sustainability

MDPI

Article

Automatic Ceiling Damage Detection in Large-Span Structures Based on Computer Vision and Deep Learning

Pujin Wang [1,2], Jianzhuang Xiao [1,3,*], Ken'ichi Kawaguchi [2] and Lichen Wang [4]

[1] Department of Structural Engineering, College of Civil Engineering, Tongji University, Shanghai 200092, China; wangpujin@tongji.edu.cn
[2] Institute of Industrial Science, University of Tokyo, Tokyo 153-8505, Japan; kawaken@iis.u-tokyo.ac.jp
[3] State Key Laboratory for Disaster Reduction in Civil Engineering, Tongji University, Shanghai 200092, China
[4] Department of Civil Engineering, School of Civil Engineering, Tianjin University, Tianjin 300350, China; wanglc@tju.edu.cn
* Correspondence: jzx@tongji.edu.cn

Abstract: To alleviate the workload in prevailing expert-based onsite inspection, a vision-based method using state-of-the-art deep learning architectures is proposed to automatically detect ceiling damage in large-span structures. The dataset consists of 914 images collected by the Kawaguchi Lab since 1995 with over 7000 learnable damages in the ceilings and is categorized into four typical damage forms (peelings, cracks, distortions, and fall-offs). Twelve detection models are established, trained, and compared by variable hyperparameter analysis. The best performing model reaches a mean average precision (mAP) of 75.28%, which is considerably high for object detection. A comparative study indicates that the model is generally robust to the challenges in ceiling damage detection, including partial occlusion by visual obstructions, the extremely varied aspect ratios, small object detection, and multi-object detection. Another comparative study in the *F*1 score performance, which combines the precision and recall in to one single metric, shows that the model outperforms the CNN (convolutional neural networks) model using the Saliency-MAP method in our previous research to a remarkable extent. In the case of a large-area ratio with a non-ceiling region, the *F*1 score of these two models are 0.83 and 0.28, respectively. The findings of this study push automatic ceiling damage detection in large-span structures one step further.

Keywords: ceiling damage detection; large-span structure; convolutional neural networks (CNN); object detection; deep learning

Citation: Wang, P.; Xiao, J.; Kawaguchi, K.; Wang, L. Automatic Ceiling Damage Detection in Large-Span Structures Based on Computer Vision and Deep Learning. *Sustainability* **2022**, *14*, 3275. https://doi.org/10.3390/su14063275

Academic Editor: Nicholas Chileshe

Received: 17 February 2022
Accepted: 9 March 2022
Published: 10 March 2022

1. Introduction

Ceilings, which serve as both structural and non-structural components in the interior space of buildings, are frequently disregarded by structural health monitoring (SHM) designers and have been shown to be dangerous to people when they collapse due to earthquakes or material degradation [1–3]. The fall of ceilings in large-span buildings, such as indoor stadiums, public buildings, and hospitals, is particularly hazardous because the remarkable height from the ground increases the impact on the human body and the vast area of the ceilings in these buildings increases the inspection costs and difficulties [4]. The detection of abnormalities in the ceilings is performed by expert inspectors and is critical for life and property preservation. Furthermore, these large-span structures serve as temporary shelters for inhabitants when calamities such as earthquakes or aftershocks strike Japan, in which case expert inspectors are unavailable and residents require immediate ceiling damage evaluation. Until present, ceiling damage detection has largely relied on on-site inspections by experts who have been properly trained to look for defects such as floating, deflection, spalling, corrosion, loose, disengagement, and deficiency in the ceilings, which is both costly and error-prone [5]. A system using a smart sensor board that collects strain gauge data and an inspection robot to analyze damage in ceiling elements has been

devised to detect the position and condition of the damage [6]. However, this approach is heavily reliant on the sensors, which only return strains in the ceiling boards and fail to identify other damage. A readily accessible, versatile ceiling damage detection approach is required to assist ceiling maintenance staff in their daily work routines and residents in severe situations.

Expert, on-site ceiling inspections are primarily an image-processing effort that can be automated using algorithms. However, until the current resurgence of machine learning, it was extremely difficult to build a comprehensive algorithm to detect all possible damage types in ceilings using traditional image processing approaches [7]. Deep learning, a branch of machine learning, has made astonishing achievements in natural language processing (NLP), speech recognition, pathological diagnoses, and object recognition [8]. A deep learning model is typically built and trained in the form of deep neural networks, through which multi-layer abstractions are performed. In the domain of image processing, the deep learning method, particularly convolutional neural networks (CNNs), has achieved astonishing achievements [8–10]. In civil engineering and structural engineering, deep learning solutions have been applied to detect concrete cracks [11–13], common damages in concrete and steel structures [14], and damages in steel surfaces [15]. The detectable defects, on the other hand, are usually limited to cracks in concrete and steel, which have common features that non-artificial-intelligence algorithms can recognize. In fact, there are a variety of ceiling damage types where deep learning can be used to its full potential in object classification and detection.

The shortage of training data is one barrier to large-scale implementations of deep learning in additional study topics. Only from adequate training data can a deep learning model really learn [16,17]. Transfer learning is an important technique for dealing with the lack of training data from decades ago, with the goal of developing lifelong machine learning algorithms that store and utilize previously learned knowledge [18]. The concept of transfer learning is simple: a high-performance machine learning model that reveals intelligent characteristics should have a high learning potential in a domain that is relevant to what it has learned. Transfer learning deep models are typically built on robust deep learning models that have been effectively pre-trained and have been applied to a variety of domains, including sentiment analysis [19], image classification [20], and medical diagnosis [21]. Ceiling damage detection is fundamentally an image processing task with the possibility of being covered by the transfer learning solution.

High prediction accuracies by the deep learning models are crucial indicators to the models. In fact, many deep learning models in various domains have played a draw to humans or outperformed humans [21–23]. The most essential criterion targeted by participants in image processing competitions is accuracy [24]. However, the inner mechanisms of the deep learning models are still unsolved mysteries of researchers [25]. Exploration and interpretation methods are required to provide trust and modification directions to the deep learning models. One possible method of interpreting the deep learning model is to use an attention map to visualize the regions contributing most to the final predictions. Recently, applications of the attention map have yielded irradiative findings in object classification and detection, allowing researchers to better understand deep learning mechanisms [26–28]. In our previous research using a CNN model with the Saliency-MAP method for ceiling damage detection [29], the CNN model obtained a prediction accuracy of 86.22% and the feature visualization results demonstrate that the CNN model can recognize the overall inside layout of the building and separate damages from the whole image by highlighting the pixels representing damage. However, the performance of the highlighted damages was strongly limited by the following two factors: the non-ceiling region area and the area ratio between the damaged part area to the whole image area. A much too sophisticated photographic method for capturing the ceiling images was suggested to comply with these restriction factors.

To overcome these restriction factors, large-span structure ceiling damage detection models based on the YOLO architectures using three alternative hyperparameters are

established, trained, evaluated, and compared in this study. The objective of this study is to develop an autonomous and robust ceiling damage detector for various types of damage in a variety of situations to secure a wide range of adaptability. The state-of-the-art YOLO architectures (You Only Look Once), namely the YOLO v4, YOLO v5, and YOLOX, are modified to establish ceiling damage detectors and then compared in terms of classification and localization performances under various circumstances. The remainder of this paper is organized as follows. Section 2 introduces the YOLO architectures and evaluation metrics. Section 3 introduces the establishment of the models by altering the hyperparameters and the training process. Section 4 demonstrates the results in the performance of the models, as well as the comparative analysis among the models, particularly the $F1$ score performance comparison between the best-performing model and the CNN model from our previous research. Section 5 presents the conclusions of this study.

2. Methodology and Algorithms

2.1. Deep Learning for Classification and Object Detection

The human brain can recognize and localize the target objects in an image instantly. However, these tasks have been difficult for computers for decades until the emergence of deep learning algorithms (DLA), particularly the convolutional neural networks (CNN). Image classification aims to separate classes of targets according to the features reflected in the images with output as one class, one scene, or one status in an image, while object detection can detect multiple objects in an image with locations and classes [30]. A modern object detection DLA usually contains of two parts: a pre-trained backbone on large-scaled datasets such as Pascal VOC (The PASCAL Visual Object Classes) [31] and COCO (Microsoft Common Objects in Context) [32] for feature extraction, and a head for class and localization prediction.

In our previous research, a deep learning model for intact and damaged ceiling classification was trained and evaluated [29]. To localize the damaged parts in a ceiling image, Grad-CAM (Gradient-weighted Class Activation Mapping) and Saliency-MAP methods were adopted to highlight the pixels contributing the most to predictions. Figure 1 shows the DLA in the classification and object detection for ceiling damage detection in large-span structures. The output of the classification DLA is a probability vector in the shape of [Intact, Damage], indicating the probabilities of intact and damage with the sum to 1. Next, the Saliency-MAP and Grad-CAM methods highlight the pixels contributing the most to the probability vector to detect the damaged parts in the image. This method is a utilization of the attention mechanisms in the DLA itself [33,34]. On the other hand, the output of the object detection DLA is a matrix of multiple detected objects with classes, confidence, and locations. The objects are detected in an end-to-end approach, which is easier to train and evaluate.

2.2. Literature Review of the YOLO Family for Object Detection

In the domain of object detection, a cutting-edge architecture consists of the two-stage detector, in which the detection process is divided into a region proposal stage and a classification stage. The R-CNN (Region Based Convolutional Neural Networks) series [35] is the most representative two-stage object detector, including Fast R-CNN [36], faster R-CNN [37], and Mask-R-CNN [38]. Even though the R-CNN series reaches relatively high results on object detection accuracy (mAP), the two-stage detector is more difficult in training and understanding [39]. The YOLO series (You Only Look Once) is an end-to-end object detector that is easier to train with high performance on accuracy and speed [40]. The YOLO series detects and localizes target objects in an image using end-to-end approaches, through which the classes and the locations are predicted at the same time. A modern YOLO object detector usually contains four parts [41]: Input, Backbone, Neck, and Head (shown in Figure 2). In the training phase, the data augmentation approach is applied in the input module to diversify the training data. These diversified data are then transferred to the backbone module for feature extraction to different levels. The neck performs feature

infusion using up-sampling and feature concatenation layers to provide more details for the last part: the head module, through which the final predictions to the classes and locations of the objects are compared with the ground-truth labels to generate the result of loss function. Next, the back-forward is performed to update the parameters in the modules of the backbone, the neck, and the head. Above is a whole training epoch with provided inputs. The training process is not complete until the loss function reaches a plateau. A well-trained detector can be evaluated by the indicator of AP (Average Precision), which comprehensively reflects the precision of the detected objects and the missing ones.

