

RESEARCH ARTICLE

Determination of the efflorescence defects on concrete and plaster surfaces using a novel deep-learning model

Safa Nayır^{1*}, Yılmaz Yılmaz¹

¹ Karadeniz Technical University, Faculty of Engineering, Department of Civil Engineering, Trabzon, Türkiye

Article History

Received 29 April 2024

Accepted 1 July 2024

Keywords

Efflorescence
Concrete surface
Plaster surface
Deep learning
Object detection
YOLOv8

Abstract

Efflorescence is a common problem in concrete and masonry structures. It occurs when soluble salts in the material are dissolved by water, and then carried to the surface where the water evaporates, leaving behind the salt deposits. In recent years, the construction industry has seen the development of deep learning methods to detect defects and damages such as efflorescence during the service life of structures. These advanced techniques offer more efficient and accurate ways to identify and address issues, improving the overall quality and durability of structures. This study aims to identify efflorescence damage using an innovative deep-learning model. In this scope, 617 different images were taken showing efflorescence defects on concrete and plaster surfaces. A custom dataset was created by labeling the efflorescence regions from the images taken and this dataset was trained using the YOLOv8x model. The accuracy, recall, precision, mAP50, and mAP50-95 performance measures obtained from the YOLOv8x model are 0.962, 0.902, 0.991, 0.851, and 0.6334 respectively. According to these performance metrics, efflorescence damage was considerably detected. Finally, the model was tested by taking efflorescence damage images from different concrete and plaster surfaces. The study indicated that detecting efflorescence defects on light-colored surfaces in the model was difficult, with a low detectability rate. Algorithms can be developed to detect efflorescence damage more effectively on light-colored surfaces.

1. Introduction

Concrete is one of the most widely used building materials today due to its strength, durability, and cost compared to other building materials [1]. Concrete and reinforced concrete structures tend to deteriorate due to internal and external factors caused by environmental conditions during service life. These internal and external factors can be of physical-mechanical as well as chemical and biological origin. Efflorescence is a type of physical and chemical deterioration that manifests itself as white spots on the surface of concrete and plaster [2-3]. Efflorescence occurs mostly because of the dissolution of the calcareous components in the cement mortar, leakage to the upper surface, and crystallization and accumulation. These deposits are generally seen on the surface of Portland concrete products such as masonry units and bricks [4]. Efflorescence requires the simultaneous presence of soluble salts and lime, as well as a transport medium

* Corresponding author (safanayir@ktu.edu.tr)

and a pathway for the solution to move to the surface. When these conditions are met, the salts can dissolve in water and be transported to the surface of the material [5].

Surface deposits of efflorescence are typically composed of calcium carbonate. Additionally, crystal structures of sodium, potassium, calcium, sulfate, and other salts may also be present. The presence of lime, which is the most water-soluble component of hydration products, should be minimized in concrete for improved durability. The use of mineral admixtures helps reduce the lime content by binding it and converting it into insoluble calcium silicates, thus promoting the formation of additional calcium silicate hydrates (C-S-H). Lime that dissolves in water can lead to white stains when it migrates to the surface of the concrete [2].

Efflorescence is generally observed in the basement and ground floors of structures with poorly insulated and constructed surfaces. Most building materials contain capillary pores. Water in these capillary pores moves upward due to capillary action, causing the surface to become damp. As the surface water evaporates, the insoluble materials carried by the water, especially lime, are left behind. The remaining hydrated lime- $\text{Ca}(\text{OH})_2$ reacts with CO_2 in the air to form carbonate [6]. This process results in the formation of powdery spots on the surface of concrete. While these spots and powdery appearances do not affect strength and durability, they are considered defects that can lead to economic losses [7-8]. Measures such as painting that can be taken for a certain period are not successful in preventing efflorescence.

Additionally, the use of unwashed sea sand and seawater, as well as environmental conditions, can trigger efflorescence [9-10]. To prevent efflorescence, it is important to prevent the structure from absorbing water from the ground, use impermeable layers along the water-rising path, and use materials with a low capillarity coefficient. Selecting concrete with a low water-to-cement ratio and ensuring adequate curing to prevent evaporation are also effective methods to prevent efflorescence.

