Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Deep Learning-based Instance Segmentation for Indoor Fire Load Recognition

Yu-Cheng Zhou[1], Zhen-Zhong Hu[2], Ke-Xiao Yan[1], and Jia-Rui Lin[1,3]
[1]Department of Civil Engineering, Tsinghua University, Beijing 100084, China
[2]Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China
[3]Tsinghua University-Glodon Joint Research Centre for Building Information Model (RCBIM), Tsinghua University, Beijing 100084, China

Corresponding author: Jia-Rui Lin (e-mail: lin611@tsinghua.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 72091512 and 51908323.

ABSTRACT Accurate fire load (combustible objects) information is crucial for safety design and resilience assessment of buildings. Traditional fire load acquisition methods, such as fire load survey, which are time-consuming, tedious, and error-prone, failed to adapt to dynamic changed indoor scenes. As a starting point of automatic fire load estimation, fast recognition and detection of indoor fire load are important. Thus, this research proposes a computer vision-based method to automatically detect indoor fire loads using deep learning-based instance segmentation. First, indoor elements are classified into different categories according to their material composition. Next, an image dataset of indoor scenes with instance annotations is developed. Finally, a deep learning model, based on Mask R-CNN, is developed and trained using transfer learning to detect fire loads in images. Experimental results show that our model achieves promising accuracy, as measured by an average precision (AP) of 40.5% and AP_{50} of 59.2%, for instance segmentation on the dataset. A comparison with manual detection demonstrates the method's high efficiency as it can detect fire load 1200 times faster than humans. This research contributes to the body of knowledge 1) a novel method of high accuracy and efficiency for automated fire load recognition in indoor environments based on instance segmentation; 2) training techniques for a deep learning model in a relatively small dataset of indoor images which includes complex scenes and a variety of instances; and 3) an image dataset with annotations of indoor fire loads. Although instance segmentation has been applied for several years, this is pioneering research on using it for automated indoor fire load recognition, which paves the foundation to automatic fire load estimation and resilience assessment for the built environment.

INDEX TERMS building resilience, deep learning, fire load recognition, fire safety, indoor scene, instance segmentation, performance-based design

I. INTRODUCTION

Building fires are one of the most frequently occurring accidents in urban environments, resulting in casualties and massive asset losses. According to the CTIF World Fire Statistics Center [1], there are 7 to 8 million uncontrolled fires in the world each year. The Chinese Fire Statistical Yearbook [2] recorded 395052 building fires in China, resulting in 1815 deaths, 1513 injuries, and a loss of 4.7 billion Chinese Yuan in 2014. Economic losses have since increased by a factor of ~2.4 compared with 2010 and appear to be increasing. Modern commercial buildings tend to be large in scale, made of synthetic materials, and are typically characterized by crowding, high electrical equipment density, and complex functional zones [3], [4]. Thus, the prevention and control of indoor fires are critically important social issues.

Fire load density (i.e., the total heat produced from combustible materials in a given space) is one of the most important factors involved in the assessment of building fire risk, exhibiting high uncertainty and strongly influencing the temperature reached during a fire event [5]. However, the determination of fire load density is not an easy task. Currently, the study of indoor fire loads is dominated by direct investigations such as field surveys [6], which have conventionally been used to assess the likelihood of a fire in a given location, and are conducted using tools such as electronic scales, tapelines, and digital cameras. Commonly used surveys include the 1) weighing method, 2) inventory method, 3) the combination of weighing and inventory methods, and 4) questionnaires [6]. These surveys represent a large range of fire load density values, from which conservative codes and standards can be recommended (e.g., 80% fractile or higher). These are intended to reflect low fire probability, which is generally a deterministic value that would be the same for any other room in a specified occupancy category (e.g., all office rooms would have the same fire load density). Estimated values can be refined by considering the compartment area and room usage [5]. However, two critical disadvantages exist in the survey-based methods. First, fire load surveys are time- and cost-prohibitive because the design and implementation of a survey requires extensive manual labor [7]. Human error, bias, and inconsistency are also difficult to avoid in field surveys, especially in complex environments. Second, for a specified room, the fire load density is a deterministic value based on a category and is thus not suitable for performance-based fire design. This highlights the urgent need for an accurate, efficient, and cost-effective approach to estimate fire loads in a dynamic changed indoor scene.

However, few studies have been conducted for automating the manual survey-based fire load estimation method. To address this problem, computer vision-based fire load detection and estimation is an efficient and promising technique. In recent years, advances in computer vision have been driven by improved deep learning techniques. In state-of-the-art studies, deep learning methods have achieved promising results for instance segmentation tasks [8] (i.e., pixelwise recognition of each instance and its class in an image). In the architecture, engineering, and construction (AEC) community, computer vision techniques have been widely applied in construction safety and health detection[9], construction progress and building indoor monitoring[10], [11], and so forth. For construction safety and health, applications of instance segmentation are emerging and demonstrate noted improvements. Examples include the detection and segmentation of ground penetrating radar (GPR) signatures to inspect civil infrastructures [12] and moisture marks of shield tunnel lining to ensure safety [13]. The application of the monitoring also produces a large amount of visual data for the inside and outside of buildings.

To achieve this, the first and most important step is recognizing fire loads (combustible objects) including their types, and then algorithms can be applied to further estimate the fire load. Thus, this research proposes a novel deep learning-based instance segmentation method to automatically detect indoor fire loads. First, indoor elements are classified into different categories according to their material composition. Then, a dataset consisting of indoor scene images with instance annotations is established. Finally, a deep learning model is developed for instance segmentation, based on a mask region-based convolutional neural network (Mask R-CNN) and transfer learning. Results show that our method can achieve promising accuracy and high efficiency for instance segmentation on the dataset. To the best of our knowledge, this is the first attempt to employ a computer vision-based method for fire load recognition, which can be utilized in the future to obtain a fire load estimation or real-time fire risk assessment, and can be applied to multiple scenarios and fields in civil engineering.

The remainder of this paper is organized as follows. Section 2 reviews conventional fire load investigation methods and provides a brief introduction to deep learning-based image recognition. Section 3 illustrates the methodology of this research. Section 4 describes the development of the dataset. Section 5 describes the proposed automated indoor fire load recognition method based on deep learning. The experiments and verification results are discussed in Section 6. The contributions, applicability, and limitations of the method are discussed in Section 7. Finally, Section 8 concludes this paper.

II. RELATED WORK

A. FIRE LOAD SURVEY

Fire load, which is defined as the quantity of energy released by the complete combustion of all combustible material in a fire compartment, has a strong influence on the temperature development during a compartment fire [7]. Therefore, the assessment of fire load as an input to model the time-temperature relationship is an important task in building fire design. Fire load can be subdivided into fixed (e.g., the floor, walls, ceiling, and structural decorations) and mobile (e.g., shelves, counters, tables, chairs, and other furniture) categories [14]. These are often recorded during the field survey. Commonly used surveys include the 1) weighing method, 2) inventory method, 3) the combination of weighing and inventory methods, and 4) questionnaires [6]. Fire load density can then be calculated after the data are collected and verified.