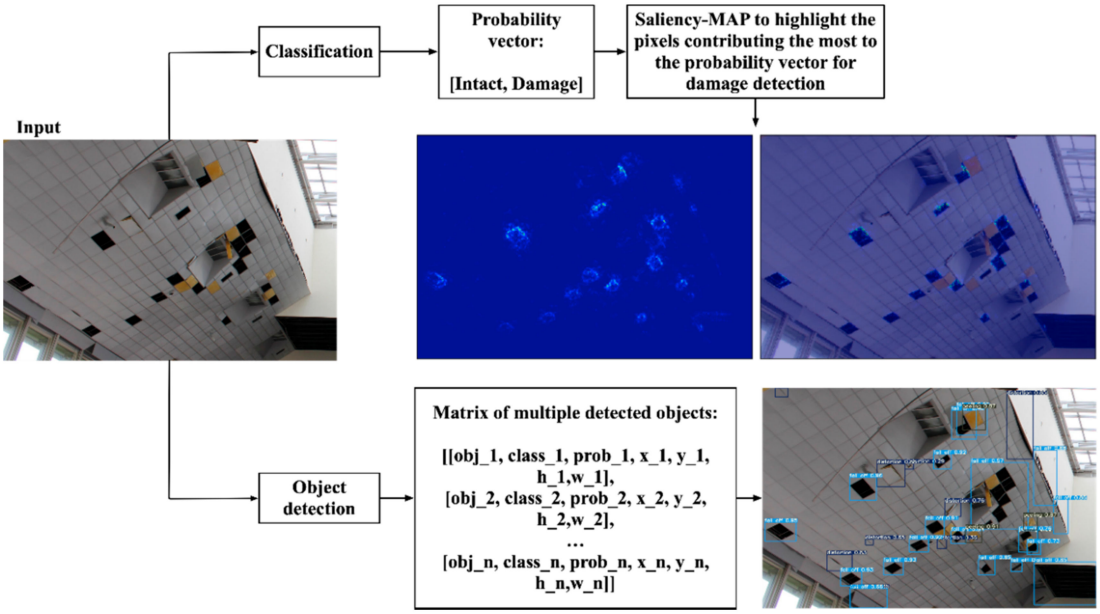

Figure 1. Image classification and object detection for ceiling damage detection in large-span structures.

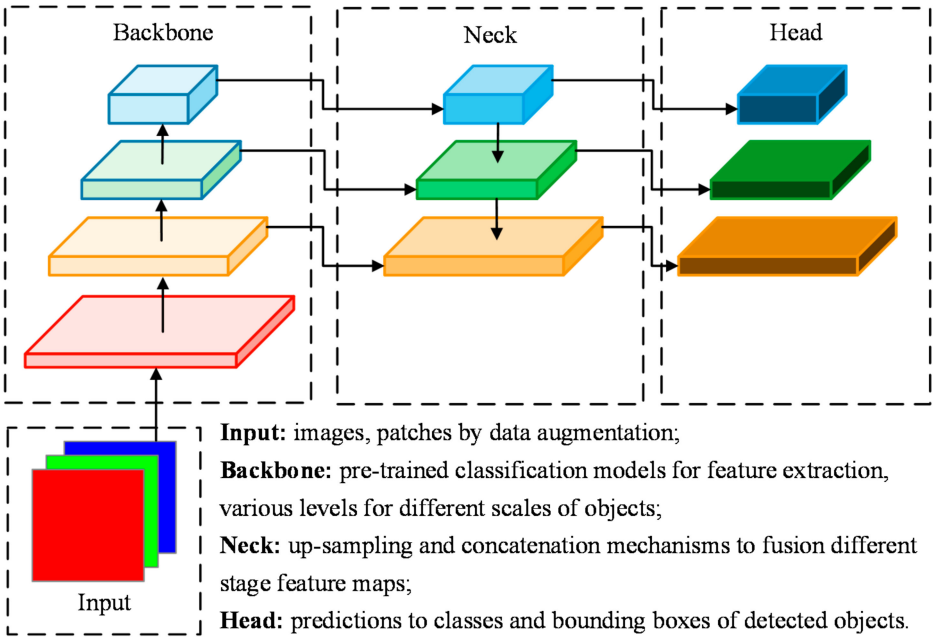

Figure 2. A YOLO architecture object detector.

A brief development history of the YOLO series is shown in Table 1. The rapid development of the YOLO series in recent years gives credit to the upgrades in these modules. Generally, it is possible for the backbone to extract more detailed features due

to the emergence of very deep convolutional networks such as ResNet (Residual neural network) [42]. The neck can concatenate multiple stage features using networks such as FPN (Feature Pyramid Networks) [43] and PAN (Path Aggregation Network) [44]. To the head module, approaches manipulating the anchors and the loss function such as multi-anchors single ground truth, decoupled anchors, and CIoU/GIoU (Complete Intersection over Union/Generalized Intersection over Union) loss can provide more compact information and improve the mAP (mean Average Precision) of the whole model.

Table 1. A brief upgrade history of the YOLO series.

YOLO Architecture	Released	Author	Upgrades in Modules and Performances (AP on COCO Test-Dev Dataset, %)	
V1 [40]	Jun. 2015	Joseph Redmon, et al.	Input	448 × 448
			Backbone	GoogLeNet [45]
			Neck	-
			Head	20 classes, MSE (Mean Squared Error) loss
			Performance	-
V2 [46]	Dec. 2016	Joseph Redmon, et al.	Input	224 × 224 for pretrain; 416 × 416 for detection
			Backbone	Darknet19
			Neck	-
			Head	20 classes, MSE loss
			Performance	21.6
V3 [47]	Apr. 2018	Joseph Redmon, et al.	Input	416 × 416
			Backbone	Darknet53
			Neck	FPN
			Head	80 classes, MSE loss
			Performance	33
V4 [41]	Apr. 2018	Alexey Bochkovskiy, et al.	Input	416 × 416, Eliminated grid sensitivity, CutMix and mosaic data augmentation, DropBlock regularization, class label smoothing
			Backbone	CSPDarknet53 (Cross Stage Partial Darknet53)
			Neck	SPP (Spatial Pyramid Pooling), FPN + PAN
			Head	YOLOv3 with multi-anchors single ground truth, self-adversarial training, cosine annealing scheduler, CIoU loss
			Performance	45.5
V5 [48]	Jun. 2020	Glenn Jocher, et al.	Input	640 × 640, adaptive anchor, adaptive image resizing
			Backbone	Focus CSPDarknet53
			Neck	SPP, cspFPN + PAN
			Head	YOLOv4 with adaptive anchor, GIoU loss
			Performance	55.4
X [49]	Jul. 2021	Zheng Ge, et al.	Input	640 × 640, strong augmentation, mosaic and mixup data augmentation
			Backbone	Darknet53
			Neck	FPN
			Head	Decoupled head, end-to-end detectors, anchor-free, multi positives, SimOTA (Simulation Optimal Transport Assignment), IoU/GIoU loss
			Performance	51.2

In this paper, three state-of-the-art architectures of the YOLO series developed in recent years (YOLO v4, YOLO v5 and YOLOX) are modified for the ceiling damage detection task.

2.3. YOLO v4

Since the announcement of YOLO v3 in April 2018, no new version has been announced for about two years. By this time, the Google EfficientDet [50] has reached higher mAP performance than YOLO v3. The EfficientDet employs EfficientNet [51] as the backbone for feature extraction and Neural Architecture Search (NAS) [52] to automatically design neural networks, which makes it seem that the development of models by human supervision has become obsolete. However, NAS requires enormous GPU power, making it difficult to win competitions in object detection research unless the participant is a large company. YOLO v4 is a significant advancement in YOLO history, launched in April 2020. Based on YOLO v3, YOLO v4 can be trained and put into practical use with only one GPU, generating a faster model with the same accuracy as EfficientDet, which is created by researchers at Google, who have huge computing power.

The architecture of YOLO v4 shown in Figure 3 consists of:

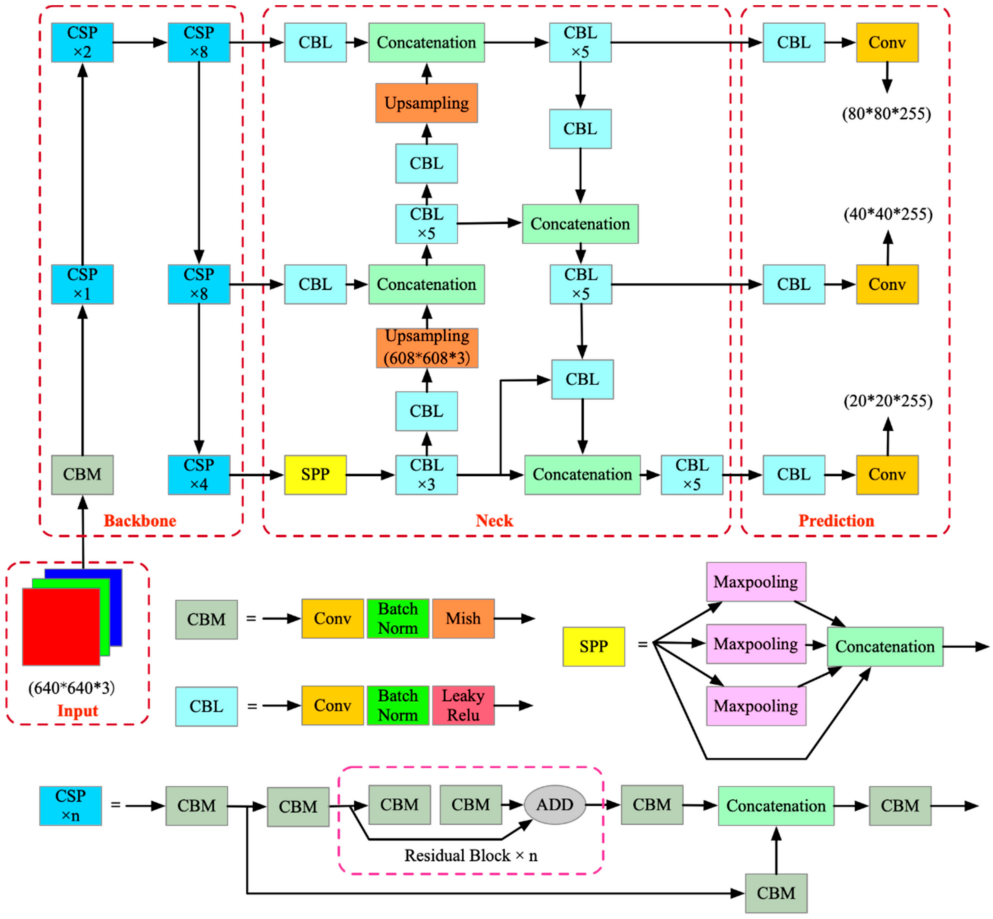

Figure 3. YOLO v4 architecture.

