humans spend most of their time in buildings like houses, schools, factories and so on, it is inevitable for people to get caught in buildings once the earthquake happens. After major earthquakes, people might not have the opportunity to leave the building and get stuck in buildings. In cases like this, it is

of vital importance for the help to be as fast as possible. For help to arrive, the detection of damaged buildings is the first step.

To ensure the success of rescue and relief operations, it is vital to follow controlled and organized steps. There are various methods available to effectively carry out these critical missions. According to International Search and Rescue Advisory Group (INSARAG) guidelines [7], Assessment, Search and Rescue (ASR) operations have different levels. The levels are given in Table 2.1. ASR Level 1 is named as Wide-Area Assessment and is used to collect general information for sectorization. In this level, it is aimed to gather information about levels of damage in regions and sectorize them accordingly. This step ensures for search and rescue teams to be distributed according to the need in each region. This step also shows that assessment of damage is a crucial first step for search and rescue operations.

Following standardized guidelines and firstly gathering information about damage levels help reduce harm caused by earthquakes as it is significantly more cost efficient to manage and distribute resources with information. It is important to acknowledge the short-term benefits of efficient and organized operations, as they can greatly improve the speed and effectiveness of search and rescue efforts. However, it is also crucial to recognize the long-term advantages of organized search and rescue operations in reducing overall damage. By streamlining processes and response times, recovery efforts can be expedited, ultimately minimizing the impact of disasters.

Table 1.2 ASR Levels

ASR Level	Title	Definition and Purpose	Responsible Institution/Organization
1	Wide Area Assessment	Conducting initial surveys aimed at sectorization of affected areas	Local Disaster and Emergency Authority (AFAD), United Nations Disaster Assessment and Coordination Team (UNDAC)
2	Sector Assessment	Preliminary surveys conducted in definite or potential working areas within the sectorized areas. The purpose of sector assessment is to…	Urban search and rescue teams assigned to the relevant sector

Table 1.2 ASR Levels (continued…)

		… create an action plan and determine priority assignments.	
3	Rapid Search and Rescue	This is a working method that aims to maximize the rescue level in the initial periods of the disaster. Alive victim rescue operations can be completed within 12 hours and are expected to be performed in a shorter time frame compared to technical operations.	Urban search and rescue teams assigned to the relevant sector
4	Full Search and Rescue	This is a working method that involves deep and repeated searches and technical operations to maximize the rescue level. Operations that require more than 12 hours due to the complexity and size of the building are covered under this level.	Urban search and rescue teams assigned to the relevant sector
5	Full Search and Rescue Recovery	This level involves activities related to the retrieval and extraction of deceased victims.	Local Disaster and Emergency Authority (AFAD)

In this thesis, a DL based system for detection and classification of damaged buildings from drone images is developed. Using drone images taken in Hatay, after the 6th February 2023 Earthquake, firstly a dataset is created. The buildings are labeled into 4 classes and segmented. Using the newly created dataset, 3 models with different architectures for both semantic segmentation and instance segmentation models are trained to perform damaged building detection. The performance of both methods is then compared to determine benefits of each one for damaged building detection. Finally, a novel method is proposed to exploit the advantages of both methods.

1.2 Related Work

Considering the global nature of earthquakes and the large area of effect their damage has; damage assessment after earthquakes is one of the popular research areas. In particular, 2023 Turkey Earthquake drew global attention [8], and it has once again come to attention that rapid damage assessment has significant effect on resource management. Following this, more researchers are leaning to this field. The studies in this area have multiple approaches for handling the detection and assessment of damage.

Different approaches can be categorized considering the source of data. One of the data sources for damage assessment after earthquakes is seismic data. Seismic data are collected from seismographs, tools that record the difference in motion on earth. Seismic signals, ground motion etc. can be extracted from seismic data. Various research is conducted using this type of data [9] [10] [11]. While this type of data is viable and used, it is rather hard to understand for researchers who do not have knowledge in seismic data.

Images are another type of data which help detecting damaged buildings. Kyrkou et al. [12] created a dataset for damaged structures caused by multiple types of disaster collected from the internet.

After earthquakes, the speed of search and rescue operations is vital. However, when the damage area is large, it is difficult to determine where the help is needed. For cases like this remote sensing image is advantageous as it can capture large areas rapidly. Remote sensing data is one of the more popular data types for assessing damage as they cover large areas, are easy to obtain from public sources and can be acquired from multiple devices. Some public datasets created from remote sensing data is listed here [13] [14].

RescueNet [15] is another dataset including drone images with buildings. It includes 10903 buildings with multiple damage degrees. It also includes other objects with segmentation. The images are taken after a hurricane.

Earthquake damage assessment from images has been a research area for a long time as it is a fundamental part of search and rescue operations. Different types of assessment methods have been carried out to determine the level of damage. For example, Sirmacek et al. [16] used shadow information extracted with greyscale histogram to segment rooftops based on color invariants. In another work, Yan et al. [17] used improved fast scanning algorithm and fuzzy logic to segment samples from remote sensing imagery after earthquake.

In recent years, with how much easier it has become to collect huge amounts of data, Machine Learning (ML) methods have been rapidly improving for different image tasks like object detection, image segmentation etc. Damaged building assessment is no exception to this. Using public remote sensing data, many ML

studies have been done to detect and assess damaged buildings after earthquakes. [18] [19] [20] [21]

While classical ML methods can be used to reach decent results, development of DL caused many researchers to prefer DL methods in their studies. Deep learning for image tasks does not require manual feature extraction unlike classical ML methods. As neural network architectures can extract useful information from data, it has become rather easy to feed the data to the model and get results. Especially with the development of Convolutional Neural Networks [22], DL models can extract and optimize filters that represent the patterns of the images.

Using DL methods some of the studies use both pre- and post-earthquake images to assess the damage. [23] [24] [25] [26] This provides the DL models to be able to detect changes and determine damages. On the other hand, this type of data is harder to find as it is a difficult task to find and match the buildings to their pre and post-earthquake versions caused from different angles, satellite revisit cycles etc.

For DL methods using remote sensing post-earthquake images, many studies have been conducted [27] [28] [29] [30]. Other studies can be split into categories according to their methods for damage assessment. These can be examined under the titles of “image classification” and “object detection” and “image segmentation”.

For image classification, one image is classified as one label. This means each image can only have one label and include one type of damage. Studies under this title, either patch the large images or use images that contain one object. One example of this is Duarte et al. [31] proposed three different multi-resolution (images with different spatial resolutions i.e. satellite, UAV) feature fusion methods for CNNs to classify building damages. A dataset with 2321 satellite images, 5455 UAV images and 5192 aerial images was created. Overall, the best model had an accuracy value of 95.4%.

Object detection on the other hand, processes an image with multiples objects in it. This method tries to detect all objects in the image and classify them. This ensures the use of large images with multiple buildings of different types. Wang et al. [32] used a modified YOLOv5 model to detect damaged building regions in real time. Using feature fusion, the model can take different resolutions as input. The dataset

used contains 3738 UAV images taken from China. Results Show that the model has 95.34% F1-score calculated from pixel accuracy. As the model will be embedded on drones, the authors have also considered the performance of inference.

Segmentation task takes an image as an input and segments the object and stuff in the image. This method is studied under different types of segmentation. Instance segmentation on the other hand is a mix of object detection and semantic segmentation. This enables for objects in the images to be detected and segmented individually. Zhan et al. [33] used a modified Mask R-CNN [34] to segment damaged buildings under instance segmentation title. Mask R-CNNs are one of the most used methods for instance segmentation. The dataset used has 667 images with 500 x 500 resolution. The modified part consists of Feature Pyramid Network and Online Hard Example Mining. The best model proposed got mean average precision of 0.365 for bounding box and 0.373 for segmentation.

Semantic segmentation labels each pixel in the image resulting in an output image in which each pixel has a label. Song et al. [35] used DeepLabV2 to segment damaged regions. Afterwards, to expand on the success of segmentation, they used simple linear iterative clustering. After this step, the results are merged, and the background noise are eliminated. The dataset used contains aerial imagery taken after an earthquake of magnitude 7.9 in China. Results show better performance to others in both convergent speed and accuracy.

In another work [36] satellite images containing damaged buildings induced from tornados are studied. The study used U-Net [37] for segmenting collapsed and non-collapsed buildings and got the mIOU result of 68.65.

While CNN based algorithms show decent results and are used commonly, with the development of Visual Transformers [38], a different type of deep learning method has become popular. Xie et al. [39] used satellite images after earthquake to segment damaged buildings. The dataset used has 3094 images taken from Haiti after 2010 earthquake. The study focuses on heterogeneity characteristics of damaged buildings after earthquake as they show large spectrum difference, uneven texture features and unusual morphological characteristics. Modifying vision

transformers, they added local global context attention mode for feature extraction and a feature fusion module with self-attention. The model has 62.43% IOU value.

In another Vision Transformer based study, Li et al. [40] used SAR imagery to develop a semantic segmentation model for damaged buildings detection called DDFormer. They emphasized damage differences with adaptive frequency and spatial modules. As a result, DDformer got mIOU of 81.81%.

In this work, firstly the buildings in drone images collected from Hatay after 2023 Kahramanmaraş Earthquake are labeled according to their damage levels and a dataset is created. Then, the created dataset is tested through multiple segmentation DL methods to assess damaged buildings. The DL methods include CNN based algorithms, but also recent Vision Transformer based methods. Through this work, both semantic and instance segmentation methods are applied, and the best performing model is chosen. On the best performing model, data augmentation methods are used to get best results. In this work, it is aimed to acquire a DL model that takes input as an image and outputs the buildings with their damage levels.

The contributions of this work can be listed as;

- A comprehensive damaged building dataset consisting of 453 images and 7293 buildings is created using drone images captured after 6th February Kahramanmaraş Earthquake. The buildings in drone images are labeled according to 4 damage classes: “No Damage”, “Minor Damage”, “Major Damage” and “Destroyed”.
- The performances of the selected instance and semantic segmentation models trained on the prepared dataset were compared with each other, considering their advantages and disadvantages for detecting damaged buildings.

The semantic segmentation approach shows better results for damage detection as it labels images pixel by pixel, whereas the instance segmentation provides more information about the number of buildings. To improve the detection of damaged buildings from drone images, a novel method is proposed that combines the advantages of both semantic and instance segmentation models.

2
METHOD

In this thesis, a deep learning-based building damage detection system is designed and developed using drone images. To this end, a comprehensive dataset consisting of buildings with 4 different types of damage is developed. After thisTo perform damaged building detection; a supervised deep learning method, image segmentation is chosen as it is a method that enables segmenting the buildings with multiple damage degrees. Two of the basic approaches in image segmentation, semantic and instance segmentation are performed on the prepared dataset. Considering two foundational architectures Vision Transformers and CNN's 3 different models are chosen. For semantic segmentation, one of the models chosen is U-Net [36] as U-Net is a well performing CNN based model in semantic segmentation especially in detailed images. Secondly, SegFormer [41] is chosen as it is a Transformer based model that is both lightweight and it outperforms its predecessors. For instance segmentation, Mask R-CNN [33] is chosen as CNN based model because of its performance compared to other instance segmentation CNN based methods. For transformer-based model BoundaryFormer is chosen because of its nature that uses polygons for segmentation. As the buildings in dataset are segmented as polygons, BoundaryFormer [42] is used. Finally, OneFormer [43] is chosen for both instance and semantic segmentation since it gives state of the art results and performs both semantic and instance segmentation.

Choosing the best performing models, semantic and instance segmentation methods are compared. Lastly, using both instance and semantic segmentation results, a method to enhance segmentation results for damaged building detection is proposed.

To perform damage assessment from drone images on damaged buildings using deep learning, the steps are followed given in the Figure 2.1. Firstly the dataset is

prepared with images captured with drones after the earthquake in Hatay. After this, under preprocessing steps, the images are resized and sampled. The processed images are then labeled according to their damage degrees. Finally, a dataset is created using the labeled images. In the end "building damage drone image dataset" containing buildings with four different damage degrees is ready for use for deep learning models.

In deep learning models section, the newly created dataset is used to train and test deep learning models.

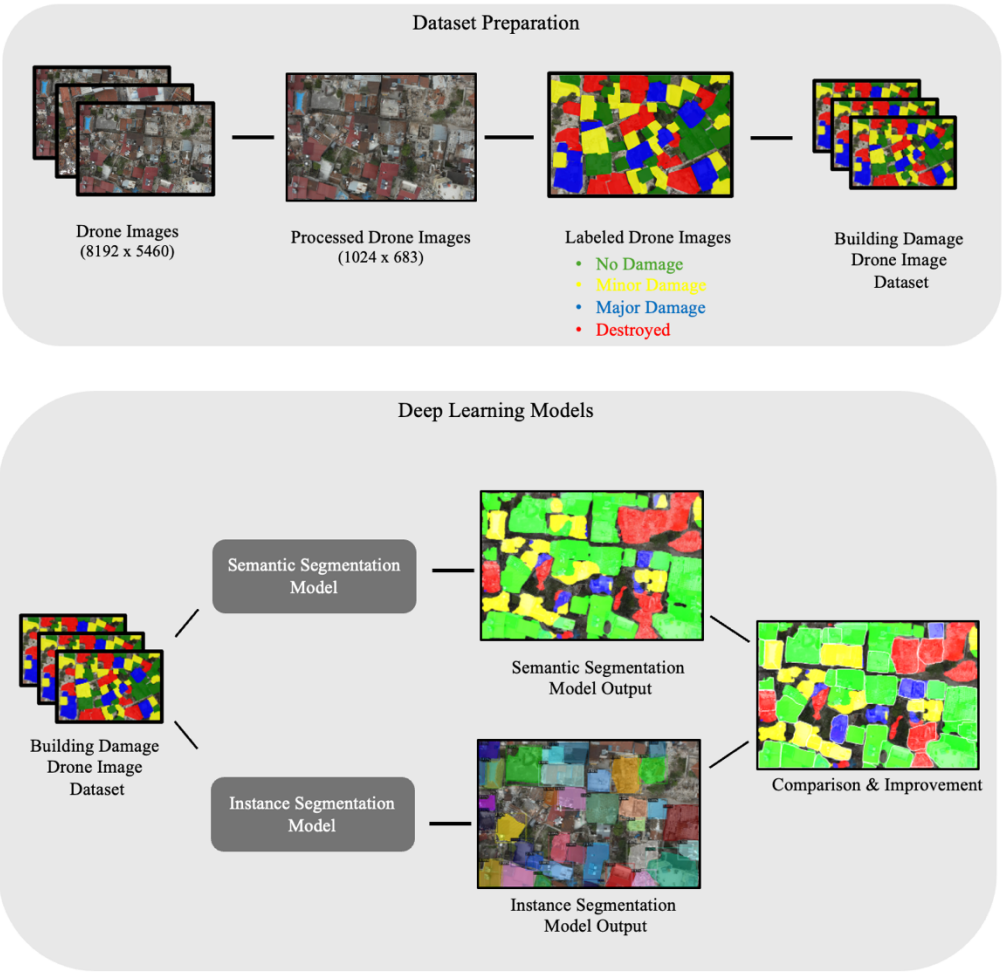


Figure 2.1 Damaged Building Detection System Block Diagram

2.1 Dataset Preparation for Damaged Building Detection

In this work, drone images are used for damaged building detection. Drone images contain detailed high-quality shots of the buildings making it easier for models to determine underlying patterns.

Data processing is an important step in all deep learning methods as they are highly dependent on the underlying data [44]. Same for damaged building detection, rather than using raw data from drone images, preprocessing steps are performed. These steps include resizing images and sampling them to make the dataset of more high quality.

Deep learning models perform feature extraction automatically from the data as opposed to older methods where feature engineering must be performed by data scientists. For damaged building detection the methods used in this project are semantic segmentation and instance segmentation. As these models are also deep learning models, the success is highly dependent on data. Thus, it is vital for data to be consistent, accurate and complete. With this goal in mind, firstly a dataset is created from drone images that include buildings with different damage levels. These damage levels correspond to the degree of the damage on buildings as they provide information about the size of the overall damage.

For deep learning models, generalization is an important factor for a model to be considered successful. Since it is mostly very difficult to feed every type of data into models, it is aimed for models to perform well on unseen data as well. To give an example, for building detection from drone images, even though multiple examples of damaged buildings exist in dataset there still might exist an unseen type of building which the model has never seen before. To adapt to this, datasets are split into multiple subsets to evaluate performance. Thus, the dataset created is then split into train, validation, and test sets for model training steps.

Multiple methods for damaged detection are possible. Image classification on patches from large images, object detection using only bounding boxes are some of the existing methods. To maximize both information about overall damage and success of models, segmentation is chosen for detecting damaged buildings. After preparation of the dataset, multiple DL segmentation models are compiled to extract the buildings. The DL models compiled are trained with train set created from dataset. During training each epoch is also validated on the validation set. Validation step ensures best hyperparameters and model parameters are chosen according to validation results. This means that the buildings the model trained with are separated from the buildings that the model will encounter through testing. This ensures that the model will not overfit and will not perform worse in real situations

when it encounters building types that are not similar to ones in dataset. Finally chosen models from validation results are trained again with combined train and validation set. Last models are tested with test set and compared to each other to examine the pros and cons of two segmentation methods.

For damaged building detection, it is no different that the methods require data. Data in this work is collected from Hatay after February 6th Kahramanmaraş Earthquake using a DJI ZenmuseP1 drone. The images taken have resolution of 8192x5460 and contain buildings from different altitudes. An example of a drone image is shown in Figure 2.2.



Figure 2.2 An example drone image captured from Hatay after the earthquake

To create a dataset, the images must be labeled according to the needs of the problem. As the main goal is to assess damage levels of buildings in images, the buildings must be labeled according to their damage levels as seen from the images.

Building damage levels are difficult to fit into a standard as it is especially hard to detect small differences in remote sensing images [45]. Thus, a damage scale is created from the images according to how distinct the damage degrees are. With this goal in mind, the building damage levels are split into 4 classes.

