REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

EARTHQUAKE DAMAGE DETECTION FROM IMAGES WITH DEEP LEARNING METHODS

18011057 — Berk SICAK
19011916 — Kemal SELÇUK

SENIOR PROJECT

Advisor
Prof. Dr. Mine Elif KARSLIGIL

June, 2023

ACKNOWLEDGEMENTS

From the start date of our project to the delivery date; We would like to express our gratitude and respect to our project consultant Prof. Mine Elif KARSLIGİL, who always gave his effort and support, contributed to our development both in our studies and in our individual development, and did not leave our questions unanswered.

Berk SICAK
Kemal SELÇUK

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once(a Convolutional Neural Network Architecture)
CNN	Convolutional Neural Network
HGM	Harita Genel Mudurlugu
GPU	Graphical Process Unit
SSD	Single-Shot Multibox Detector
R-CNN	Region-based Convolutional Neural Network
GDBDA	Ground-level Detection in Building Damage Assessment Dataset
CBAM	Convolutional Block Attention Module
Bi-FPN	Bi-Directional Feature Pyramid Network
CAE	Image Convolution Auto-Encoder
vCPU	Virtual Central Processing Unit
RAM	Random Access Memory
vRAM	Virtual Random Access Memory

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

EARTHQUAKE DAMAGE DETECTION FROM IMAGES WITH DEEP LEARNING METHODS

Berk SICAK
Kemal SELÇUK

Department of Computer Engineering
Senior Project

Advisor: Prof. Dr. Mine Elif KARSLIGIL

One of the biggest problems of mankind at all times, one of the most dangerous natural disasters, is earthquake. A lot of research is being done on the earthquake, engineers and scientists have searched for a solution. However, no solution has been found to prevent this natural disaster. Pre-earthquake precautions is increased, and progress continues in areas such as post-earthquake coordination and rescue.

This project was carried out in order to minimize post-earthquake losses and to react quickly. The destroyed building was determined by the aerial UAV and drone images. This topic is a complex object recognition project. The reason why it is complex is that the details and textures are similar to each other in the images taken from high between the collapsed buildings and the ones that are not collapsed. Previously similar projects were examined and it was decided that the most effective solution was deep learning algorithms.

Deep learning algorithms are increasing their speed and accuracy day by day. YOLO has been accepted as the first model to be tried in object recognition and classification projects, works faster than its competitors and gives good results. In this project, YOLOv8, the version of YOLO released in 2023, was used and YOLOv7 was also used for comparison purposes.

In the study, it was researched to determine the places where there might be a need for rescue in the fastest way after an earthquake disaster. Deep learning methods, data collection, data labeling, data augmentation and hyperparameter optimization have been studied.

Keywords: Earthquake, Natural Disaster, UAV, Deep Learning, Convolutional Neural Network, Dataset Preparation, Data Augmentation.

ÖZET

DERİN ÖĞRENME YÖNTEMLERİYLE GÖRÜNTÜLERDEN DEPREM HASAR TESPİTİ

Berk SICAK
Kemal SELÇUK

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Prof. Dr. Mine Elif KARSLIGİL

İnsanoğlunun her zaman en büyük sorunlarından biri, doğal afetlerin en tehlikelilerinden biri depremdir. Deprem üzerin bir çok araştırma yapılmaktadır, mühendisler, bilim adamları çözüm aramışlardır. Ancak bu doğal afeti engelleyebilecek bir çözüm bulunamamıştır. Deprem öncesi önlemler arttırılmakta, deprem sonrası koordinasyon, kurtarma gibi alanlarda ilerlemeler devam etmektedir.

Bu proje de, deprem sonrası kayıpları minimuma indirmek, hızlı reaksiyon vermek amacıyla gerçekleştirilmiştir. Havadan alınan UAV ve drone görüntüleriyle, yıkılan bina tespiti yapılmıştır. Bu konu, complex bir nesne tanıma projesidir. Complex olmasının sebebi, yıkılan binalar ile yıkılmayan binalar arasında yüksekten çekilmiş görüntülerde, detayların azalması ve dokuların birbirine benzer olmasıdır. Önceden benzeri yapılmış projeler incelenip, en efektif çözümün derin öğrenme algoritmaları olduğuna karar verildi.

Derin öğrenme algoritmaları, gün geçtikçe hızını ve doğruluğunu arttırmaktadır. YOLO, nesne tanıma, sınıflandırma projelerinde ilk denenen, rakiplerine kıyasla hızlı çalışan ve iyi sonuçlar veren model olarak kabul görmüştür. Bu projede de YOLO’ nun 2023 yılında çıkan versiyonu YOLOv8 kullanılmış olup karşılaştırmak amaçlı YOLOv7 de kullanılmıştır.

Yapılan çalışamada, deprem afeti sonrası en hızlı şekilde kurtarılma ihtiyacının olabileceği yerleri saptamak konusu araştırılmıştır. Derin öğrenme metotları, veri

toplama, veri etiketleme, veri arttırma ve hiper parametre optimizasyonu konularında çalışılmıştır.

Anahtar Kelimeler: Deprem, Doğal Afet, İHA, Derin Öğrenme, Evrişimsel Sinir Ağı, Veri Kümesi Hazırlama, Veri Arttırımı.