Input: Mosaic data augmentation shows the model multiple, resized images with different combinations at one time (Figure 4);

Backbone: CSPDarknet53 [53] is a unique backbone that augments the learning capacity of the CNN and mitigates the problem that heavy inference computations required in previous work;

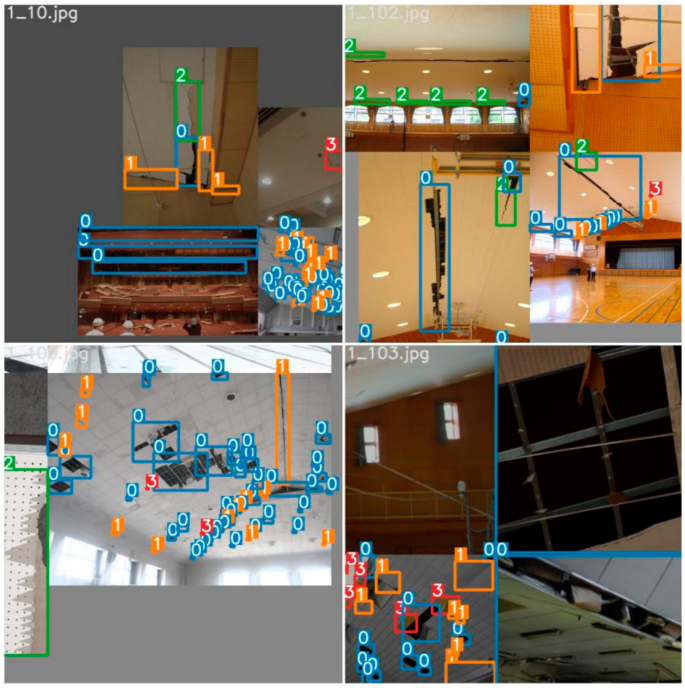

Figure 4. Mosaic data augmentation.

Neck: Spatial Pyramid Pooling (SPP) [54] is attached to CSPDarknet53 in order to improve the receptive field and distinguish the highly important context features while generating a fixed-length representation regardless of image size/scale. The Path Aggregation Network (PAN) [44] boosts information flow in a proposal-based instance segmentation framework by deploying in terms of the method for parameter aggregation for distinctive detector levels. The Feature Pyramid Network (FPN) [43] uses the inherent multi-scale pyramidal hierarchy of deep convolutional networks to construct feature pyramids with a marginal extra cost;

Head: The YOLO v3 head uses the output of the FPN for multi-scale object detection.

YOLO v4 has the following advantages: Anyone with a GTX 1080 Ti or 2080 Ti GPU can train a fast and accurate object detector; the influence of state-of-the-art "Bag-of-Freebies" and "Bag-of-Specials" object detection methods to train the detector has been verified and carefully selected to build the final YOLO v4 model; the modified state-of-the-art methods are more efficient and suitable for single GPU training.

YOLO v4 achieved an AP value of 45.5% over the Microsoft COCO dataset, and 65 FPS (Frame per second) in real-time detection performance using the GPU of Tesla V100, which is two times faster than EfficientDet with the same accuracy. In comparison with YOLO v3, the AP and FPS of YOLO v4 have been enhanced by 12.5% and 12%, respectively. YOLO v4's exceptional speed and accuracy with detailed and structured explanations are excellent contributions to the scientific realm of computer vision.

2.4. YOLO v5

YOLO v5 was released by Glenn Jocher, the Founder & CEO of Utralytics, in June 2020, only a few weeks after the release of YOLO v4. The YOLO v5 is a natural extension of the YOLO v3 PyTorch implementation. Even though it remains controversial to include YOLO v5 in the YOLO family due to the lack of a published paper, YOLO v5 is a lightweight, easy-to-modify, quick-inferenced, and well-performed object detection architecture. As shown in Figure 5, the YOLO v5 architecture has the following improvements:

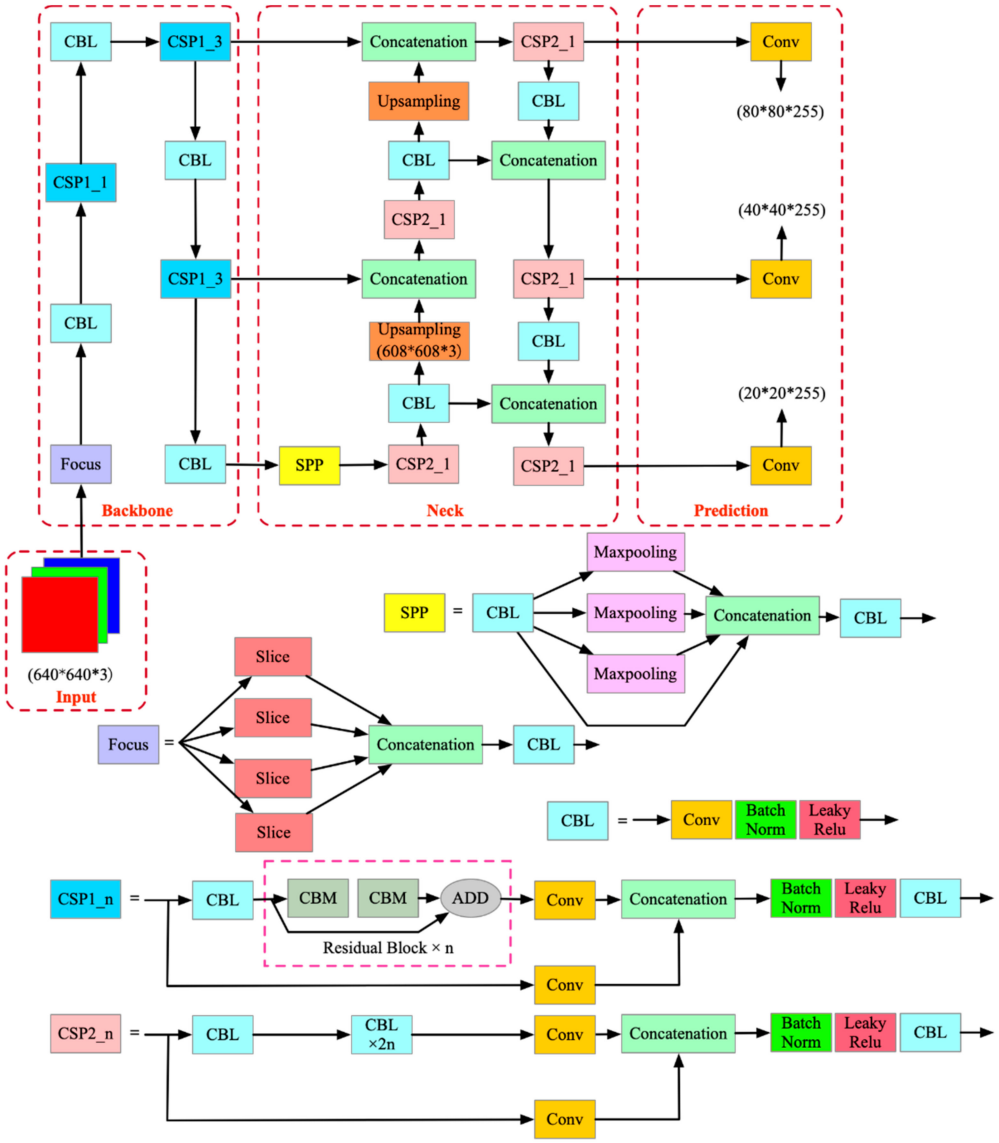

Figure 5. YOLO v5 architecture.

Input: Adaptive anchors by K-means can calculate the anchor boxes automatically suitable for the training data set. All anchor boxes are auto-learned in YOLO v5 to the custom data. A self-adaptive image scale adds the least black edges adaptively to the original image to accelerate the inference speed;

Backbone: Focus CSPDarknet53, where the focus mechanism concatenates higher resolution features with the lower ones by stacking into different channels instead of spatial locations. The focus mechanism is better for the model to learn small object features;

Neck: Cross stage partial connections (CSP) [53] is introduced to the FPN to shorten the feature extraction networks for higher speed;

Head: The same as YOLO v4.

YOLO v5 is easy to use for a developer implementing object detection into an application compared to other object detection frameworks with the following qualities: Positive sample augmentation using a neighboring positive sample anchor matching strategy; through variable configuration parameters, different levels of the model can be obtained; improves the overall performance through built-in hyperparameter optimization strategies.

2.5. YOLOX

YOLOX [49] is a state-of-the-art object detection model released in August 2021 by Megvii Technology with performance beyond YOLO v5. YOLOX was awarded first place in the Streaming Perception Challenge (Workshop on Autonomous Driving at CVPR 2021) with features including anchor-free, a decoupled-head, and SimOTA. YOLOX has been improved with YOLO v3-SPP [47] as a baseline compared to the over-optimized, anchor-based detectors with hand-crafted assigning rules for training such as YOLO v4 and YOLO v5. The confident title of YOLOX, "Exceeding YOLO Series in 2021", is supported by the marks of 47.3% AP on COCO, surpassing the current best practice by 3.0% AP and the first place of the Streaming Perception Challenge. As shown in Figure 6, the YOLOX architecture consists of:

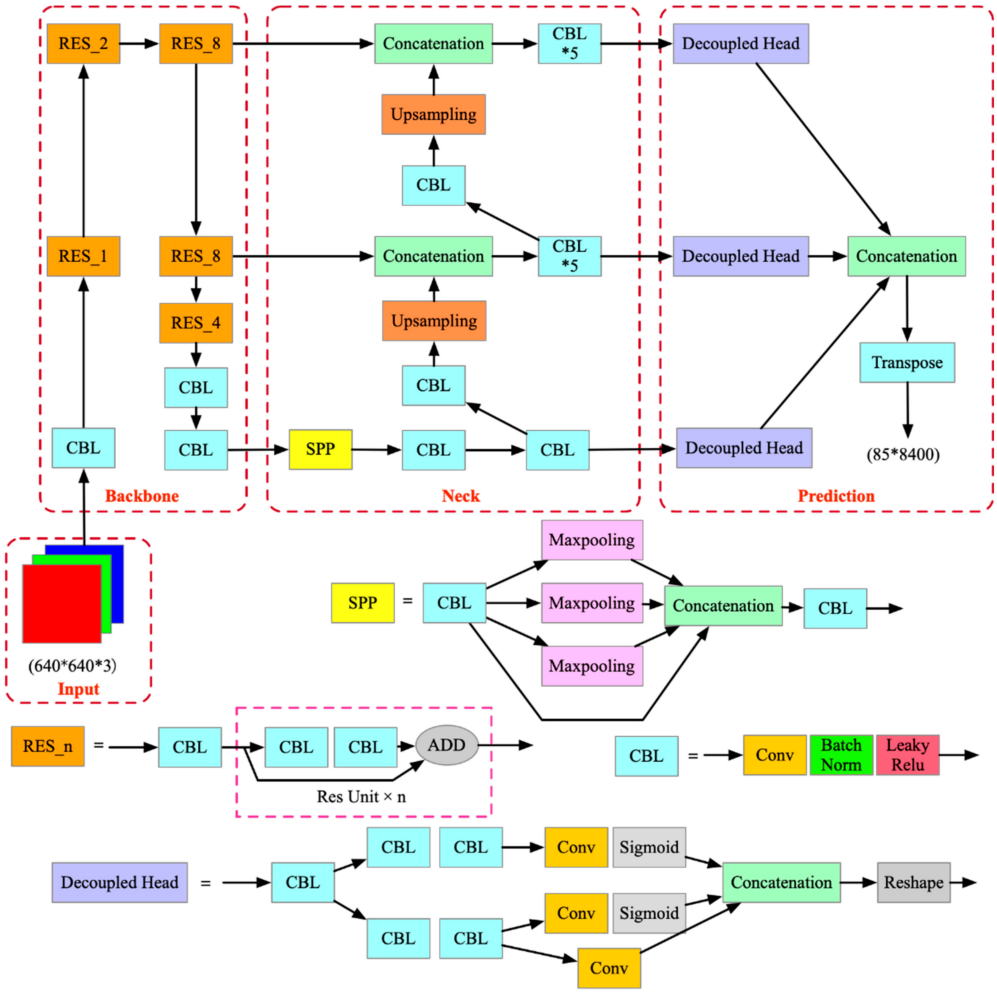

Figure 6. YOLOX architecture.