The classes are given in Table 2.1. As it is shown in the table, first class is “No Damage” class and it represents the buildings with no damage that can be detected from drone imagery. Second class is “Minor Damage”, and it contains buildings which show no major damage but are not completely steady either; some damage can be seen especially on roofs. Third class is “Major Damage” class, and the damage can be seen in the buildings. The main point of this class is that both damaged and steady part of the buildings can be seen. And lastly, “Destroyed” class shows only the debris of the destroyed buildings.

Table 2.1 Classes in dataset with examples

No Damage	
Minor Damage	
Major Damage	

Table 2.1 Classes in dataset with examples (…continued)

Destroyed	

Labeling data requires determining the buildings in an image and assigning classes to each object. Images in the dataset are labeled using Label Studio [46] tool. Label Studio enables both labeling with polygons and classifying objects with boundary boxes. It also offers exporting data in multiple formats like COCO, JSON making it easier to prepare data for different input types. To make labeling and image processing easier, each image is resized to 1024x683 resolution. This size enables ease of use during segmentation part of labeling while also protecting the details for classifying. In Figure 2.3, an example of labeled image is given.

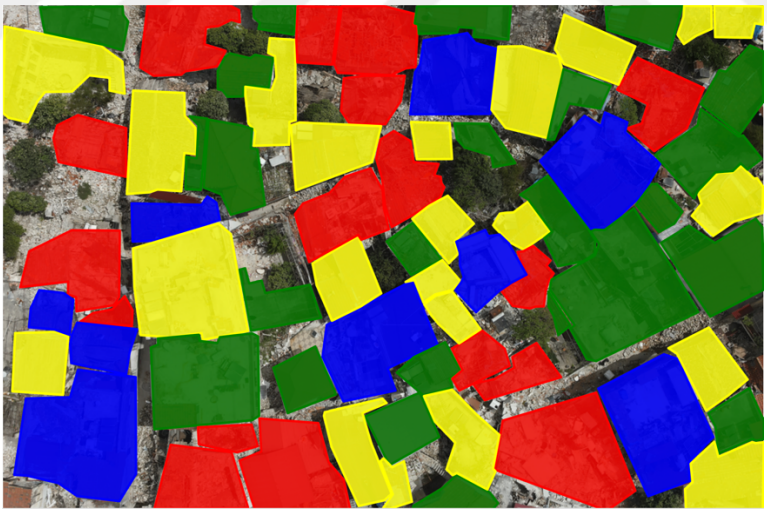


Figure 2.3 Labeled drone image from created dataset

The prepared dataset created contains 453 images and 7293 buildings. The class distribution of the dataset after labeling is shown in Table 2.2.

Table 2.2 Building type distribution

Class	No Damage	Minor Damage	Major Damage	Destroyed	Total
Building Count	3039	1963	1151	1140	7293

2.2 Damaged Building Detection with Image Segmentation

In this thesis detection of damaged buildings from drone images is aimed to help search and rescue operations after earthquakes. With the goal of assessing the severity of damage in buildings, the drone images are classified based on the level of damage they exhibit. Building detection is achieved through the use of image segmentation methods.

Image segmentation is the task of segmenting things (objects like cars, animals and so on) and/or stuff (sky, sea and so on) in an image. Image segmentation can be examined under three different titles: semantic segmentation, instance segmentation and panoptic segmentation. Semantic segmentation labels each pixel in the image. Instance segmentation partitions each object in the image with both object detection and semantic segmentation. Panoptic segmentation on the other hand, does both what semantic and instance segmentation do and classify and segment both “things” and “stuff” [47].

For semantic segmentation U-Net [36] is chosen as CNN based model for its success in semantic segmentation tasks. Even though it excels in medical image segmentation, it performs well on other domains as well. In recent years, Transformers [48] gained popularity with their success in NLP tasks. With release of Vision Transformers paper, the success of Transformers is applied to image tasks as well. For Transformer based model to perform semantic segmentation, SegFormer is chosen [41] as it is a lightweight Transformer architecture that outperforms previous architectures.

For instance segmentation, CNN based model Mask R-CNN [33] is chosen for both its lightweight architecture and performance on instance segmentation tasks. BoundaryFormer [42] is another Transformer based instance segmentation model

and it focuses on polygons. Since the buildings are segmented as polygons in prepared dataset, it makes a good choice for building damage detection.

OneFormer [43] on the other hand, is a Transformer based architecture for both semantic and instance segmentation. Since the work on this paper focuses on comparison between to methods, it is a convenient model to perform both methods on same architecture.

2.2.1 Semantic Segmentation

Semantic segmentation is a method to segment objects in images. This is conducted with labeling each pixel in the image with a class. Semantic segmentation performs localization, meaning it can detect boundaries of objects. This way, it provides more information compared to image classification or object detection.

As semantic segmentation provides pixel-level detailed information, it is used widely in real-world applications. Some examples of this includes; pedestrian detection [49], self-driving cars [50] and so on [51].

Multiple methods for semantic segmentation exist. CNN based methods are one of the most popular methods as CNN architecture performs well on feature extraction from images.

In order to see the performance of the two approaches, models from each architecture were selected to apply damage detection to the buildings using semantic segmentation. The approaches chosen are U-Net, SegFormer and OneFormer.

2.2.1.1 Semantic Segmentation with U-Net

U-Net [36], is a CNN based architecture for semantic segmentation. It is widely used in damaged building detection [23][35]. It is firstly presented for medical image segmentation but gained popularity across other domains as well. The work focuses on creating a pipeline that captures the context as well as localization. Unlike image classification tasks, image segmentation needs localization as it classifies each pixel separately. To ensure this, instead of using architecture of CNNs for image classification tasks, proposed architecture has a contracting path and an expansive path. Using contracting path, the context is propagated to upper

layers of the network. The paths are symmetrically structured in a U shape, giving the network its name “U-Net”. The architecture for U-Net is given in Figure 2.4

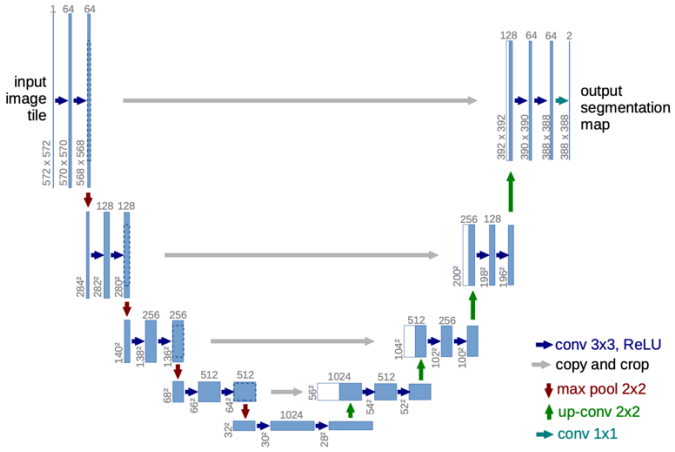


Figure 2.4 U-Net architecture taken from original paper [36]

U-net builds up on fully convolutional networks, so that the model works with even smaller datasets and results in more precise segmentations. As seen from Figure 2.5 the architecture consists of multiple layers. In FCN architecture, the layers increase output’s resolution thanks to the upsampling operators. To perform localization, the features from corresponding path are aggregated with the upsampled outputs. Different from FCNs, U-Net offers large number of feature channels during the upsampling, enabling the context information to propagate to higher resolution layers.

U-Net is chosen in this work as it is widely used in multiple tasks across semantic segmentation. It is also not complex and easily trainable because of small training times.

2.2.1.2 Semantic Segmentation with SegFormer

SegFormer [41] is a semantic segmentation DL model that combines MLP and transformer architecture. The architecture uses Transformer encoders to output multiscale features. Instead of using complex decoders of usual Transformer architecture, SegFormer makes use of proposed MLP decoders. The MLP decoder combines multiple information from different layers enabling collection of both local and global attention. SegFormer’s simple and effective design reached significant success on semantic segmentation tasks. The architecture for SegFormer is given in Figure 2.5

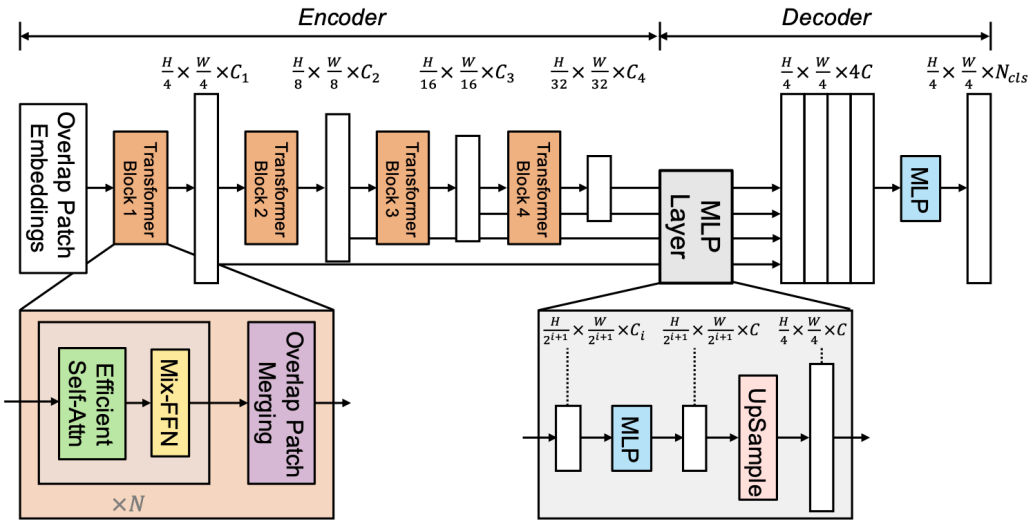


Figure 2.5 SegFormer architecture taken from original paper [41]

Shown in Figure 2.5 two main modules are the base for SegFormer architecture. The first module is a hierarchical transformer encoder. This module enables extracting coarse and fine features by using the patch inputs to get multi-level and multi-scale features with different resolutions shown in Figure 2.5.

The second module is a lightweight All-MLP decoder for creating the semantic masks. The outputs of the previous module are fed into to second module. To fuse the multi-level features from the encoder, four steps are conducted using MLP's instead of other methods to avoid complexity. The first step is to unify the channel dimensions. Second, the inputs are upsampled to ¼'th and concatenated. Thirdly the outputs from second step are fused using an MLP layer. In last step, another MLP layer uses the fused features to predict a segmentation mask.

SegFormer is chosen in this work due to its simple architecture while combining Transformers with MLPs. On top of this, short training times of the model is a plus.

2.2.1.3 Semantic Segmentation with OneFormer

OneFormer [43], is a segmentation model that tries to unify different segmentation tasks together under the name of Universal Image Segmentation. Most universal image segmentation models need to be trained individually for each task. OneFormer proposes a method to unify all tasks with one time training design. To do this, the proposed method introduces task-conditioned joint training and task tokens that determines the task at hand. On top of these, query-text contrastive loss is added to get better distinction between tasks and classes. This way, OneFormer

proposes a method for all segmentation tasks. The architecture for OneFormer is given in Figure 2.6

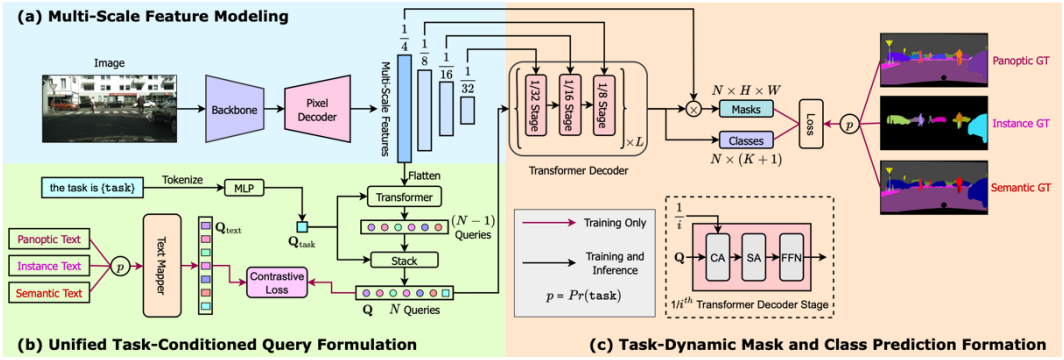


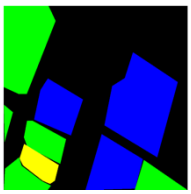
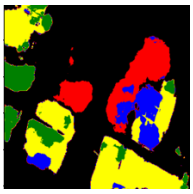
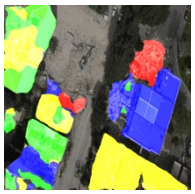
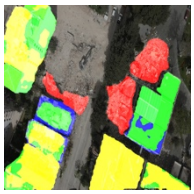
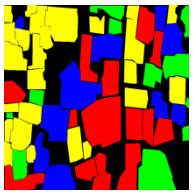
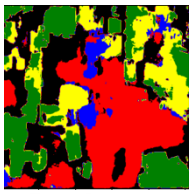
Figure 2.6 OneFormer architecture from original paper [43]

The architecture is as follows; firstly, examining the Multi-Scale Feature Modeling section multi-scale features are extracted using a backbone for an image. The outputs are then fed into a pixel decoder. In Unified Task-Conditioned Query Formulation step, task-conditioned object queries are created using the Transformer. The tasks here refer to semantic, instance and panoptic segmentation. The queries are then concatenated with processed task tokens. Using a text mapper, text queries are created with sampling the task. For learning inter-task distinctions, a query-text contrastive loss is calculated. Finally, in Task-Dynamic Mask and Class Prediction Formation a Transformer decoder is used for class and mask predictions.

OneFormer is chosen in this work for being able to practice universal segmentation tasks. It can perform both semantic and instance segmentation. It is also recently developed Transformer based model.

The chosen models' inference results on test images are given in Table 2.3. The colors green, yellow, blue, red represent class labels "No Damage", "Minor Damage", "Major Damage", "Destroyed" respectively.

Table 2.3 Semantic segmentation model outputs

Image	True Mask	U-Net	SegFormer	OneFormer
				
				

Examining the results, the models are able to detect the buildings in images. As classification of buildings are done per pixel, some buildings have multiple labels causing the misclassification. U-Net especially seems to have this problem more as one building includes multiple class labels. While OneFormer and Segformer models' results show similarity, main problem is the struggle to separate damage degrees. Both models detect the non-damaged and damaged parts of the buildings as "No Damage" and "Destroyed" but have hard time combining into "Major" damage class.

2.2.2 Instance Segmentation

Instance segmentation combines object detection and semantic segmentation to partition objects in images. Unlike semantic segmentation where pixels are labeled in a whole image, in instance segmentation pixels are labeled in an object. This ensures an object is classified as one class only.

The following are the multiple instance segmentation methods utilized in this study. The methods are chosen similarly to semantic models. One CNN based and two Transformer based models are chosen. The chosen models are Mask R-CNN, OneFormer and BoundaryFormer.

2.2.2.1 Instance Segmentation with Mask R-CNN

Mask R-CNN [33], is a general framework for object instance segmentation. Extending Faster R-CNNs [52], Mask R-CNNs add an extra step to predict object

masks parallel with bounding box recognitions. It is also more flexible as the model can easily generalize. The architecture is given in Figure 2.7.

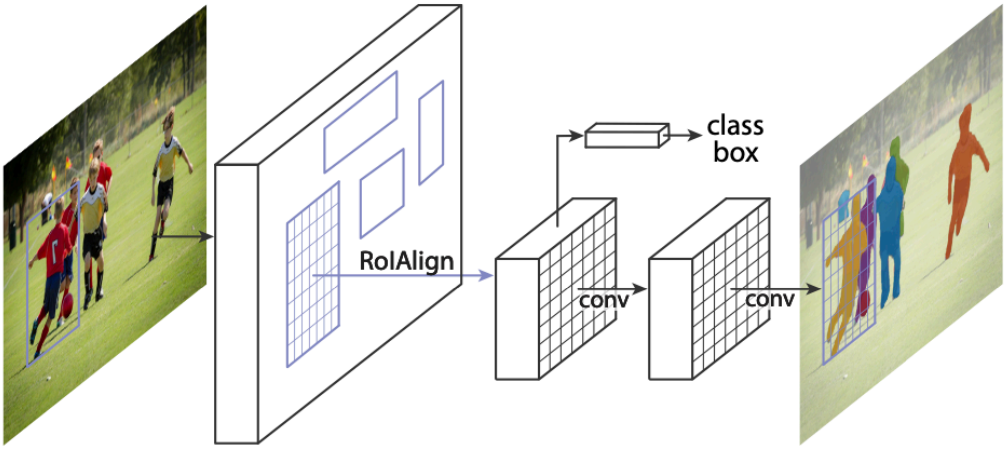


Figure 2.7 Mask R-CNN framework from original paper [33]

Mask R-CNN architecture is built upon Faster R-CNN. Faster R-CNN is made of two stages: one stage for proposing candidate object bounding boxes. This stage is called Region Proposal Network. The other stage is for classification and bounding-box regression to extract features using RoIPool -a procedure to extract small feature maps- from each box. For faster inference, the features are shared among the stages.

Mask R-CNN uses the same two stages from Faster R-CNN with some changes. Firstly, in second stage, during prediction of class and box offsets, a binary mask for all RoI's is also created. The parallelism in this step for bounding box classification and regression enables simplification of the pipeline. This third branch that creates the mask enables generating masks for every class without causing competition among classes. Thus, mask and class prediction are decoupled.

Another point in Mask R-CNN is that instead of RoIPool, a new method called RoIAlign is proposed. RoIAlign avoids the quantization with aligning the extracted features to the input.

Mask R-CNN is a powerful CNN based model and is widely used for instance segmentation tasks. Thus, it is chosen as one of the models to evaluate damaged building detection success in this work.

