# Crack Detecting Method With YOLOv3


1st Kang Lee
*Computer Software Engineering)*
*Wonkwang University*
Iksan, Republic of Korea
river449@gmail.com

2nd Dohyun Kim
*Computer Software Engineering*
*Wonkwang University*
Iksan, Republic of Korea
flash6471@naver.com

3rd Sanha Choi
*Digital Contents Engineering*
*Wonkwang University*
Iksan, Republic of Korea
0412csh@gamil.com

4th Christian Elfreich
*Computer and Information Technology*
*Purdue University*
West Lafayette, Indiana
celfreic@purdue.edu

5th Kichang Kang
*Computer and Information Technology*
*Purdue University*
West Lafayette, Indiana
kang408@purdue.edu

6th Evan McClain
*Network Engineering Technology*
*Purdue University*
West Lafayette, Indiana
mcclai13@purdue.edu

4th Christian Elfreich
*Computer and Information Technology*
*Purdue University*
West Lafayette, Indiana
celfreic@purdue.edu

4th Christian Elfreich
*Computer and Information Technology*
*Purdue University*
West Lafayette, Indiana
celfreic@purdue.edu

4th Christian Elfreich
*Computer and Information Technology*
*Purdue University*
West Lafayette, Indiana
celfreic@purdue.edu



***Abstract*—There are several types of cracks that can occur on structures. These cracks can cause structural problems like water leakage, corrosion, and even total collapse. This project aims to detect cracks on structures with You Only Look Once Version 3 (YOLOv3) which is based on the Darknet framework. It will be able to detect cracks on structures in real-time and can help with responsible maintenance. This project utilizes Unmanned Aerial Vehicles (UAVs) to detect damage that is in difficult or dangerous to access locations. This project has the potential to have more diverse functionality such as classifying and detecting other features and objects.**

***Index Terms*—crack, deep learning, darknet, YOLO**


## I. Introduction

In today's society, civil features such as sidewalks, buildings, bridges, etc., are very prevalent. While building new structures is important, it is also critical to focus on maintaining existing architectural structures and buildings. Some might see the cracks on the buildings which will further lead to the aging of the structures and cause a consecutive failure according to it. According to Buyers Ask, approximately 60% of homes in the United States are built on soils with clay content and 98% of homes are constructed out of concrete, masonry blocks, or brick and are more likely to experience damages [1][1]. There are several different types of cracks that appear on the structures and the location for these different types of cracks varies. Figure 1 below shows the types of cracks in the structure and Figure 2 provides the following chart showing the information depending on the type of cracks [2].

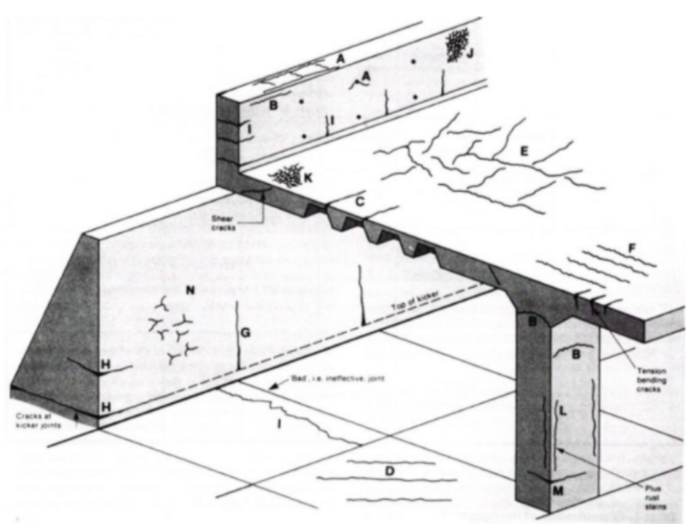


Fig. 1. Types of cracks in concrete [2].

TABLE I

| Letter | Type of Cracking | Subdivision | Most Common Location | Primary Cause (excluding restraint) | Secondary Causes/Factors | Time of Appearance |
|---|---|---|---|---|---|---|
| A | Plastic settlement | Over reinforcement | Deep sections | Excess bleeding | Rapid early drying conditions | Ten minutes to three hours |
| B | | Arching | Top of columns | | | |
| C | | Change of depth | Trough and waffle slab | | | |
| D | Plastic shrinkage | Diagonal | Roads and slabs | Rapid early drying | Low rate of bleeding | Thirty minutes to six hours |
| E | | Random | Reinforced concrete slabs | | | |
| F | | Over reinforcement | Reinforced concrete slabs | Ditto plus steel near surface | | |
| G | Early thermal contraction | External restraint | Thick walls | Excess heat generation | Rapid cooling | One day or two or three weeks |
| H | | Internal restraint | Thick slabs | Excess temperature gradients | | |
| I | Long-term drying shrinkage | | Thin slabs (and walls) | Inefficient joints | Excessive shrinkage inefficient curing | Several weeks or months |
| J | Crazing | Against formwork | "Fair faced" concrete | Impermeable formwork | Rich mixes | One to seven days, sometimes much later |
| K | | Floated concrete | Slabs | Over troweling | Poor curing | |
| L | Corrosion of reinforcement | Natural | Columns and beams | Lack of cover | Poor quality concrete | More than two years |
| M | | Calcium chloride | Precast concrete | Excess calcium chloride | | |
| I | Alkali-aggregate reaction | | Damp locations | Reactive aggregate plus high-alkali cement | | More than five years |

Fig. 2. Table supporting Fig 1. [2].

While there are multiple reasons that cracks can appear on structures, lack of inspection on the structures is one of the most important factors that is overlooked. Cracks on structures can cause further deterioration of architecture, causing corrosion and leakage within the structure. Therefore, it is important to not neglect or ignore the inspection of cracks on the structures. Understanding that damage can occur to the interior of the structures encourages the prevention of the factors that cause these problems. The existing traditional image-processing-based crack detection methods can confuse patterns on the wall or paint marks with cracks due to inaccurate empirical and experimental threshold settings. The reasons behind low-quality detection include difficulties in identifying cracks due to obstacles and debris, confusion with changing background, and variation in lighting conditions. This project overcomes these disadvantages and aims to more accurately detect cracks on multiple types of structures such as sidewalks, buildings, and bridges through machine learning technology. Using machine learning technology will provide a more objective and convenient method of determining cracks on structures while helping to visualize the results of the YOLO model and Darknet framework. Using public pre-trained images of cracks to refine and implement the data, several sets of primary collected photographs and videos were added to ensure the inclusion of real-world diversity of the dataset. With comparative verification and cross-learning methods, the data will be analyzed using YOLO V3 to create a model for the .mjpg video stream to be compared against.

## II. Literature Review

Based on the thorough review of sample research papers, and through an explicit comparison with the current research question, the research gap was concluded. Since the labor force's safety has been a priority and a global concern in construction fields, there has to be an advancement in technology in order to reduce fatalities. According to a research paper on hazard detection, a hazard was detected through a first-person view of workers as they moved within the construction site and marked potential hazardous locations and objects [3]. The technology utilized in this paper is SLAM, a visual simultaneous localization, and mapping which is cost-effective and suitable for hazard detections. The SLAM technology and sensors are used in multiple cameras that are located in hazardous places marked by the workers. The gap between this paper and utilizing a UAV in hazard detection is about how the camera is placed on the drone appropriately with suitable technology.

Another paper approaches how the camera can detect hazardous smoke in a home environment [4]. The paper broke down the smoke dataset into pixel size where cameras can possibly differentiate other objects and smokes. This utilized the Raspberry Pi and was written in Python as well as its graphical user interface, Qt, which also supports the YOLO format. This system was created in order to prevent fire by detecting early signs of smoke that traditional smoke detectors cannot detect. The research gap between this paper and utilizing a UAV in hazard detection has similarities in which smoke detection in construction areas is a must. Thus, in order to reduce the gap, applying the technology (Raspberry Pi/Python) utilized in smoke detection to UAV cameras is needed.

There exists previous research to detect the damage of buildings; for example, Francesco Nex et al. have proposed real-time building damage mapping with UAV solutions [5]. The occurrence of disasters can lead buildings to be damaged, and attempting to notice these damages with the naked eye can lead to difficulties. Utilizing the UAVs to take real-time images of the accident site with an attached camera in real-time can solve this issue; to detect property damage, this device is able to create georeferenced orthophotos by using SfM(Structure for Motion), Visual SLAM, and BA (Bundle Adjustment). Also, further research for detecting the damage of the structure is suggested as well, as Idris Geelani et al. propose the Hazard Detection of the steel sea-cross bridge; this utilizes elastic modulus and the bearing yield response to measure the bearing capacity of steel structures [6].

Sankarasrinivasan, S., et al. concluded that crack and surface damage detection can be analyzed by examining captured images via a method called the hat transform approach. This method is able to accurately predict damage on buildings, for in this application UAV devices can collect real-time image data and allow for instant image capture and analysis; this provides on-site utility for damage detection in dangerous or unreachable locations [7]. Another method for damage detection includes a deep learning model, which can be more plausible for different types of surfaces such as spalling, defacement, and efflorescence. This, however, requires having access to a pre-trained network and may not be able to recognize less common types of building damage, such as destruction or stagnant water [8].

In terms of detecting property damage, a method that utilizes a Darknet neural network via YOLO has also been researched [9]. S. Carata et al. determined that object detection can still be a difficult task to perform, so they chose to design an interface that can more easily allow for network creation within this method of deep learning. This can further aid in the project's goals of detecting property damage via the use of a UAV, particularly with regards to improving the deep learning algorithm used. The creation of an algorithm primarily involves the use of both image processing and image classification [10]. in order to properly detect cracks within structures by using filters as well as image morphing and subtraction methods, an image can be automatically processed until key points of interest are found by the algorithm. Then, these points are determined to be cracks of varying levels through a trained and fine-tuned neural network. Still, there can be challenges regarding the varying sizes and general unpredictability of cracks, so the use of deep convolutional neural networks (CNN) may allow for increased accuracy when analyzing property damage [11].

Detecting damages within structures can be difficult and very expensive, so the general use of UAVs has proven to be of great benefit [12]. They have been shown to significantly shorten inspection times and also provide considerably less risk compared to in-person examinations. UAVs are also capable of navigating to areas and spaces much more easily and efficiently than through the use of other types of vehicles, further showing their effectiveness. All of these factors have also proven to greatly reduce inspection costs as well. The utility of UAVs has been proven to be of great aid, even when these vehicles are manually operated and the footage taken is reviewed by the human eye; if deep learning methods could be implemented to automatically review the numerous amounts of footage taken, their effectiveness and potential would be even further realized.

## III. Research Method

This paper develops a deep learning crack detecting method. YOLO v3 will be utilized and the Darknet framework will train the model. Deep learning works similarly to a neural network by attempting to mimic the human brain with many layers of interconnected neurons. It differs from machine learning in the sense that its algorithms allow it to function more independently with less human input. In his project, The YOLO scheme is used as the main framework to compute the deep learning information as it is much more efficient at detecting objects as opposed to a CNN and R-CNN. This makes it the better choice for crack detection.

YOLO utilizes a singular convolutional network in order to aid the program in forecasting the bounding boxes within the image and then calculating class probabilities for the corresponding boxes within the image as shown in Fig 2. The bounding boxes are then analyzed for the probability of objects within them, and if that probability is higher than the programmed threshold, the object(s) will be identified by the program [13]. As shown in Fig 3, this computational process makes YOLO much faster running at an average frame rate of 45 frames per second, compared to R-CNN and CNN's 5 frames per second. YOLO outperforms CNN and R-CNN in almost every criterion except for small object detection [17]. The YOLO framework can miss detection of small objects due to spatial constraints within the algorithm, but in most situations, YOLO will work well [14].

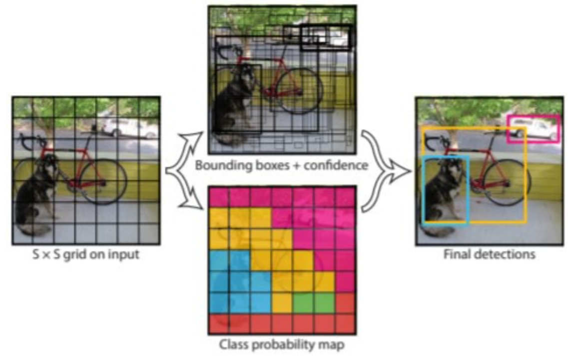


Fig. 3. Process of YOLO bounds box being established

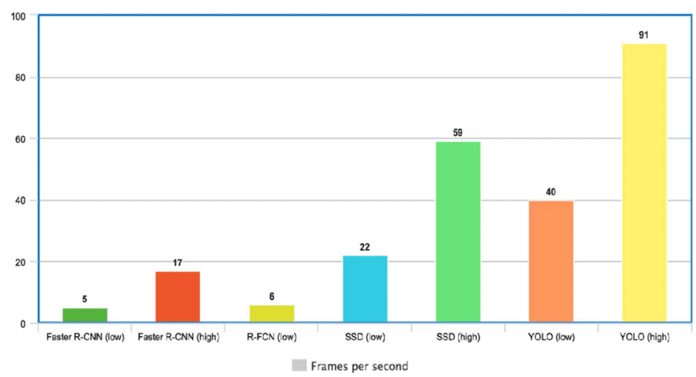


Fig. 4. FPS Comparison

### A. Learning environment setting

YOLO model through a DarkNet framework to detect the cracks was designed. The structure of YoloV3 utilizes a variant of Darknet, which originally has a 53 layer network trained on Imagenet, but an additional 53 layers were stacked on top, providing full 106 convolutional layer architecture for YOLO v3. The boosts in the architecture layer catch detections at three different scales. The detection is done by the application of 1 x 1 detection kernels on feature maps of three distinct sizes at three different locations in the network. Here, YOLO vs passes images to a convolutional neural network (CNN). For calculations, YOLO v3 utilizes binary cross-entropy for calculating the classification loss for each label while object confidence and class predictions are predicted through logistic regression [15].

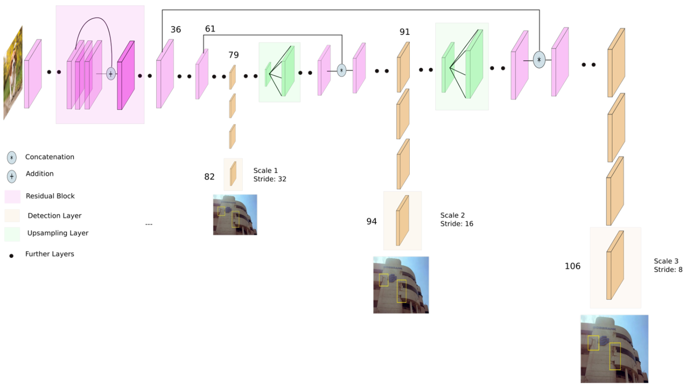


Fig. 5. Yolo layer architecture [15]

cfg file settings which define the layers of CNN had been simultaneously changed throughout the project. This becomes the basic structure of the darknet53 architecture, and later layers will be dealt with or added by pretreatment or fine-tuning. One class called crack is defined to set the CNN layer structure, such as appropriate max batch, subdivision, and filter values. In addition, Yolo predicts the final object of the object by being resized according to the size of the detected object based on the anchor box defined as the initial value [ ]. Therefore, the initial value of the anchor box was set for the set cfg file. These settings were focused on one accuracy in model training for crack detection. Existing weight values for model learning are trained using the darknet53.conv.74 file of alexyab [ ].

### B. Datasets

TABLE II
USED CRACK DATASETS

| | | |
|---|---|---|
|  |  |  |
| *METU* | *SDNET* | *COLLET* |
| *20000 images* | *8000 images* | *700 images* |
| *Mainly consist of Concrete* | *Bridge decks, wall, pavements.* | *considering various shooting environments* |

The datasets used for model learning used datasets collected in various environments to detect various environments and crack forms. The learning dataset consists of metu, sdnet, and additionally collected custom data. Each dataset has its own characteristics, and the metu dataset is a dataset for cracks generated based on concrete, and the shape and location of cracks are also clear. In the case of sdnet, it has crack images of various environments and includes images in which discrimination against cracks is unclear. Next, for the directly collected crack images, various photographing environments and angles are considered and collected. Each dataset is basically used to determine whether there is a crack, and the quality and shape of the data are different due to the different collection paths. Therefore, the image is handled through a preprocessing process before image learning.

### C. preprocessing

*1) labeling:* labeling - As a preprocessing process of data for machine learning, cracks in the image were classified through a bounding box and annotated with the annotation ”crack”. Labeling was carried out using an online tool provided by “Roboflow”. The file labeled for darknet learning consists of center x, center y, width, height values for the entire image size.

*2) image processing:* The quality of the image is improved so that yolo can easily analyze the input image frame accepted for crack detection. For preprocessing through Keras, a powerful library of Python, in a darknet environment operated based on C language, a model that processes weight values through darknet learning as Keras is created. Using the Yad2k converter, the learned weight file is converted into a form of h5 supporting Keras and used [for the git hub citation] Outputs that have gone through the convolution layer have different size information for variable input images, which can create issues about the location inference and probability of bounding boxes in Yolo’s fully connected layer. The input image is applied in the same shape through the resizing of the image and the featurescaling.

## IV. EXPERIMENT

The learning of the model proceeded in the form of a mixture of metu, sdnet, and collected image datasets, and with learning time limitations arising from the size of the dataset and GPU performance of the computer. This was adjusted by the number of datasets and learning batches for each experiment. Through the conclusions obtained through this, modifications for the development of the model were made. all datasets were basically labeled and resized, and the shape proceeded to 416*416.

Used 2200 images from the METU dataset.

Showed reasonable numerical value with low single average data loss. Did not accurately detect the cracks yet, but had shown rare occasions in misclassifying cracks which is a good start.

Used 5600 images from the METU dataset.

There have been many cases in which it detects not only cracks but also non-cracks that are classified as crack as well. The possible cause of this problem was assumed to be over-fitting of the data because of the learning process being biased in the way of only training the common characteristics among the cracks.

Used 9000 mixed images from METU and SDNET. It mostly focused on non-concrete materials and included custom data. The average of single data was low as well as the map. Compared to the 2nd test of data processing, its own self-performance did not noticeably get better but reduced in the number of misclassifying wrong objects as a crack. The next process performed was fine-tuning. Because it did not show reasonable performance itself, fine-tuning was processed to freeze the weight until 74th layers and re-train the later 32 layers and check the performance once again. Overall improvement has shown in the crack detection part [16]. YOLOv3 utilizes Darknet-53 as a backbone which has an improved power efficiency than the previous Darknet-19. For comparison, Darknet-52 is 1.5 times faster than ResNet101 and doesn't trade off between accuracy and speed. Since the training time of YOLO is shorter compared to other object detection algorithms, YOLOv3's accuracy could be made up for when a larger dataset is utilized to train. Cracks, which could be seen as small objects in the YOLO, the upgraded v3 increased the AP (average precision) to 13.3 for small objects [**?**]. There are techniques currently existing such as the dynamic discarding technique to increase speed and preserve accuracy for YOLOv3. Compared to the standard execution, after processing the dynamic discarding technique achieved, an increase of 22

## V. Conclusion

This project focuses on making a model which can detect many different crack types. 2,200 images of the METU dataset were trained on the first model and the average data loss rate was low. It can detect a few cracks but there is no phenomenon that detects other objects as a crack. Second model trained 5,600 images of METU datasets. It detects not only cracks but other objects like walls, concrete, and even humans as a crack. It seems that the model trains common and biased characteristics of cracks, so there is a possibility that the model is overfitted. A phenomenon in which the model is trained closer to the distribution of training samples than to the actual distribution is overfitting. Can prevent this overfitting with regularization. Third model trains 9,000 images of mixed METU datasets which are not on concrete, SDNET and custom datasets which we take a photo on concrete structures, sidewalks. It's average data loss rate is kept lower and there is no phenomenon that detects other objects as a crack but performance of the model is not as good as the first model. This project can be improved as the solution below. There are several solutions to improve our model. The first solution moving forward is fine tuning by adjusting loss rate and accuracy in weight file which is utilized at darknet-53, repeating training to find the optimal values of loss rate and accuracy. The second solution would be to collect more datasets that have different shapes as current models learn a lot of cracks that have similar shapes which can make models overfitted.

## References


[1] "Foundation Cracks: Causes, Seriousness and When To Worry", Jul. 2021. [Online]. Available: https://buyersask.com/structural/foundation-cracks-causes-seriousness-and-when-to-worry/

[2] "Types of Cracks in Concrete", Mar. 2017. [Online]. Available: https://cemsolutions.org/types-of-cracks-in-concrete/

[3] Alon, A. S., Dioses, J. L., Diamante, R. A., Dellosa, R. M., amp; Montalbo, F. J. P. Mobilenet SSDv2 inference approach of Smoke Hazard Detection and alert system: A smoke-induced simulated home-environment. International Journal of Advanced Trends in Computer Science and Engineering, vol 9. no 1.3. p. 197–202, 2020.

[4] Jeelani, I., Asadi, K., Hariharan Ramshankar, Han, K., Albert, A. Real-time vision-based worker localization hazard detection for construction. Automation in Construction, vol 121. p. 103448 Jan. 2021.

[5] F. Nex, D. Duarte, A. Steenbeek, and N. Kerle, "Towards Real-Time Building Damage Mapping with Low-Cost UAV Solutions," Remote Sensing, vol 11. no. 3. p. 287, Feb. 2019.

[6] Qixin Liu and Chaoan Li. Journal of Coastal Research, Special Issue No. 93: Advances in Water Resources and Exploration. vol 93. p. 969-974, 2019.

[7] Sankarasrinivasan, S., Balasubramanian, E., Karthik, K., Chandrasekar, U., Gupta, R. Health monitoring of civil structures with integrated UAV and image processing system. Procedia Computer Science, vol 54. p. 508-515, 2015.

[8] Kung, R. Y., Pan, N. H., Wang, C. C., Lee, P. C. Application of Deep Learning and Unmanned Aerial Vehicle on Building Maintenance. Advances in Civil Engineering, 2021. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, p. 740–741, Aug. 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982.

[9] H.-G. Moon and J.-H. Kim, "Intelligent crack detecting algorithm on the concrete crack image using neural network," 28th International Symposium on Automation and Robotics in Construction (ISARC 2011), 2011.

[10] D. Cao, Z. Chen, and L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional Neural Networks," Human-centric Computing and Information Sciences, vol. 10, no. 1, 2020.

[11] S.-suk Choi and E.-kon Kim, "Building Crack Inspection Using Small UAV," 2015 17th International Conference on Advanced Communication Technology (ICACT), 2015.

[12] S. Carata, R. Mihaescu, E. Barnoviciu, M. Chindea, M. Ghenescu, and V. Ghenescu, "Complete Visualisation, network modeling and training, web based tool, for the Yolo Deep Neural Network model in the Darknet Framework," 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), 2019.

[13] P. Kumar, S. Batchu, N. Swamy S. and S. R. Kota, "Real-Time Concrete Damage Detection Using Deep Learning for High Rise Structures," in IEEE Access, vol. 9, p. 112312-112331, 2021.

[14] Pulkit Sharma Journal of Analytics Vidhya, No. 3 : Practical Guide Object Detection YOLO Framework Python p. 1-2, 2018.

[15] "ArcGIS API for python YOLOv3 Object Detector", [Online]. Available:https://developers.arcgis.com/python/guide/yolov3-object-detector/

[16] Meel, V. (2021, August 6). Yolov3: Real-time object detection algorithm (what's new?). DEEP LEARNING. Retrieved December 10, 2021, from https://viso.ai/deep-learning/yolov3-overview/. s

[17] Martinez-Alpiste, I., Golcarenarenji, G., Wang, Q. et al. A dynamic discarding technique to increase speed and preserve accuracy for YOLOv3. Neural Comput Applic 33, 9961–9973 (2021). https://doi.org/10.1007/s00521-021-05764-7