1 Introduction

In earthquakes, the detection of debris is vital to save lives. Our motivation in the project is to determine the structures that are destroyed in the fastest and most accurate way in disaster situations such as earthquakes. In these cases, it was decided that using UAVs would give the best results for the fastest detection. The reason for detecting over UAVs is that satellite images are affected by adverse weather conditions and may cause delays.

Traditional machine learning methods are insufficient in tasks such as object detection and classification through images. Therefore, the use of deep learning methods was deemed appropriate in the project. Since the data we will use when training the model will be images, convolutional neural networks are the best method to extract useful information from it. Objects to be detected will be indicated with bounding boxes on the images, and the model will show the building that it has detected damage with bounding boxes on the image while making estimations. In this study, the YOLO neural network, which has been very popular in recent years, is used for real-time object detection. The factors in using this network were its speed, accuracy and successful results in previous studies on UAV images.

When training machine learning models, datasets are used for the model to learn patterns and information. In deep learning methods, which is a type of machine learning, the importance of using the right data set is very high. In the project, the data set was created with the images taken by the UAVs from the earthquake zone centered in Kahramanmaras/ Turkey on Feb/06/2023. The dataset consists of approximately 200 labeled UAV images and approximately 200 drone images.

1.1 Natural Disaster

Natural disasters are dangerous events that occur mostly beyond the control of people, can cause loss of life and property, and affect large masses. Natural disasters are grouped under 2 groups, these are; Slow-developing natural disasters: severe cold, drought, famine, etc. Rapidly developing natural disasters are: earthquake, flood, landslide, avalanche, storm, fire etc. Earthquakes are the type of natural disaster that have caused the most loss to human beings throughout history.

1.2 Earthquake

Earthquake can be termed as producing high level vibration by the sudden release of energy accumulated underground. Earthquakes happen very often, most of them are not felt by people, but some of them are high in size and they can even demolish buildings. Examples of these have occurred many times in Turkey and also other countries. Also earthquakes can

2
Literature Review

It is very important to quickly identify the places that are needed after the earthquake. In the [1] project, images after the 2010 Haiti earthquake and 3829 undamaged, 962 slightly damaged and 545 damaged labels were used as training data to identify damaged buildings. In the first stage, the Mask Region-based Convolutional Neural Network model was trained by dividing it into 3 classes with damaged, little damaged and no damaged, and the accuracy value was 58.62%. In the following trials, 83.53% success was obtained by dividing them into two groups, undamaged and damaged.

[2] demonstrates a technique for identifying building damage using terrestrial images and an enhanced YOLOv5. The researchers created the GDBDA dataset, which includes annotated images of debris, collapse, spalling, and cracks. They improved the YOLOv5 model's accuracy and detection speed by incorporating a lightweight Ghost bottleneck, CBAM, and Bi-FPN for multi-scale feature fusion. The model achieved over 90% detection accuracy for different types of damage and had the smallest weight size and fastest detection speed compared to other methods. The proposed model can be applied to different regions and satisfies the need for future lightweight embedding.

[3] proposes a damaged building assessment method using the SSD algorithm with pretraining and data augmentation. The method categorizes buildings as ruins, damaged, or undamaged and uses an unlabeled post-disaster CAE to initialize the SSD model's weights. Data augmentation techniques like image rotation, gaussian blur, and noise processing increase the training dataset. The model achieved an mAP value of 72% when looking at the results.

In a project running on UAV in real time, blurry and noisy images need to be estimated. Work has been done at [4] on real-time estimation of damaged areas in areas rejected by GPS. Using ResNet-101 as its core network performance, the faster R-CNN detects small and fuzzy defects in video frames captured by autonomous UAVs, with 93.31% mAP and average IOU of 92.16%.

3
System Analysis and Feasibility

3.1 System Analysis

In this section, the path to be followed in the development of the project is briefly stated. The data to be worked on in the project were obtained by taking screen recordings of the UAV images taken from the districts affected by the earthquake in Turkey via the HGM Küre application and drone images taken in the regions. Afterwards, the collapsed buildings were labeled in these images. Traditional machine learning algorithms would fall short as the project work involved object detection on image data. Therefore, deep learning algorithms were used in the project.

3.2 Feasibility

In this part, technical, legal and financial researches were conducted within the scope of the project and the requirements were determined.

3.2.1 Technical Feasibility

The software and hardware requirements are specified in the following section.

3.2.1.1 Hardware Feasibility

To develop the project, a computer with the resources to run any browser (Safari, Google Chrome, Mozilla Firefox etc.) and any operating system (MacOS, Windows, Linux etc.) is required. Because deep learning models do a lot of processing during training, training periods lasting days or weeks can be encountered. Therefore, the training is completed with the GPU support provided by Google by using Google Colab over the browser.

Hardware features provided by Google Colab:

- an Intel Xeon CPU with 2 vCPUs,
- 83.5 GB of RAM, and
- NVIDIA A100 GPU with 40GB of VRAM.

3.2.1.2 Software Feasibility

While creating the data set in the project, HGM KURE application and Atlas Map website were used to collect UAV and UAV images. In addition, an image search was made on the internet via the browser. The browser can be any browser compatible with your operating system (Safari, Google Chrome, Mozilla Firefox etc). LabelStudio application and Roboflow website were used to label the images in YOLO format while preparing them for training.