2.2.2.2 Instance Segmentation with OneFormer

OneFormer [43], as explained in Section 2.3.1.3, is a universal segmentation model. It proposes to perform all segmentation tasks with only panoptic segmentation data given as input. On top of this, OneFormer also offers segmentation task separately for semantic and instance. In this work OneFormer is used for both semantic and instance segmentation. Using same model for both methods enables comparison which is aimed to achieve.

2.2.2.3 Instance Segmentation with BoundaryFormer

BoundaryFormer [42] is a transformer-based instance segmentation model. It is a regression-based model as it uses transformers to regress a point along a polygonal boundary. Instance segmentation methods usually evaluate performance using segmentation masks even though the annotations are in form of polygonal boundaries or segmentation masks. BoundaryFormer proposes to directly predict polygons while using mask segmentations as ground truth for computing loss. The architecture for BoundaryFormer is given in Figure 2.8

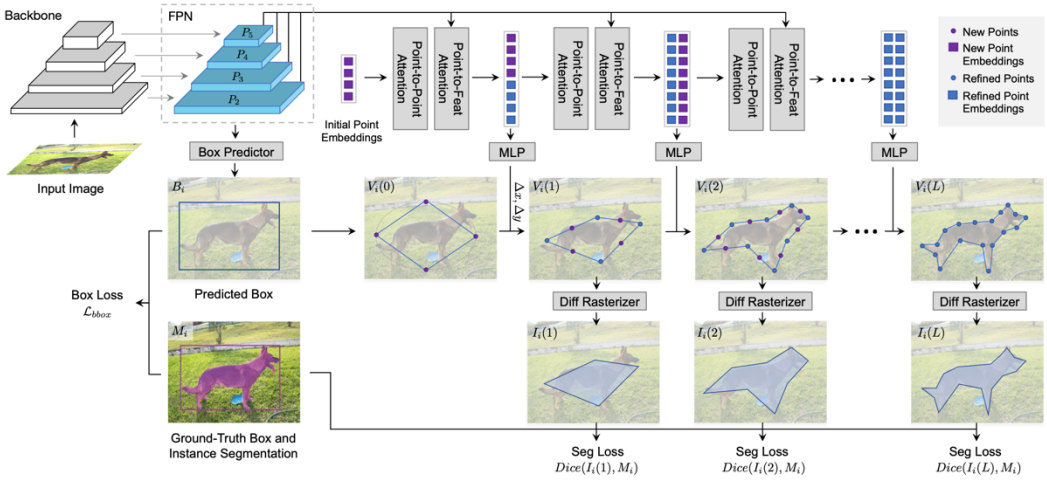


Figure 2.8 Architecture of BoundaryFormer from original paper [42]

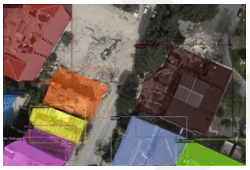
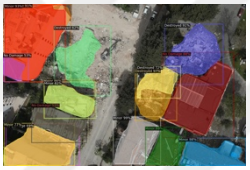
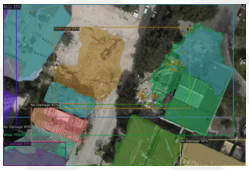
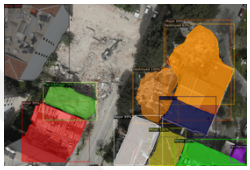
As shown in Figure 2.8 the architecture is conducted as follows; firstly, the images go through an underlying and multi-scale features and boxes for each object are extracted. Secondly, an ellipse shape is initialized as a polygon for each box. Going through layers, with the aim of refining polygons, attention within between polygons and attention from points to image features are performed. The refinement process includes adding new points to produce higher resolution polygons. Every time after refinement, the created polygon is rasterized in a differentiable way.

Finally, the polygon is compared to ground truth mask to calculate loss using a mask based loss.

BoundaryFormer is a recent transformer-based instance segmentation model. The images in the damaged building drone image dataset are labeled in polygons as the buildings are shaped like polygons. As BoundaryFormer makes use of polygon masks, the model is chosen to apply to the problem at hand.

In Table 2.4 the results for mentioned architectures are given. Each color represents a building and the labels with confidence values are given on top of the bounding box. The threshold for outputs is 0.50.

Table 2.4 Instance segmentation model outputs

Ground Truth	Mask RCNN	OneFormer	BoundaryFormer
			

Examining the results, Mask RCNN and BoundaryFormer show similar results in case of bounding boxes as they seem to successfully detect the buildings mostly. There are some cases where two buildings are detected as one. OneFormer, compared to other models struggle to detect buildings and cover large areas as bounding boxes. As for classification, mostly minor damage and no damage classes are misclassified caused by small differences in details.

2.3 Performance Metrics

Performance metrics are measurements that enable evaluating models' performances. Evaluation is a critical step in model selection as it provides ways to compare different methods. The success of damaged building detection methods can be measured using metrics for each model.

2.3.1 Pixel Accuracy

Pixel accuracy metric is used for semantic segmentation model evaluation. It is a metric that is similar to classic classification accuracy metric. It calculates the ratio of correctly classified pixels to all classified pixels [53]. For multi-class case, class

average accuracy is calculated. Pixel accuracy in this work is calculated according to Ref [54].

The calculation is for accuracy is done using True Positive (TP), False Positive (FP), False Negative values whose examples are given in Table 2.5.

Table 2.5 Confusion matrix

		Ground Truth	
Prediction		**Positive**	**Negative**
	Positive	TP	FP
	Negative	FN	TN

The formula for accuracy is given in Formula 2.1.

$$\text{Accuracy} = \frac{TP + TN|}{TP + TN + FP + FN} \quad (2.1)$$

2.3.2 Intersection Over Union

Intersection over union is a metric that is used in multiple image tasks such as object detection, semantic segmentation and instance segmentation. It is also called Jaccard Index. It is main goal is to evaluate how well the model can distinguish objects. It is calculated as the ratio of intersection of two areas to their union. IOU metric can be visualized as shown in Figure 2.9

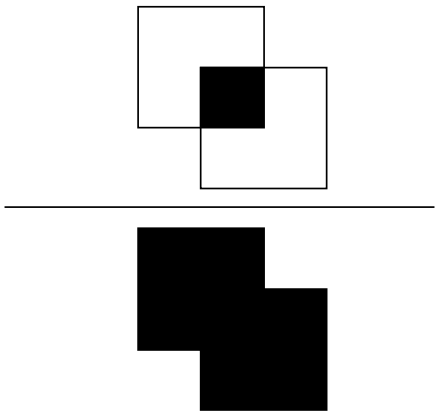


Figure 2.9 IOU = Intersection / Union visualized

The IOU for semantic segmentation is calculated as the ration of the intersection of the pixel-wise classification results with the ground truth labels, to their union. The formula is given in Formula 2.1.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.2}$$

Mean intersection over union, on the other hand, is the average of the class IOUs [53].

mIOU in this work is calculated according to Ref [54].

2.3.3 Average Precision

Average precision and mean average precision are metrics widely used in object detection tasks. As the object detection part of instance segmentation uses bounding boxes, it requires an evaluation metric. AP is calculated both using precision and recall metrics using IOU and a threshold.

Precision and recall metrics are calculated using a threshold. Figure 2.10 shows the difference threshold causes while determining true positives, false positives and false negatives. In Figure 2.10, the red line is the ground truth, and the blue line is model's prediction for bounding box.

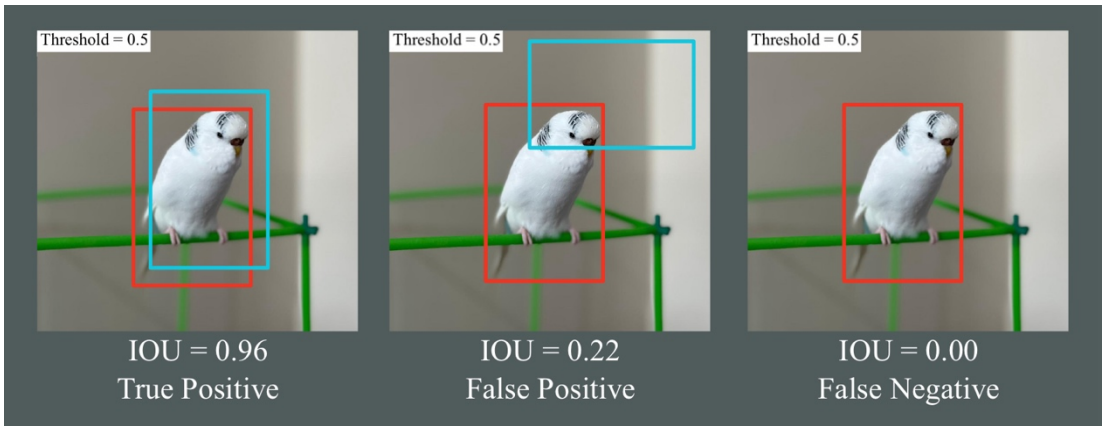


Figure 2.10 Effect of threshold on calculation of AP

The AP is calculated using precision and recall. The formula for precision is given in Formula 2.3.

$$Precision = \frac{T}{TP + FP} \tag{2.3}$$

The formula for recall is given in Formula 2.4.

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

The formula for AP is given in Formula 2.5

$$AP = \int_0^1 p(r)\, dr \tag{2.5}$$

AP is calculated accordingly to COCO dataset standards [55] in Detectron2. AP value is calculated in 3 different ways. First is "AP" which is AP values' average through threshold values between 0.50 to 0.95 increasing by 0.05. Second is "AP-50" representing the threshold value of 0.50. Lastly, "AP-75" shows the AP values for threshold 0.75.

2.4 Combining Semantic Segmentation and Instance Segmentation for Enhancing Detection of Damaged Buildings

For building damage detection after earthquakes, both the accuracy of the models and the amount of information gained are important. Thus, in this section a method to combine both methods' advantageous points is proposed.

The proposed method's diagram is given in Figure 2.11. Firstly, the information extracted from inference on models is acquired. For semantic segmentation, this information consists of labels for each pixel in the image. For instance segmentation, the information is location of each detected object -building in this case-. Using these acquired information, a method to aggregate the inference results is developed.

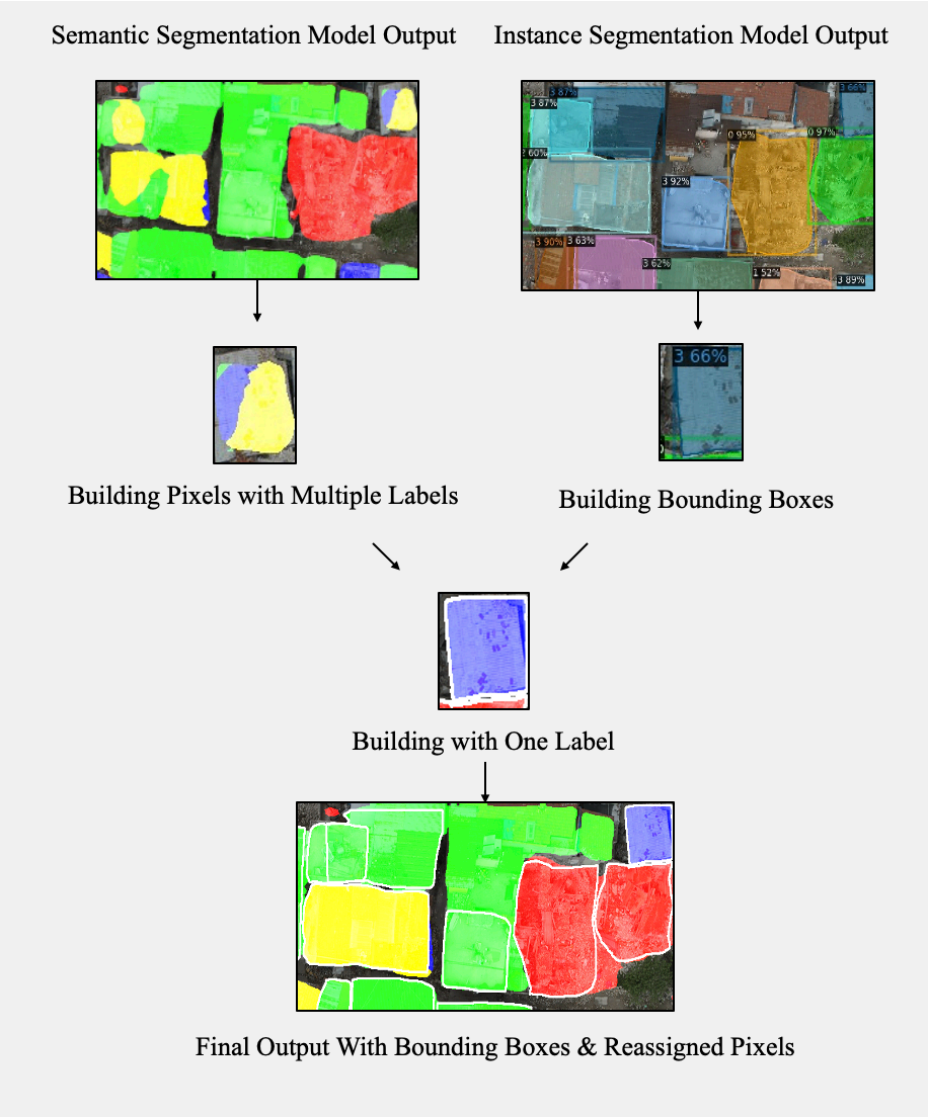


Figure 2.11 Flowchart for combining semantic and instance results

Semantic segmentation model's possibility of missing buildings is lower compared to instance segmentation as it labels each pixel in every image. Thus, it is chosen as the base for results. After getting label class results for each pixel in image, the segment results from instance segmentation model are acquired. For each instance segment in instance segmentation results, the labels for that specific segment from semantic segmentation results are iterated through. The process that every semantic label in instance segment goes through proposed method. If all labels belong to same class, this means that the semantic inference had no trouble classifying one building with one class and there is no need for improvement on results. In this case, only the border of the segment is outlined, and the building information is gathered.

For cases when there are multiple class labels for one segment, the process is different. Firstly, the labels which are less than chosen threshold of total labels (except for background) are eliminated as they are accounted as noise and not worth of considering. After this step, from the remaining classes the class with largest damage degree is chosen. This is because during the labeling phase, similar characteristics were considered. To explain more, during creation of dataset, in a case where there is both major damage and minor damage in a building, the building is labeled as major damage.

The only exception for choosing the largest damage degree is for “Destroyed” class. Destroyed class represents buildings that are no longer in shape and consists of debris. Thus, containing multiple labels like “Minor” or “Major” damage might mean the objects includes a part of a building which puts the object in “Major Damage” class according to labeling procedures. On the other hand, a huge debris with a small part of building is not considered “Major Damage”. To apply both these conditions, a specific case for objects with multiple labels that include “Destroyed” class is used. The building is labeled as “Destroyed” if and only if the largest damage degree is “Destroyed” and the label class ratio is more than chosen threshold. Similarly to previous case, this number is chosen according to labeling rules.

3
EXPERIMENTS

This section provides a detailed explanation of the steps involved in detecting damaged buildings from drone images, as well as showcasing the results of our experiments.

To conduct damaged building detection using deep learning, the first requirement is data. Specifically, the data needed for this particular problem is in the form of images. The images captured by drones in Hatay following the earthquake on February 6, 2023, serve as the dataset for all deep learning models utilized in this study. This dataset is utilized for training, validating, and testing the models.

To perform damaged building detection, image segmentation is used. Image segmentation focuses on segmenting the objects in the image. Firstly, semantic segmentation models is developed on newly created drone image dataset. Semantic segmentation performs damaged detection by classifying all pixels in image one by one. Three different deep learning models including CNN based and Transformer based were developed. From resulting metrics, the best performing model on validation set is chosen. After this, hyper-parameter tuning is performed to get the best performing model perform better. With optimized best model, the combination of train and validation set is trained. The results are examined on metrics extracted from evaluation on test set. Finally, labeled classes in dataset are grouped differently for multiple damage degrees to analyze the impact of class labels on the model performance.

For second part of the experiments, instance segmentation models are trained on drone dataset. Different from semantic segmentation, instance segmentation combines object detection with segmentation. Thus, the damaged buildings will be detected first. Three different models including CNN based and Transformer based deep learning models undergo training. After this, the same steps from semantic segmentation are applied to get the best model on instance segmentation model.

Finally, best semantic segmentation model and instance segmentation model is compared to analyze which model has advantages or disadvantages over each other. A method to combine and reduce the misclassification of both models is proposed.

3.1 Data Splitting

Input data needs to be processed before using. First, the labelled data must be split into sets. As the existing data is limited and is needed for training, validation and testing, it needs to be split accordingly.

Training is one of the fundamental parts as it is where the model learns the underlying patterns in data. Thus, it is important that there is enough training data for the model to learn. If the training data is not enough, then the model might underfit. Underfitting is when the model hypothesis is rather simple and fails to determine the patterns. Underfitting can be examined in validation results and inference on new images.

Validation set enables examining whether the model training is going smoothly. In each epoch, model's metrics are tested with validation data. Since the model has never seen the validation data, examiners can infer model's status from metrics. Underfitting as well as overfitting can be seen from validation results. Another benefit of using validation set is to determine best hyperparameters for the model.

Test set is used for testing the models' success and comparing. Test data is never used during training. Thus, model's success for real-world data can be inferenced.

According to the need of every set, the data set is split into sets as shown in Figure 3.1.

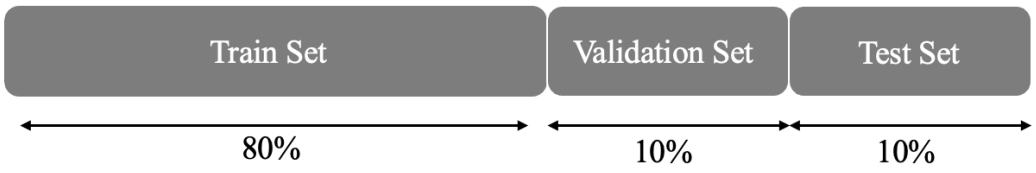


Figure 3.1 Data split for supervised deep learning models

After the distribution, training set has 361 images, validation set has 46 images and test set has 46 images. Object count per class is given in Table 3.1.