Traditional efflorescence detection methods are laborious, time-consuming, and require expertise. Instead, currently, image-based object detection mitigates these problems considerably. The traditional method involves the detection of staining in situ, generally by authorized persons in basements and ground floors. This method focuses on visually inspecting and assessing signs such as staining and lumps, which are particularly common in such areas. Significant progress has been made recently in the field of deep learning-based object detection, especially in large and complex datasets [11]. Object detection models based on a single network are characterized by their multi-scale object detection and fast processing capabilities [12]. In addition, the use of different deep learning architectures leads to the discovery of more complex and deeper features in the field of object detection [13]. The use of techniques such as transfer learning enables faster and more efficient customization of object detection models [14]. The use of large and various datasets often enables object detection models to perform better in different and complex scenarios. These developments have taken important steps in the field of deep learning-based object detection and allow the development of even more advanced and effective object detection systems in the future. With technological developments, object detection studies have started to be used in the construction sector. It has been significantly utilized in structural health monitoring [15-18]. As an example, Bai et al. [19] employed end-to-end deep learning methodologies to automatically identify cracks and spalling, crucial types of structural deterioration in buildings and bridges, particularly during significant events like large earthquakes. They annotated and utilized 2229 images to train and validate three newly developed Mask RCNN models. Furthermore, they tested the accuracy of these models on three sets of publicly available images representing various disasters. One of their proposed methods successfully detected and labeled cracks and spalling, achieving 67.6% and 81.1% accuracy on low- and high-resolution images from field inspections, respectively. These findings highlight the potential of their end-to-end approach in aiding human experts during disaster scenarios. Hacıefendioğlu et al. [20] aimed to swiftly and cost-effectively detect existing cracks. They employed a descriptive approach utilizing a pre-trained Faster R-CNN model to detect cracks on concrete road surfaces.

Their experimental setup utilized 323 images with a resolution of 4128×2322 pixels and an aspect ratio of 16:9. Through parametric studies altering shooting conditions on concrete roads, crack detection analyses were conducted. Additionally, the impact of weather conditions and illumination levels was analyzed. Results from crack detection analyses under various weather conditions, including sunny, cloudy, foggy, sunset, moonlight, and dark periods, revealed consistent crack detection on sunny days, a 50% decrease at sunset and moonlight, and varying changes at certain hours. Xu et al. [21] introduced YOLOv5-IDS, a novel deep learning model based on Look Only Once network v5 (version 6.2), intending to address inefficiencies and multi-step detection challenges seen in traditional concrete crack detection methods. Their approach integrated extended convolution, pyramid pooling module, and attention enhancement module, in conjunction with a bilateral segmentation network. Furthermore, they introduced crack parameter measurement algorithms based on the micro-element method. Their proposed methodology showcased notable accuracy and efficiency in crack detection and segmentation, along with swift crack parameter measurement. Experimental findings for the YOLOv5-IDS model revealed promising performance metrics: The average mean average precision at intersection over union (mAP@0.5) for crack detection achieved 84.33%, with a frames per second (FPS) rate of 159. Similarly, the average intersection over union (mIoU) for crack segmentation reached 94.78%, with an FPS rate of 8. Compared to existing methods, the proposed approach significantly improves both accuracy and efficiency. Additionally, the computation of crack parameters is demonstrated to be precise and rapid. Qiu et al. [22] propose the integration of the You Only Look Once (YOLO) model into an unmanned aerial vehicle (UAV) system to detect cracks in pavement tiles in real time. They explore several network architectures, including YOLOv2-tiny, Darknet19-based YOLOv2, ResNet50-based YOLOv2, YOLOv3, and YOLOv4-tiny, and analyze their performance to improve both the accuracy and speed of crack detection. The results indicate that ResNet50-based YOLOv2 and YOLOv4-tiny achieve remarkable sensitivity (94.54% and 91.74% respectively) and high processing speeds (71.71 fps and 108.93 fps respectively), while effectively detecting small cracks. Moreover, these models exhibit robustness in various environmental conditions, including shadows, rain, and motion-induced blur. The assessment offers valuable insights and recommendations for ensuring remote, dependable, and swift crack detection by optimizing the altitude and scanning parameters of the YOLO-UAV system.

Image processing techniques are used to recognize cracks, defects, etc. in concrete and reinforced concrete structures, and the acquired and processed images are widely used for quality assessment. However, although image processing is widely used in the evaluation of defects such as cracks, etc., studies and research on the use of images for the detection of efflorescence in concrete and plaster are very limited. In this study, a special dataset was obtained with images of efflorescence-induced defects in structures using a digital camera. Then, the custom dataset is trained with the innovative deep learning algorithm YOLOv8, and performance scales are determined. Finally, the trained model is tested and generalized by using images of efflorescence-induced defects from different concrete and plaster surfaces.

2. Method and material

2.1. YOLOv8 algorithm

Object detection is a technology in computer vision that enables machines to identify and position various objects within digital images or video frames [23]. This process not only recognizes the presence of objects but also determines the boundaries of objects by drawing precise lines around the object [24]. Object detection is used very effectively in many sectors such as health, automobile, defense industry, and packaging. This effective technology represents the point of changing the way machines understand and interact with the visual environment. Thanks to advances in technology and software, many object detection algorithms have been developed [25]. YOLO (You Only Look Once) stands out as a groundbreaking method

in the field of object detection. Unlike traditional methods for object detection, this innovative approach estimates bounding boxes and class probabilities directly from full images using a single deep neural network [26]. The basic principle of YOLO is to partition the input image into a grid of a given size, with each grid cell responsible for estimating the position and class of objects. In this way, the detection of objects is realized by estimating multiple bounding boxes for each grid cell and the confidence scores of these boxes (Fig. 1) [26]. The unique structure of YOLO offers great advantages in real-time applications by providing fast and efficient object detection with a single network evaluation. In addition, after the predictions are made, more accurate results are obtained by eliminating double detections with a method called non-max suppression and improving the final bounding boxes. The output of YOLO includes the bounding boxes, class labels, and confidence scores of the detected objects in the image. The method is optimized during training on a loss function directly related to detection performance and offers significant advantages over other methods in the industry by treating the entire image globally for real-time object detection. YOLO's comprehensive and effective approach sets a new standard in object detection, providing a significant improvement in speed, accuracy, and general applicability.[27]

YOLOv8 [28] is an object detection algorithm based on deep learning and is specifically built on Convolutional Neural Networks (CNNs). YOLOv8, part of the YOLO family of object detection algorithms, is known for detecting objects in real-time and offers fast and accurate results in this area. YOLOv8 leverages previous YOLO versions, providing a unified framework for training models and achieving faster and more accurate results. This is a significant advance in the field of object detection. This new version outperforms many other object detection algorithms in terms of speed and accuracy [29]. Developed using new techniques in deep learning and computer vision, YOLOv8 offers improved performance compared to previous versions. This is a revolutionary step in the field of object detection and can provide significant benefits in real-time applications. YOLOv8 is a deep-learning model for object detection. This model processes an image in a single iteration and instead of analyzing each pixel multiple times, it processes the entire image from a single moment. The input image is converted into predefined dimensions and color channels. Then, the image is given as input to the YOLOv8 model. The model consists of deep learning layers such as convolution-based layers and pooling layers. These layers are used to extract features of the image hierarchically. In YOLOv8, the image is divided on a grid. Each grid cell represents a region where the model will predict to detection of objects [30]. Each cell is assigned anchor boxes, usually 3 or more, of different sizes and proportions. These anchor boxes represent the sizes and proportions of different objects. The model makes object predictions for each grid cell. Each prediction contains the class probabilities of a potential object in that grid cell and the coordinates of the bounding box. The coordinates specify the position and size of the predicted object. The output of the model contains the predictions of all grid cells. However, since these predictions are in the form of class probabilities and bounding box coordinates, it is necessary to interpret the output. In these steps, the predictions with class probabilities greater than a certain threshold are filtered out and then the overlapping predictions are reduced by Non-Maximum Suppression (NMS). As a result, the output of the model is presented as a list in which objects are detected and classified [31]. This list contains information such as the class label, location, and confidence score of each detected object. This information is useful for detecting objects and using them in targeted applications. The architecture of YOLOv8 offered by RangeKing is given in Fig. 2 [32].

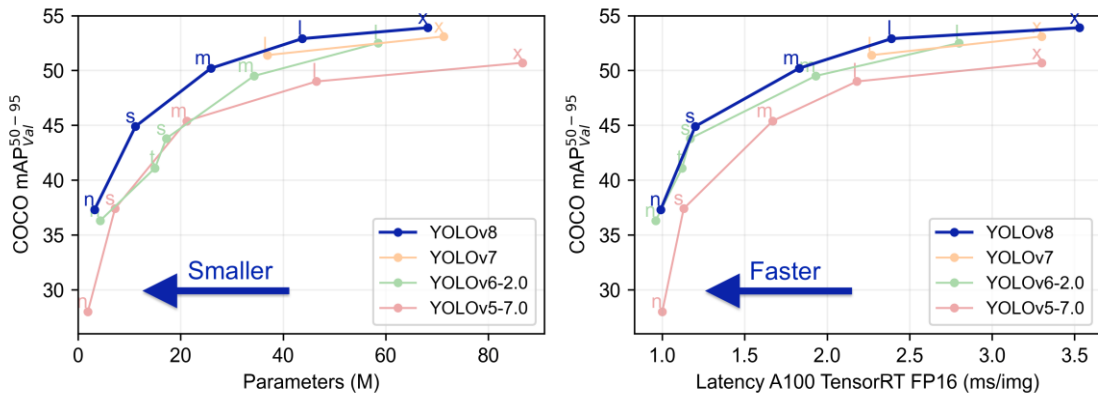


Fig. 3. YOLO versions comparison [26]

YOLOv8 brings certain significant improvements compared to previous YOLO versions (Fig. 3). In particular, in terms of object detection performance, YOLOv8 provides higher accuracy and faster processing speed. For example, compared to YOLOv7, YOLOv8 can achieve up to a 10 percent increase in object detection accuracy and at the same time achieve a 20 to 30 percent acceleration in processing speed. This is because YOLOv8 has a deeper and more complex architecture and uses more effective optimization techniques. In addition, YOLOv8 can learn larger data sets more effectively, which increases overall accuracy and generalization capability. Especially in resource-constrained environments such as mobile and embedded devices, YOLOv8's reduced memory usage and faster prediction capability are advantageous. For these reasons, YOLOv8 offers a better object detection solution compared to previous versions, providing higher accuracy, faster processing speed, and better scalability.

2.2. Object detection

2.2.1. Creation of the dataset

Within the scope of the study, a total of 3024 x 3024-pixel resolution and square aspect ratio was taken from different concrete and plaster surfaces, from different distances and angles, using an iPhone 11 mobile phone. Sample images used for training from the data set are given in Fig. 4. These images were then transferred to the Roboflow [33] application running on the internet. In the Roboflow environment, the inflorescence regions in the images were manually boxed and labeled. Then, data augmentation was applied to the images to increase the number of available data. Rotation, blur, and noise operations were applied to the images and the total number of images 617 was increased. Afterwards, the total data 617 was divided into 3 parts 80% train data, 10% validation data, and 10% test data. This labeled data set was created to be imported into Google Colaboratory [34] and exported as Python 3.10 [35] code. Label distributions for the efflorescence dataset are given in Fig. 5(a) and label correlations are given in Fig. 5(b).

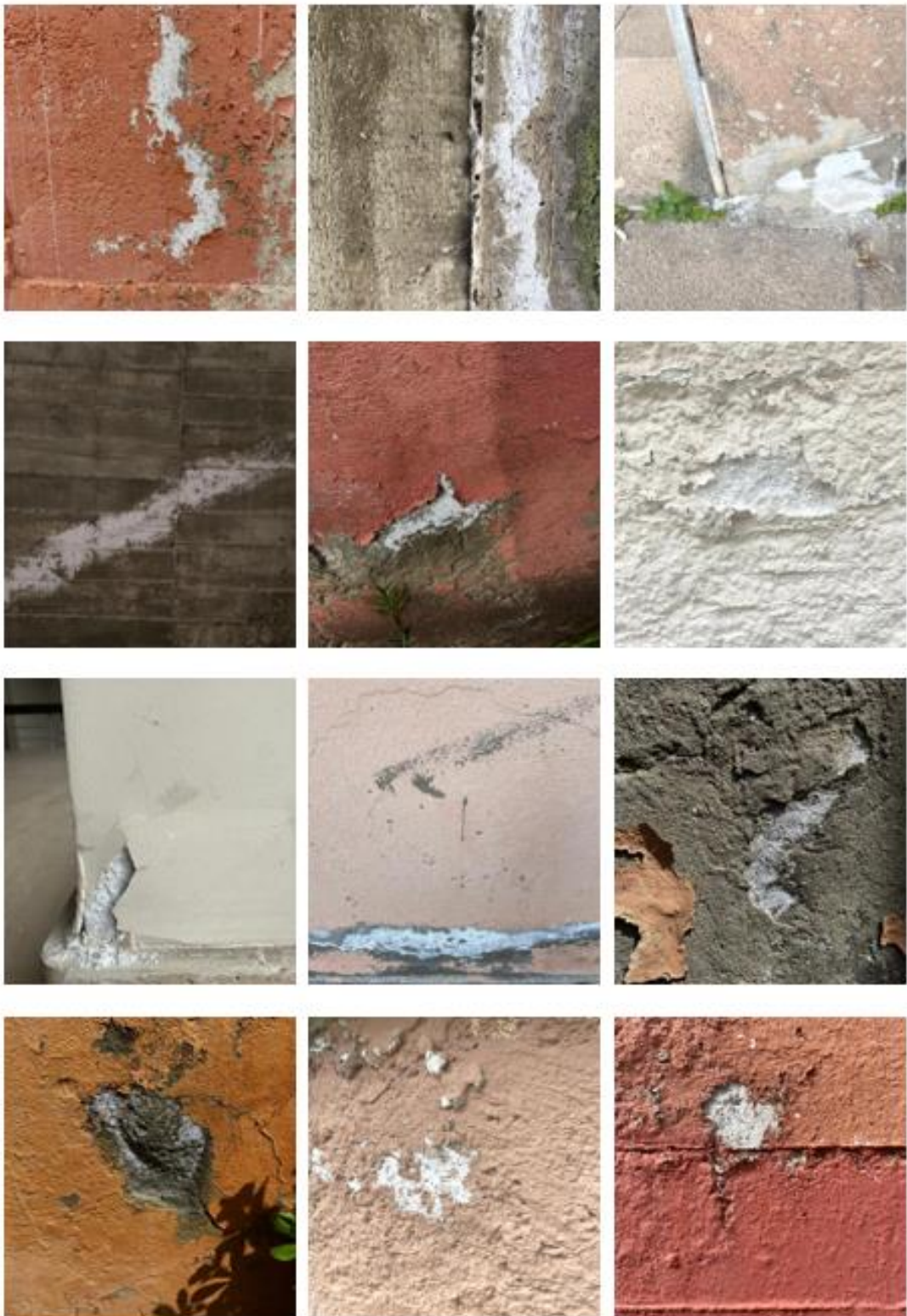


Fig. 4. Sample images used in the training set

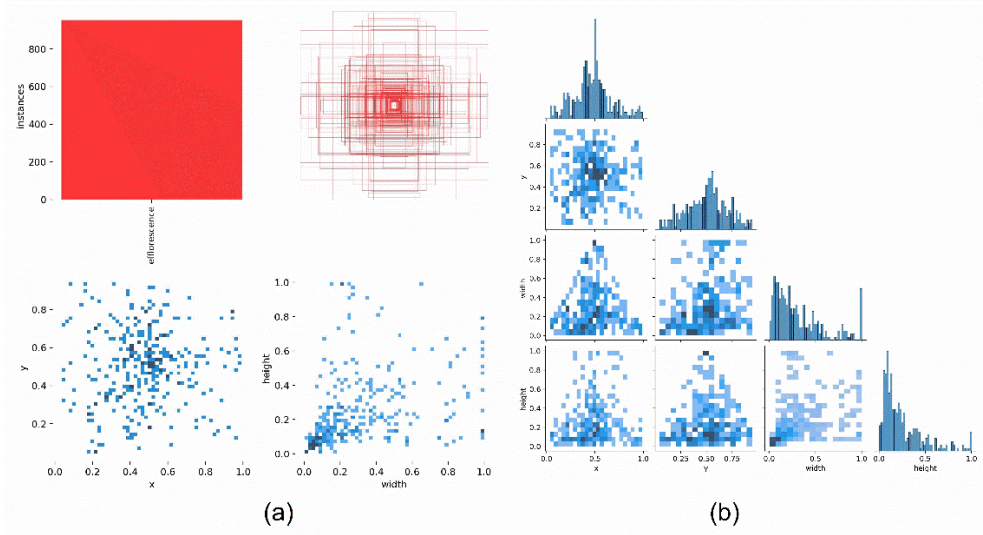


Fig. 5. Label distributions for the efflorescence (a), and label correlations (b)

2.2.2. Design and implementation

YOLOv8 is a transfer learning model, which operates by taking a pre-trained YOLO model and customizing it for a specific application or dataset. This process involves adjusting the general object detection capabilities of a pre-trained model to better suit the target application or dataset. In the study, NVIDIA Tesla T4, developed by NVIDIA, and T4, a graphics processor specially designed for GPU-accelerated computing, were used. It is commonly used in cloud-based services, servers, and data centers for artificial intelligence, deep learning, machine learning, data analytics, and other high-performance computing tasks. Initially, the YOLOv8x [28] model was pre-trained with known data, and then a training study of 300 epochs was performed using the custom data set of the study. This model was then tested with test data. These include an input image size of 640×640 pixels, an initial learning rate of 0.01, a momentum of 0.937, weight decay set at 0.0001, AdamW as the optimizer, and a batch size of 16.

3. Efflorescence detection results

The main metrics used to evaluate the performance of the object detection model include accuracy (1), precision (2), recall (3), and F1 score (4). Accuracy refers to the percentage of objects that the model correctly classifies, while precision refers to the percentage of items that the model predicts as positive that are positive. Sensitivity indicates the rate at which the model correctly detects items that are positive. The F1 score balances the classification performance of the model by expressing the harmonic mean of precision and sensitivity. This metric is calculated under the recall and precision curve and converted to an average value based on different averaging methods.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

where TP (True Positive) refers to the number of instances that the model correctly classifies as positive. That is, it represents the cases where positive instances are correctly identified. FP (False Positive) refers to the number of instances that the model incorrectly classifies as positive. That is, it represents cases where negative instances are incorrectly identified as positive. FN (False Negative) refers to the number of instances that the model incorrectly classifies as negative. That is, it represents cases where positive instances are incorrectly identified as negative. TN (True Negative) refers to the number of samples that the model correctly classifies as negative.

mAP50 and mAP50-95 (average precision) are two important metrics used to evaluate the performance of object detection models. mAP50 refers to the average precision of the model under a 50% cut-off threshold. This metric is obtained by calculating the area under the Precision-Recall curve for each class and averaging across classes. On the other hand, mAP50-95 indicates the average precision of the model under cut-off thresholds between 50% and 95%. This metric provides a more detailed performance analysis by evaluating the performance of the model at different cut-off thresholds over a wider range. The training and validation processes were carefully monitored to avoid overfitting and underfitting problems of the model. In the training and validation processes of the YOLOv8x model, four different types of losses were observed: bounding box loss, segmentation loss, and object loss. Object loss measures the probability of objects within the predicted bounding box; a higher object score indicates a higher probability of an object being inside the box. Bounding box loss determines how effectively the predicted bounding box covers an object. However, segmentation loss calculates the difference between the predicted target mask and the actual ground truth target mask on a pixel-by-pixel basis [36]. During the training and validation phases of the YOLOv8x [28] model, the three different loss plots showed a continuous decreasing trend, and no overfitting of the model was observed. Moreover, the training loss curve of the algorithm seems to converge at the initial stages of training. On the other hand, the validation loss curve of the algorithm started to converge at the end of the training process. Finally, it was observed that both training and validation loss curves reached minimum values after the completion of 300 training epochs. The training loss graphs, validation loss graphs, and performance metrics obtained from the training results are given in Fig. 6.

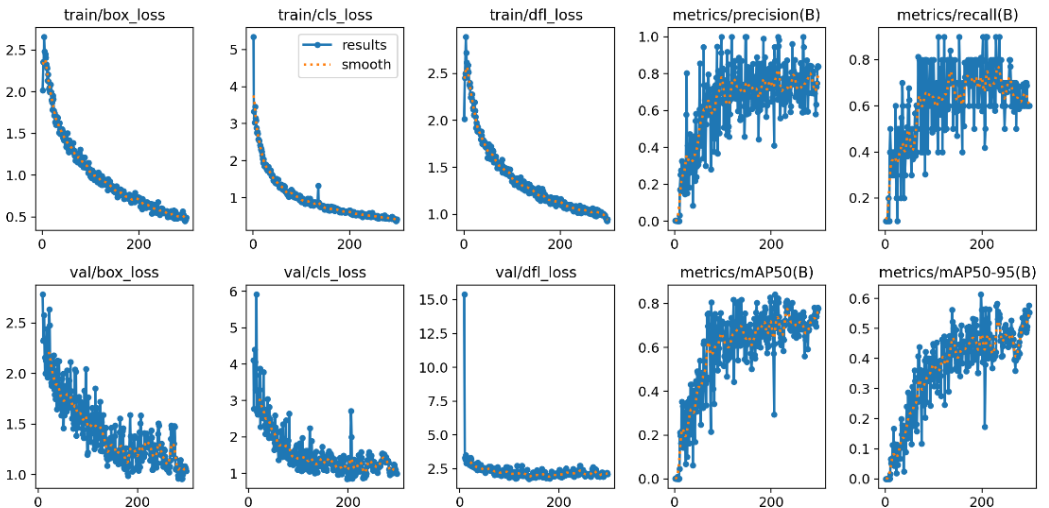


Fig. 6. Training performance metrics results



Fig. 7. Sample images of efflorescence detection

The test results showed that the efflorescence detection varies depending on the concrete surface texture, surface color, size of the efflorescence area, image acquisition angle, and distance. Especially if the surface color is unplastered or painted white, the object detection is significantly reduced. When the test images are examined, the efflorescence scores on dark-colored surfaces are on average 2 times better than light-colored surfaces. When the object detection model is evaluated with the studies in the literature, it is concluded that the effectiveness of the model in object detection is strong [37-39]. In addition, the algorithm was a little weak in detecting the effect of large and diffuse efflorescence. Apart from these cases, the algorithm can detect a high rate of efflorescence detection. As a result of the test run, six of the efflorescence detection scores are given in Fig. 7.

4. Conclusions

Using the YOLOv8x algorithm, a comprehensive efflorescence detection study was carried out on 617 photographs taken from regions where efflorescence is seen intensively. Based on the results, the following conclusions can be drawn:

- The accuracy, recall, precision, mAP50, and mAP50-95 performance measures obtained from the YOLOv8x model are 0.962, 0.902, 0.991, 0.851, and 0.6334 respectively. According to these performance metrics, efflorescence damage was considerably detected.
- The efflorescence detection score on colored surfaces is higher than on light-colored surfaces.
- Significant differences in the images taken from the surfaces (area, shape, color) caused fluctuations in the mAP50 and mAP50-95 values.
- Efflorescence detection was significantly affected by the angle of acquisition, distance of image, and area of efflorescence region. Efflorescence was detected at a higher rate in images taken perpendicular to the surface.

As a result of the study, it was determined that efflorescence was considerably detected using deep learning. Thanks to this study, efflorescence areas can be detected and repaired in a short time. In future studies, different durability situations can be detected with different deep learning models, and the coordinates of the areas that need maintenance and the durability problem can be determined.

Author contributions

Safa Nayır: Conceptualization, Data curation, Supervision, Writing, Reviewing and Editing, Investigation Methodology, Yılmaz Yılmaz: Conceptualization, Software, Data curation, Writing-Original Draft Preparation, Methodology, Visualization, Investigation, Validation.

Conflict of interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research received no external funding.

Data availability statement

The data presented in this study are available upon request from the corresponding author.

References

- [1] Monteiro NBR, Moita Neto JM, da Silva EA (2021) Environmental assessment in concrete industries. *J. Clean. Prod.* 327:129516. <https://doi.org/10.1016/j.jclepro.2021.129516>.
- [2] Baradan B, Yazıcı H, Ün H (2010) Beton ve Betonarme Yapılarda Kalıcılık (Durabilite). Türkiye Hazır Beton Birliği, Istanbul (in Turkish)
- [3] Zhang S, Ghoulah Z, Shao Y (2020) Effect of carbonation curing on efflorescence formation in concrete paver blocks. *J. Mater. Civ. Eng.* 32(6):04020127. [https://doi.org/10.1016/\(ASCE\)MT.1943-5533.0003210](https://doi.org/10.1016/(ASCE)MT.1943-5533.0003210).
- [4] Brocken H, Nijland TG (2004) White efflorescence on brick masonry and concrete masonry blocks, with special emphasis on sulfate efflorescence on concrete blocks. *Constr. Build. Mater.* 18(5):315–323. <https://doi.org/10.1016/j.conbuildmat.2004.02.004>.
- [5] Chin IR, L Petry (1993) Design and Testing to Reduce Efflorescence Potential in New Brick Masonry Walls. ASTM, West Conshohocken, PA.
- [6] Andrés A, Díaz MC, Coz A, Abellán MJ, Viguri JR (2009) Physico-chemical characterisation of bricks all through the manufacture process in relation to efflorescence salts. *J. Eur. Ceram. Soc.* 29(10):1869–1877. <https://doi.org/10.1016/j.jeurceramsoc.2008.11.015>.
- [7] Dow C, Glasser FP (2003) Calcium carbonate efflorescence on Portland cement and building materials. *Cem. Concr. Res.* 33(1):147–154. [https://doi.org/10.1016/S0008-8846\(02\)00937-7](https://doi.org/10.1016/S0008-8846(02)00937-7).
- [8] Neville AM (2002) Efflorescence: Surface blemish or internal problem. I: The knowledge. *Concrete International* 24(8):86–90.
- [9] Brocken H, Nijland TG (2004) White efflorescence on brick masonry and concrete masonry blocks, with special emphasis on sulfate efflorescence on concrete blocks. *Constr. Build. Mater.* 18(5):315–323. <https://doi.org/10.1016/j.conbuildmat.2004.02.004>.
- [10] Allahverdi A, Kani EN, Hossain KMA, Lachemi M (2015) Methods to Control Efflorescence in Alkali-Activated Cement-Based Materials. In: *Handbook of Alkali-Activated Cements, Mortars and Concretes* (Eds: F. Pacheco-Torgal, JA Labrincha, C Leonelli, A Palomo, P Chindapasirt). Woodhead Publishing. <https://doi.org/10.1533/9781782422884.3.463>.
- [11] Zhang L, Yang F, Zhang YD, Zhu, YJ (2016) Road crack detection using deep convolutional neural network. In: 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, pp. 3708–3712. <https://doi.org/10.1109/ICIP.2016.7533052>.
- [12] Silva WRL, Lucena DS (2018) Concrete cracks detection based on deep learning image classification. *Proceedings* 2:489. <https://doi.org/10.3390/ICEM18-05387>.

- [13] Park SE, Eem SH, Jeon H (2020) Concrete crack detection and quantification using deep learning and structured light. *Constr. Build. Mater.* 252:119096. <https://doi.org/10.1016/j.conbuildmat.2020.119096>.
- [14] Wang Z, Su C, Wang W (2020) Concrete cracks detection using convolutional neural network based on transfer learning. *Math. Probl. Eng.* 2020:7240129. <https://doi.org/10.1155/2020/7240129>.
- [15] Yoon J, Lee J, Kim G, Ryu S, Park J (2022) Deep neural network-based structural health monitoring technique for real-time crack detection and localization using strain gauge sensors. *Scientific Reports* 12(1):20204. <https://doi.org/10.1038/s41598-022-24269-4>.
- [16] Iannelli P, Angeletti F, Gasbarri P, Panella M, Rosato A (2022) Deep learning-based structural health monitoring for damage detection on a large space antenna. *Acta Astronaut.* 193:635-643. <https://doi.org/10.1016/j.actaastro.2021.08.003>.
- [17] Ye XW, Jin T, Yun CB (2019) A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Struct. Syst.* 24(5):567–585. <https://doi.org/10.12989/sss.2019.24.5.567>.
- [18] Jia J, Li Y (2023) Deep learning for structural health monitoring: Data, algorithms, applications, challenges, and trends. *Sensors* 23:8824. <https://doi.org/10.3390/s23218824>.
- [19] Bai Y, Sezen H, Yilmaz A (2021) Detecting cracks and spalling automatically in extreme events by end-to-end deep learning frameworks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2:161–168. <https://isprs-annals.copernicus.org/articles/V-2-2021/161/2021/>.
- [20] Hacıfendioğlu K, Başağa HB (2022) Concrete road crack detection using deep learning-based Faster R-CNN method. *Iranian Journal of Science and Technology – Transactions of Civil Engineering* 46(2):1621-1633. <https://doi.org/10.1007/s40996-021-00671-2>.
- [21] Xu G, Yue Q, Liu X (2023) Deep learning algorithm for real-time automatic crack detection, segmentation, qualification. *Eng. Appl. Artif. Intell.* 126:107085. <https://doi.org/10.1016/j.engappai.2023.107085>.
- [22] Qiu Q, Lau D (2023) Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* 147:104745. <https://doi.org/10.1016/j.autcon.2023.104745>.
- [23] Jiang P, Ergu D, Liu F, Cai Y, Ma B (2022) A review of Yolo algorithm developments. *Procedia Comput. Sci.* 199:1066-1073. <https://doi.org/10.1016/j.procs.2022.01.135>.
- [24] Kim JH, Kim N, Won CS (2023) High-speed drone detection based on Yolo-V8. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, Rhodes Island, Greece, pp.1-2. <https://doi.org/10.1109/ICASSP49357.2023.10095516>.
- [25] Hussain M (2023) YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* 11(7):677. <https://doi.org/10.3390/machines1107067>.
- [26] Redmon J, Divvala SK, Girshick RB, Farhadi A (2015) You only look once: unified, real-time object detection. <https://arxiv.org/abs/1506.02640>.
- [27] YOLO: Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/>.
- [28] Ultralytics. <https://www.ultralytics.com/tr/yolo>.
- [29] Lou H, Duan X, Guo J, Liu H, Gu J, Bi L, Chen H (2023) DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics* 12(10):2323. <https://doi.org/10.3390/electronics12102323>.
- [30] Reis D, Kupec J, Hong J, Daoudi A (2023) Real-time flying object detection with YOLOv8. <https://arxiv.org/abs/2305.09972>.
- [31] Yolo V8: A deep dive into its advanced functions and new features. <https://medium.com/@mujtabaraza194/yolo-v8-a-deep-dive-into-its-advanced-functions-and-new-features-f008599fe604>.
- [32] Range King. <https://github.com/RangeKing>.
- [33] Everything you need to build and deploy computer vision models. <https://roboflow.com/>.
- [34] URL-1. <https://colab.research.google.com/>.
- [35] URL-2. <https://www.python.org/>.
- [36] Yıldırım E, Sefercik UG, Kavzoğlu T (2024) Automated identification of vehicles in very high-resolution UAV orthomosaics using YOLOv7 deep learning model. *Turkish Journal of Electrical Engineering and Computer Sciences* 32:1. <https://doi.org/10.55730/1300-0632.4060>.

- [37] Diniz JDCN, de Paiva AC, Junior GB, de Almeida JDS, Silva, AC, Cunha AMTDS, Cunha SCAPD S (2024) A one-step methodology for identifying concrete pathologies using neural networks—using YOLO v8 and dataset review. *Applied Sciences* 14:10. <https://doi.org/10.3390/app14104332>.
- [38] Alipour M, Harris DK (2020) Increasing the robustness of material-specific deep learning models for crack detection across different materials. *Eng. Struct.* 206:110157. <https://doi.org/10.1016/j.engstruct.2019.110157>.
- [39] Jitendra MS, Srinivasu PN, Shanmuk Srinivas A, Nithya A, Kandulapati SK (2020) Crack detection on concrete images using classification techniques in machine learning. *J. Crit. Rev* 7(9):1236-1241. <https://doi.org/10.31838/jcr.07.09.224>.