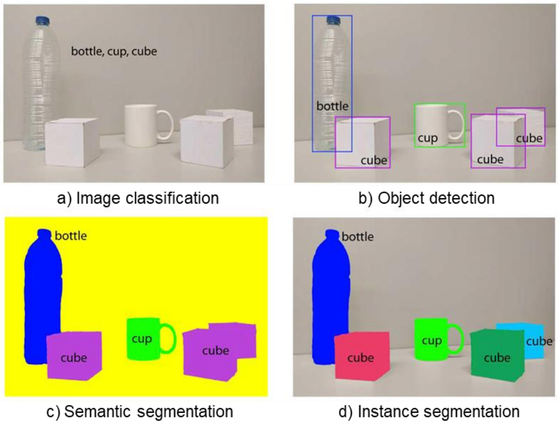
Fire load density for a specified room, evaluated based on codes and standards, is generally a deterministic value that is the same for any other room in a specified occupancy category (e.g., all office rooms have the same fire load density value) [5]. Statistics collected from fire load surveys form the basis of design codes and standards. These surveys indicate a large range of fire load density values; based on that, codes and standards can recommend a conservative value (e.g., 80% fractile or higher) having a small probability of reaching for fire design [5].

B. DEEP LEARNING-BASED INSTANCE SEGMENTATION

Instance segmentation is the task of detecting and delineating each distinct object of interest appearing in an image [15]. In the context of image recognition, object recognition or scene understanding has evolved from coarse-grained to fine-grained tasks: image classification, object detection/localization, semantic segmentation, and instance segmentation [16], as illustrated in Figure 1. Compared to object detection and semantic segmentation, instance segmentation is more challenging because it requires the correct detection of all objects in an image while also precisely segmenting each instance. It therefore combines elements from computer vision tasks used in object detection (which classifies individual objects and localizes each using a bounding box) and semantic segmentation (which classifies each pixel into a fixed set of categories without differentiating object instances) [8].

Mask R-CNN, proposed by He et al. [8], is the most accepted state-of-the-art model for instance segmentation. Mask R-CNN achieves high-quality instance segmentation by combining object detection and semantic segmentation: it uses object detection to localize and classify each object in an image and uses semantic segmentation to predict a binary mask inside each detected bounding box to identify the instance. Specifically, Mask R-CNN extends Faster R-CNN [17] by adding a mask branch for predicting segmentation masks on each region-of-interest (RoI), in parallel with the existing branch for classification and bounding box regression. However, unlike Faster R-CNN which uses RoIPooling, Mask R-CNN uses RoIAlign, which provides a more reliable way of matching bounding boxes in a feature map to the input image by using a bilinear interpolation algorithm. The mask branch is a small fully convolutional network (FCN) [18] applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner. The loss of the mask (L_{mask}), classification (L_{cls}), and bounding box (L_{box}) are calculated by average binary cross-entropy loss, classic cross-entropy, and smooth L_1 loss, respectively. Thus, the total loss function of Mask R-CNN is defined as $L = L_{cls} + L_{box} + L_{mask}$.

Figure 2 shows the architecture of Mask R-CNN. In object detection, a general deep learning algorithm first produces a large number of candidate boxes, and then revises these boxes by applying classifiers and regressors. This process, called two-stage detection, has achieved top-tier performance (in terms of accuracy) with respect to several benchmarks [19]. The representative approach in object detection is the region-based convolutional neural network (R-CNN) [20], which has adopted a convolutional neural network (CNN) to extract image features and a support vector machine (SVM) to classify candidate boxes. This two-stage detection method has been improved by other techniques, such as the RoI pooling of Fast R-CNN [21], region proposal network (RPN) of Faster R-CNN [17], and feature pyramid network (FPN) [22], for better feature extractions in the backbone. The RPN can greatly reduce the computational complexity of generating bounding box candidates, compared with its predecessor (selective search) [17]. The first step in the RPN involves generating anchor boxes of different sizes and length-width ratios using a sliding window [23]. Each sample is then mapped onto a probability value and four coordinate values, which represent the probability that an anchor box includes a target object. Finally, the loss of binary classification and coordinate regression is unified during target training of the RPN network. The FCN is a pioneering algorithm in semantic segmentation that still serves as a blueprint for most modern approaches [24]. The basic idea of the FCN is designing a network as a bunch of convolutional layers, with differentiable down-sampling (convolution) and up-sampling (transpose convolution) inside the network. Thus, the "funnel-like" FCN can make predictions for pixels all at once. Based on this, the accuracy of semantic segmentation has been further improved (though sometimes with decreased efficiency) by introducing skip connections between different layers, such as in U-Net [25], and by using atrous (dilated) convolution and atrous spatial pyramid pooling (ASPP), such as in DeepLab [26].



a) Image classification
b) Object detection
c) Semantic segmentation
d) Instance segmentation

FIGURE 1. Object recognition or scene understanding from coarse-grained to fine-grained: a) classification, b) detection/localization, c) semantic segmentation, and d) instance segmentation (adapted from [16]).

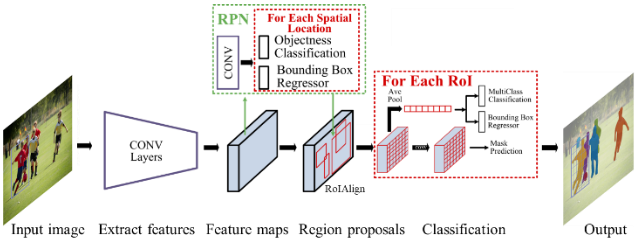


FIGURE 2. Architecture of Mask R-CNN for instance segmentation (adapted from [8] and [27]).

III. METHODOLOGY

The objective of this research is to propose an automated indoor fire load recognition method based on image recognition. As such, a deep learning-based instance segmentation approach is adopted because instance segmentation combines the advantages of both semantic segmentation and object detection: 1) detecting instances in an image in a pixelwise manner, and 2) differentiating all instances of a class.

Figure 3 provides an overview of the research methodology. In our summary of related work, we first reviewed studies about fire load estimation and conclude that most of them are calculated based on field surveys, which are time- and cost-prohibitive. Then, a brief illustration of deep learning-based instance segmentation is provided. To achieve the research objective, the following steps are conducted:

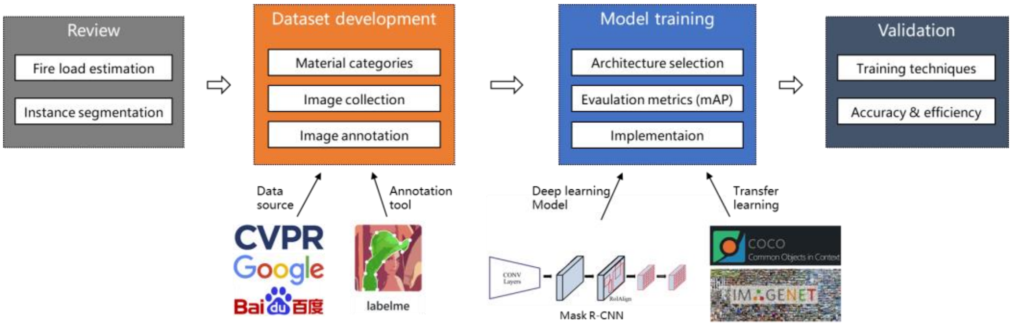


FIGURE 3. Overview of the research methodology.