Table 3.1 Building count distribution per class

Class	No Damage	Minor Damage	Major Damage	Destroyed	Total
Building Count	**Train Set**				
	2216	1337	833	783	5169
	Validation Set				
	450	295	164	169	1078
	Test Set				
	373	331	154	188	1046

3.2 Results for Damaged Building Detection with Semantic Segmentation

In this section the experiments done, and their results will be shown. The training is done in epochs. An epoch is one cycle where all training data is given as input into the model.

3.2.1 Results for Damaged Building Detection with U-Net

U-Net results are given in Table 3.2. For U-Net, no prior weight information is used, and the weights are all trained using the images in dataset. The batch size is given as 4 and the training is done for 100 epochs.

Table 3.2 U-Net pixel accuracy and IOU results

		Class					
		No Damage	Minor	Major	Destroyed	Back-ground	Mean
Metric	Accuracy	0.46	0.34	0.32	0.42	0.82	0.67
	IOU	0.30	0.23	0.23	0.32	0.66	0.35

As seen from Table 3.2, U-Net results show an mIOU of 0.35 and average pixel accuracy of 0.67. The background class has the highest accuracy and IOU. IOU results show worse values compared to accuracy results. This is expected as accuracy does not consider the class distribution. Apart from back-ground, the highest mIOU score is obtained from "Destroyed" class. Even though "No Damage" class has highest accuracy, its IOU is not the highest. This might be due to the fact that the dataset contains more "No Damage" buildings.

The inference results for U-Net is given in Figure 3.2. The colors green, yellow, blue, red represent class labels "No Damage", "Minor Damage", "Major Damage", "Destroyed" respectively. The results show that U-Net struggles to separate the buildings and labels one building with multiple labels excessively.

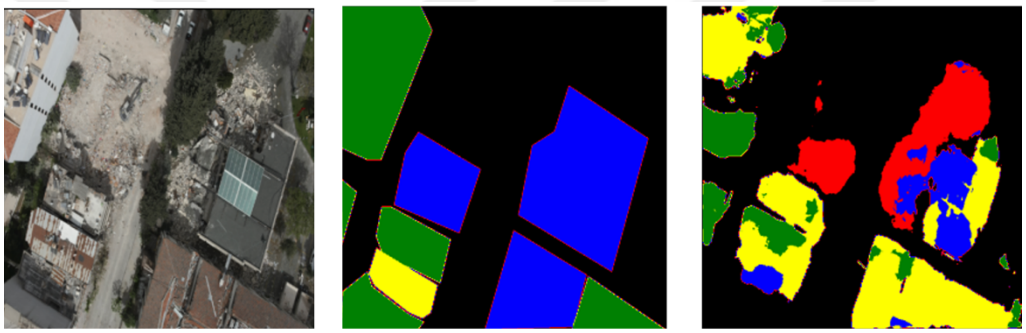


Figure 3.2 From left to right, original image, true mask and U-Net model output

3.2.2 Results for Damaged Building Detection with SegFormer

SegFormer results are given in Table 3.3. The model is trained using batch-size of 4. Epoch count is 100. The weights are initialized from pre-trained models and fine-tuned on the dataset created in this work.

Table 3.3 SegFormer pixel accuracy and IOU results

		Class					
		No Damage	**Minor**	**Major**	**Destroyed**	**Back-ground**	**Mean**
Metric	**Accuracy**	0.65	0.56	0.45	0.67	0.80	0.70
	IOU	0.45	0.44	0.35	0.49	0.71	0.49

Table 3.3 shows that the mIOU for SegFormer model is 0.49 and the average PA is 0.70. Class scores for IOU shows similarity except for “Major” class. Both accuracy and IOU for back-ground is high. Apart from back-ground, the highest score for both IOU and accuracy is obtained from “Destroyed” class.

The inference results for SegFormer is given in Figure 3.3. The colors green, yellow, blue, red represent class labels “No Damage”, “Minor Damage”, “Major Damage”, “Destroyed” respectively. Main problem of semantic segmentation which is labeling one building with multiple classes can be seen in SegFormer results. Apart from this, the building shapes are segmented correctly.

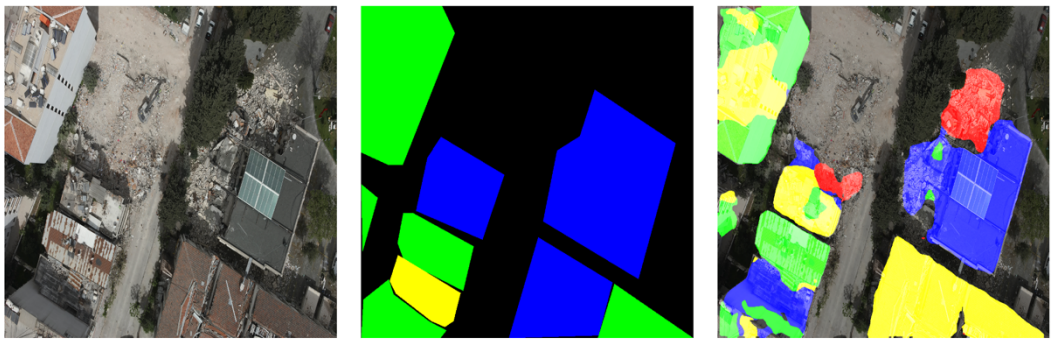


Figure 3.3 From left to right, original image, true mask and SegFormer model output

3.2.3 Results for Damaged Building Detection with OneFormer

OneFormer results are shown in Table 3.3 Due to the complex nature of the model and hardware constraints, batch-size is given as 2. Train epoch count is 100. The backbone for model is chosen as Swin Transformer [56]. The weights are initialized from a pre-trained model and fine-tuned.

Table 3.4 OneFormer pixel accuracy and IOU results

		Class					
		No Damage	**Minor**	**Major**	**Destroyed**	**Back-ground**	**Mean**
Metric	**Accuracy**	0.52	0.42	0.30	0.41	0.85	0.67
	IOU	0.37	0.31	0.21	0.27	0.73	0.38

As seen from Table 3.4, OneFormer model got an mIOU of 0.38 and a mean pixel accuracy of 0.67. Apart from back-ground, the highest score for both IOU and accuracy is obtained from “No Damage” class.

The inference results for OneFormer is given in Figure 3.4. The colors green, yellow, blue, red represent class labels “No Damage”, “Minor Damage”, “Major Damage”, “Destroyed” respectively. Main problem of semantic segmentation which is labeling one building with multiple classes can be also seen in OneFormer results. Apart from this, some debris is segmented as destroyed buildings.

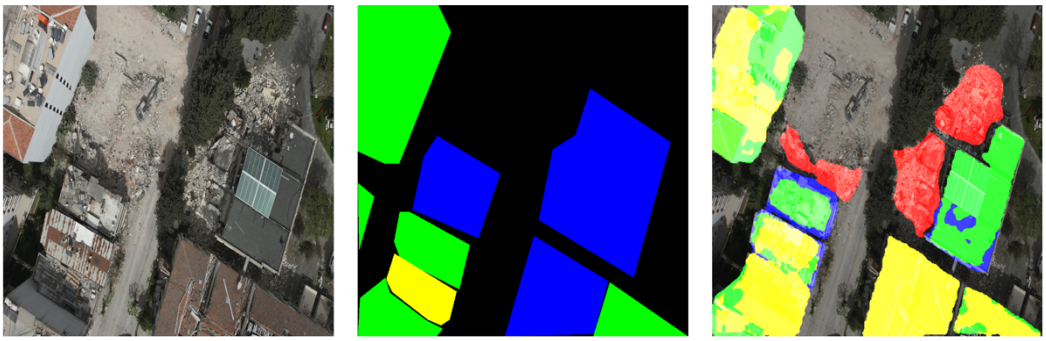


Figure 3.4 From left to right, original image, true mask and OneFormer model output

3.2.4 Comparing Semantic Segmentation Models for Damaged Building Detection

All semantic segmentation models’ mIOU and mean pixel accuracy results are shown in Table 3.5. mIOU metric considers more variables compared to pixel accuracy. Thus, it is chosen as primary metric to evaluate models.

Table 3.5 Comparison of semantic segmentation models

Model	mIOU	Mean Pixel Accuracy
U-Net	0.35	0.67
SegFormer	0.49	0.70
OneFormer	0.38	0.67

According to Table 3.5, SegFormer is the best performing model with an mIOU of 0.49 and a mean pixel accuracy of 0.70. OneFormer is the second best and U-Net is the last. Considering U-Net model not having initial weights, its results are not bad. OneFormer and U-Net results are very close to each other.

Because SegFormer model showed the best results, the following semantic segmentation tests are done using SegFormer.

3.2.5 Making The Best Semantic Segmentation Model Better

In this section, the best model chosen for semantic segmentation is evaluated and optimized to have better results.

3.2.5.1 HyperParameter Tuning for Best Semantic Segmentation Model

Hyperparameter tuning is the process of choosing best parameters for the model to perform better. The model is trained with different parameters on validation set and the results are evaluated.

Different hyperparameters for SegFormer model like pre-trained model, learning rate and augmentations are tested and the highest scoring one is chosen.

One of the crucial hyperparameters is epoch count. Epoch affects the model's performance as it represents how long the training process lasts. For choosing the best epoch parameter, the model is trained with 100 epochs and the graphs for loss and accuracy are examined. In Figure 3.5 the loss graph is shown. The loss function is Cross Entropy loss.

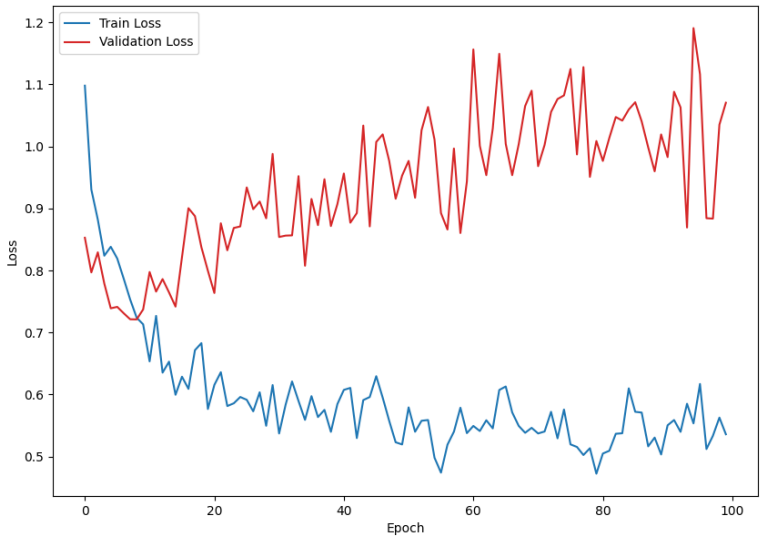


Figure 3.5 Loss graph for SegFormer model

Examining Figure 3.5, the red line shows validation loss and the blue line represents training loss. The graph show how the loss values change during training epoch by epoch. While training loss is expected to get smaller over epochs, validation loss should also be considered as it shows the models ability to generalize. In this particular graph, even though the training loss is going down, validation loss starts to go up. This might show the over-fitting of model and it is important to stop training before the model learns and memorizes patterns of the training data. The continuous increase of validation loss might be due to the fact that model struggles to differentiate between classes and the confidence is low while making decisions. Since it is hard to determine the stopping point only from loss graph, accuracy graph is also used. The model's accuracy graph is shown in Figure 3.6.

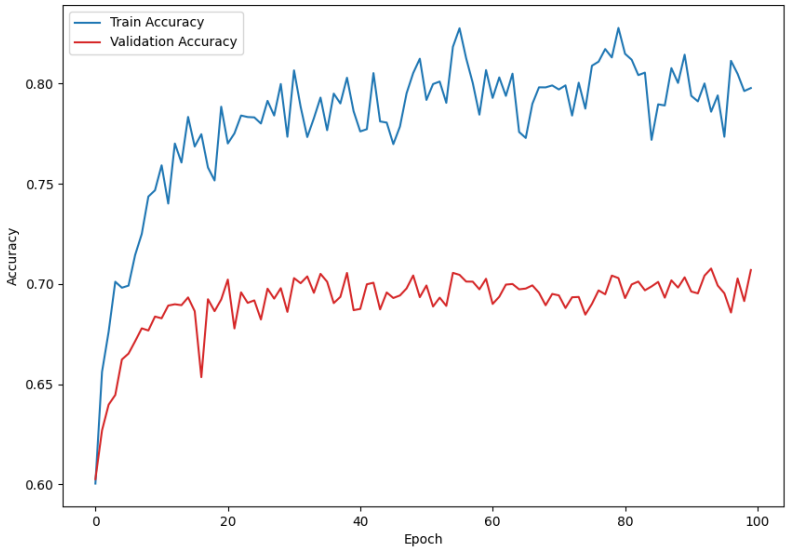


Figure 3.6 Accuracy graph for SegFormer model

Figure 3.6 shows the average pixel accuracy metric for SegFormer model over 100 epochs. Starting from Epoch 55, it can be seen from the graph that after a drop and rise the accuracy starts to converge and does not change for validation loss. This is a good stopping point. Thus, the epoch is chosen as 55.

During training, to prevent overfitting, multiple methods are conducted. These include training with no pretrained weights, using simpler pretrained models and augmentation. Best results are gained with augmentation. For augmentation, Albumentations [57] library is used. The augmentations used are rotation, flip, contrast, fog rain and shadow with 0.1 probability on images. With these augmentations, noise is added to train images, preventing model from memorizing the patterns in train images.

After choosing best hyperparameters for the model, validation set's main objective which is finding best models and hyperparameter is done. Thus, the validation data can be used for training. Combining validation and training set the model with best parameters is trained for 55 epochs. The results on test set are given in Table 3.6.

Table 3.6 SegFormer best pixel accuracy and IOU results

		Class					
		No Damage	**Minor**	**Major**	**Destroyed**	**Back-ground**	**Mean**
Metric	**Accuracy**	0.53	0.57	0.61	0.83	0.75	0.72
	IOU	0.44	0.45	0.42	0.57	0.70	0.51

Given in Table 3.6, the mIOU of the model is 0.51 and average PA is 0.72. Compared to previous models, an increase in metrics can be seen. This is due to hyperparameter tuning as well as the increase in training data.

To get a better understanding of the metric results, inference on test imaged is performed. The original image, true mask and model output is given in Figure 3.7. Green colour represents the class “No Damage”, yellow represents “Minor”, blue represents “Major” and red represents “Destroyed” class.

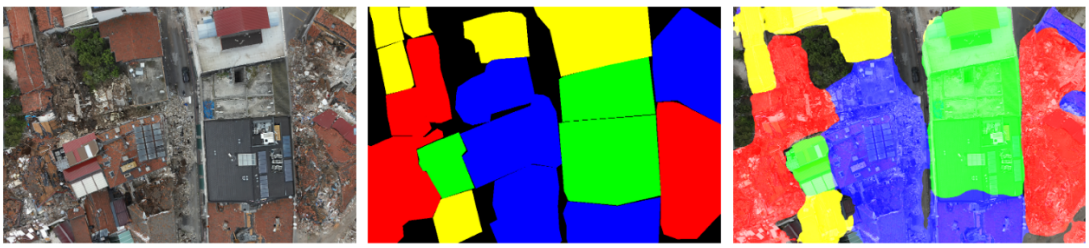


Figure 3.7 Original image, ground truth labels and Segformer model output

Examining Figure 3.7, the difference between the ground truth labels and model output labels can be seen. Firstly, when compared generally, the model seems to classify labels mostly correctly. Especially “Destroyed” and “No Damage” classes which represent the two extremes of damage degrees are classified correctly very frequently.

One main effect that decreases the metrics is that the pixel labels predicted by the model are not as sharp as the ground truth labels. While this can be ignored for overall damage assessment it negatively affects the model’s metric results.

In Figure 3.8, a building is zoomed in to show another significant point to be examined. In ground truth labels, it can be seen that the building is labeled as "Major" class, and it represents one object. In model output labels, the pixels in the region are labeled with multiple classes even though it represents one object and needs one class label. This is because semantic segmentation does not consider objects while labeling, it only labels the pixels.



Figure 3.8 Misclassified building from SegFormer inference

The regions shown in Figure 3.9, shows one of the examples of model's misclassifications. "No Damage" and "Minor" classes have very small difference in damage degree seen from drone images. Thus, they are frequently misclassified as each other in inference results.



Figure 3.9 Minor and No Damage labels from SegFormer inference

Other classes which result in misclassification are "Major" and "Destroyed". This is because majorly damaged buildings have parts that are destroyed. The destroyed parts of the buildings get classified as "Destroyed" when in fact, they should be included in "Major" labeled building.

Multiple examples from SegFormer results are given in Figure 3.10. The colors green, yellow, blue, red represent class labels "No Damage", "Minor Damage", "Major Damage", "Destroyed" respectively.