Python was chosen as the programming language to be used for the development of the project. Python; It is used very often due to the rich libraries it provides in the fields of artificial intelligence, deep learning, machine learning and data visualization. As a result of the research done for the project, it was determined to use the PyTorch library. Since there will be used deep learning networks like YOLO, PyTorch is a good option since it has useful functions and is a relatively new library.

3.2.2 Time Feasibility

For the time management project, work will be carried out for a period of approximately 100 days. Article and report submission is planned from the beginning (March) to be approximately 3.5 months later (June). The tasks to be done will be completed by sharing skills, knowledge and time between the 2 people responsible for the project. Both people are equally responsible for the entire project.

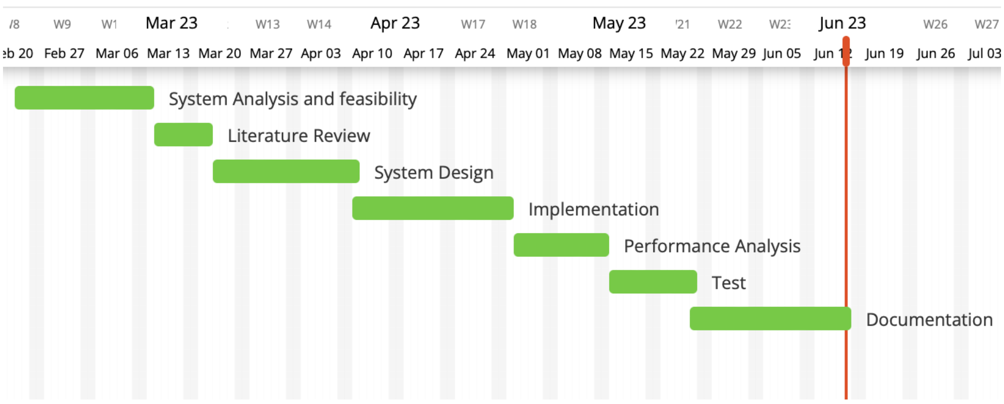

Figure 3.1 Gantt Chart

3.2.3 Economic Feasibility

using one computer per developer for expenditures throughout the project costs a total of approximately 25000 * 2 = 50000TL. It is calculated from 20000TL per developer, taking into account the salaries of the 2 developers working on the project. Developers work at half performance due to school, 20000TL * 0.5 developer/month * 2 people * 3.5 months = 70000TL total salary expense is calculated. Invoice prices seem to be 1000 TL. Google Colaboratory, PyTorch and Youtube are other tools used, they are calculated as 0 TL because they are free. When the total expense is calculated, it turns out to be 121000TL.

Category	Calculation
Hardware	2 computer * 20000TL
Software	2 Colab Pro Account * 168TL
Developer	0.5 developer/months * 2 developer * 3.5 months * 20000TL
Total	110336 TL

Table 3.1 Project Expenses

3.2.4 Legal Feasibility

The system complies with existing laws and regulations. It does not violate existing patents and intellectual industrial rights. A resource or method to be used in the system does not require any special license or permission.

4
System Design

In this section, the architecture used in the project is explained.

4.1 Data Annotation

Data annotation is one of the important steps of the training phase. Labels show the model where and what the important elements to learn are. In this study, since the collapsed building will be determined on a single class, collapsed buildings were identified on the images in the data set and annotated under the 'collapsed' class using LabelStudio and Roboflow's annotation tool [5]. Since YOLO models do not accept explanations with polygonal shapes, collapsed buildings are labeled with rectangular bounding boxes as seen in the figure 4.1.

Figure 4.1 Sample Annotated UAV image

4.2 Data Augmentation

Data augmentation, cropping, rotating, flipping, blurring, shifting, fading, scaling for images of existing training data samples; frequency masking, scaling for sounds; is to create synthetic copies of texts processed by methods such as random word deletion and synonym replacement.The main purpose of data augmentation is to support the model with more data and enable it to make more accurate predictions. With data augmentation, the model can also learn images taken from different angles, with noise or with different color levels. In addition, overfitting is avoided by differentiating similar images that may overlap with the images in the train dataset and real-world examples. In this study images were only flipped and rotated in different directions and added to the training set.

Figure 4.2 Image Flipped in y Direction

Figure 4.3 Image Flipped in x Direction

Figure 4.4 Image Flipped in x and y Directions

Figure 4.5 Image Rotated 90 degrees

4.3 YOLO

You only look once (YOLO) is a state-of-the-art, real-time object detection model. This model detects objects using convolutional neural networks (CNNs). The reason for its name is because of the working logic of the model. As the name suggests, this model performs the detection and classification process on the image in a single step by looking at an image only once. Briefly the advantages of the YOLO models are its high speed, high accuracy and its learning capabilities.