1. *Develop a dataset for instance segmentation of indoor fire loads.* First, several common material categories are selected for the study, representing different types of fire loads. Next, images are collected from existing studies and online datasets. Finally, after proposing image annotation criteria, the collected images are annotated by ground truth of instance segmentation, which forms the dataset for model training and evaluation.
2. *Train a deep learning model for instance segmentation.* First, the Mask R-CNN is adopted and various CNNs are selected to build its backbone for comparison. Evaluation metrics are then defined for model training and validation. Finally, the transfer learning technique is utilized to improve the performance, and the training of deep learning model is implemented.
3. *Validate the proposed method.* Various training techniques and strategies are experimented and discussed. The accuracy and efficiency of the method are demonstrated by comparing the performance of different models and with humans. The advantages and limitations of the method are discussed and analyzed.

IV. DATASET DEVELOPMENT

A dataset containing images of indoor scenes and annotations of instance segmentation is developed in this research. Indoor scenes typically contain a variety of materials with complicated layout relationships. As such, four material categories are identified as labels to represent each indoor object based on its primary composition. Selected categories included: 1) fabric, 2) wood, 3) plastic, and 4) glass (see Table 1). These materials are commonly used and often the primary components of indoor scenes. They also exhibit varying degrees of flammability, as three are combustible (flammable) and one is noncombustible (nonflammable). The noncombustible material, glass, is chosen for comparison and used to test the robustness of the proposed method.

A. IMAGE COLLECTION

Indoor images are collected from two sources: an existing dataset acquired from the MIT-67 Indoor Scene Recognition Database [28] (http://web.mit.edu/torralba/www/indoor.html) and online images identified using the Google and Baidu search engines. Only images containing at least one combustible object and exhibiting proper resolution are selected. The dataset is also required to be of sufficient size, offering balanced sample types, broad diversity, and multiple categories.

Finally, 1015 images are selected to form the dataset, distributed across five classes: bedroom, dining room, hospital, living room, and office. Figure 4 shows some sample images from the final dataset.

TABLE I
FOUR SELECTED MATERIAL CATEGORIES FOR IMAGE RECOGNITION

Material category	Example	Flammability
Fabric	Fabric, also known as textile, is the main component of clothes, bedclothes, pillows, curtains, etc.	Highly flammable
Wood	Wood is widely used in furniture such as chairs, desks, cabinets, bookcases, and wardrobes.	Flammable
Plastic	Plastic is widely used in facilities and equipment, such as bottles, chairs, computers, printers, and air conditioners	Flammable
Glass	Glass is often transparent and can be the main composition of windows, screens, desktops, frosted glass partitions, etc.	Non-flammable

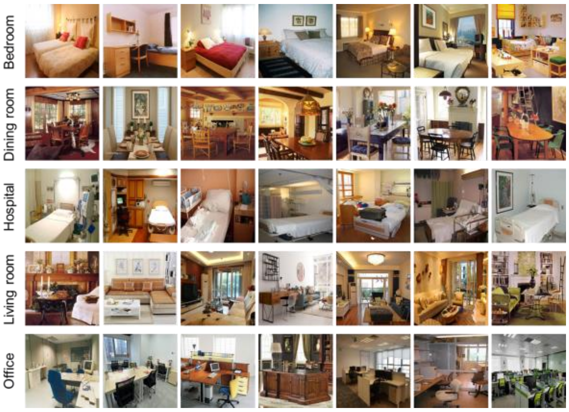


FIGURE 4. Example images in the built indoor scene dataset.

B. IMAGE ANNOTATION

LabelMe [29], a graphical image polygonal annotation tool in the Python environment, is chosen for image annotation.

The following criteria are applied to the annotation process to ensure quality and consistency:

1. *Objects are inside the room*. Only objects inside the room should be annotated, whereas the wall, floor, and ceiling are not considered.
2. *Sufficient size*. Each annotated object in an image should be large enough in both dimensions; otherwise, it will be discarded (e.g., for being too thin or too small). The minimum size is 1/50 of the image width. This is a soft regulation and is manually estimated.
3. *Explicit boundaries*. Each annotated object should exhibit an explicit boundary. Implicit borders or edges obscured by shadows are discarded.
4. *Definitive classification*. Each annotated object should belong to one predefined material category; otherwise, it will be ignored.
5. *Non-overlapping objects*. Objects exhibiting large overlap (i.e., major parts of some objects are obscured), will be annotated as a single object rather than multiple objects.

Each object (or visible part of an object) satisfying the criteria above is annotated by a polygon to denote its boundary. Annotation results are saved in a JSON file in accordance with the Microsoft COCO format [30]. Finally, each annotated image is cross-validated by two of the authors to achieve a consensus. Figure 5 shows some examples of annotated images, where each labeled instance is indicated by a colored mask and a bounding box with its category name in the top-left corner.

C. DATASET STATISTICS

After image annotation, the dataset of 1015 images included 8858 annotated instances. Table 2 shows the number of instances for each category in the dataset. The fabric and wood categories exhibited ~3000 instances, while glass and plastic categories are identified 1000–2000 instances. Figure 6 shows the number of annotated instances per image in our dataset, MS COCO [30], and PASCAL VOC [31]. The median number of instances per image of our dataset is 8, which is twice that of COCO's and eight times that of VOC's. On average, our dataset has 8.7 instances per image, whereas COCO and VOC have 7.3 and 2.3 instances per image, respectively.

This statistical result shows that our dataset has a more balanced distribution of instances per image and thus has higher diversity. This may be because our dataset is composed of indoor images while the latter two datasets mainly contain outdoor images. Indoor scene images are more complicated than outdoor images, due to their complex backgrounds, complex indoor decorations, heavy occlusion, and different viewpoints, scales, and textures changes across scenery and cluttered environments [32]. However, the distribution of the categories in our dataset is unbalanced, which may be because fabric and wood objects are more commonly found in indoor scene. This unbalanced distribution will bring difficulties to the training and optimization of deep learning models.

A repository containing the dataset is established on GitHub and can be found at https://github.com/Zhou-Yucheng/fire-load-detection. This repository also contains documents and codes used in the model training and evaluation steps below.

TABLE II
NUMBER OF ANNOTATED INSTANCES OF EACH CATEGORY IN THE DATASET

Category	Number of instances
Fabric	3176
Wood	2823
Plastic	1077
Glass	1782
Total	8858

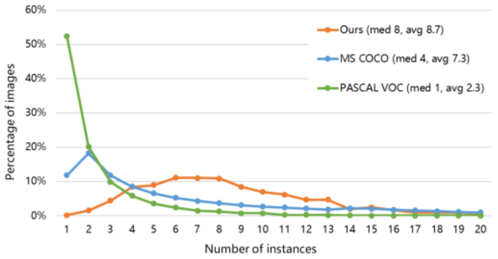


FIGURE 6. Number of annotated instances per image for MS COCO, PASCAL VOC, and our dataset (the median and average number of instances are shown in parentheses).