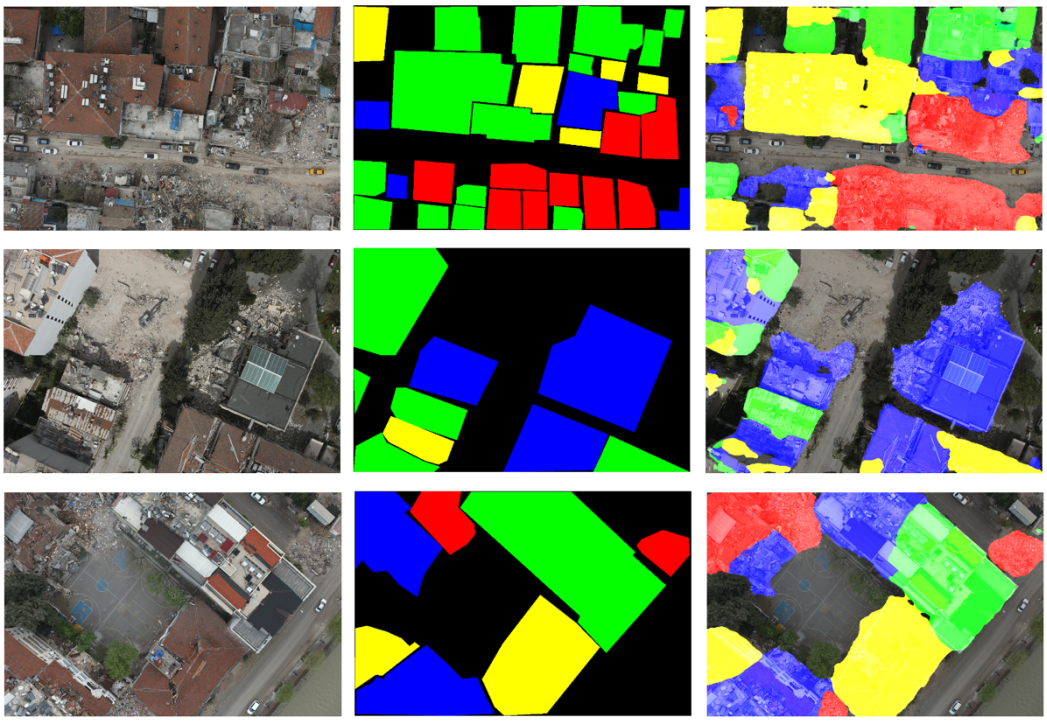


Figure 3.10 Output examples from Segformer model. From left to right; original image, true masks, SegFormer output

3.2.5.2 Grouping Classes for Semantic Segmentation

From the results from previous sections, it can be seen that "Minor" and "Major" classes are most likely to get misclassified. To get better results for semantic segmentation, the classes are merged.

- **Stable and Damaged Classes**

In this section, "Minor" and "Major" classes are removed. Minor class is merged with "No Damage" class. The new class is named "Stable" because the buildings are either not damaged or slightly damaged. "Major" class containing buildings that has large damage is merged with "Destroyed" class and the new class is named "Damaged". "Damaged" class has buildings with major damage and are destroyed.

The newly grouped dataset is trained on the model with the same settings as best SegFormer model.

Table 3.7 SegFormer pixel accuracy and IOU results for 2 classes

		Class			
		Stable	**Damaged**	**Back-ground**	**Mean**
Metric	**Accuracy**	0.78	0.80	0.77	0.80
	IOU	0.70	0.64	0.70	0.68

The result SegFormer model trained on 2-class-dataset are given in Table 3.7. The average PA is 0.80 and mIOU is 0.68. Both values are significantly higher than 4 class case. The difference in average PA is 0.8 and difference in mIOU is 0.17. As the wrong classification of eliminated classes affect the mIOU more, the difference is higher for mIOU.

Examining the class distribution for metrics, stable buildings are classified correctly more frequently compared to damaged buildings.

The original image for inference is given in Figure 3.11. The colour green represents "Stable" class and the colour red represents "Damaged" class.

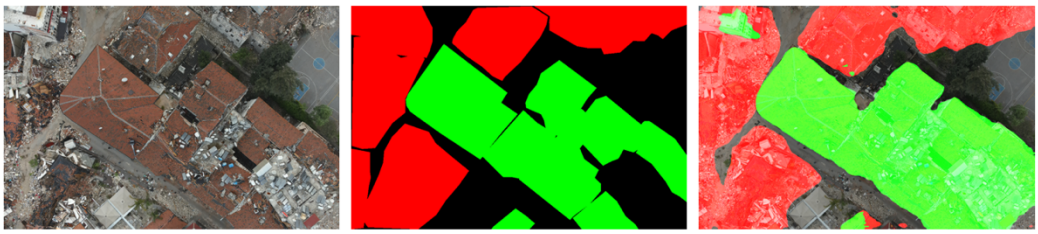


Figure 3.11 Original image, ground truth and model output for 2-class

Compared to ground truth labels, model output labels are pretty similar to each other. Stable and damaged buildings are determined correctly. Similar to 4-class case, some buildings are labeled as 2 different classes. However, it is much less frequent compared to 4 class case.

As seen from the model output image, another misclassification case is labeling regions that are too small to be a building. Even though they might be considered debris, they are too small to be considered as buildings. The model struggles to differentiate this case.

Lastly, the edges are not as sharp in the ground truth labels similar to 4 class case. Multiple examples from the model with 2 classes are given in Figure 3.12. The colors green and red represent class labels “Stable” and “Damaged” respectively.

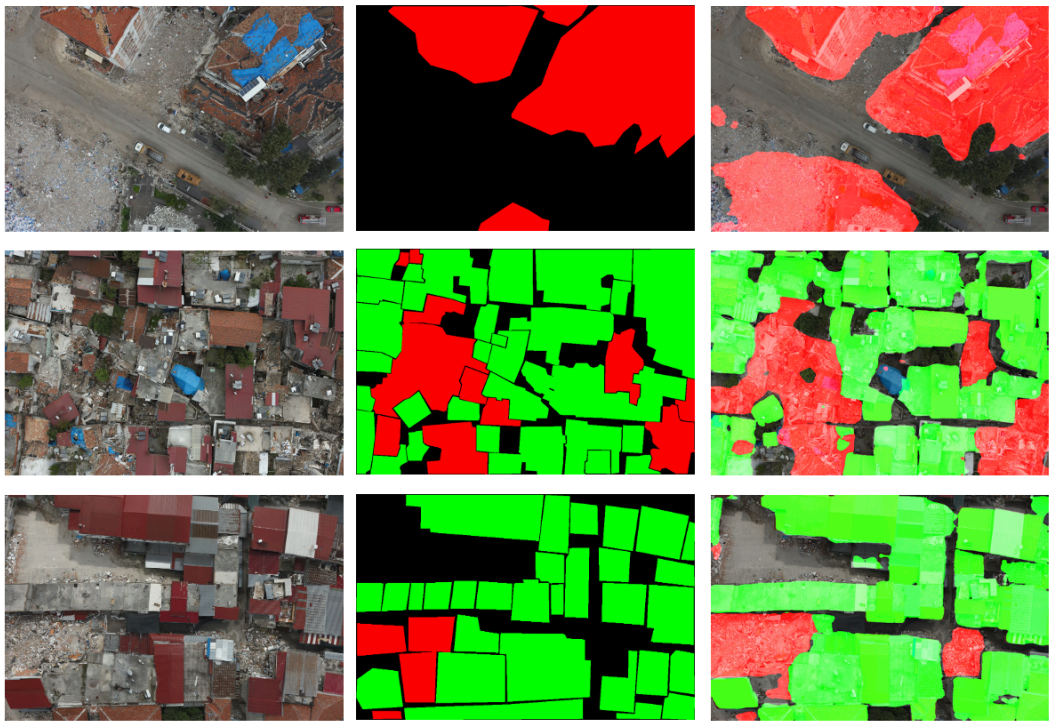


Figure 3.12 Output examples from SegFormer model with 2 classes. From left to right; original image, true masks, SegFormer output

- **Damaged Class Only**

Considering the fact that the main goal of the work is evaluating overall damage after earthquakes, detecting only damaged buildings is also a method. Even though this method provides less information about the damage as it does not have damage degrees, it still can help evaluate the regions according to damage.

For this, the “No Damage” and “Minor” classes are eliminated and “Major” and “Destroyed” classes are merged to create “Damaged” class. In the end, a dataset consisting of segmented damaged buildings is acquired. The dataset is trained on best SegFormer model in same setting as previous models.

Table 3.8 SegFormer pixel accuracy and IOU results for 1 class

		Class		
		Damaged	**Back-ground**	**Mean**
Metric	**Accuracy**	0.73	0.87	0.84
	IOU	0.64	0.82	0.73

The results for 1-class-dataset is shown in Table 3.8. The mIOU of the model is 0.73 and the average PA is 0.84. The results for this model shows the best metrics for all models tested. Inference from test images is done on SegFormer model.

The original image given as input, ground truth mask and inference output is shown in Figure 3.13.

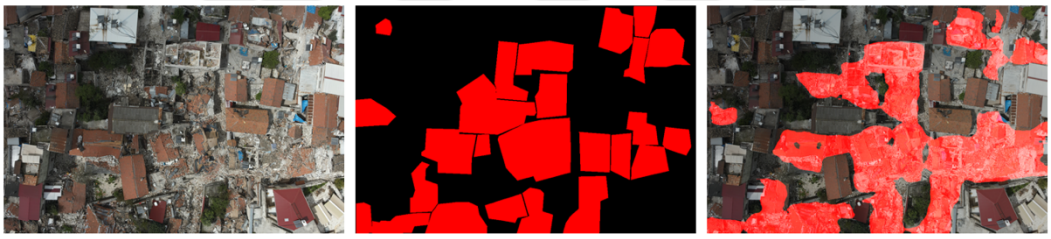


Figure 3.13 Original image, ground truth and model output 1-class

For 1 class case, the main problem is splitting the buildings. As “Major” class consists of buildings half damaged the non-damaged parts are not predicted by the model. On top of this, the model struggles to find borders between buildings as they are closely located to each other. Apart from these problems, the model for predicting only damaged buildings is able to detect successfully.

Multiple examples for 1 class model are given in Figure 3.14.

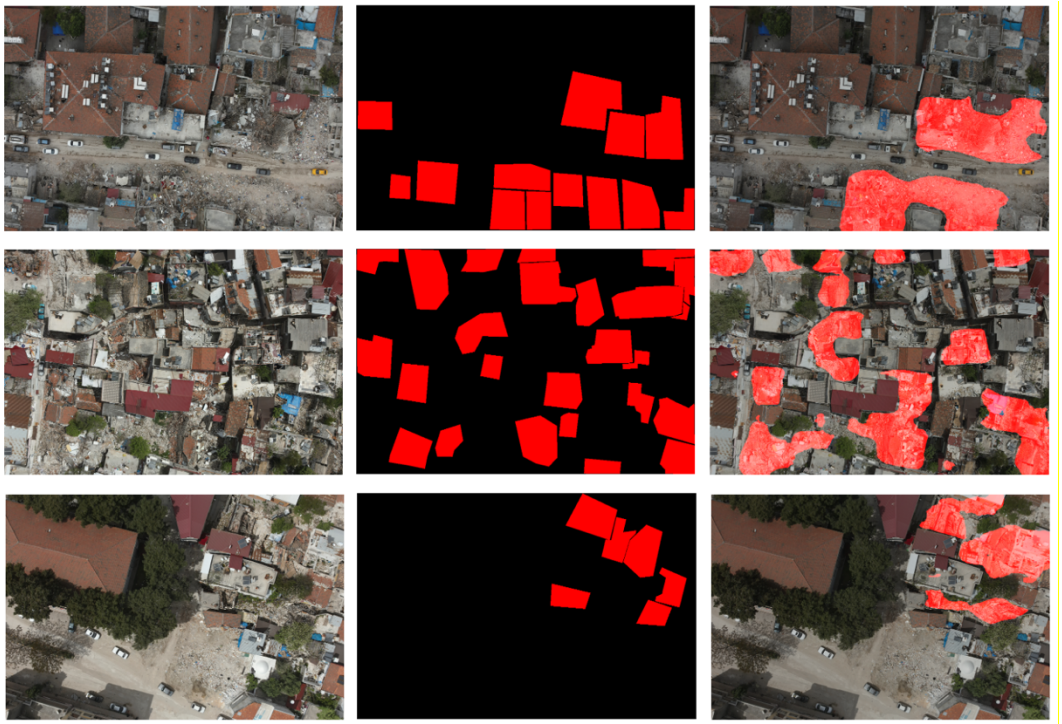


Figure 3.14 Output examples from SegFormer model with 1 class. From left to right; original image, true masks, SegFormer output

3.3 Results for Damaged Building Detection with Instance Segmentation

In this section, instance segmentation evaluation metrics on validation set for each model is shown. The training is done in iterations since Detectron2 library uses iteration as training process. Each iteration represents the process of model's weight parameters' update. Since instance segmentation performs both object detection and segmentation, both metrics for bounding box and segmentation is evaluated separately. For comparison, average AP-50 metric is used. Through inference, a threshold value of 0.50 is used as it results in better detection.

3.3.1 Damaged Building Detection Mask R-CNN

Mask R-CNN AP results for instance segmentation is given in Table 3.9. The model is fine-tuned using pre-trained model weights. The batch size is 4 and the iteration count is 10000.

Table 3.9 Mask R-CNN bounding box and segmentation results

	AP	**AP50**	**AP75**
BBox	0.143	0.260	0.140
Segm	0.134	0.257	0.127

Examining Table 3.9, the model's mean AP for bounding box is 0.143 and 0.134 for segmentation. The model seems to get better bounding box results compared to segmentation. This is because even though buildings are detected, segmenting is worse compared to detection. Class distribution for metrics is given in Table 3.10.

Table 3.10 Mask R-CNN bounding box and segmentation results per class

	AP (%)			
	No Damage	**Minor**	**Major**	**Destroyed**
BBox	18.3	10.4	11.7	16.8
Segm	18.1	9.8	11.0	14.9

The results in Table 3.10 show that "No Damage" class is classified most accurately. Similar to semantic segmentation results, "Major" and "Minor" classes show worse performance compared to others. Example image from Mask R-CNN model is given in Figure 3.15.

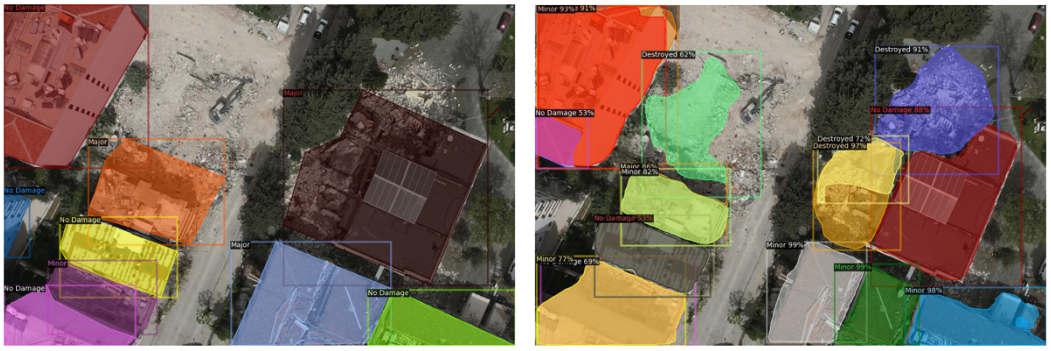


Figure 3.15 From left to right: Ground truth, Mask R-CNN inference result

In output images, Mask R-CNN detects most buildings as expected. Some buildings are detected as two buildings instead of one. Especially for "Major

Damage" cases, they are detected as to different buildings with "No Damage" and "Destroyed".

3.3.2 Damaged Building Detection OneFormer

OneFormer model results are given in Table 3.11. The model is fine-tuned on initialized pre-trained weights. The backbone chosen is DiNAT [58]. The batch size is chosen as 2 due to model complexity and hardware constraints. The iteration count for OneFormer model is 50000.

Table 3.11 OneFormer bounding box and segmentation results

	AP	**AP50**	**AP75**
BBox	0.030	0.069	0.027
Segm	0.046	0.110	0.038

Table 3.11 results show that for bounding box mean AP is calculated as 0.030 and for segmentation mean AP is calculated as 0.046. These results show significantly worse performance compared to other models. This is due to long training times of the model and its inability to converge quickly caused by hardware constraints. To make better comparison, iteration count is increased only for this model. Under better circumstances, OneFormer model might be able to show better results.

Table 3.12 OneFormer bounding box and segmentation results per class.

	AP (%)			
	No Damage	**Minor**	**Major**	**Destroyed**
BBox	2.7	2.1	2.3	5.1
Segm	5.1	2.9	4.3	6.0

Given in Table 3.12 are class distributions for OneFormer AP metric. Destroyed class has the best AP value for both BBox and Segm.

Example image from OneFormer model is given in Figure 3.16.

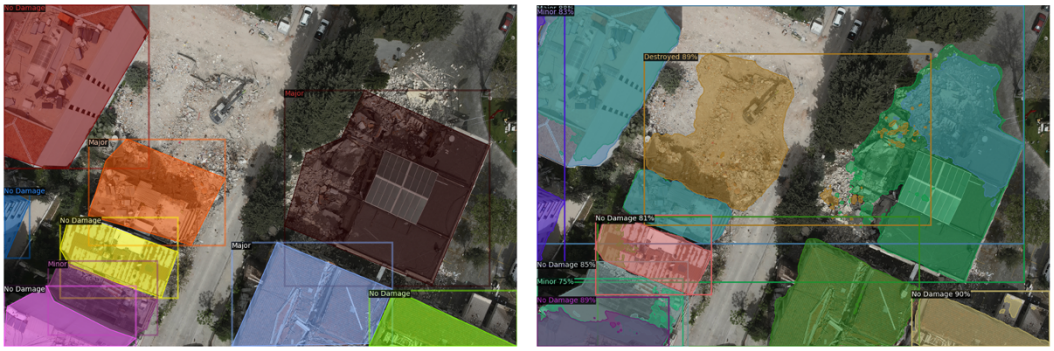


Figure 3.16 From left to right: Ground truth, OneFormer instance segmentation inference result

The results from OneFormer show that the model struggles to detect the buildings. Large bounding boxes are drawn as the model detects small pieces far from each other as one building. Still, some simple buildings are detected correctly.

3.3.3 Damaged Building Detection BoundaryFormer

BoundaryFormer instance segmentation model results are shown in Table 3.13. The model is fine-tuned on pre-trained model. The batch-size is 4 and the iteration count is 10000.