Input: Strong data augmentation with Mosaic and MixUp strategies makes training the whole model from scratch possible;

Backbone: Darknet53 for feature extraction;

Neck: FPN is the same as YOLO v3 and YOLO v4;

Head: Anchor-free mechanism significantly reduces the heavy parameter tuning with hand-crafted assigning rules and makes the training and decoding process simpler. Multi-positives assign the center 3 × 3 area as positives to alleviate the extreme imbalance between positive and negative samples during training. SimOTA is utilized as the advanced label assignment strategy and optimizes the loss function. The decoupled-head accelerates the convergence speed in training and improves the accuracy in detection.

2.6. Object Detection Evaluation Metrics

The predictions for an object detection task contain multiple detected objects with classes, confidence, and location, which are more sophisticated than those for a classification task. To evaluate these results, the most common metric used to measure the accuracy of the detections is the AP (average precision) [55]. Before the introduction of AP, some shared metrics between classification and object detection are:

***TP* (True Positive):** A correct prediction to a ground-truth label;

***FP* (False Positive):** An incorrect prediction to a non-existent label or a misplaced prediction to an existing object;

***FN* (False Negative):** A failure in predicting an existing object.

where: a label is a pre-defined class or an object with a bounding box.

In object detection, a *TN* (True Negative) prediction does not apply as a metric because there are infinite numbers of objects that should not have been detected within an image. To define "a correct prediction", "an incorrect prediction", and "a failure in prediction", the metric of *IOU* (intersection over union) is calculated to quantify the overlapping area between the prediction bounding box, B_p, and the ground-truth bounding box, B_{gt}, divided by the union area between them:

$$IOU = \frac{area\left(B_p \cap B_{gt}\right)}{area\left(B_p \cup B_{gt}\right)} \tag{1}$$

By a given threshold of *IOU* (which is 0.5 in this study), if $IOU \geq 0.5$ and the classification is correct, the prediction is correct and a *TP* is concluded. If $0 < IOU < 0.5$ or the classification is incorrect, the prediction is incorrect and a *FP* is concluded. If $IOU = 0$ and there exists a ground-truth bounding box, a *FN* is concluded. As stated above, additional evaluation metrics, precision *(P)* and recall *(R)*, can be calculated as:

$$P = \frac{TP}{TP+FP} = \frac{TP}{all\ detections} \tag{2}$$

$$R = \frac{TP}{TP+FN} = \frac{TP}{all\ grund-truths} \tag{3}$$

$$F1 = \frac{2 \cdot P \cdot R}{P+R} \tag{4}$$

Precision evaluates the percentage of correct positive predictions in all the detected objects. Recall evaluates the ability of the model to correctly find all the ground-truth bounding boxes. The precision and the recall have a trade-off relationship as follows: A low *FP* suggests that the model is good at correctly predicting the objects found by the model, but it could be the result of a high *FN* because the model is missing too many objects. Therefore, a well-functioning object detector requires its precision to stay high as its recall increases. The *F*1 score considers both *P* and *R* by computing the harmonic mean of them, which is easier to focus on. A more accurate indicator is *AP* (average precision), which summarizes the *P*-*R* curve into a single value by calculating the area using:

$$AP = \int_0^1 p(r)dr \tag{5}$$

P and *R* are always in the range of 0 to 1. As a result, *AP* also ranges from 0 to 1. *AP* is the evaluation of the detector for one class. For multiple class object detection, the evaluation of the model is *mAP* (mean Average Precision), which is the sum of *AP* divided by the number of classes.

3. Dataset Processing, Hyperparameter Tuning, and Approach Implantation

3.1. Dataset Generation

Since the Great Hanshin-Awaji Earthquakes in 1995, the Kawaguchi Lab has been collecting intact and damaged ceiling images in large-span structures such as indoor stadiums, natatoriums, concert halls, and traffic stations. These images were divided into intact and damage classes in our previous research for the classification DLA with the

original resolutions ranging from 5184 px × 3456 px to 3072 px × 2304 px. However, in the domain of object detection, negative samples, i.e., the intact ceiling images, are excluded from dataset formation because the models only need to learn to identify the damaged parts from the backgrounds. Figure 7 shows the intact ceilings and damaged ceilings, the latter ones are kept for the dataset formation with the total number of 914. Four main damage forms in ceilings, being peeling, crack, distortion, and fall-off, are summarized as target classes and labeled for training and testing from these images by experts. These four damage classes are shown in Figure 8, and details are shown in Table 2. In one image, it is possible to contain multiple damage classes and damage targets for detection. The total number of damaged objects for training, validation, and testing is 6809.

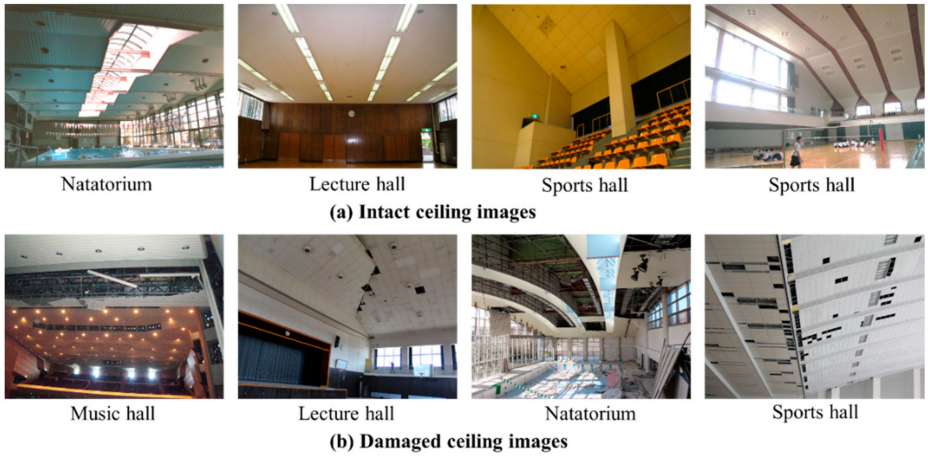

Figure 7. Examples of ceiling images.

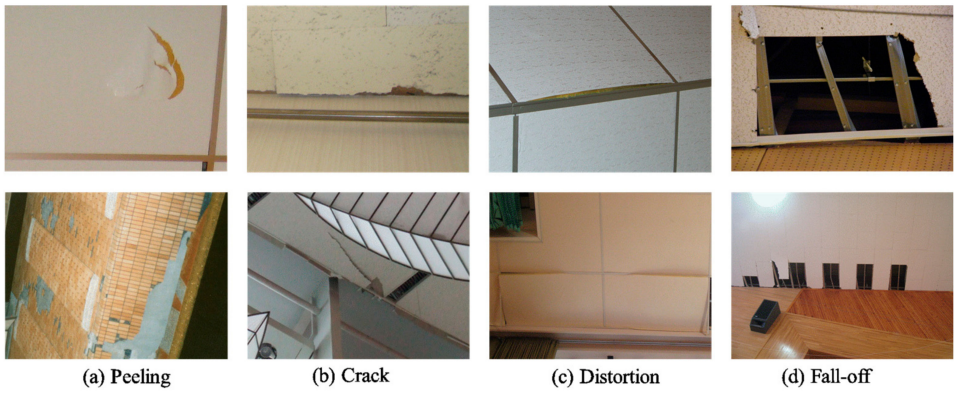

Figure 8. Damaged ceiling classes.

Table 2. Damaged ceiling classes for detection.

Class	Typical Causing Factors and Classification Criteria	Total Number
Peeling	Aging in the components, moisture and temperature cause peeling in the surface [56]. The surface comes off in strips or small pieces.	591
Crack	Squeezing and stretching in the ceilings. An uneven line on the surface of the ceilings along which have split without breaking apart.	574
Distortion	Earthquake, wind or other stress causing the ceilings crushing each other. The ceilings are wrenched or twisted out of shape with uniform edges.	2101
Fall-off	Severe external force or corrosion in the materials cause the failure of fall-off. A decrease in the ceilings leaves a void among the boards.	3843

3.2. Loss Function

In the YOLO series, the loss function is comprised of three parts:

1. the L_coor loss is responsible for the correctness to the coordinate prediction of a box covering an object;
2. the L_obj loss is for the confidence of the network that the predicted box covering the object;
3. the L_class loss is for deviations from predicting "1" for the correct classes and "0" for other classes for the object in the prediction box.

All of these loss functions are mean-squared error losses used to calculate the differences between the prediction and ground truth. As shown in Table 1, the YOLO v4 loss function uses L_{CIOU} as L_{coor}, while YOLO v5 uses L_{DIOU} as L_{coor}, and YOLOX uses SimOTA based on $L_{IOU/GIOU}$ as L_{coor} [41,48,49].

3.3. Hyperparameters in the Experiment Models

In the field of computer vision (CV), competitions for object detection are associated with specific metrics and datasets such as VOC, COCO, and ImageNet [57]. These datasets are typically made up of a huge number of everyday objects such as animals, transportation, and food. Even though the backbones in the DLAs perform well in classification, it is still unknown whether the hyperparameters in a particular DAL are appropriate for the designated detection targets, which are the four damage forms in this research. Therefore, different YOLO architectures, the volume of the networks, and the input/detection resolution for training and testing are chosen as the three hyperparameters for the following reasons: 1. Architectures of the DLAs determine the feature extraction process and information combination in the models and affect the performance extraordinarily. In this study, YOLO v4, YOLO v5, and YOLOX are chosen as the architecture hyperparameter. 2. The network volume size is a trade-off between speed and quality, where a larger volume is prone to possess more learnable layers, even though it is not a guarantee of greater mAP. Two weights, the small weight and the extra-large weight, with a volume ratio about 1:10 for each YOLO architecture are chosen as the weight scale parameter. 3. The dataset of a large-span structure contains many small objects to learn and detect, so a higher input and detection resolution is a potential favorable hyperparameter for the task in this research. The input/detection resolutions of 640 × 640 and 1280 × 1280 are chosen to compare the object detection performance in small and multi-object detection. In Table 3, 12 ceiling damage detection models are established due to the listed hyperparameters for training.