V. MODEL TRAINING METHODS

Mask R-CNN, a state-of-the-art deep learning model for instance segmentation, is utilized in this study. The Mask R-CNN includes a CNN backbone for feature extraction and a head net (the network after RoIAlign) for bounding box classification and regression and mask segmentation. Three CNN backbones are selected as a basis for comparing the accuracy and time efficiency (denoted by network-depth-feature), forming three Mask R-CNNs: ResNet-50-FPN, ResNet-101-FPN, and ResNeXt-152-FPN. The selected model architecture is shown in Figure 7. ResNet and ResNeXt are state-of-the-art CNN models and widely used backbones in Mask R-CNN [8], where ResNeXt has some advantages over ResNet by repeating a building block that aggregates a set of transformations with the same topology [33]. Generally, a deeper model is expected to achieve higher accuracy than a shallow model, while requiring more computational time. An FPN [22] is added to all backbone models as it can extract more feature maps from different layers and thus improve instance segmentation.

The transfer learning technique is utilized to reduce training difficulty for our relatively small dataset, as well as improving performance. Transfer learning is a promising method that can reduce dependence on a large number of target domain data, by transferring knowledge contained in different but related source domains [34]. The deep learning model will first be pretrained on a large general dataset COCO, and then trained and tested on our dataset.

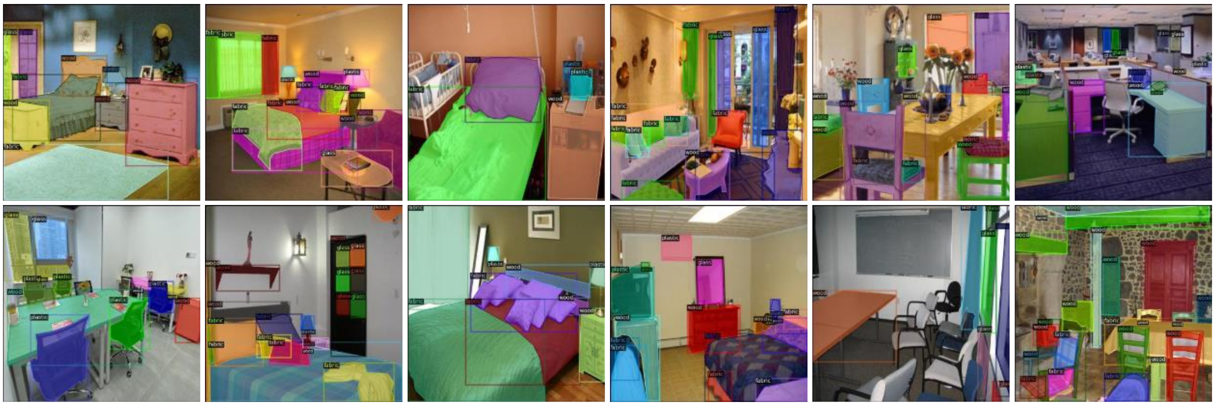


FIGURE 5. Examples of annotated images.

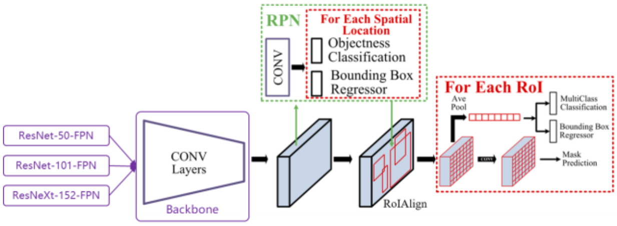


FIGURE 7. The selected model architecture Mask R-CNN with different backbones.

A. EVALUATION METRICS

Mean average precision (mAP, or AP for brevity) is the evaluation metric used to describe the accuracy of instance segmentation and is calculated using a precision-recall curve [35]. Equations 1–4 describe the calculation of mAP [19]. TP (true positive) is the number of correctly predicted masks. The correct mask is determined by the intersection over union (IoU) metric [36]. If the IoU of a ground truth mask and a predicted mask is greater than an IoU threshold (e.g., 0.5), the predicted mask is considered correct. FP (false positive) is the number of incorrectly predicted masks. FN (false negative) is the number of masks that have been incorrectly ignored (not predicted). The 11 in (3) represents 11 used recall values from 0, 0.1, 0.2 to 1. N is the number of predefined classes/categories. By adjusting the IoU threshold value, different metrics can be obtained.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,0.2,\ldots,1\}} Precision(r) \tag{3}$$

$$mAP = \frac{1}{N} \sum AP \tag{4}$$

In accordance with the detection evaluation metrics used by COCO [37], this research uses AP to represent the averaged mAP at the IoU threshold=0.5, 0.55, 0.6, …, 0.95, AP_{50} to represent the mAP at IoU threshold=0.5, and AP_{75} to represent the mAP at IoU threshold=0.75. For denoting AP of a specific class, AP_{class} is used.

B. IMPLEMENTATION

Implementation is conducted in a Python environment using the Pytorch deep learning package, performed on a single NVIDIA RTX 3090 GPU. The Detectron2 API [38] is utilized for model training and validation. Detectron2 provides several Mask R-CNN models that have been pretrained on the COCO dataset. Thus, model training originated from the COCO-pretrained Mask R-CNN models using the three backbones. Images in our dataset are randomly split into training (80%) and validation (20%) sets using a 0.8:0.2 ratio. Models are trained on the training dataset and tested on the validation dataset.

In the training process, images are resized to 512 × 512 pixels for input. The optimizer of the three models is a stochastic gradient descent (SGD) with a momentum of 0.9 and a weight decay factor of 0.0001 used for L2 regularization. SGD is chosen over adaptive optimization methods (e.g., AdaGrad, RMSProp, or Adam) as it is more likely to achieve a higher test accuracy, converge to a flat minimum, and thus provide higher generalizability [39]. Data augmentation is utilized during training, including horizontal flipping and random cropping. Training is performed over 6000 iterations with a batch size of 1 and an initial learning rate of 0.001, which is decreased by 10 at the 4000 iterations. These hyperparameters, which significantly affect model performance, will be evaluated and discussed in the next section.

VI. EXPERIMENTS AND RESULTS

A. EXPERIMENTS: EVALUATION OF HYPERPARAMETERS

Typical training strategies for a deep learning model first require an appropriate learning rate, which is decreased (e.g., by 10) when the model's validation accuracy stops increasing after several training iterations [40]. A large batch size is often used to make full use of GPU memory [8]. For example, training of the Mask R-CNN with the COCO dataset involved a batch size of 8, 270k training iterations, and a learning rate of 0.02 that was decreased by 10 after 210k iteration [38]. When training the Mask R-CNN model with our dataset, the model exhibits sensitivities to hyperparameters of the learning rate, learning rate decrease step, and batch size, as some widely adopted practices do not yield better results. Thus, an experiment is conducted to evaluate these hyperparameters and optimize model performance.

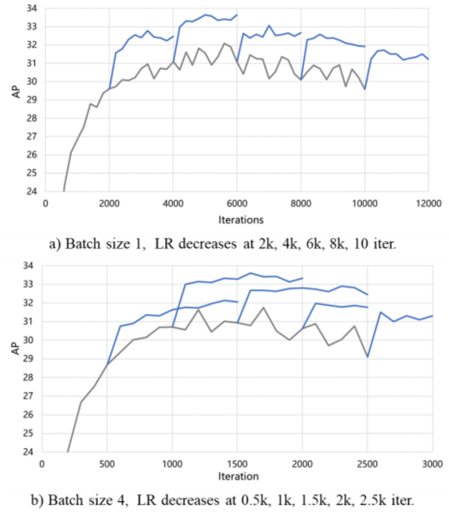
First, the (initial) learning rate is tested under various conditions, which shows that the optimal learning rates for batch sizes of 1, 2, 4, and 8 are 0.001, 0.002, 0.004, and 0.008, respectively. This result is reasonable according to [41], where comprehensive experiments demonstrated that the learning rate should be linearly scaled with batch size. Next, learning rate decrease strategies are tested for different batch sizes using optimal initial learning rates. Figure 8 shows validation AP curves for training the ResNet-50-FPN backbone Mask R-CNN (denoted by R50) using batch sizes of 1 and 4, with varying learning rate decrease steps (decreased by 10). Corresponding max AP values are summarized in Table 3. It is evident the best learning rate decrease steps for batch sizes of 1 and 4 are 4000 and 1000 iterations, respectively, wherein the two highest APs (33.7% and 33.6%) are achieved. In contrast, decreasing the learning rate too early or too late results in a lower AP. After 6000 or 1500 training iterations (for batch sizes of 1 and 4), the model encounters overfitting problems that cause the AP to decrease. This condition worsened for training beyond 10000 or 2500 iterations, as the AP will always be less than the previously achieved maximum value, even after a learning rate decrease. Choosing an appropriate learning rate decrease step produce AP improvements of 5.0% (32.1% to 33.7%) and 5.7% (31.8% to 33.6%).

Compared to the conventional training strategy discussed above, training a model on a small dataset is more likely to result in overfitting problems, as the optimal learning rate decrease step also occurs earlier. For example, training of the Mask R-CNN on the COCO dataset proceeds for 210k iterations before the learning rate decreased, representing approximately 29 epochs (one epoch means all images in the training set are processed). However, when training with our dataset, the best learning rate decrease step occurred at 4000 iterations or 5 epochs (since the training set has about 800 images), which is much lower than 29 epochs. In addition, as shown in Figure 8, the maximum AP without a learning rate decrease is achieved near 6000 or 1500 iterations, but decreasing the learning rate at 4000 or 1000 iterations is a better choice.

This experiment is also conducted for different batch sizes (2 and 8) and different model backbones (ResNet-101-FPN and ResNeXt-152-FPN), producing the same conclusion. The validation AP achieves a maximum when decreasing the learning rate after training 4000 images (i.e., 2000 or 500 iterations for batch sizes of 2 or 8) and starts to decline after training 6000 images. After training 10000 images, the AP will always be less than the previously achieved maximum value, even after a learning rate decrease. Hence, this experiment suggests an inversely proportional scaling rule for the learning rate decrease step and batch size: when the batch size is multiplied by *k*, divide the learning rate decrease step (iteration) by *k*.

In addition, we find that using a smaller batch size produces slightly better performance compared with larger batch sizes. For instance, if we scale the x-axis of Figure 8b by a factor of 4 (or change the x-axis to trained images), the AP values will become lower than those in Figure 8a at the same position. Our analysis suggests this phenomenon is related to the characteristics of the dataset. Generally, a larger batch size reduces the fluctuation in optimizing training loss and speeds up training by improving the utilization of GPU; meanwhile, a smaller batch size introduces noise in the training, which helps the model exit from a sharp minimum and obtain a higher generalization ability [42]. The dataset used in this research includes a relatively small number of annotated images compared to existing datasets. Therefore, the optimal batch size should be lower to avoid overfitting and achieve higher generalizability. Furthermore, the dataset includes more instances in each image (as indicated in Figure 6), so a small batch size is sufficient for training the model.

In summary, during model training, using a batch size of 1, a learning rate of 0.001, and decreasing the learning rate by 10 at 4000 iterations enables the model to achieve the best performance. These hyperparameters are also optimal for all three models.



a) Batch size 1, LR decreases at 2k, 4k, 6k, 8k, 10 iter.

b) Batch size 4, LR decreases at 0.5k, 1k, 1.5k, 2k, 2.5k iter.

FIGURE 8. Validation AP (%) curves of the R50 model using different batch sizes and learning rate (LR) decrease steps (LR is decreased by 10; the optimal initial LR for each batch size is used; the grey/blue curve denotes APs before/after LR decrease).

TABLE III
VALIDATION AP (%) OF THE R50 MODEL USING DIFFERENT BATCH SIZES AND LEARNING RATE DECREASE STEPS

Batch size	Step iteration	AP [a]
1	2000	32.8
	4000	33.7
	6000	33.1
	8000	32.6
	10000	32.1 (31.7)
4	500	32.1
	1000	33.6
	1500	32.9
	2000	32.0
	2500	31.8 (31.5)

[a] Max AP after learning rate decrease is shown in parentheses if it is lower than the max AP obtained previously.

B. EXPERIMENTAL RESULTS

Table 4 shows the models' instance segmentation results, measured by AP and test times on the validation dataset. Test time augmentation (TTA) is utilized in the last entry, where the test is performed on multiple altered versions of the same image and predictions are then aggregated for higher accuracy. Validation AP curves are shown for the training process of all three models in Figure 9.

As seen in Table 4, the best model (ResNeXt-152-FPN) achieves very promising results: 40.5% AP, 59.2% AP_{50}, and 44.2% AP_{75} on the validation dataset. Without TTA, the best model also achieves a high AP of 38.8% and the test time is 0.16 s/image, which means it could be used in real-time scenarios. The other two smaller models, ResNet-50-FPN and ResNet-101-FPN, achieve lower APs of 33.7% and 34.7%, respectively. However, these models require much lower computational time as they can be run at very high speeds of 24 to 30 frames per second (fps), which may make them more suitable for small or mobile devices. The APs of different categories are also shown in Table 4. Fabric and glass always have higher APs (about 8% absolutely) than wood or plastic, which indicates varied difficulties in detecting different kinds of instances. The research suggests two reasons. 1) The shape variability of categories. Instances of wood and plastic are much more likely to have variable and flexible shapes and geometries, which increases the difficulty to precisely detect them. 2) The unbalanced number of instances of categories. A category having more instances in the dataset provides more training examples for deep learning models and can result in better accuracy. Thus, to address this issue, images containing more instances of wood and plastic can be included in the dataset, and some advanced training techniques can be utilized such as under-sampling and over-sampling [43] to alleviate the unbalanced category distribution and improve the performance.

Figure 10 shows examples of model prediction output (listed in the same order as Figure 5), wherein each predicted instance is indicated by a colored mask and a bounding box with its category name and confidence score in the top-left corner. Note that only masks with a confidence score of no less than 0.5 are shown and all images are in the validation dataset. Figure 10a–g exhibit good results as they are highly similar to the annotations in Figure 5. In contrast, Figure 10h–l represent non-ideal examples. In Figure 10h, several separated instances are predicted to be a single instance by the model, such as pillows on the bed and glass in the door. This may be a result of edges along adjacent instances being blurred or unclear. Figure 10i includes several overlapping regions among instances (i.e., pillows on the bed). These overlapped instances are annotated as a single instance but are separated in the model prediction. This may be due to a non-overlapping bias in the dataset (i.e., non-overlapping instances are dominant in the dataset), as the model is likely to separate as many instances as possible. In Figure 10j, the mirror is not recognized by the model as glass. This is a difficult case because the mirror reflects an image. In Figure 10k, the chairs and blackboard are not annotated because their materials are not definite and the model predicts them to be fabric. We suggest this may be because the model is likely to recognize and classify an object based on its color, as the dataset may not be large enough to fully train the model. Figure 10l is a complex indoor scene example, where wood instances in the center-left and bottom-right are not recognized and a fabric instance (tablecloth) is segmented by the rails of a chair's back. The unrecognized instances are thin or obscure, which increases the difficulty of detection. The model also lacks the ability to understand context and semantics and thus separates parts of an object as independent instances.

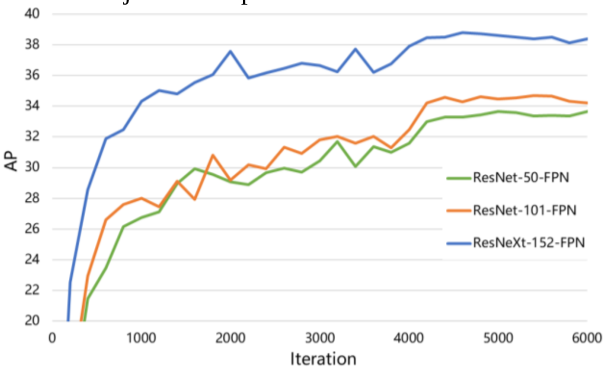


FIGURE 9. Validation AP (%) curves of the three models.

TABLE IV
RESULTS OF MODEL SEGMENTATION AP (%) AND TEST TIME (S/IMAGE) ON THE VALIDATION DATASET

Model backbone	AP	AP_{50}	AP_{75}	AP_{fabric}	AP_{wood}	$AP_{plastic}$	AP_{glass}	Time
ResNet-50-FPN	33.7	52.8	35.6	37.7	28.4	29.9	38.7	0.033
ResNet-101-FPN	34.7	54.0	36.7	39.6	29.3	31.2	38.6	0.041
ResNeXt-152-FPN	38.8	57.7	41.8	42.5	35.0	34.4	43.4	0.16
ResNeXt-152-FPN (TTA)	40.5	59.2	44.2	44.9	37.2	35.2	44.8	5.43

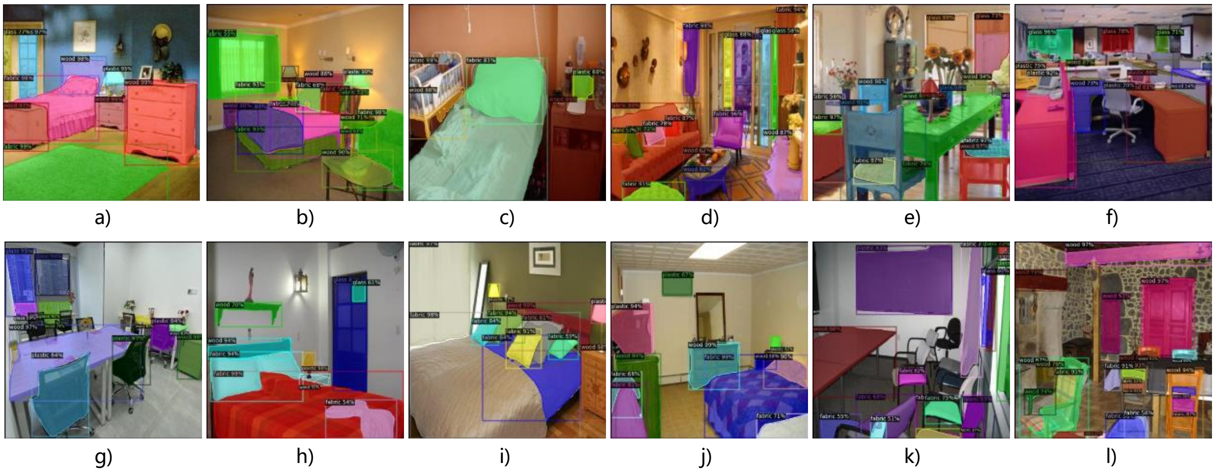


FIGURE 10. **Examples of the Mask R-CNN model prediction output (the annotations of these images are referred to Figure 5 in the same order).**

C. COMPARISON RESULTS WITH HUMANS

To better illustrate and validate the performance of the proposed method, an experiment is conducted to compare the method with human performance. In the experiment, a random sample of 100 images is selected from the dataset and further divided into two groups where each has 50 images. Then, the two sets of images are annotated by two of the authors individually, and the annotation time for each image is recorded. Finally, the annotation result is checked with existing annotated images which exhibits consistency, and the statistics of the human annotation performance is summarized and shown in Table 5.

As shown in Table 5, the averaged annotation time per image and instance of the two annotators are 144.6 s and 21.4 s, respectively. In contrast, the ResNeXt-152-FPN model achieves a detection time of 0.16 s/image (Table 4), which means 0.018 s/instance since the dataset has 8.7 instances per image on average. This result indicates that the proposed method can detect indoor fire load 1200 times faster than manual annotation, which demonstrates the high efficiency of the method and suggests it can be used in real-time detection scenarios or massive image-based indoor fire load detection to support fire load estimation and risk assessment.

TABLE V
RESULTS OF HUMAN ANNOTATION PERFORMANCE

Annotator	#1	#2	Avg.
Total annotation time (min)	110	131	121
Mean annotation time per image (s)	132.6	156.6	144.6
SD of annotation time per image (s)	77.4	94.2	85.8
Mean annotation time per instance (s)	20.5	22.3	21.4
SD of annotation time per instance (s)	12.5	10.6	11.6

VII. DISCUSSION

The proposed Mask R-CNN model achieves a 40.5% AP for instance segmentation on the indoor scene dataset. This is a very promising result because indoor scene recognition is a more changeling task than other recognition scenarios due to the complex background, layout, and cluttered environments [32]. As a comparison, the original Mask R-CNN achieved a 37.1% AP on the COCO test-dev dataset [8]; and recently, the COCO challenge 2020 winner achieved a 52.5% AP on the COCO test dataset using an architecture enhanced by the Mask R-CNN and additional training data [44]. In the AEC community, an improved Mask R-CNN achieved a 29.12% AP_{75} for ground penetrating radar signature detection [12].