Table 3.13 BoundaryFormer bounding box and segmentation results.

	AP	**AP50**	**AP75**
BBox	0.169	0.314	0.168
Segm	0.159	0.308	0.143

As shown in Table 3.13, BondaryFormer's Bbox mean AP is 0.169 and Segm mean AP is 0.159. Bounding box AP is higher than segmentation AP. AP calculated with 0.50 threshold on the other hand, shows bounding box AP of 0.314 and segmentation AP of 0.308. Class distribution of AP's are given in Table 3.14.

Table 3.14 BoundaryFormer bounding box and segmentation results per class

	AP (%)			
	No Damage	**Minor**	**Major**	**Destroyed**
BBox	18.5	17.8	17.1	17.0
Segm	18.4	14.4	15.0	15.7

Examining class distribution results from Table 3.14, the highest AP score for bounding box and segmentation belongs to “No Damage” class.

Example image from OneFormer model is given in Figure 3.17.

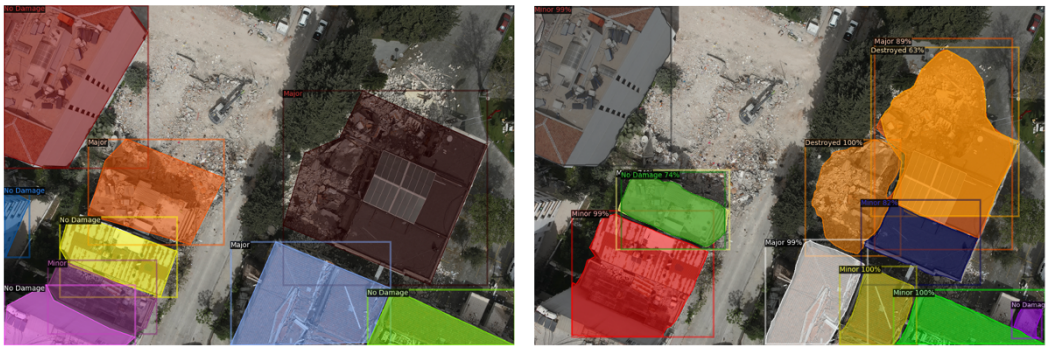


Figure 3.17 From left to right: Ground truth, BoundaryFormer inference result

The inference results show that similar to other models BoundaryFormer also struggles on some buildings as it detects one building as two buildings. One advantage of BoundaryFormer compared to other models, it does not misclassify debris as a destroyed building.

3.3.4 Comparing Models

All instance segmentaion models’ and their mean AP-50 for bounding box and segmentation is given in Table 3.15. Both BBox and Segm AP’s are considered while comparing.

Table 3.15 Models' AP results

Model	BBox AP	Segm AP
Mask-RCNN	0.143	0.134
OneFormer	0.030	0.046
BoundaryFormer	0.169	0.159

As seen from Table 3.15 Transformer based BoundaryFormer model shows the best results. Second best model is Mask R-CNN. It also shows noteworthy results and can be used for damaged building detection. OneFormer is last in metrics, most likely due to not being trained under optimum circumstances.

Following tests will be done using BoundaryFormer model.

3.3.5 Making The Best Model Better

In this section, the steps followed for semantic segmentation model optimization will be done for best instance segmentation model, BoundaryFormer.

3.3.5.1 HyperParameter Tuning for Best Instance Segmentation Model

Different hyperparameters for BoundaryFormer model like pre-trained model configuration, learning rate and augmentations are tested and the highest scoring one is chosen.

For BoundaryFormer, firstly the model is trained for 10000 iterations. The loss used is BoundaryFormer's own mask-based loss. The resulting loss graph is given in Figure 3.18.

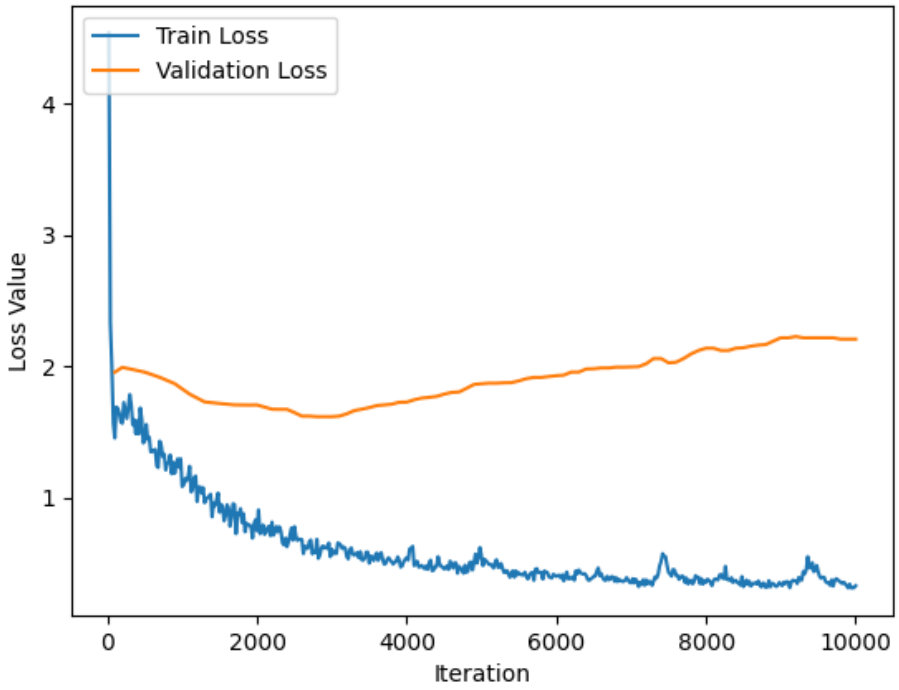


Figure 3.18 Loss graph for instance segmentation model BoundaryFormer.

Examining Figure 3.18, the train loss is shown in blue coloured line and validation loss is shown in orange coloured line. The model's validation loss seems to converge during 2500th iteration. Thus, iteration count for best model is chosen as 2500. Training more than selected iteration count may lead to overfitting.

After finding the best hyper-parameters, the model is trained on combined train and validation sets. The results on test set are given in Table 3.16.

Table 3.16 Best BoundaryFormer bounding box and segmentation test results.

	AP	**AP50**	**AP75**
BBox	0.173	0.340	0.160
Segm	0.174	0.329	0.168

The results in Table 3.16 show that the optimized model trained with more images show better results for AP. The bounding box mean AP is 0.173 and segmentation mean AP is 0.174. This is 0.004 increase for bounding box and 0.015 increase for segmentation. The increase of data and model optimization effects can be seen through results.

Class AP distributions are given in Table 3.17.

Table 3.17 Best BoundaryFormer bbox and segmentation test results per class

	AP (%)			
	No Damage	**Minor**	**Major**	**Destroyed**
BBox	16.7	17.0	16.8	18.8
Segm	17.9	16.3	16.7	18.8

Table 3.17 shows that the best metrics are from "Destroyed" class. As the results are close to each other, it is hard to say that either bbox AP or segm AP is better.

To get a better understanding of the results, inference is done on test images. Ground truth labels are given in Figure 3.19.

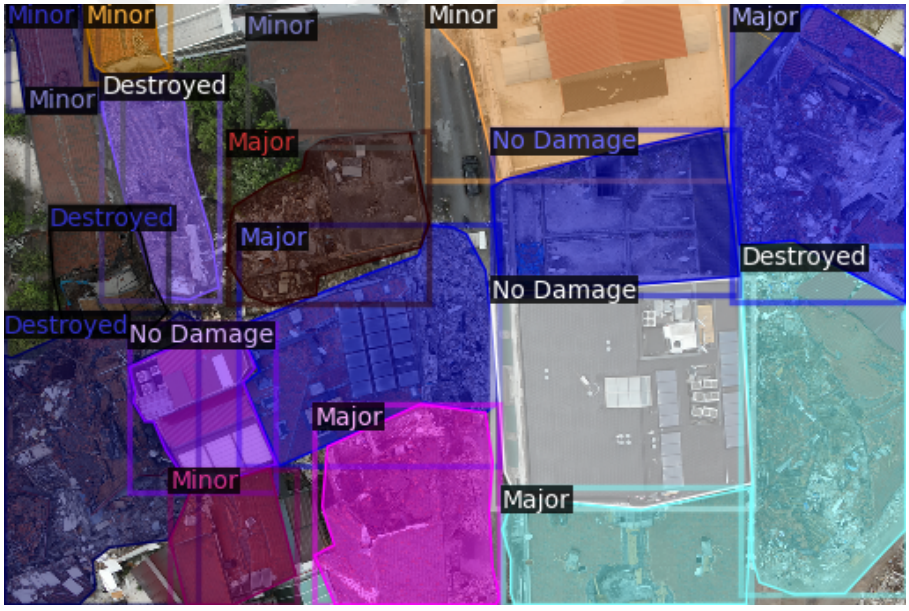


Figure 3.19 Ground truth labels for instance segmentation

BoundaryFormer model output is given in Figure 3.20.

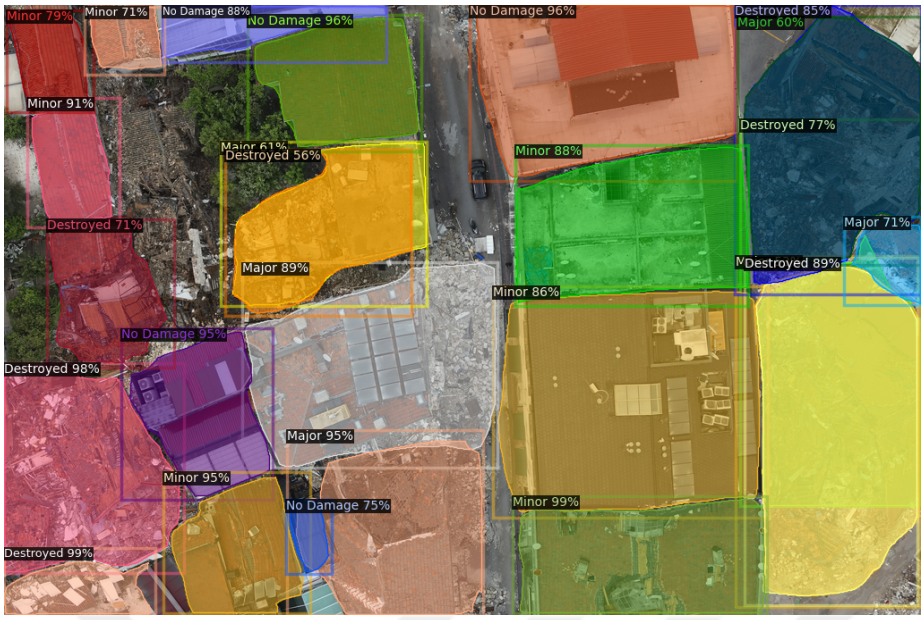


Figure 3.20 BoundaryFormer model inference outputs on test image

Comparing the ground truth to model outputs, the model can generally detect buildings. The objects are detected separately, providing deep understanding of the damage. Building count and the class distribution information can be extracted from inference, meaning the information gained can be useful for rescue operations after earthquake.

Same as in semantic segmentation, “Minor” and “No Damage” is predicted as each other frequently causing the metrics to go down.

On rare occasions, some buildings in image are detected twice. Mis-interpreted buildings are in multiple bounding boxes causing wrong classifications.

One big problem that is crucial for damage assessment is that some destroyed buildings are not detected by the model. This might be due to extra ordinary shapes of the destroyed buildings as the other building classes share similar shapes. This problem can be seen better in Figure 3.21



Figure 3.21 BoundaryFormer model missing buildings on inference

In Figure 3.21, some destroyed buildings are not detected. Common properties of these buildings can be listed as: having a colour similar to roads, being too closely located to other destroyed buildings, having shapes that are difficult to be detected by bounding boxes.

Multiple examples from BoundaryFormer model are given in Figure 3.22. In left side the original images with true labels are given. On the right-side model output is given. Each colour represents one building and class labels and their confidence values are given in box corners.

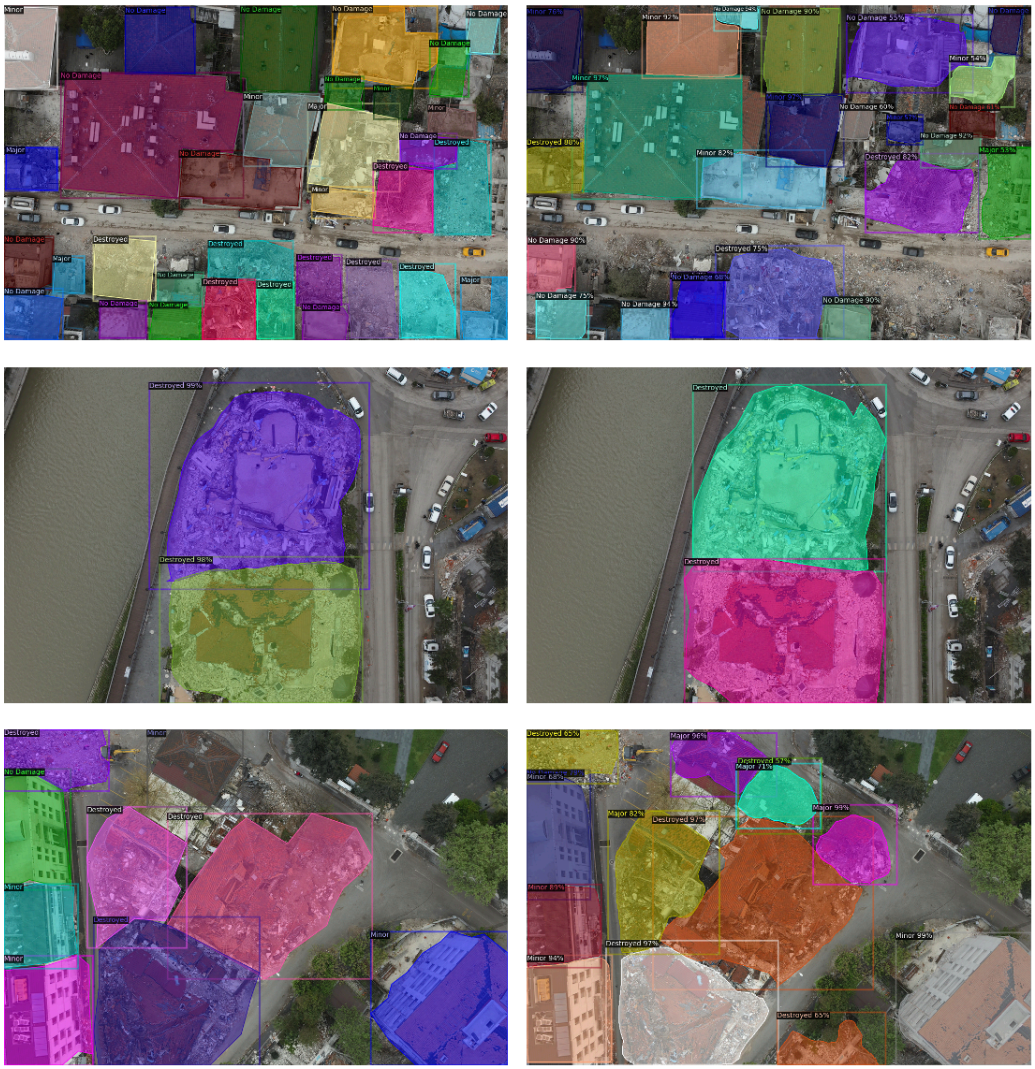


Figure 3.22 Examples from model From left to right: True labels, BoundaryFormer outputs

3.3.5.2 Grouping Classes for Instance Segmentation

In this section, the same process for grouping classes as in Semantic Segmentation is applied.

- **Stable and Damaged Classes**

After groping the classes, the newly created train set has 4298 stable and 1949 damaged buildings. As for test set, it has 704 stable, 342 damaged buildings.

The results for BoundaryFormer model trained on 2-class-dataset is given in Table 3.18.

Table 3.18 BoundaryFormer bounding box and segmentation results for 2 class labels

	AP	**AP50**	**AP75**
BBox	0.244	0.479	0.224
Segm	0.245	0.475	0.228

Results from Table 3.18 show better metrics compared to 4 class case. The mean AP for BBox is 0.244 and for Segm is 0.245. Class distribution is given in Table 3.19.

Table 3.19 BoundaryFormer model results per class for 2 class labels

	AP (%)	
	Stable	**Damaged**
BBox	25.3	23.4
Segm	26.6	22.4

Considering class AP values in Table 3.19, the detection of "Stable" and "Damaged" classes are quite similar. However, "Stable" class shows higher AP. This is due to the fact that the model find it easier to classify detected building as they belong to either of the 2 classes.

In Figure 3.23, ground truth labels for 2-classes are given. The buildings are labeled with "Stable" and "Damaged" classes.



Figure 3.23 2-class ground truth labels in test image

In Figure 3.24 the predictions of BoundaryFormer model is given.

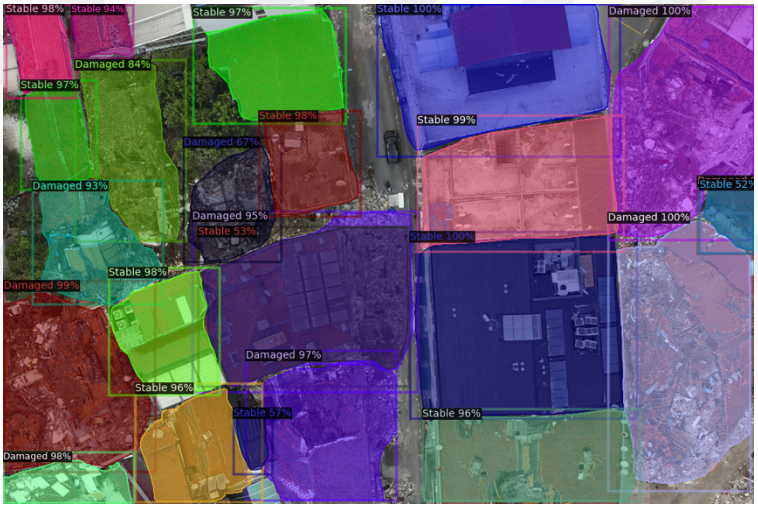


Figure 3.24 2-class BoundaryFormer predictions

Compared to 4-class case, 2-class case detects the buildings better. The missing damaged building count is lower in this case. There are still multiple detections for one object even though it is a rare occurrence. As the metrics show, 2-class case show better performance as the model is not struggling with classification of medium-degree-damage levels.