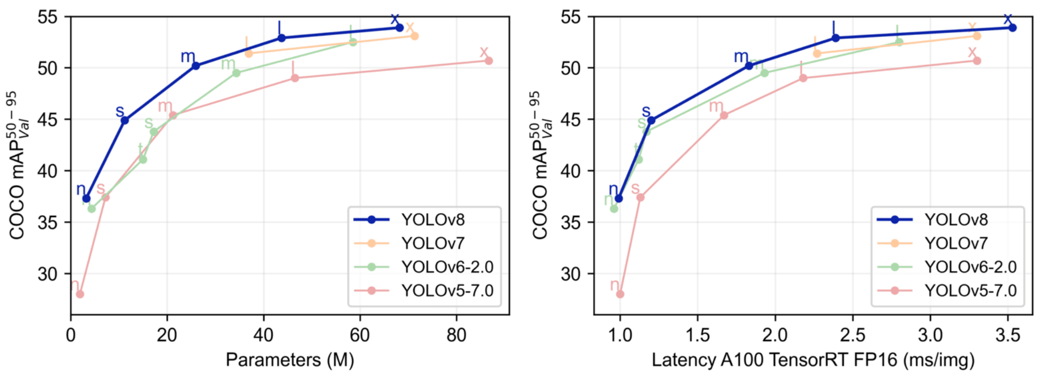

Figure 4.6 Comparison of YOLO versions [6]

The YOLO algorithm operates by splitting the picture into N tiny sections, each with an equal SxS size, to create the final image. These N areas each processes and questions objects existence there. The method creates the bounding box for the object if it is discovered inside the region and its center is within its area. However, duplicate predictions will be produced if an object is detected by multiple regions with different bounding box predictions. In this case, bounding boxes are created. The Non Maximum Suppression (NMS) approach is used to filter them.

4.3.1 YOLOv8

YOLOv8 is a state-of-art framework developed by Ultralytics and it introduced on January 10, 2023. As can be seen in the graph in Figure 5.1, it can make faster and more accurate predictions. It can perform the following tasks:

- Object Detection,
- Instance Segmentation, and
- Image Classification.

YOLOv8 provided five scaled versions:

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 4.7 YOLOv8 models [7]

According to Figure 5.2, as the model grows, the speed decreases, so model selection should be made according to the need.

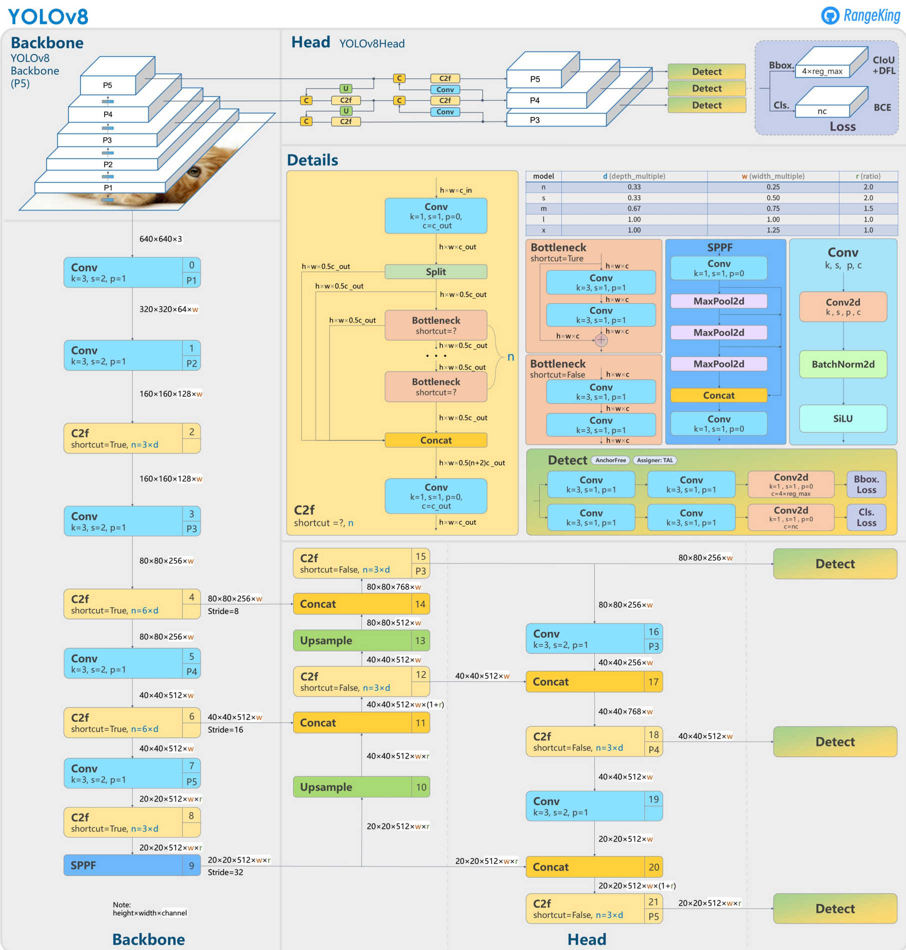

Figure 4.8 YOLOv8 architecture [7]

4.3.2 YOLOv7

YOLOv7 is one of the versions of a popular object detection approach known as "You Only Look Once". YOLOv7 has the ability to detect and classify objects using bounding boxes like other versions. YOLOv7 was introduced in 2022 by WongKinYiu and Alexey Bochkovskiy as a development based on previous YOLO versions.

YOLOv7 extracts features using convolutional neural networks (CNN) and then uses these features for classification and bounding box estimation. The model is usually trained on large datasets with a large number of images. Also, with transfer learning, it can achieve good results on small datasets with fewer images.

In addition, one of its most important features is that it can work on many devices even phones. Some of those:

- Neural compute stick or NCS (Intel),
- AI edge devices (Nvidia),
- Coral Edge TPU (Google),
- Apple neural engine (Apple), and
- Neural Processing Engine (Qualcomm).

It has been measured by experiments that you get more successful results than other object recognition models. It showed very good results from the working speed and performance trade off. For these reasons, it has become one of the most popular object recognition and classification models today.

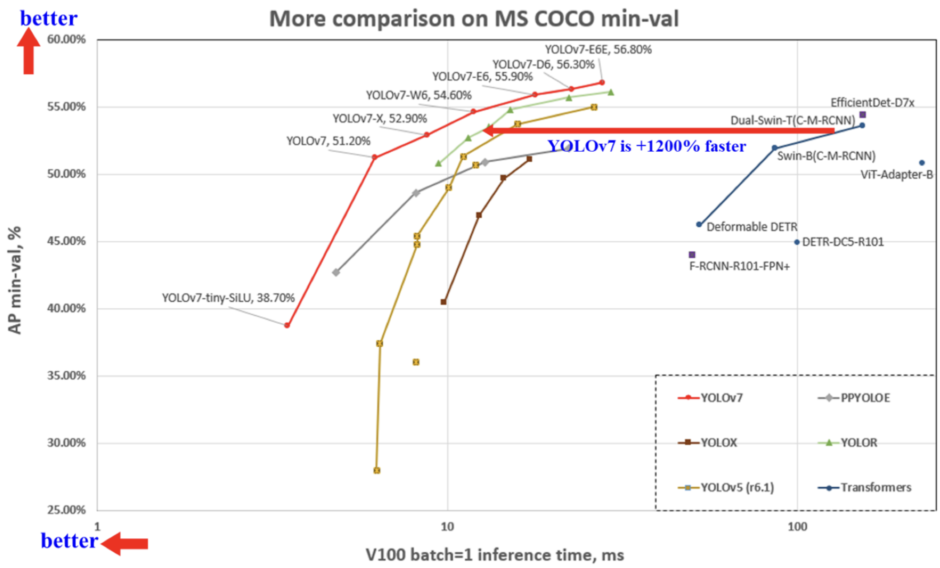

Figure 4.9 Comparison with other object detectors. [8]

4.4 Evaluation Metrics

In object recognition projects, different evaluation metrics such as mAp and IOU are used compared to other machine learning projects. The most commonly used of these is the mAP metric. There are multiple metrics used to obtain the mAP, these are IOU, precision and recall.

4.4.1 Intersection Over Union

The Intersection Over Union metric quantifies the degree overlap between two region. It evaluates to correction of the prediction. Values range from 0 to 1 where 1 completely overlaps and 0 means no intersection.

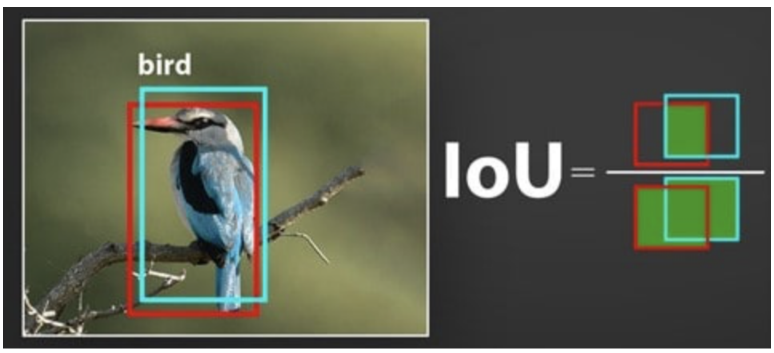

Figure 4.10 visualization of IOU [9]

With the determined IOU threshold value, it is decided that the prediction is true positive, false negative or false positive .

4.4.2 Confusion Matrix

Similarly, these terms apply to object detection. However, their exact meanings are not the same. In object detection, the accuracy of the prediction (TP, FP or FN) is decided with the help of the IoU threshold. In object detection, the decision is made by looking at ground truth, that is, known object pixels.

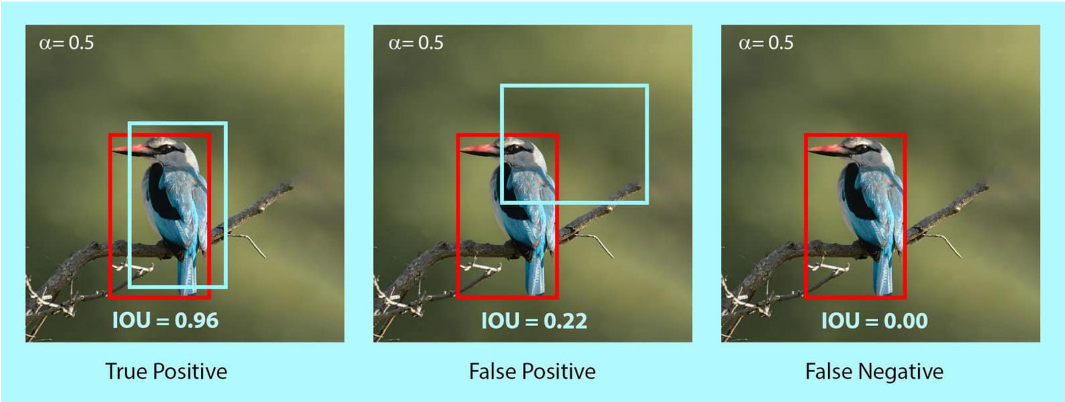

Figure 4.11 visualization of Confusion Matrix [9]

4.4.3 Precision and Recall

Precision and Recall are metrics obtained from confusion matrix. Precision measures how accurate positive predicted values are. Recall, on the other hand, measures how many of those that are positive according to ground truth are predicted to be positive.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Figure 4.12 Formulas of Precision and Recall [9]

4.4.4 Average Precision

Contrary to its name, average precision refers to the underlying area of the precision-recall curve, not the average of the precision values. A table is created with the predictions we have obtained. The rows of the table contain the predictions, confidence values, and IOU results that show whether the estimates match ground truth. Each column is sorted in ascending order of confidence.