Our experiments show the hyperparameters of a deep learning model can significantly affect its performance and typical training strategies are not suitable for training with a small dataset. These results suggest the following.

1. Training with a small dataset makes a model more likely to encounter overfitting problems that cause the validation AP to decrease. Hence, the total number of

training iterations/epochs and learning rate decrease steps should be lower.

2. There is an inversely proportional scaling rule for optimal learning rate decrease step and batch size: when the batch size is multiplied by *k*, divide the learning rate decrease step (iteration) by *k*. When training on our dataset, the optimal learning rate decrease steps are 4000 and 1000 iterations for batch sizes of 1 and 4, which lead to relative AP improvements of 5.0% and 5.7%, respectively.
3. Using a small batch size yields slightly better performance than larger batch sizes. The reason for this is related to the dataset: 1) a smaller batch size introduces more noise during training, which can help prevent overfitting [42], and 2) each image in the dataset has many instances on average, which makes a smaller batch size sufficient for training.

The proposed method provides an automated algorithm for indoor fire load recognition, which could establish a foundation for further fire load estimation. Once all combustible objects in an indoor scene are recognized and segmented, their sizes, quantity, and distributions can be utilized to estimate fire load, which is more accurate than the load obtained solely from room usage or building type. Furthermore, if the geometry of recognized instances can be measured, the fire load can be estimated more precisely.

The proposed method can be run at a high speed of 0.16 s/image with an AP of 38.8%, and the comparison with manual recognition also demonstrates the method's high efficiency as it can detect fire load 1200 times faster than humans. Therefore, this method can be applied to real-time detection/monitoring for fire prevention. For example, it could be run on a device to automatically and continuously monitor combustion sources in a building. Cameras could be installed for auto-marking of combustible objects, source tracking, and calculating fire loads in real-time. This method can also be used to digitize combustible and fire-related building information for easy visualization.

Error analysis of the results suggests that the following conditions will bring difficulties in recognition and may decrease the model's accuracy. 1) An unbalanced data distribution: instances or scenes occurring less in the dataset are likely to have poorer accuracies. 2) Shape variability of a category: instances of a category having variable and flexible shapes are more difficult to recognize. 3) Blurred or unclear images, such as those with blurred edges of adjacent instances. 4) High-level semantics, such as a mirror reflecting an image. 5) Complicated spatial distributions or layouts, such as the scene in Figure 10l. To address these issues, the size of the dataset should be enlarged to include more annotated images, especially for categories having fewer instances to balance the distribution; and the quality of images and annotations (e.g., image resolution and annotation consistency) should be improved to make the recognition target clearer and the model training easier.

Two primary limitations are identified in the proposed method which need to be improved in future research. First, the method only recognizes fire loads but not provides estimation. To address this, the geometry of instances should be measured. For example, the algorithm could be combined with building information modelling (BIM) technology [45] to calculate instance geometries, and multi-view images or point cloud techniques for 3D scene understanding [46] could be leveraged to measure instance geometries. Second, the classification of material should be enriched and refined. Currently, indoor instances are classified into only four categories according to their material. More material categories should also be considered to acquire a more accurate estimation of fire load.

The contribution of this research can be further enhanced in a couple of ways. First, the discussion of model hyperparameters is not exhaustive; therefore, further tuning of hyperparameters and additional training techniques can be explored in future studies. Second, more model architectures and backbones can be experimented to obtain a higher AP or faster speed. Third, the dataset can be further enriched, in terms of quantity, variety, and annotations, and this is expected to enhance the model's performance.

VII. CONCLUSION AND FUTURE WORK

Efficient and dynamic estimation of fire load is important to ensure the safety and assess the resilience of built environments. As a promising technique to address this problem, computer vision can be utilized for the fire load estimation, where the first step is recognizing fire loads (combustible objects) and their types, i.e., fire load recognition and then consequent estimation algorithms can be applied. Thus, this research proposes a deep learning-based instance segmentation method for indoor fire load recognition. First, indoor scene images are collected and all instances in the images are annotated and labeled with a material category, forming a dataset. Second, three Mask R-CNN models are developed using a transfer learning technique and trained with the dataset for instance segmentation. Finally, results show the model can achieve validation AP, AP_{50}, and AP_{75} of 40.5%, 59.2%, and 44.2% on the dataset, respectively, which are very promising results. A comparison with manual detection further demonstrates the method's high efficiency as it can detect fire load 1200 times faster than humans.

This research contributes to the body of knowledge in three aspects. First, an accurate and efficient method for automated indoor fire load recognition is proposed based on image instance segmentation, which provides a foundation for further fire load estimation, real-time fire load monitoring, and risk assessment. Second, training techniques for a deep learning model in a relatively small dataset of indoor images, such as how to overcome the overfitting problem and optimization of hyperparameters, which can provide insights and guidelines for future related studies. Finally, an indoor image dataset with pixel-level annotations of fire load instances and its annotation criteria, which could be used as a baseline dataset to assist in the future development of indoor fire load recognition methods or datasets.

Future work remains to be done to enhance and expand the contribution of this research. First, the proposed method can be combined with BIM or 3D reconstruction technologies to measure the instance geometries and thus estimate fire loads. Second, the method can be integrated with simulation methods for performance-based building fire design and resilience assessment. Third, hyperparameters and training techniques for the deep learning model could be further optimized, as well as exploring more model architectures and backbones. Finally, the developed dataset could be further enriched in terms of material classification, quantity, variety, and available annotations.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (No. 72091512, No. 51908323).

REFERENCES

[1] N. N. Brushlinsky, M. Ahrens, S. V. Sokolov, and P. Wagner, “World fire statistics,” 13, 2006.

[2] Fire Department of Ministry of Public Security of China, *China Fire Services*. Yunnan Renmin Press, 2015.

[3] M. Liang, “Large Commercial Building Fire Regulations and Provisions Related to Research and Policy Analysis of Fire Protection Design,” South China University of Technology, 2013.

[4] Y. Yang and L. Cao, “Preparatory Study on Scenario Design for Subway Fire,” *Journal of Natural Disasters*, vol. 15, no. 4, 2006.

[5] N. E. Khorasani, M. Garlock, and P. Gardoni, “Fire Load: Survey Data, Recent Standards, and Probabilistic Models for Office Buildings,” *Engineering Structures*, vol. 58, pp. 152–165, 2014, doi: https://doi.org/10.1016/j.engstruct.2013.07.042.

[6] Q. Xie, J. Xiao, P. Gardoni, and K. Hu, “Probabilistic Analysis of Building Fire Severity Based on Fire Load Density Models,” *Fire Technol*, vol. 55, no. 4, pp. 1349–1375, Jul. 2019, doi: 10.1007/s10694-018-0716-0.

[7] M. J. Hurley *et al.*, *SFPE Handbook of Fire Protection Engineering*. Springer, 2015.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. doi: 10.1109/ICCV.2017.322.

[9] J. Seo, S. Han, S. Lee, and H. Kim, “Computer vision techniques for construction safety and health monitoring,” *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 239–251, Apr. 2015, doi: 10.1016/j.aei.2015.02.001.

[10] A. Dimitrov and M. Golparvar-Fard, “Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections,” *Advanced Engineering Informatics*, vol. 28, no. 1, pp. 37–49, 2014, doi: https://doi.org/10.1016/j.aei.2013.11.002.

[11] H. Deng, H. Hong, D. Luo, Y. Deng, and C. Su, “Automatic Indoor Construction Process Monitoring for Tiles Based on BIM and Computer Vision,” *Journal of Construction Engineering and Management*, vol. 146, no. 1, p. 04019095, Jan. 2020, doi: 10.1061/(ASCE)CO.1943-7862.0001744.

[12] F. Hou, W. Lei, S. Li, J. Xi, M. Xu, and J. Luo, “Improved Mask R-CNN with distance guided intersection over union for GPR signature detection and segmentation,” *Automation in Construction*, vol. 121, p. 103414, Jan. 2021, doi: 10.1016/j.autcon.2020.103414.

[13] S. Zhao, D. M. Zhang, and H. W. Huang, “Deep learning–based image instance segmentation for moisture marks of shield tunnel lining,” *Tunnelling and Underground Space Technology*, vol. 95, p. 103156, Jan. 2020, doi: 10.1016/j.tust.2019.103156.

[14] C. Thauvoye, B. Zhao, J. Klein, and M. Fontana, “Fire Load Survey and Statistical Analysis,” *Fire Safety Science*, vol. 9, pp. 991–1002, 2008, doi: 10.3801/IAFSS.FSS.9-991.

[15] B. Romera-Paredes and P. H. S. Torr, “Recurrent Instance Segmentation,” in *ECCV 2016*, Cham, 2016, pp. 312–329. doi: 10.1007/978-3-319-46466-4_19.

[16] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A Survey on Deep Learning Techniques for Image and Video Semantic Segmentation,” *Applied Soft Computing*, vol. 70, pp. 41–65, Sep. 2018, doi: 10.1016/j.asoc.2018.05.018.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[18] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. doi: 10.1109/CVPR.2015.7298965.

[19] B. Xiao and S.-C. Kang, “Development of an Image Data Set of Construction Machines for Deep Learning Object Detection,” *Journal of Computing in Civil Engineering*, vol. 35, no. 2, p. 05020005, 2021, doi: 10.1061/(ASCE)CP.1943-5487.0000945.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[21] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125. doi: 10.1109/CVPR.2017.106.

[23] F. Deng, H. Hu, S. Chen, Q. Guan, and Y. Zou, “Rich feature hierarchies for cell detecting under phase contrast microscopy images,” in *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*, Nov. 2015, pp. 348–353. doi: 10.1109/ICICIP.2015.7388195.

[24] N. Atif, M. Bhuyan, and S. Ahamed, “A Review on Semantic Segmentation from a Modern Perspective,” in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Nov. 2019, pp. 1–6. doi: 10.1109/UPCON47278.2019.8980189.

[25] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[26] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[27] L. Liu *et al.*, “Deep Learning for Generic Object Detection: A Survey,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020, doi: https://doi.org/10.1007/s11263-019-01247-4.

[28] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 413–420. doi: 10.1109/CVPR.2009.5206537.

[29] K. Wada, *Labelme: Image Polygonal Annotation with Python*. 2016. [Online]. Available: https://github.com/wkentaro/labelme

[30] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *ECCV 2014*, Cham, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.

[31] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[32] M. Afif, R. Ayachi, Y. Said, and M. Atri, "Deep Learning Based Application for Indoor Scene Recognition," *Neural Process Lett*, vol. 51, no. 3, pp. 2827–2837, Jun. 2020, doi: 10.1007/s11063-020-10231-w.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," Apr. 2017. doi: 10.1109/CVPR.2017.634.

[34] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.

[35] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," in *Proceedings of the 15th ACM international conference on Information and knowledge management*, 2006, pp. 102–111. doi: https://doi.org/10.1145/1183614.1183633.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," 2019, pp. 658–666. doi: 10.1109/CVPR.2019.00075.

[37] "Detection evaluation metrics used by COCO," 2020. https://cocodataset.org/#detection-eval (accessed Dec. 24, 2020).

[38] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. https://github.com/facebookresearch/detectron2

[39] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2017, pp. 4151–4161. doi: 10.5555/3294996.3295170.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016. doi: 10.1109/CVPR.2016.90.

[41] P. Goyal *et al.*, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," Apr. 2018. Accessed: Jan. 31, 2021. [Online]. Available: https://research.fb.com/wp-content/uploads/2017/06/imagenet1kin1h5.pdf

[42] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," presented at the ICLR 2017, Feb. 2017.

[43] N. Junsomboon and T. Phienthrakul, "Combining Over-Sampling and Under-Sampling Techniques for Imbalance Dataset," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, New York, NY, USA, Feb. 2017, pp. 243–247. doi: 10.1145/3055635.3056643.

[44] Z. Li, Y. Ma, Y. Chen, X. Zhang, and J. Sun, "Joint COCO and Mapillary Workshop at ICCV 2019: COCO Instance Segmentation Challenge Track," Oct. 2020. Accessed: Feb. 05, 2021. [Online]. Available: https://cocodataset.org/files/panoptic_2019_reports/Innovation_coco_report_latest_version.pdf

[45] J.-R. Lin and Y.-C. Zhou, "Semantic classification and hash code accelerated detection of design changes in BIM models," *Automation in Construction*, vol. 115, p. 103212, Jul. 2020, doi: 10.1016/j.autcon.2020.103212.

[46] M. Jaritz, J. Gu, and H. Su, "Multi-View PointNet for 3D Scene Understanding," 2019. doi: 10.1109/ICCVW.2019.00494.

YU-CHENG ZHOU was born in Guizhou, China. He received the B.S. degree from the Department of Civil Engineering, Tongji University, China, in 2015. He is currently an Undergraduate Student in the Department of Civil Engineering, Tsinghua University, China.

His research interests are in building information model (BIM) and information technologies in civil engineering.

ZHEN-ZHONG HU was born in Guangdong, China. He received the B.S. and Ph.D. degrees from the Department of Civil Engineering, Tsinghua University, China, in 2005 and 2009, respectively. He is currently an Associate Professor with the Department of Civil Engineering, Tsinghua University.

His research interests include information technology in civil engineering, building information model, and digital disaster prevention and mitigation.

KE-XIAO YAN received the Master degree from the School of Civil and Environment Engineering, University of New South Wales, Australia, in 2019. He is currently a Research Assistant with the Department of Civil Engineering, Tsinghua University.

His research interests are information technology for building and civil engineering, including building information model (BIM), deep learning, and data mining.

JIA-RUI LIN received the B.S. and Ph.D. degrees from the Department of Civil Engineering, Tsinghua University, China, in 2011 and 2016, respectively. He is currently a Research Assistant Professor with the Department of Civil Engineering, Tsinghua University.

His research interests are information technology for building and civil engineering, including building information model (BIM), augmented reality (AR), machine learning, and internet of things (IoT).