Multiple examples from BoundaryFormer model with 2 classes are given in Figure 3.25. In left side the original images with true labels are given. On the right-side model output is given. Each colour represents one building and class labels and their confidence values are given in box corners.

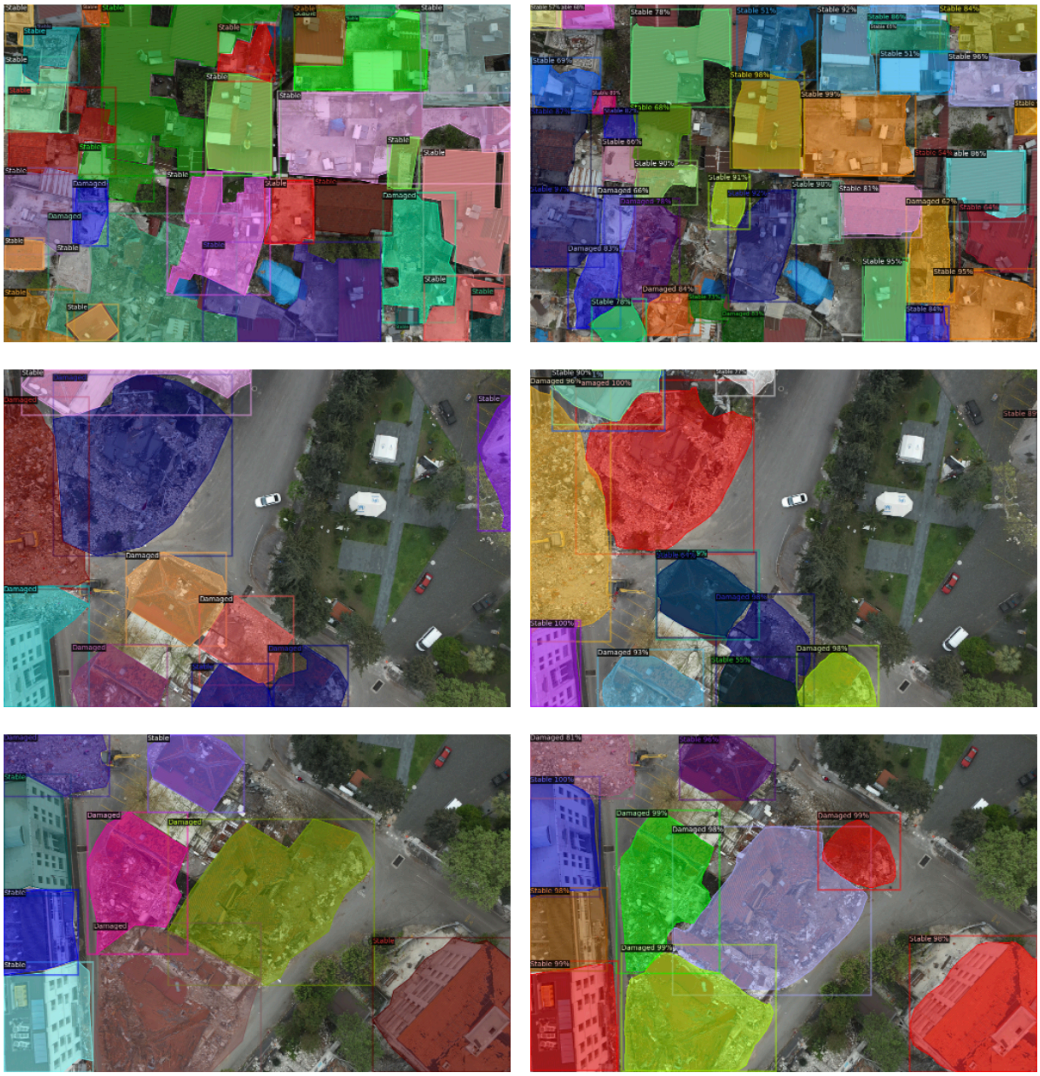


Figure 3.25 Examples from mode with “Stable” and “Damaged” classes. From left to right: True labels, BoundaryFormer output

- **Damaged Class Only**

After eliminating 2 classes and combining “Major” and “Destroyed” class into 1 class of “Damaged” the dataset contains 1949 objects in train from 364 images. The image count decreases since some images have neither “Major” nor “Destroyed” buildings.

The results for BoundaryFormer model trained on 1 class is given in Table 3.20. As there is only 1 class, there is no class distribution table.

Table 3.20 BoundaryFormer bounding box and segmentation results for "Damaged" class only

	AP	**AP50**	**AP75**
BBox	0.215	0.450	0.172
Segm	0.217	0.443	0.189

Examining Table 3.20, the results show worse performance than 2 class version. Mean AP for bbox is 0.215 and mean AP for segm is 0.217. Unlike semantic segmentation model, making the dataset 1 class only did not result in better results for all thresholds. Examining AP-50 shows that with lower threshold, the results are better compared to 2-class case. This might be because of instance segmentation's detection section. Less objects result in harder detection of objects causing the model to miss some buildings.

In Figure 3.26, the ground truth labels, and model predictions are given. When the results on inference images are compared to previous versions, the result show similar outputs for detecting the buildings.

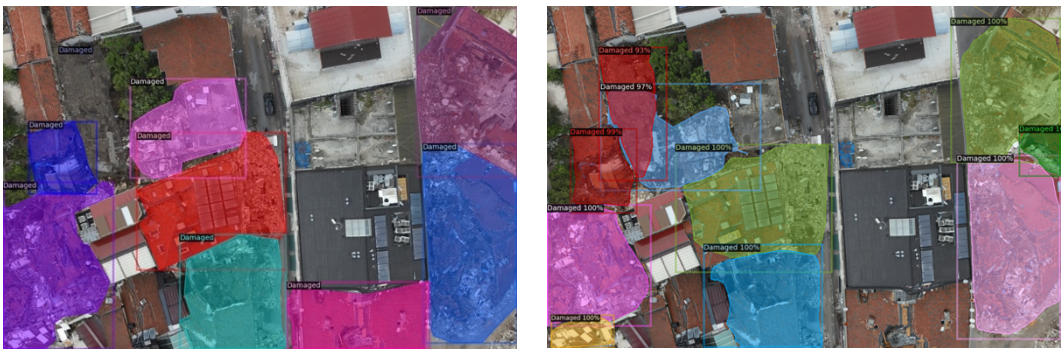


Figure 3.26 1-class ground truth labels and model output

Multiple examples from BoundaryFormer model with 2 classes are given in Figure 3.27. In left side the original images with true labels are given. On the right-side model output is given. Each colour represents one building and class labels and their confidence values are given in box corners.

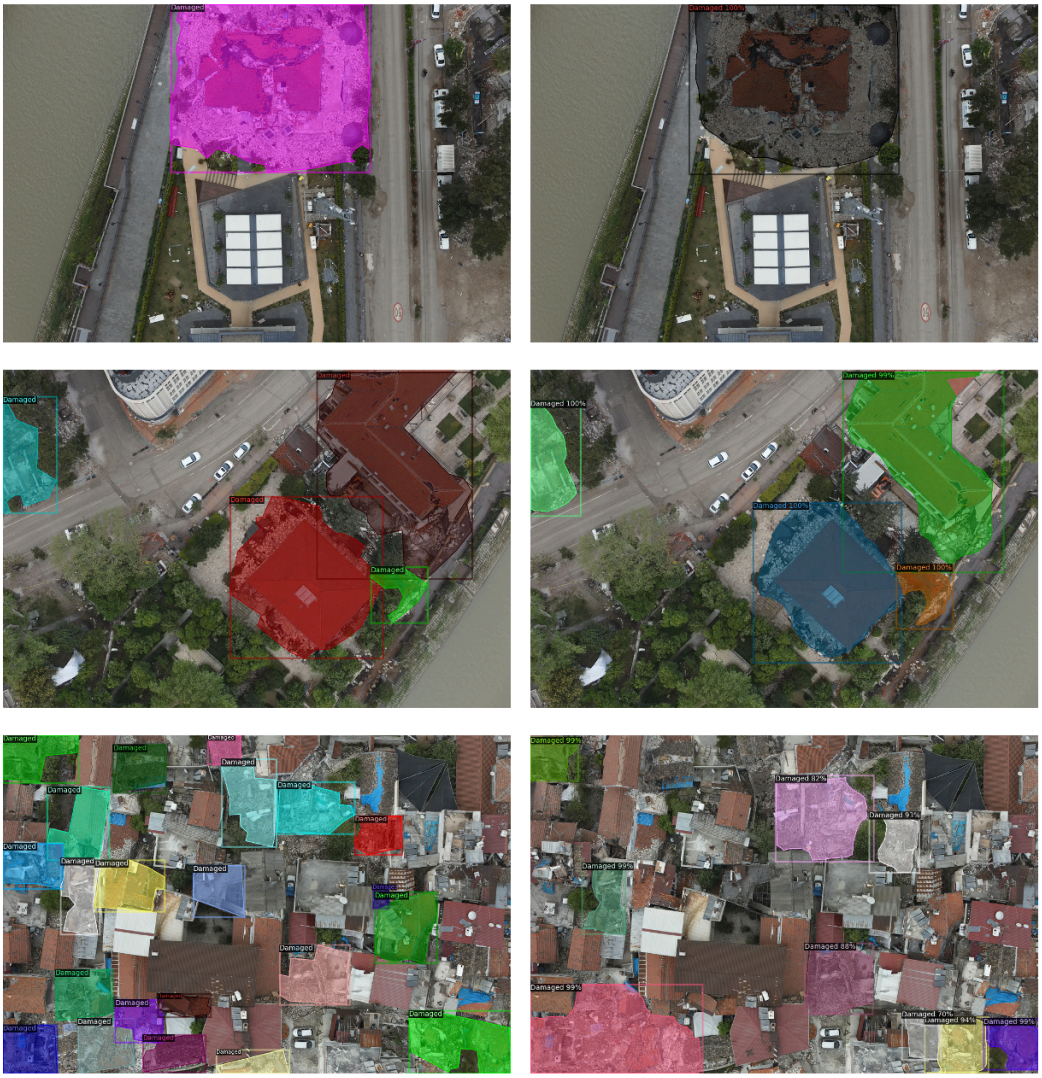


Figure 3.27 Examples from model with "Destroyed" class only. From left to right: True labels, BoundaryFormer outputs

3.4 Comparing Semantic and Instance Segmentation Best Models For Damaged Building Detection

After applying both semantic and semantic segmentation methods on damaged building drone image dataset, the outputs are compared. In Figure 3.28 model outputs for both methods are given. The first image in Figure 3.28 is the output of the BoundaryFormer instance segmentation model. Each detected building is in different colour and shown in bounding boxes. For each bounding box, a class label with confidence value is written. In second image, the output for SegFormer semantic segmentation model is given. As semantic segmentation labels each pixel separately, buildings are not separated perfectly. The colors green, yellow, blue, red

represent class labels “No Damage”, “Minor Damage”, “Major Damage”, “Destroyed” respectively.

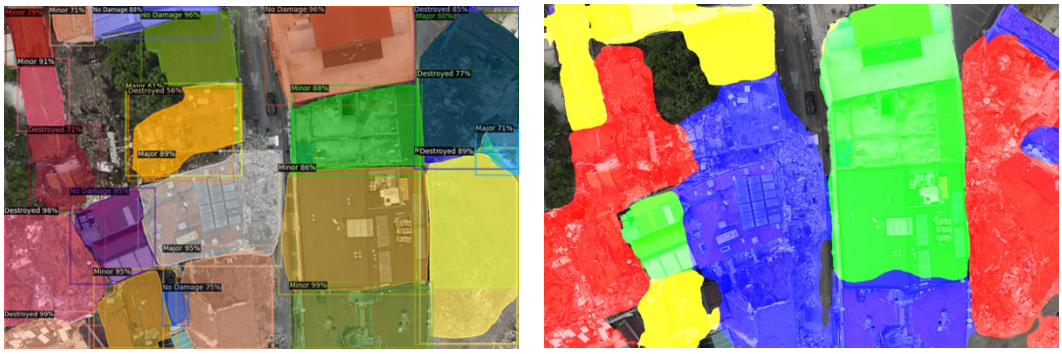


Figure 3.28 Comparison of semantic and instance segmentation outputs

The images in dataset contain many buildings closely located to each other. Also, the bounding boxes in labeling process are created automatically by the polygons segmented. This means that some bounding boxes must have multiple objects/buildings inside them. As instance segmentation uses bounding boxes, the model’s bounding box success is affected. Semantic segmentation does not make use of bounding boxes as it only labels pixels, so it does not get affected.

While bounding boxes may seem to cause problems as mentioned above, they provide extra information. Instance segmentation’s main advantage is to be able to differentiate same class objects into partitions. Object detection part in instance segmentation provides the extra dimension to detecting damaged buildings which semantic segmentation does not have. This causes different problems in semantic segmentation. As it does not detect different buildings, the model labels pixels belonging to one building differently. This means one building has multiple labels which does not exist in ground truth labels. Especially “Major Damage” class has buildings half damaged and half stable. These kinds of buildings get labeled by multiple classes resulting in false classification and bad accuracy.

Another advantage of semantic segmentation is that all pixels in image gets labeled. This means that each pixel goes through the process of classification and the only reason for missing buildings is classifying a pixel wrongly as background. On the other hand, instance segmentation detection process might miss buildings resulting in missing out on classifying some buildings. This especially becomes crucial for

evaluating overall damage after earthquakes, as it is important to not miss destroyed buildings.

Instance segmentation provides more detailed information compared to semantic segmentation about objects in the image. Examining the model outputs, semantic segmentation provides areas according to their damage degree while instance segmentation provides the number of buildings in each class as well as the boundaries of each building. After earthquakes, it might be crucial to have information about the number of buildings and their damage degree.

Generally comparing the two methods, it can be said that there is a tradeoff between accuracy and detail. Semantic segmentation performs better detecting the damaged buildings while instance segmentation provides more information regarding the building count.

3.5 Combining Semantic and Instance Segmentation Models for Enhancing Detection of Damaged Buildings

In section 3.4, the model outputs for semantic and instance segmentation are compared. The comparison results show that both methods have advantages and disadvantages over each other. To reiterate briefly, the main advantage of semantic segmentation is that it labels each pixel separately making it less possible to miss damaged buildings. On the other hand, the main advantage of instance segmentation is that the model extracts each building separately, having that extra dimension of information.

In Figure 3.29 the images from are combined using the proposed method. The left image is output of semantic segmentation model, the middle image is the output of instance segmentation model and the left image is the combined image. From the resulting image, both inferences from semantic and instance segmentation methods can be seen. The segments with outlines are the result of instance segmentation giving information about the number of buildings. They also provide information to semantic labels to classify one building with one class only. With proposed method, the buildings previously labeled with multiple labels from semantic segmentation are now labeled with one class only thanks to information from instance segmentation model.

While the proposed method provides improvement, it does not solve all the problems the models have. The buildings with no instance segment counterparts still remain with multiple labels. On top of this, the pixels out of the segment borders remain as previous labels causing noise. Another point is that the wrongly labeled damage degrees might cause the buildings to get labeled with worse damage degree than they already are.

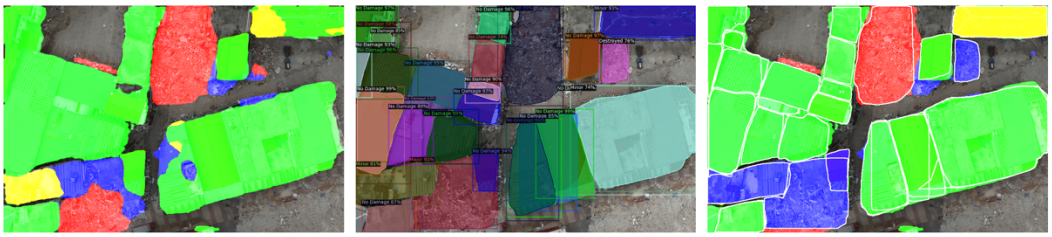


Figure 3.29 The resulting image for enhancement method

To examine the method in more depth, one building example is given in Figure 3.30. In the figure, resulting bounding box from instance segmentation model and pixel labels from semantic segmentation model for one building is given. After the proposed method is applied, the output can be compared to ground truth. The method successfully corrects the wrong pixels for the specific building in this case.

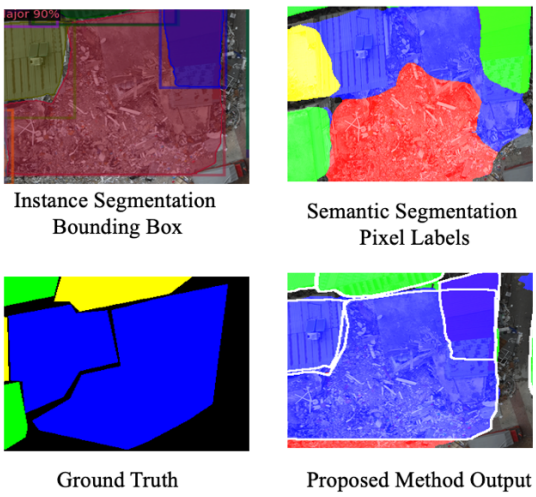


Figure 3.30 Successful building example for proposed method

On the other hand, when the buildings are detected incorrectly by the instance segmentation model this may cause problems. This is because when two buildings get detected as one, more pixels than needed get relabeled. Another important point is that when semantic segmentation model misclassified pixels in a building as

higher damage degree, whole building gets relabeled wrongly. These cases can be seen in Figure 3.31.