Detections							
Conf.	0.63	0.77	0.92	0.86	0.88	0.58	0.91
Matches GT by IoU?	TP	TP	TP	FP	TP	TP	FP

Figure 4.13 Sample AP table [9]

Precision callback values are then calculated for each column. Formulations are:

$$Precision = CumulativeTP/(CumulativeTP + CumulativeFP)$$

$$Recall = CumulativeTP/TotalGroundTruths$$

Preds.	Conf.	Matches	Cumulative TP	Cumulative FP	Precision	Recall
	0.92	TP	1	0	1/(1+0) = 1	1/16 = 0.08
	0.91	FP	1	1	1/(1+1) = 0.5	1/16 = 0.08
	0.88	TP	2	1	2/(2+3) = 0.66	2/16 = 0.16
	0.86	FP	2	2	0.5	0.16
	0.77	TP	3	2	0.6	0.25
	0.63	TP	4	2	0.66	0.33
	0.58	TP	5	2	0.71	0.41

Figure 4.14 Sample AP table [9]

Obtained precision values are interpolated across 11 Recall values, i.e., 0, 0.1, 0.2, 0.3,...,1.0. The interpolated Precision is the maximum Precision corresponding to the Recall value greater than the current Recall value. The intention in interpolating the precision/recall curve in this way is to reduce the impact of the wiggles in the precision-recall curve, caused by small variations in the ranking of examples.

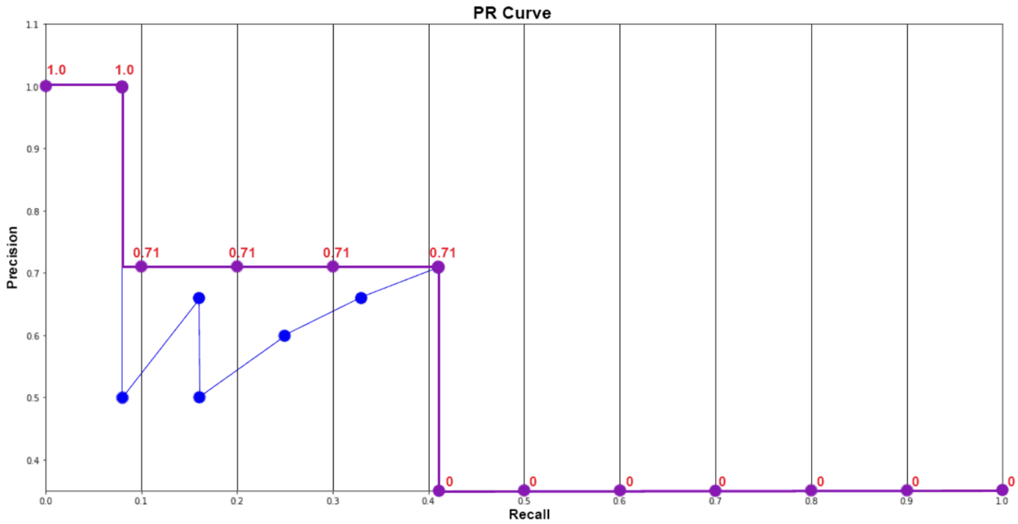

Figure 4.15 Interpolation Graph [9]

When calculating AP, it can be calculated with "1/11 * (Sum of 11 interpolated Precision values)". In this example, the values "1/11 * (1 + 4*0.71 + 6*0)" are added in the formula and the AP value is obtained.

When calculating AP, it is calculated separately for each class, because there may be an imbalance in the data set. In training images, it is possible to understand the classes that cannot be learned when there is an unbalanced distribution.

4.4.5 Mean Average Precision

Unlike average precision, mean average precision is the average of AP values. If we formulate it, it becomes:

$$sum(AP)/(number of classes)$$

5
Experimental Results

In this chapter, the result images and evaluation metrics (mAP) obtained after the training for the detection of collapsed buildings are shown.

As a result of the researches, the hyper parameters that provide the most optimum training in similar studies using the YOLO model to detect collapsed buildings were used for both YOLOv8 and YOLOv7 in this study. It is indicated in Table 5.1.

Parameter	Value
Epochs	100
Image Size	640,640,3
Batch Size	16
Optimizer	SGD
Initial Learning Rate	0.01
Box Loss Gain	7.5
Cls Loss Gain	0.5
Dfl Loss Gain	1.5

Table 5.1 Hyperparameters for YOLOv8 and YOLOv7

Both models were first trained on the raw data set without data augmentation. Then augmentation was applied to the training set for both models. The following Table 5.2 indicates the time taken to train the data set and the resulting evaluation metrics.

Model	Preprocess	mAP50	mAP50-95	Training Time
YOLOv8	Non-Augmented	%75.65	%47	26 min.
YOLOv8	Augmented	%77.6	%54	45 min.
YOLOv7	Non-Augmented	%76.1	%42	2 hours
YOLOV7	Augmented	%77.8	%45	3 h. 35 min.

Table 5.2 Evaluation Results

mAP50 is a variant of mAP where the IoU threshould is fixed and its value is 0.5, mAP50-95 takes into account a range of IoU thresholds from 0.5 to 0.95. It calculates the average precision across this range, considering multiple IoU thresholds to determine the accuracy of the object detection model at different levels of overlap between the predicted bounding box and the ground truth bounding box.

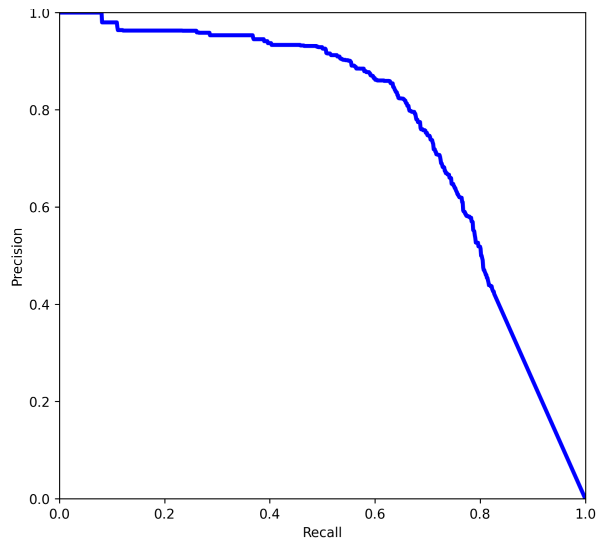

Figure 5.1 Precision Recall Curve for YOLOv8 (Augmented Data)

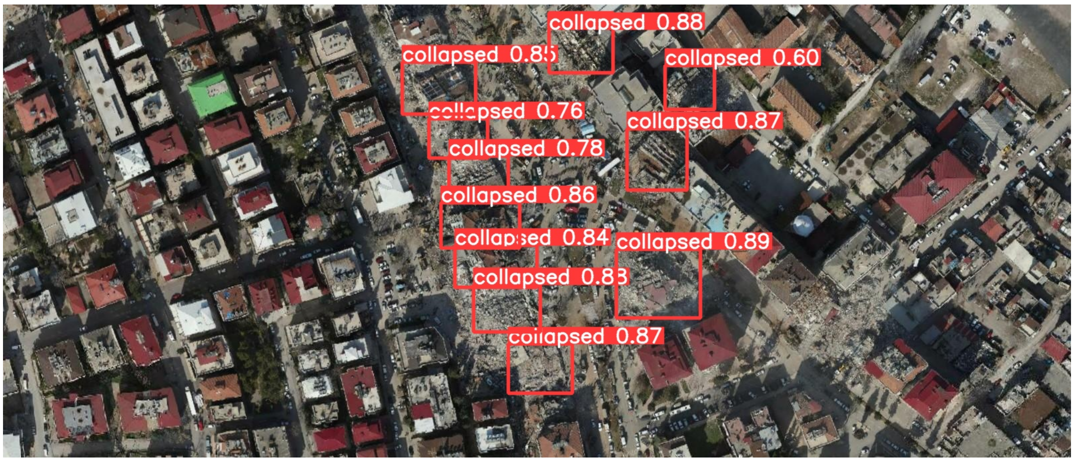

Figure 5.2 Sample Collapsed Building Detection Result for YOLOv8

Following images are samples of detection results for training with augmented data set and training with non-augmented data set.

Figure 5.3 Trained On Augmented Data Set

Figure 5.4 Trained On Non-Augmented Data Set

As it turns out, the model trained on the augmented data made more successful detections for this image. However, this is not true for every case.

Figure 5.5 Trained On Augmented Data Set

Figure 5.6 Trained On Non-Augmented Data Set

As it can be seen in the figure 5.10 and figure 5.11, the model trained on the non-augmented data set was able to detect the 2nd collapsed building on the right by finding the confidence value above 0.5 for this image. The model trained with the augmented data set failed to detect collapsed building on the right.

The following images are samples of detection results of YOLOv7 model.

Figure 5.7 Sample Collapsed Building Detection Result for YOLOv7

In some cases YOLOv7 model made more accurate detections than the YOLOv8 model.

Figure 5.8 YOLOv7 Detection Result

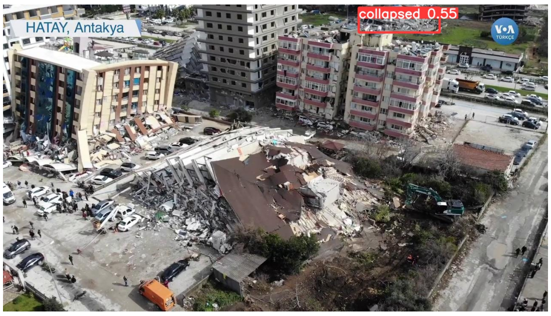

Figure 5.9 YOLOv8 Detection Result

As it can be seen in the figures, YOLOv8 was unable to detect the collapsed building at the center but YOLOv7 model was ablo to detect it with 0.89 confidence value. In general YOLOv8 was the more accurate model but not by far.

There were some cases where both model's detections were very accurate.

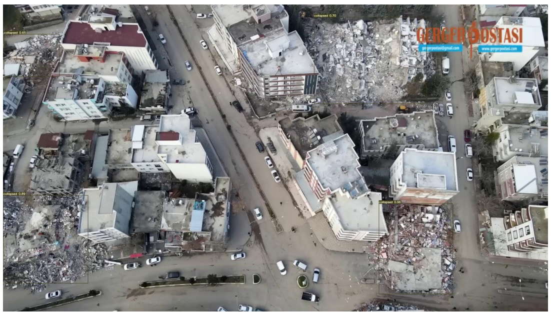

Figure 5.10 YOLOv7 Accurate Detection

Figure 5.11 YOLOv8 Accurate Detection

In some cases model made wrong detections.

Figure 5.12 Wrong Detection for YOLOv8

6
Performance Analysis

First of all, separate training on UAV images and separate training on drone images were carried out. In the successes measured after these trainings, the mAP could not exceed %50. Therefore, these two data sets were combined as a solution. As a result of the measurements, mAP values were obtained as indicated in the Table 5.2. Although there was not much difference in the achievements, the biggest factor separating YOLOv8 from YOLOv7 was training time.It was seen that the training time of YOLOv7 took approximately 4.7 times longer than YOLOv8. While YOLOv8 was trained on the raw data set for 25 minutes, and it completed the training on the augmented data set in 45 minutes, in v7 these time measurements were 2 hours and 3 hours 35 minutes. We can say that v8 is more successful in this respect, since the training period is a very large factor in determining the model. Although the training time naturally increased in augmented data sets as the number of data increased, there was a noticeable increase in the success of the mAP50-95 metric in particular. Based on the mAP50-95 output, it was seen that the YOLOv8 augmented model predicted TP estimates with higher confidence values. In some cases v8 did better capturing collapsed buildings and in some cases v7 did better. The model made some wrong detections for example in Figure 5.12 a globe like big object is considered a collapsed building. The reason to this is that there is no before earthquake data and the model can't determine if there was this object before the disaster. If there was before earthquake data or more and better quality images the model could make more correct detections.

7 Conclusion

Since rapid detection is very important after an earthquake, a research project that can make real-time predictions and work with speed and accuracy has been implemented. With this project, it has been shown that saving human lives after an earthquake can only be achieved with a drone, and it is one of the contributions that artificial intelligence can provide to human life.

In the project, UAV and drone images that can take images in any situation were used in order not to be affected by the adverse conditions caused by the weather conditions. Data augmentation technique has been applied to give successful results in noisy, blurry or images that may come from different angles.

As a result, it was seen that YOLOv8 is much faster than YOLOv7 and provides higher confidence values in correct predictions. It has been observed that data augmentation improves performance, especially in datasets with a large number of samples. It has been observed that the use of deep learning algorithms is successful in detecting post-earthquake damage. It has been seen that YOLOv7 and YOLOv8 can be preferred as deep learning algorithms in this subject.

References

[1] E. E. Maraş and H. İ. Sarıyıldız, "İha ile derin öğrenme algoritmaları kullanılarak hasarlı yapıların tespit edilmesi," *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, vol. 23, no. 2, p. 427, 2023.

[2] C. Liu, H. Sui, J. Wang, Z. Ni, and L. Ge, "Real-time ground-level building damage detection based on lightweight and accurate yolov5 using terrestrial images," *Remote Sensing*, vol. 14, no. 12, p. 2763, 2022.

[3] Y. Li, W. Hu, H. Dong, and X. Zhang, "Building damage detection from post-event aerial imagery using single shot multibox detector," *Applied Sciences*, vol. 9, no. 6, p. 1128, 2019, ISSN: 2076-3417. [Online]. Available: https://www.mdpi.com/2076-3417/9/6/1128.

[4] R. Ali, D. Kang, G. Suh, and Y.-J. Cha, "Real-time multiple damage mapping using autonomous uav and deep faster region-based neural networks for gps-denied structures," *Automation in Construction*, vol. 130, p. 103 831, 2021, ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2021.103831. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092658052100282X.

[5] Roboflow, *Roboflow annotate*, 2023. [Online]. Available: https://roboflow.com/annotate (visited on 06/16/2023).

[6] S. Rath, *Yolov8 ultralytics: State-of-the-art yolo models*, Mar. 2023. [Online]. Available: https://learnopencv.com/ultralytics-yolov8/%5C#What's-New-in-YOLOv8.

[7] Ultralytics, *Ultralytics-yolov8-github*, YYYY. [Online]. Available: https://github.com/ultralytics/ultralytics.

[8] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. arXiv: 2207.02696 [cs.CV].

[9] Kukil, *Mean average precision (map) in object detection*, Aug. 2022. [Online]. Available: https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Berk SICAK
Birthdate and Place of Birth: 03.07.2000, Kırklareli
E-mail: berk.sicak@std.yildiz.edu.tr
Phone: 0554 428 58 29
Practical Training: Yapı Kredi Teknoloji Şirketi Yazılım Geliştirme Departmanı

SECOND MEMBER

Name-Surname: Kemal SELÇUK
Birthdate and Place of Birth: 01.03.2000, İstanbul
E-mail: kemal.selcuk@std.yildiz.edu.tr
Phone: 0541 952 97 29
Practical Training: Jotform Yazılım A.Ş. Veri Bilimi Departmanı
Yapı Kredi Teknoloji Şirketi Yazılım Geliştirme Departmanı

Project System Informations

System and Software: Windows İşletim Sistemi, Python
Required RAM: 4GB
Required Disk: 10GB