Table 3. Establishment of ceiling damage detection models.

Model	YOLO Architecture	Backbone	Weight Scale	Weight Size (M)	Input and Detection Resolution
v4_s_640	YOLO v4	CSPDarknet53-Tiny	Small	22.4	640 × 640
v4_s_1280	YOLO v4	CSPDarknet53-Tiny	Small	22.4	1280 × 1280
v4_x_640	YOLO v4	CSPDarknet53	Extra-large	244	640 × 640
v4_x_1280	YOLO v4	CSPDarknet53	Extra-large	244	1280 × 1280
v5_s_640	YOLO v5	Focus CSPDarknet53	Small	14.4	640 × 640
v5_s_1280	YOLO v5	Focus CSPDarknet53	Small	14.4	1280 × 1280
v5_x_640	YOLO v5	Focus CSPDarknet53	Extra-large	148	640 × 640
v5_x_1280	YOLO v5	Focus CSPDarknet53	Extra-large	148	1280 × 1280
X_s_640	YOLOX	Darknet53	Small	34.3	640 × 640
X_s_1280	YOLOX	Darknet53	Small	34.3	1280 × 1280
X_x_640	YOLOX	Darknet53	Extra-large	378	640 × 640
X_x_1280	YOLOX	Darknet53	Extra-large	378	1280 × 1280

3.4. Approach Implantation

The YOLO architectures shown in Table 3 are established, trained, and evaluated on a computer with a Core i9-10850K @3.60 GHz CPU, 32 GB of RAM, and an Nvidia

GeForce GTX3090 24GB GPU. The deep learning environment is Compute Unified Device Architecture (CUDA) v.11.2 with PyTorch 1.9.1. The training process is on the GPU. The whole damaged ceiling image dataset is divided into the training set, the validation set, and the test set with a ratio of 0.8: 0.1: 0.1. Each model is trained for 300 epochs to ensure that the loss of the whole model is optimized to a plateau. Figure 9 shows the loss-epoch curves in the training process for each model, where the loss is calculated by validating the validation dataset after each epoch. Even though the loss functions are different in the three architectures, the loss tendency is to achieve a plateau with no over-fitting presence. The batch size and iteration in each epoch are automatically determined for appropriate GPU memory usage (maximum 24 GB). Validation is conducted at the end of each epoch to evaluate the current training result. The weight file that generates the best result is saved as the final weight for one model. The training process details are shown in Table 4, where the weight size, batch size, and the resolution are three important parameters in GPU memory usage. The training time varies substantially among the YOLO architectures. Although YOLO v4 has the fastest training speed, the v4_x_1280 training time is twice as long as that of the v4_s_1280. The training time of the YOLOX models is slightly longer than the YOLO v4 models, but much faster than the YOLO v5 models.

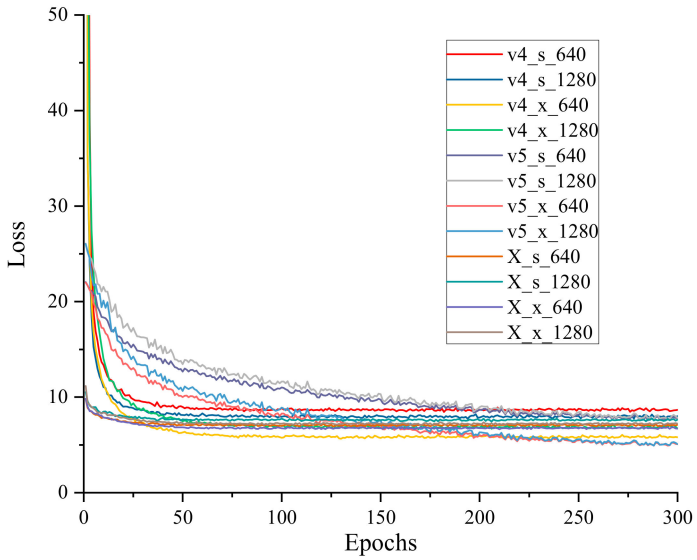

Figure 9. Loss-epoch curves.

Table 4. Training details.

Model	Weight Size (MB)	Batch Size	GPU Memory Usage (GB)	Training Time (h)
v4_s_640	22.4	64	16.99	5.92
v4_s_1280	22.4	16	17.14	7.15
v4_x_640	244	8	20.67	5.08
v4_x_1280	244	2	22.19	14.17
v5_s_640	14.4	128	23.14	24.68
v5_s_1280	14.4	32	23.69	27.33
v5_x_640	148	32	20.20	26.84
v5_x_1280	148	8	19.33	30.92
X_s_640	34.3	32	17.54	7.42
X_s_1280	34.3	8	18.95	11.58
X_x_640	378	8	20.72	9.40
X_x_1280	378	2	20.86	15.85

4. Results and Discussion

4.1. Evaluation and Comparison of the Models

The AP is defined as the area under the P-R curve shown in Equation (5). The mAP is the mean value of APs for all the categories. The P-R curves of all the damage classes in the models are shown in Figure 10. The AP and mAP are shown in Figure 10. The model X_s_1280 outperforms all other models in the AP and mAP evaluations. An ideal object detector has the characteristic of an AP area of 1. In Figure 10a peeling and Figure 10d fall-off, the P-R curves cover larger areas than those of the Figure 10b crack and Figure 10c distortion P-R curves, indicating that the models perform better in detecting fall-off and peeling damage. Despite the fact that the distortion samples are three times larger than the peeling samples, the peeling AP is substantially greater. One explanation for this phenomenon is to take the shape of the damage classes into account. The aspect ratio (the ratio of width to height) of the fall-off and peeling damage in the ground-truth data is closer to a square (1:1) than that of the crack and distortion damage. Square-shaped damage is easier for the model to learn due to the rotation of the object remaining a square-shaped aspect ratio.

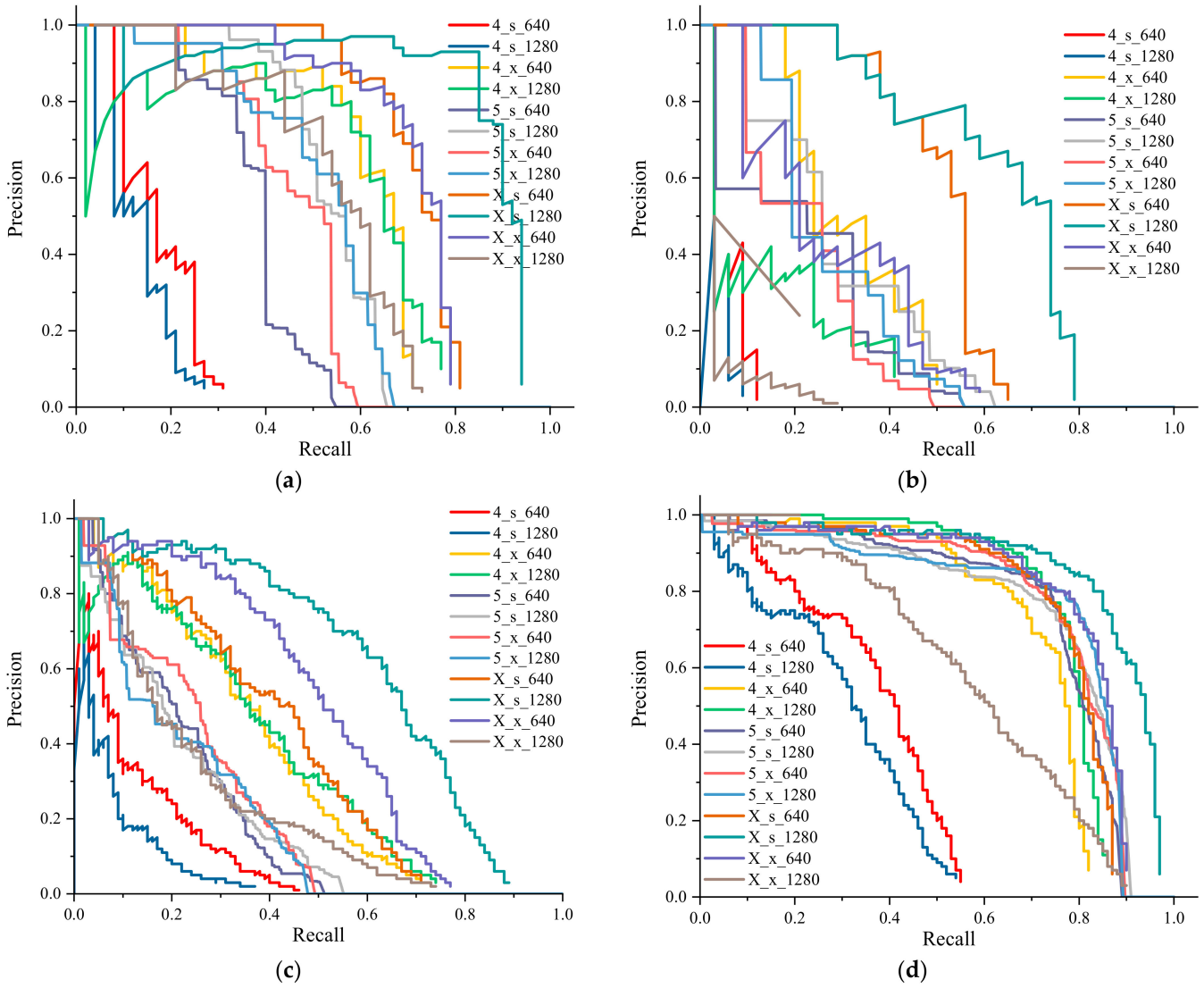

Figure 10. P-R curves of damaged ceiling classes: (**a**) Peeling, (**b**) Crack, (**c**) Distortion, and (**d**) Fall-off.

The AP of the four damage classes varies drastically amongst the models. For example, the distortion AP ranges from 6.80% to 62.39%, with the maximum value being 9.18 times

the minimum value. In the establishment of ceiling damage detection models (shown in Table 3), the YOLO architecture, the volume size of the networks, and the input/detection resolution are three hyperparameters for multi-factor analysis.

In architecture selection, the highest mAP of YOLO v4, YOLO v5, and YOLOX are 50.22%, 45.03%, and 75.28%, respectively, as shown in Table 5 and Figure 11. In fact, all of them are adequate when compared to the performances of the YOLO family on the COCO dataset in Table 1.

Table 5. AP and mAP of the models.

Model	AP (Average Precision) (%)				mAP (%)
	Peeling	Crack	Distortion	Fall-off	
v4_s_640	17.77	5.91	11.92	37.07	18.17
v4_s_1280	13.16	2.94	6.80	30.59	13.37
v4_x_640	60.97	32.13	36.01	71.75	50.22
v4_x_1280	59.58	14.71	37.24	77.63	47.29
v5_s_640	37.87	20.90	21.44	75.71	38.98
v5_s_1280	53.98	29.04	21.27	75.83	45.03
v5_x_640	45.90	22.18	23.75	77.28	42.28
v5_x_1280	51.20	27.28	21.18	76.39	44.00
X_s_640	72.15	51.18	40.33	77.36	60.26
X_s_1280	87.70	63.83	62.39	87.21	75.28
X_x_640	71.38	28.72	49.45	80.38	57.48
X_x_1280	55.10	4.76	24.62	58.30	35.69

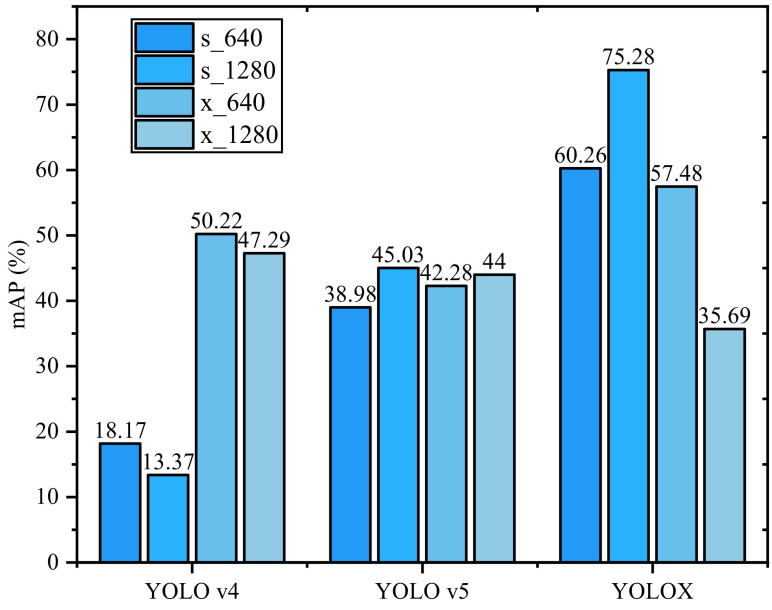

Figure 11. mAP of the YOLO series for large-span structure ceiling damage detection.

Furthermore, when compared to the model performance in the literature for pavement distress detection using Faster-RCNN, YOLO v3, and YOLO v4 architectures [58], the YOLOX_s_1280 model in this study achieves a mAP of 75.28%, which is a remarkable improvement of 18.68% over the best mAP of 56.6% in the literature. In this study, the YOLOX_s_1280 model outperforms the other two models significantly, indicating that the YOLOX architecture with new features, such as an anchor-free mechanism, a decoupled-head, and SimOTA, is provided with better classification and localization performance.

From the perspective of the second hyperparameter in the establishment of the models, which is the network volume, the size of all the extra-large weights are over 10 times more than those of the small weights shown in Table 3. A larger weight usually indicates a

higher mAP [49]. The maximum improvement in mAP when extra-large weights are adopted takes place in the group of YOLO v4 models. The mAP of (v4_s_1280, v4_x_1280) are (13.37%, 47.29%) with a 33.92% improvement, indicating that an extra-large weight improves the YOLO v4 architecture performance tremendously. However, the mAP of (v5_s_1280, v5_x_1280), (X_s_640, X_x_640), and (X_s_1280, X_x_1280) are (45.03%, 44.00%), (60.26%, 57.48%), and (75.28%, 35.69%), respectively. In these models, the mAP of the small weight models exceeds those of the extra-large weight models, indicating that an extra-large weight is not essential in the YOLO v5 and YOLOX architectures.

The input/detection resolution is the last hyperparameter in model configuration. Similar to the phenomenon that a larger weight usually suggests a higher mAP, a higher resolution for input/detection yields a higher mAP as well [41]. In this study, even though exceptions occur that the 640 × 640 models yield a higher mAP than the 1280 × 1280 models, such as (v4_s_640, v4_s_1280), (v4_x_640, v4_x_1280), and (X_x_640, X_x_1280), the X_s_1280 model with the highest mAP of 75.28% is 15.02% higher than the comparative X_s_640 model at 60.26%. The 1280 × 1280 input and detection resolution elevates the mAP performance extensively.

The YOLO v4, YOLO v5, and YOLOX architectures have similar features in data augmentation, backbones, and FPN neck. The YOLOX architecture, with features such as the decoupled-head, end-to-end detectors, the anchor-free mechanism, and SimOTA, yields a remarkable mAP compared to other models, indicating that the YOLOX architecture is the most appropriate for the large-span structure ceiling damage detection in this study.

4.2. Ceiling Damage Detection Results

The AP of peeling and fall-off is generally higher than the AP of crack and distortion, as shown in Table 5. This phenomenon occurs in the best-performed X_s_1280 model; in addition, the aspect ratios in peeling and fall-off are closer to 1. The results of ceiling damage detection using the X_s_1280 model are shown in Figure 12. The results cover most of the real-world working conditions in ceiling damage detection, including long-distance detection for small objects, partial occlusion detection, and multi-object detection, etc. Each result consists of two parts: the bounding boxes of the damaged class and the probabilities of the class being predicted by the model. The results show that in most cases, the damaged ceiling regions are correctly located and classified with a high prediction probability.

Figure 12a,b demonstrate the typical cases in the detection process where intricate decorative or structural patterns may mislead the algorithms if not properly trained. The diamond-shaped gratings with hanging ceiling voids in Figure 12a and the hollow-like multiple ventilators are distractions in Figure 12b, even to human detectors. The X_s_1280 model successfully detected most of the fall-off ceiling boards and the distortion damage in the ceiling against the wall. It fails to detect one distortion damage in the upper right of Figure 12a, spotted by an orange ellipse. This is an FN that fails to detect an existing damaged object. This FN failure is generally explained by it being a distortion ceiling board with the diamond-shaped gratings as surroundings, so the deformation has the character of a triangle that matches the diamond-shaped surroundings. Moreover, the area of the FN accounts for only 0.4% of the total image area. In Figure 12b, the model successfully detected all of the severe fall-off damages. Furthermore, the two distortion damages in the upper ceiling with longitudinal trends are detected as well, even though the probabilities are relatively lower. Nevertheless, the model has detected too many fall-off damages so that the microphone in the upper left corner is mistaken as a fall-off damage, resulting in a FP failure. The probability is 0.55, indicating that the model is not particularly sure about the detection.

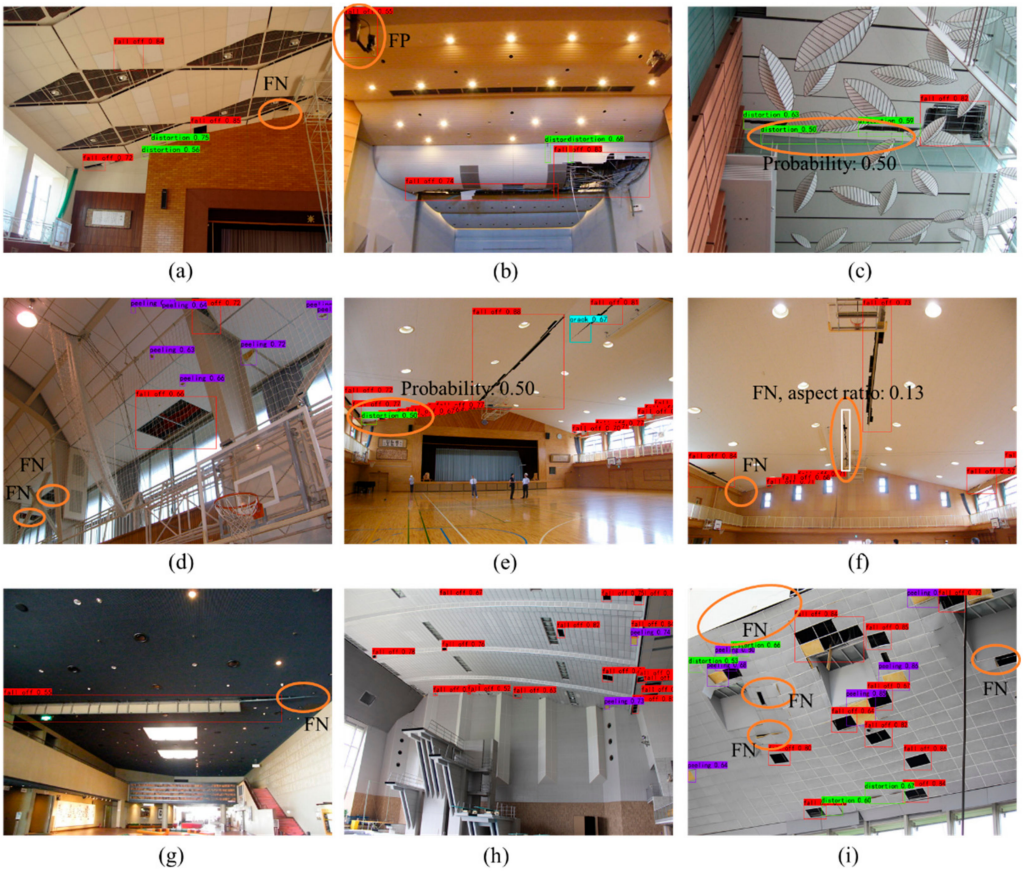

Figure 12. (**a**–**i**) Examples of ceiling damage detection using the X_s_1280 model.

In Figure 12c,d, the model is challenged by another typical distraction in ceiling damage detection: the partial occlusion by visual obstructions. In Figure 12c, all the distortion and fall-off damages are correctly detected by the model without any FP or FN. However, the detection of the distortion damage with a long horizontal extension (labeled by the orange ellipse) is on the brink of failure: the prediction probability is only 0.50, which means if the probability is 0.49 or lower, the model will ignore the object and that results in an FN failure. In fact, the distortion damage in the orange ellipse is trivial and difficult to detect even by human experts. In Figure 12d, the whole ceilings are protected by the fall protection net. The damage detection process has to go through them. The results indicate that the model successfully detects the fall-offs and the peelings in the boards nearer to the camera but fails to detect two fall-offs further in the ceilings, which result in two FN failures. The two missed fall-offs are partially blocked by both the fall protection net and the inclined steel frames. To elevate the performance of the model, more training dataset images containing damages in such forms are required.

In Figure 12e–g, the model faces the challenges of the extremely varied aspect ratios and small object detection. The drastic distortions evolve into transversely distributed damages in the ridge line and fall-offs in the ceiling-wall junction as shown in Figure 12e and f. The detection model correctly localizes the fall-off and crack damages with an aspect ratio (the ratio of width to height) ranging from 0.25 to 10, but fails to localize the crack damage with the aspect ratio of 0.13 as identified in the white rectangle in Figure 12f. This FN failure can be classified as small object detection, with the area ratio to the whole image as 0.68%. Another FN failure in Figure 12f is the missed detection of distortion damage in the lower left quarter. However, this damage is successfully detected in Figure 12e with the probability at the threshold value of 0.50, indicating the model needs more training data for damages with the characteristics of extreme aspect ratios and small objects. In Figure 12g, the model detects the fall-off damage with an aspect ratio of 10 until the ceiling boards halt

in falling, resulting in a FN in the corner. This FN is acceptable as a failure on account of the smooth edges and the triangle damage shape that are very rare, even in the real world.

Figure 12h,i demonstrate the same damaged structure from a long shot and a close shot. In Figure 12h, the small object detection ability is fully proven for the smallest area, as the ratio of the fall-off damage is merely 0.01%. Figure 12i shows the multiple object detection capability of the model with more than 20 TP. However, four FN are missed because these damage forms require more contextual information, which is lacking in the image.

In general, the X_s_1280 model with the mAP of 75.28% has shown good performance in real-world ceiling damage detection, which is challenged by the requirements of long-distance detection for small objects, partial occlusion detection, and multi-object detection.

4.3. Comparative Studies with Our Previous Research

In our previous research, a CNN model was proposed to detect the damaged ceilings of large-span structures using the Saliency-MAP method. A classification accuracy of 86.22% was obtained by dividing the training dataset into intact and damaged classes. In this study, a prediction accuracy of greater than 98% was obtained under the circumstances of no non-ceiling regions and an area ratio ranging from 20% to 30%, where the area ratio is the damaged-ceiling region to the overall image area [29].

However, the evaluation criteria in the previous research were based on the classification task, which is totally different from the criteria in the object detection performance, such as that performed by the YOLO family. To compare these results, the testing results of the damaged cases are classified into eight categories as with the previous research, while the performance is replaced with the $F1$ score shown in Equation (4), with the prediction confidence threshold 0.50. The statistics of the eight categories of damaged ceiling images are shown in Table 6, where the large-area ratio is defined as greater than or equal to 10%. The categories of the damaged ceiling images in Table 6 are identical to our previous research. Figure 13 and Table 7 show the comparison example results of the X_s_1280 model and the CNN model with the Saliency-MAP method with the FP results, labeled by the white ellipses, and the FN results, labeled by the orange ones. The comparisons of the results indicate that the X_s_1280 model outperforms the CNN model with remarkable improvements.

Figure 13a,b show that the CNN model using the Saliency-MAP method is well-trained for identifying the damages that are triangle-shaped, but is inferior in identifying damages with a regular-shape for the outlines of the fall-offs that are dim in Figure 13b-Saliency, which were confused with the sloped beams. The CNN model performs poorly in detecting small objects as well because it was trained by using the whole image labeled by a bivector as an input. Figure 13c shows that in the prediction of a close-up image, the X_s_1280 model can recognize more details of the damaged region, while the CNN model focuses on the damages with more area ratio. Figure 13d is a comparison for small object detection performance. The X_s_1280 model wins again because it can localize most of the damage. While the CNN model can clearly delineate the contours of the fall-offs, it can provide more visual and spatial information to the human user. In Table 7, the $F1$ scores of the X_s_1280 model are higher than those of the CNN model in the first three images with a remarkable value of 0.4, indicating the new X_s_1280 model is better at ceiling damage detection.

Figure 14 shows the $F1$ score performance of the two models under different influencing factors. The $F1$ score performance of the Saliency-MAP method is inferior to that of the X_s_1280 model in general. The prediction accuracy of the CNN model using the Saliency-MAP method has been proved susceptible to two important factors, being the non-ceiling region and the area ratio between the area of the damaged ceiling region in the image and the overall image area in the previous research. As shown in Figure 14, the $F1$ score performance of the Saliency-MAP method exhibits similar characters: a smaller, non-ceiling region and larger area ratio are two favorable factors that generate the highest $F1$ score, 0.53, in the category of a large area ratio without a non-ceiling region; on the

opposite spectrum, the small area ratio with a non-ceiling region generates the lowest *F*1 score of 0.13. These two factors have less influence on the *F*1 score performance of the X_s_640 model because the YOLO architecture has been particularly tuned for small object detection and object-wise based training.

Table 6. Statistics of eight categories of damaged ceiling images (identical to the previous research).

Categories of Images	Number of Images in Different Area Ratio Range								Total Number
	(0%, 5%)	(5%, 10%)	(10%, 20%)	(20%, 30%)	(30%, 50%)	(50%, 70%)	(70%, 90%)	(90%, 100%)	
With non-ceiling region	135	123	43	72	72	20	10	3	478
Without non-ceiling region	65	84	46	53	32	23	20	5	328
Large area ratio	-	-	89	125	104	43	30	8	399
Small area ratio	200	207	-	-	-	-	-	-	407
Large area ratio with non-ceiling region	-	-	43	72	72	20	10	3	220
Small area ratio with non-ceiling region	135	123	-	-	-	-	-	-	258
Large area ratio without non-ceiling region	-	-	46	53	32	23	20	5	179
Small area ratio without non-ceiling region	65	84	-	-	-	-	-	-	149

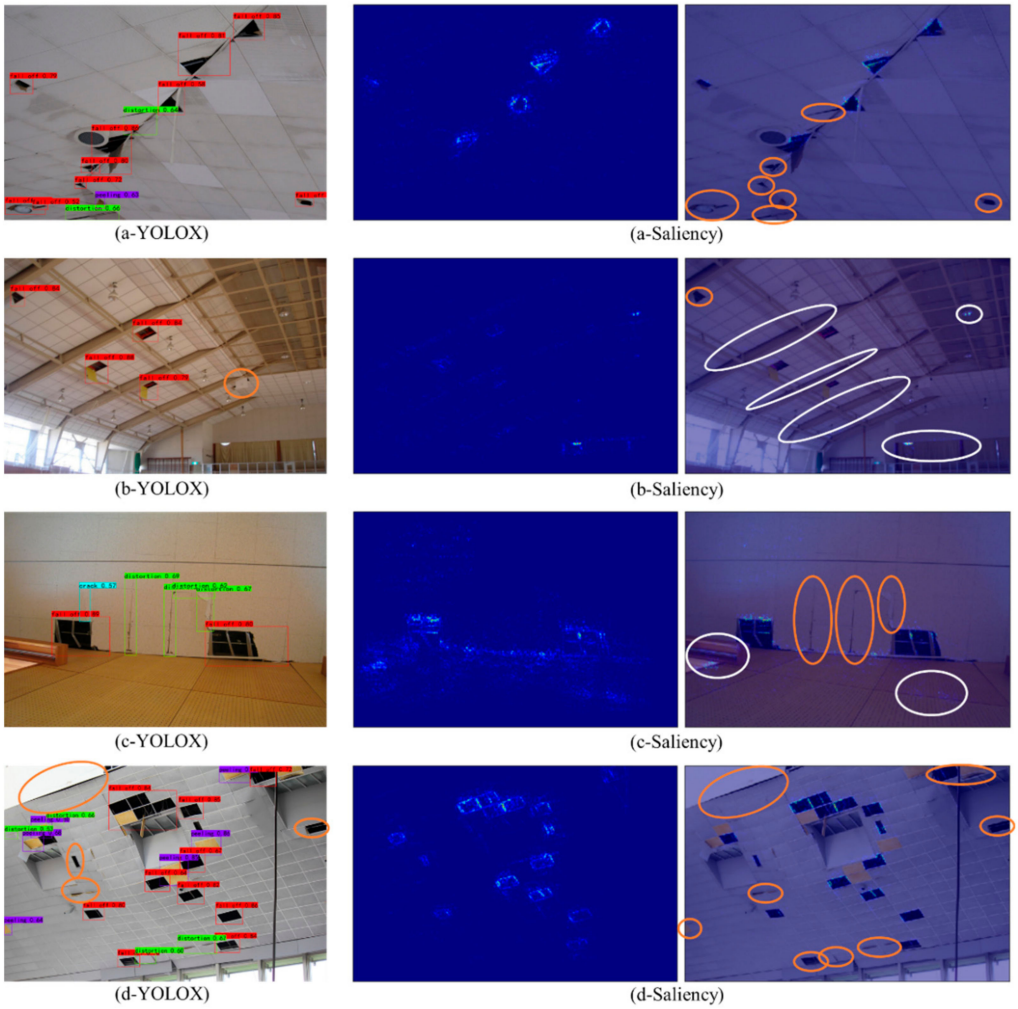

Figure 13. Examples of the X_s_1280 model and the CNN model with the Saliency-MAP method.

Table 7. The *F*1 results from Figure 13.

Figure	*TP*	*FP*	*FN*	*P* = *TP*/(*TP* + *FP*)	*R* = *TP*/(*TP* + *FN*)	*F*1 = 2·*P*·*R*/(*P* + *R*)
a-YOLOX	12	0	0	1.00	1.00	1.00
a-Saliency	5	0	7	1.00	0.42	0.59
b-YOLOX	4	0	1	1.00	0.80	0.89
b-Saliency	3	5	1	0.38	0.75	0.50
c-YOLOX	7	0	0	1.00	1.00	1.00
c-Saliency	4	2	3	0.67	0.57	0.62
d-YOLOX	20	0	4	1.00	0.83	0.91
d-Saliency	16	0	8	1.00	0.67	0.80

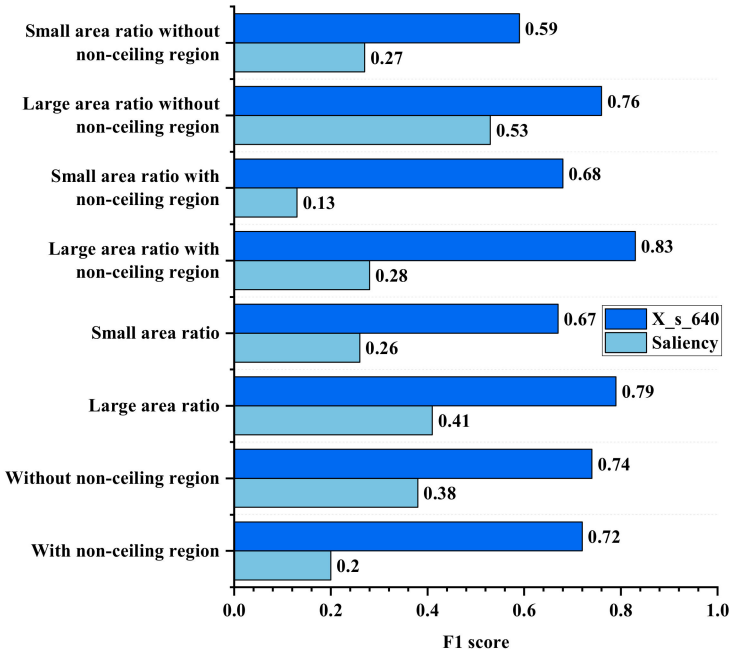

Figure 14. *F*1 score under different influencing factors.

In the previous research, the CNN model can obtain the best prediction without non-ceiling regions and with an area ratio ranging from 20% to 30%. A photographic method that improves the prediction accuracy of the CNN model is suggested in the following procedures: (1) Take an initial image of the entire ceiling region without non-ceiling regions. (2) The initial image is fed into the CNN model to generate the highlighted result for further investigation. (3) More test images are captured in the highlighted regions. (4) These test images are fed into the CNN model to obtain the prediction results and the damage locations. (5) Repeat this circulation until satisfactory results are finally obtained. In fact, this photographic method is a compromise when the CNN model performance is limited in small object detection and under complex circumstances, such as the existence of excess non-ceiling regions. These limitations are weakened or avoided by the YOLOX architecture for ceiling damage detection using an end-to-end approach.

4.4. Discussion on the Improvement of the YOLOX Model

The ceiling damage detection using YOLO series architectures has dramatically improved the performance in the multi object detection, the small object detection, and the *F*1 score compared to the CNN model using the Saliency-MAP method. It overcomes the difficulties in the existence of non-ceiling regions and small objects. The results of the detection are bounding boxes with the probabilities to one class, through which it is difficult to evaluate the severity of these damages. A more detailed classification in the

dataset generation can improve the recognition of the YOLO model. A future study can aim to collect more damaged ceiling images to provide severity information for the model to learn. Another improvement is to study the damage development in the ceilings by setting fixed detection cameras. This investigation helps to classify and detect the damages when they are trivial, but will deteriorate rapidly.

5. Conclusions

Ceiling damage detection in large-span structures is critical for safety inspection and maintenance. The key contribution of this study is the development of a YOLOX-based architecture that detects four major damage forms in the ceilings. Vision-based ceiling damage detection models using the YOLO series architectures were established, trained, and evaluated. In this study, a comparative analysis of different YOLO series models is undertaken to establish the best-performing model for the ceiling damage detection task. The comparative analysis of the *F*1 score performance between the best-performing model and the CNN model using the Saliency-MAP method is carried out as well. The following conclusions can be drawn from this study:

(1) Three hyperparameters, namely the YOLO architecture, the weight scale, and the input/detection image resolution, are chosen to establish 12 YOLO series models for the four classes of ceiling damage detection. The mAP performances of the 12 models are compared to find the best model was the X_s_1280 model, where "X" represents YOLOX, "s" represents the small weight scale, and "1280" stands for the 1280 × 1280 input/detection image resolution. Furthermore, a greater mAP is not guaranteed by a larger weight scale or a higher resolution.

(2) The mAP of the best-performing X_s_1280 model is 75.28%, with APs of 87.70%, 63.83%, 62.39%, and 87.21% for peeling, crack, distortion, and fall-off, respectively. The mAP is 15.02% higher than the second-placed X_s_640 model (640 refers to the input/detection image resolution of 640 × 640), which is 60.26%. Furthermore, the X_s_1280 model has shown a remarkable improvement with a 18.68% higher result than the best mAP of 56.6% in literature that applies YOLO v3 for pavement distress detection.

(3) The performance of the X_s_1280 model is generally robust to the challenges of partial occlusion by visual obstructions, the extremely varied aspect ratios, small object detection, and multi-object detection.

(4) The comparative study between the performances of the X_s_1280 model and the CNN model using the Saliency-MAP method demonstrates that the X_s_1280 model outperforms the CNN model to a remarkable extent and that the non-ceiling region and the area ratio are no longer strict constraints to the ceiling damage detection. In the case of a large-area ratio with a non-ceiling region, the *F*1 scores of these two models are 0.83 and 0.28, respectively. The sophisticated photographic method of the CNN model for ceiling damage detection is no longer essential and can be substituted with an end-to-end approach.

(5) The results of the detection are bounding boxes with the probabilities to one class. One downside of these results is that it is difficult to evaluate the severity in these detections because the probabilities are simply the model confidence. A more detailed classification in the dataset generation and the collection of more damaged-ceiling images are necessary to provide severity information to the model to learn in future studies.

Author Contributions: Conceptualization, P.W.; methodology, P.W.; software, P.W.; validation, P.W. and L.W.; resources, K.K.; data curation, L.W.; writing—original draft preparation, P.W.; writing—review and editing, P.W.; supervision, J.X.; funding acquisition, J.X. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China: 52078358; Taisei (Japan): Development of ceiling damage detection system using image database by deep learning approach; Science and Technology Committee of Shanghai Municipality: 21DZ1200403.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ogi, Y.; Kawaguchi, K.; Oya, S.; Katayama, S.; Kumagai, S.; Sakurai, S. Damage to non-structural components in large roof buildings failed during the Iwate-Miyagi Nairiku earthquake in 2008 or an earthquake in the north shore of Iwate prefecture in July 24th of 2008. *AIJ J. Technol. Des.* **2010**, *16*, 821–826. (In Japanese) [CrossRef]
2. Kawaguchi, K. Damage to non-structural components in large rooms by the Japan earthquake. In Proceedings of the Structures Congress 2012, American Society of Civil Engineers (ASCE), Chicago, IL, USA, 29–31 March 2012; pp. 1035–1044.
3. Farrar, C.R.; Worden, K. An introduction to structural health monitoring. *Philos. Trans. R. Soc.* **2007**, *365*, 303–315. [CrossRef] [PubMed]
4. AIJ. *Guidelines for Safety Measures against Accidental Fall of Ceilings and Other Non-Structural Components*; AIJ: Tokyo, Japan, 2015; ISBN 978-4-8189-4206-6. (In Japanese)
5. Ministry of Education, Culture, Sports, Science and Technology. Japan Guidebook for Earthquake Protection for Nonstructural Members of School Facilities (Revised Edition) Protecting Children from Falling and Tumbling Objects Due to an Earthquake—Implementing Earthquake Resistance Inspection 2015. Available online: https://www.nier.go.jp/shisetsu/pdf/e-gijyutsu2.pdf (accessed on 15 October 2021).
6. Nitta, Y.; Iwasaki, A.; Nishitani, A.; Wakatabe, M.; Inai, S.; Ohdomari, I.; Tsutsumi, H. Development of the damage assessment methodology for ceiling elements. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2012, SPIE-Int Soc Optical Engineering, Bellingham, DC, USA, 6 April 2012; Parts 1 and 2. Volume 8345. [CrossRef]
7. Alpaydin, E. *Introduction to Machine Learning*, 3rd ed.; MIT Press: Cambridge, UK, 2014; ISBN 978-0-262-02818-9.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
9. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
10. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Object detectors emerge in deep scene cnns. *arXiv* **2014**, arXiv:1412.6856v2.
11. Butcher, J.; Day, C.; Austin, J.; Haycock, P.; Verstraeten, D.; Schrauwen, B. Defect detection in reinforced concrete using random neural architectures. *Comput. Civ. Infrastruct. Eng.* **2013**, *29*, 191–207. [CrossRef]
12. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
13. Lin, Y.-Z.; Nie, Z.-H.; Ma, H.-W. Structural damage detection with automatic feature-extraction through deep learning. *Comput. Civ. Infrastruct. Eng.* **2017**, *32*, 1025–1046. [CrossRef]
14. Cha, Y.-J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput. Civ. Infrastruct. Eng.* **2017**, *33*, 731–747. [CrossRef]
15. Soukup, D.; Huber-Mörk, R. Convolutional neural networks for steel surface defect detection from photometric stereo images. *Database Syst. Adv. Appl.* **2014**, *8887*, 668–677. [CrossRef]
16. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
17. Raudys, S.; Jain, A.K. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 252–264. [CrossRef]
18. Pan, S.J.; Yang, Q. A Survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
19. Guerra, P.H.C.; Veloso, A.; Meira, W.; Almeida, V. Almeida, from bias to opinion: A transfer-learning approach to real-time sentiment analysis. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 150–158. [CrossRef]
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556v6.
21. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef] [PubMed]
22. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]

23. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
24. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.-H. Challenges in representation learning: A report on three machine learning contests. In Proceedings of the International Conference on Neural Information Processing; Springer: Berlin/Heidelberg, Germany, 2013; pp. 117–124. [CrossRef]
25. Castelvecchi, D. Can we open the black box of AI? *Nature* **2016**, *538*, 20–23. [CrossRef]
26. Das, A.; Agrawal, H.; Zitnick, L.; Parikh, D.; Batra, D. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Comput. Vis. Image Underst.* **2017**, *163*, 90–100. [CrossRef]
27. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. *arXiv* **2017**, arXiv:1704.06904v1.
28. Zheng, H.; Fu, J.; Mei, T.; Luo, J. Learning multi-attention convolutional neural network for fine-grained image recognition. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5219–5227.
29. Wang, L.; Kawaguchi, K.; Wang, P. Damaged ceiling detection and localization in large-span structures using convolutional neural networks. *Autom. Constr.* **2020**, *116*, 103230. [CrossRef]
30. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
31. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Computer Vision—ECCV 2014*; Part V 8693; Springer: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
33. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034v2.
34. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]
35. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
36. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497v3. [CrossRef]
38. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969. [CrossRef]
39. Sultana, F.; Sufian, A.; Dutta, P. A Review of object detection models based on convolutional neural network. In *Intelligent Computing: Image Processing Based Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–16. [CrossRef]
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
41. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-J.M. YOLOv4 Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934v1.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
43. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
44. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. [CrossRef]
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
46. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. *arXiv* **2016**, arXiv:1612.08242v1.
47. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767v1.
48. Jocher, G.; Changyu, L.; Hogan, A.; Lijun, Y.; Rai, P.; Sullivan, T. ultralytics/yolov5: Initial Release. *Zenodo* **2020**. [CrossRef]
49. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO series in 2021. *arXiv* **2021**, arXiv:2107.08430v2.
50. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. *arXiv* **2020**, arXiv:1911.09070v7.
51. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv* **2020**, arXiv:1905.11946.
52. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *J. Mach. Learn. Res.* **2019**, *20*, 1997–2017. Available online: https://arxiv.org/abs/1808.05377v3 (accessed on 15 October 2021).

53. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580. [CrossRef]
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*; Springer International Publishing: Cham, Switzerland, 2014; pp. 346–361. [CrossRef]
55. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
56. Charoenvai, S.; Khedari, J.; Hirunlabh, J.; Asasutjarit, C.; Zeghmati, B.; Quénard, D.; Pratintong, N. Heat and moisture transport in durian fiber based lightweight construction materials. *Sol. Energy* **2005**, *78*, 543–553. [CrossRef]
57. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; Da Silva, E.A.B. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
58. Zhu, J.; Zhong, J.; Ma, T.; Huang, X.; Zhang, W.; Zhou, Y. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* **2021**, *133*, 103991. [CrossRef]