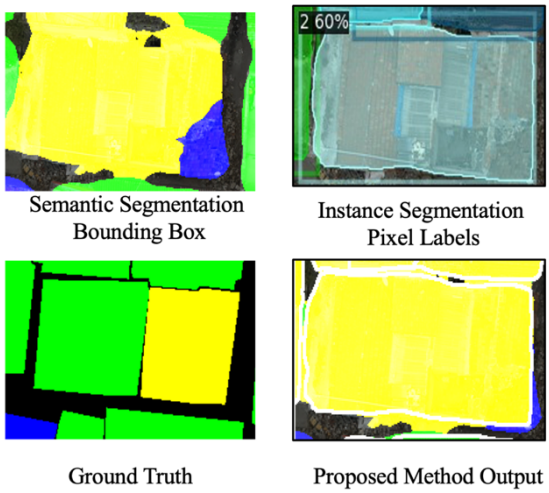


Figure 3.31 Failed building example for proposed method

The result table for enhanced method is given in Table 3.21.

Table 3.21 Enhanced method accuracy and IOU results

		Class						
		No Damage	**Minor**	**Major**	**Destroyed**	**Back-ground**	**Mean**	**Seg-Former Mean**
Metric	**Accuracy**	0.38	0.40	0.57	0.67	0.72	0.72	0.72
	IOU	0.32	0.30	0.34	0.50	0.68	0.43	0.51

As seen from Table 3.21, accuracy for the new method is the same as semantic segmentation results given in Table 3.6 while the mIOU is lower. The proposed method shows better result for accuracy. Failure cases mentioned below cause the drop in mIOU. While the mIOU is lower it should also be considered that the proposed method also enables extra information about the count of the building thanks to instance segmentation. On top of this, the success of the model is dependent on the performance of semantic and instance segmentation models.

3.6 Comparing Model Results with Human Inference

The success of the model in this problem relies on how distinguishable the degrees of damage of buildings are in the images. To compare the models in this work with humans a survey is conducted. A randomly selected group of 116 buildings are shown to 4 individuals, and they are asked to classify the damaged buildings. After explaining each damage class, 116 different buildings in images are shown to 4 different people. The buildings' boundaries are shown, and the damage degree is asked. After their input, the same buildings are fed into the proposed method with boundaries. These results are then compared with the system's outputs.

Examples of the survey are given in Table 3.22. P1, P2, P3 and P4 shows each person's inference and GT is the ground truth.

Table 3.22 Survey Examples for Human – Model Inference Comparison

Building	P1	P2	P3	P4	Model	GT
	No Damage	No Damage	No Damage	No Damage	No Damage	No Damage
	No Damage	No Damage	No Damage	Minor	Minor	No Damage
	Destroyed	Major	Major	Destroyed	Destroyed	Destroyed
	Major	Major	Major	Minor	Major	Major
	No Damage	Minor	No Damage	Minor	No Damage	No Damage

As results of the survey, the average accuracy of humans is found to be 0.52 and the accuracy of the model is found to be 0.77. The reason for humans' inference

results to be lower is because the decisions are dependent on the interpretation of each individual especially for buildings with small damage differences. Nevertheless, the fact that humans' inference results being higher than the model's output indicates that the model's performance cannot get better.

4
CONCLUSION

4.1 Conclusion

In this work, deep learning methods are used for detecting damaged buildings from drone images after earthquakes. For rescue operations after earthquakes, it is vital to detect the damaged areas. To fulfill this goal, firstly a dataset from drone images is created. The drone images taken from Hatay after February 6th Kahramanmaraş Earthquake are used. The buildings in images are labeled according to their damage degree in 4 classes. After dataset creation, deep learning methods are used to perform damage detection. Two approaches are used: semantic segmentation and instance segmentation. Three models are tested and the best ones according to their performance and metrics are chosen. For semantic segmentation, the best model SegFormer mIOU results are 0.51, 0.68 and 0.73 for 4 damage degree classes, 2 damage degree classes and 1 damage degree -damaged- class respectively. For instance segmentation the best model BoundaryFormer segmentation AP results are 0.17, 0.25 and 0.22 for 4 damage degree classes, 2 damage degree classes and 1 damage degree -damaged- class respectively For bounding box the results are 0.17, 0.24 and 0.22 respectively.

As given in results, different groupings of classes are performed in models. Firstly the 4-class case for 4 different damage degrees is used as dataset. Secondly, combining 4 classes into 2 classes, stable and damaged buildings are classified. And finally, only 1 class for damaged buildings is used for training.

Examining the results, for semantic segmentation the less class count resulted in better performance. As less classes mean less decision-making semantic segmentation model's metrics became higher.

For instance segmentation, best results are achieved with 2-class case. After this, 1-class case had the best results. 4-class case is worst performing case. The reason for 1-class case not being the best performer might be due the fact that destroyed

buildings have indefinite shapes causing bounding boxes to include nearby buildings as well.

In both cases 4-class case had the worst results. This is causes by similarities between buildings in different damages degrees. As all objects are buildings in essence, it is hard for models to differentiate small details between classes.

Even though 1-class grouping shows good results, it causes loss of information. Because the models do not detect stable buildings, there is no information about the ratio of damaged buildings to non-damage buildings.

In conclusion, a balance exists for classifying buildings in damage degrees. Small details might be misclassified models. On the other hand, less classes means less information for damaged regions. To dive further into this study, different labeling techniques, like only segmenting debris, can be done.

In future work, when determining the extent of damage to a building during the labeling process, a voting system can be implemented to reach a consensus. By utilizing this type of dataset, the outcomes can be analyzed to assess the impact of annotations.

The results show that drone images are suited for overall damage detection after earthquakes. As drone images contain more detailed information unlike satellite images, it is possible to classify multiple damage types. On the other hand, some handicaps like rain, snow and other weather conditions may disrupt the detection of damaged buildings in drone images.

One thing to consider while using drone images is that some buildings might collapse vertically. Vertically collapsed buildings might not be detected by deep learning models from drone images as the angle is not suitable. These cases might require different angled images for deep learning models to learn.

Comparing semantic and instance methods for damaged building detection, both methods have pros and cons. Overall, semantic segmentation shows better result for detecting the buildings as it labels each segment. On the other hand, instance segmentation gives better information as it also detects each. Detecting building separately makes it possible to count the buildings as well. After performing semantic and instance segmentation and comparing the two methods, a novel algorithm to combine both methods to enhance building detection is developed. The

proposed method is aimed to solve main issues for both tasks and focus on the advantageous sides. To reduce misclassified pixels and gain information from detected buildings; the pixels obtained from the semantic segmentation and the building boundaries obtained from the instance segmentation are combined. While the method improves main problems like labeling one building with multiple classes, the success is dependent on the performance of instance and semantic segmentation models.

The models show promising results for evaluating overall damage after earthquake as they are successful in detecting damaged buildings. As for detecting small details and classifying buildings into more classes, the models struggle to differentiate small differences like “No Damage” and “Minor Damage” classes, caused by the minimal difference between the two classes. Considering the fact that wide area damage assessment is a priority for rapid rescue and search operations post-earthquakes, the results show that the models in this work would be fit to be used in real world scenarios.

REFERENCES

[1] K. Rehdanz, , H. Welsch, D. Narita, and T. Okubo, “Well-being effects of a major natural disaster: The case of Fukushima,” Journal of Economic Behavior & Organization, vol. 116, pp. 500-517, 2015.

[2] J. F. Bird, J. J. Bommer, "Earthquake losses due to ground failure," Engineering geology, vol. 75.2, pp. 147-179, 2004.

[3] F. Finazzi, "Fulfilling the information need after an earthquake: statistical modelling of citizen science seismic reports for predicting earthquake parameters in near realtime," Journal of the Royal Statistical Society Series A: Statistics in Soc, vol. 183, pp. 857-882, 2020.

[4] Earthquake Magnitude Scale, https://www.mtu.edu/geo/community/seismology/learn/earthquake-measure/magnitude/, 10.06.2024.

[5] B. Adriano, N. Yokoya et al, "Learning from multimodal and multitemporal earth observation data for building damage mapping," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 175, pp. 132-143, 2021.

[6] K.A. Korkmaz, "Earthquake disaster risk assessment and evaluation for Turkey," Environmental Geology, vol. 57, pp. 307-320, 2009.

[7] INSARAG GUIDELINES 2020, https://www.insarag.org/methodology/insarag-guidelines/, 10.06.2024.

[8] J. Jia, W. Ye, "Deep Learning for Earthquake Disaster Assessment: Objects, Data, Models, Stages, Challenges, and Opportunities," Remote Sensing, vol. 15.16, pp. 4098, 2023.

[9] X. Lu, Y. Xu, et al, "A deep learning approach to rapid regional post-event seismic damage assessment using time-frequency distributions of ground motions," Earthquake Engineering & Structural Dynamics, vol. 50.6, pp. 1612-1627, 2021.

[10] X. Yuan, D. Tanksley, L. Li, H. Zhang, G. Chen, D. Wunsch, “Faster Post-Earthquake Damage Assessment Based on 1D Convolutional Neural Networks,” Applied Sciences, vol. 11, pp. 9844, 2021.

[11] Y. Yu, C. Wang, X. Gu and J. Li, "A novel deep learning-based method for damage identification of smart building structures," Structural Health Monitoring, vol. 18.1, pp. 143-163, 2019.

[12] C. Kyrkou, T. Theocharides, "EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 1687-1699, 2020.

[13] R. Gupta, B. Goodman, N. Patel, et al, "Creating xBD: A dataset for assessing building damage from satellite imagery," In Proceedings of the

IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 10-17, 2019.

[14] S.A. Chen, A. Escay, C. Haberland et al, "Benchmark dataset for automatic damaged building detection from post-hurricane remotely sensed imagery," arXiv preprint arXiv:1812.05581, 2018.

[15] M. Rahnemoonfar, T. Chowdhury, and R. Murphy, “RescueNet: a high resolution UAV semantic segmentation dataset for natural disaster damage assessment,” Scientific data, vol .10(1), pp. 913, 2023.

[16] B. Sirmacek, and C. Unsalan, "Damaged building detection in aerial images using shadow information," 2009 4th International Conference on Recent Advances in Space Technologies IEEE, pp. 249-252, 2009.

[17] Z. Yan, C.D. Sheng, and R.H. Zhong, "The research of building earthquake damage object-oriented segmentation based on multi feature combination with remote sensing image," Procedia Computer Science, vol. 154, pp. 817-823, 2019.

[18] Y. Bai, B. Adriano, et al, "Machine learning based building damage mapping from the ALOS-2/PALSAR-2 SAR imagery: Case study of 2016 Kumamoto earthquake," Journal of Disaster Research, vol. 12.sp, pp. 646-655, 2017.

[19] M.H. Rafiei, H. Adeli, "A novel machine learning-based algorithm to detect damage in high-rise building structures," The Structural Design of Tall and Special Buildings, vol. 26.18, pp. e1400, 2017.

[20] S. Naito, H. Tomozawa, et al, "Building-damage detection method based on machine learning utilizing aerial photographs of the Kumamoto earthquake," Earthquake Spectra, vol. 36.3, pp. 1166-1187, 2020.

[21] S. Mangalathu, H. Sun, et al, "Classifying earthquake damage to buildings using machine learning," Earthquake Spectra vol. 36.1, pp. 183-208, 2020.

[22] Y. LeCun, and Y. Bengio, "Convolutional networks for images, speech, and time series," The handbook of brain theory and neural networks, vol. 3361.10 1995.

[23] B. Kalantar, N. Ueda, et al, "Assessment of convolutional neural network architectures for earthquake-induced building damage detection based on pre- and post-event orthophoto images," Remote Sensing, vol. 12.21, pp. 3529, 2020.

[24] C. Wu, F. Zhang, et al, "Building damage detection using U-Net with attention mechanism from pre-and post-disaster remote sensing datasets," Remote Sensing, vol. 13.5, pp. 905, 2021.

[25] M. Presa-Reyes, and S. C. Chen. "Assessing building damage by learning the deep feature correspondence of before and after aerial images," 2020 IEEE conference on multimedia information processing and retrieval (MIPR) IEEE, pp. 43-48, 2020.

[26] S. Ghaffarian, N. Kerle, et al, "Post-disaster building database updating using automated deep learning: An integration of pre-disaster OpenStreetMap and multi-temporal satellite data," Remote sensing, vol. 11.20, pp. 2427, 2019.

[27] C. Xiong, Q. Li, X. Lu. “Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network,” Automation in Construction, vol. 109, pp. 102994, 2020.

[28] Y. Qing, D. Ming, Q. Wen, et al, “Operational earthquake-induced building damage assessment using CNN-based direct remote sensing change detection on superpixel level,” International Journal of Applied Earth Observation and Geoinformation, vol. 112, pp. 102899, 2022.

[29] Z. Hong, H. Zhong, et al, “Classification of building damage using a novel convolutional neural network based on post-disaster aerial images,” Sensors, vol. 22.15, pp. 5920, 2022.

[30] T. Ci, Z. Liu and Y. Wang, “Assessment of the degree of building damage caused by disaster using convolutional neural networks in combination with ordinal regression,”, Remote Sensing, vol. 11.23, pp. 2858, 2019.

[31] D. Duarte, F. Nex, N. Kerle and G. Vosselman, “Multi-Resolution Feature Fusion for Image Classification of Building Damages with Convolutional Neural Networks,” Remote Sens, vol. 10, pp. 1636, 2018.

[32] Y. Wang, W. Feng, K. Jiang, Q. Li, R. Lv, J. Tu, "Real-Time Damaged Building Region Detection Based on Improved YOLOv5s and Embedded System From UAV Images,", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 4205, 2023.

[33] Y. Zhan, W. Liu, and Y. Maruyama, "Damaged building extraction using modified Mask R-CNN model using post-event aerial images of the 2016 Kumamoto earthquake," Remote Sensing, vol. 14.4, pp. 1002, 2022.

[34] K. He, G. Gkioxari, et al, "Mask r-cnn," Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.

[35] D. Song, X. Tan, B. Wang, L. Zhang, X. Shan and J. Cui, "Integration of super-pixel segmentation and deep-learning methods for evaluating earthquake-damaged buildings using single-phase remote sensing imagery," International Journal of Remote Sensing, vol. 41.3, pp. 1040-1066, 2020.

[36] F. Xiong, H.Wen, et al, “Semantic segmentation recognition model for tornado-induced building damage based on satellite images,” Journal of Building Engineering, vol. 61, pp. 105321, 2022.

[37] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation," Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III, pp. 234-241, 2015.

[38] A. Dosovitskiy, L. Beyer, et al, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[39] Y. Xie, D. Feng, H. Chen, et al, "Damaged building detection from post-earthquake remote sensing imagery considering heterogeneity characteristics," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-17, 2022.

[40] T. Li, C. Wang, F. Wu and X. Zheng, “DDFormer: A dual-domain transformer for building damage detection using high-resolution SAR

imagery,” IEEE Geoscience and Remote Sensing Letters, vol. 20, pp. 1-5, 2023.

[41] E. Xie, W. Wang, et al, "SegFormer: Simple and efficient design for semantic segmentation with transformers," Advances in neural information processing systems, vol. 34, pp. 12077-12090, 2021.

[42] J. Lazarow, W. Xu, and Z. Tu, "Instance segmentation with mask-supervised polygonal boundary transformers," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4382-4391, 2022.

[43] J. Jain, J. Li et al, "Oneformer: One transformer to rule universal image segmentation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2989-2998, 2023.

[44] A. Munappy, J. Bosch, et al, “Data management challenges for deep learning,” In 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) IEEE, pp. 140-147, 2019.

[45] L. Dong, J. Shan. "A comprehensive review of earthquake-induced building damage detection with remote sensing techniques," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 84, pp. 85-99, 2013.

[46] Label Studio, https://labelstud.io, 10.06.2024.

[47] S. Minaee, Y. Boykov, et al, "Image segmentation using deep learning: A survey," IEEE transactions on pattern analysis and machine intelligence, vol. 44.7, pp. 3523-3542, 2021.

[48] A. Vaswani, N. Shazeer, et al, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[49] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection & segmentation," Proceedings of the IEEE international conference on computer vision, pp. 4950-4959, 2017.

[50] B. Li, S. Liu, W. Xu, W. Qiu, "Real-time object detection and semantic segmentation for autonomous driving," MIPPR 2017: Automatic Target Recognition and Navigation, vol. 10608, pp. 167-174, 2018.

[51] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," Neurocomputing, vol. 406, pp. 302-321, 2020.

[52] R. Girshick, "Fast r-cnn," Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.

[53] I. Ulku, E. Akagündüz, "A survey on deep learning-based architectures for semantic segmentation on 2d images," Applied Artificial Intelligence, vol. 36.1, pp. 2032924, 2022.

[54] S. Mehta, M. Rastegari et al, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," Proceedings of the european conference on computer vision (ECCV), pp. 552-568, 2018.

[55] COCO, https://cocodataset.org/#detection-eval, 10.06.2024.

[56] Z. Liu, Y. Lin, et al, "Swin transformer: Hierarchical vision transformer using shifted windows," Proceedings of the IEEE/CVF international conference on computer vision, pp. 10012-10022, 2021.

[57] A. Buslaev, VI Iglovikov, et al, "Albumentations: Fast and Flexible Image Augmentations," Information, vol. 11, 2020.

[58] A. Hassani, and H. Shi, "Dilated neighborhood attention transformer," arXiv preprint arXiv:2209.15001, 2022.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. Gürer B., Karslıgil M.E., “Deprem Sonrası Drone Görüntülerinden Derin Öğrenme ile Hasarlı Bina Tespiti”, SIU 2024 32. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı