https://doi.org/10.1016/j.autcon.2020.103185

DETECTING SEMANTIC REGIONS OF CONSTRUCTION SITE IMAGES BY TRANSFER LEARNING AND SALIENCY COMPUTATION

Ling CHEN[1], Yuhong WANG[2], Ming-Fung Francis SIU[3*]

1. Postdoctoral Fellow, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

(email: ling.a.chen@connect.polyu.hk)

2. Associate Professor, Department of Civil and Environmental Engineering, Faculty of Construction and Environment, The Hong Kong Polytechnic University; Hung Hom, Kowloon, Hong Kong

(email: ceyhwang@polyu.edu.hk)

3. Assistant Professor (corresponding author), Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University. Hung Hom, Kowloon, Hong Kong*

(email: francis.siu@polyu.edu.hk; phone: +852-27665820; *fax:* +852-27645131*)*

Abstract: Effective use of massive construction site images and videos requires an efficient storage and retrieval method. However, significant portions of the image regions contain little useful information to project engineers and managers. To reduce resource waste in data storage and retrieval, we developed a new semantic region detection approach using transfer learning and modified saliency computation method without the need to specify targetted objects. In the new approach, the saliency matrix is generated using labelled bounding boxes, and the semantic regions are selected using a developed algorithm. The proposed method was applied to case studies based on two image datasets. The case studies suggest that the proposed method can efficiently detect semantic regions in site images and detect construction events from other image datasets without a modifying or re-training process. The research contributes to construction image analytics academically by advancing the context-based semantic region detection method and practically by facilitating the effective storage and processing of the massive site images and videos.

Keywords: semantic region detection; image/video retrieval; adaptive site image/video cropping; image saliency analysis

1. INTRODUCTION

With the widespread use of visual recording equipment, a large collection of construction images and videos are generated from monitoring jobsite activities. Those videos are usually deleted after a short period of time mainly due to the limit of storage capacity. The visual data, however, contain useful project information that can serve a variety of purposes, such as the post-project performance evaluation, forensic analysis, job training. The duration of a construction project is typically long (i.e., up to several years). To keep the visual information, it is critically important to store the massive data systematically and concisely for efficient retrieval. However, a close look at continuous construction images from a monitoring system can quickly reveal two facts: (1) dynamic activities that are of interest to project personnel are usually concentrated in certain regions of the images, and (2) some regions of the images are lack of changes and repeat themselves continuously across the recorded images. The relatively static and highly redundant regions, which can be treated as the background of the dynamic activities, consume a large amount of storage space. Information in the background regions may be sufficiently represented by a few keyframes [1], hence eliminating the necessity to store them continuously. Monitoring cameras in a construction site are typically installed at locations that can capture the overall view of the site. Consequently, only small regions in those site images capture active changes, i.e., high semantic information in connection with the projects. As such, given the existence of those patches with low semantic information, extracting only the regions with high semantic information will significantly increase the efficiency of processing site image and video datasets.

Semantic regions reveal information of important objects and events. Applications of semantic region detection such as object detection have been extensively used in computer vision. Construction applications are also reported for work entity and work defect detection [2,3], site image classification [4,5], and content-based site image retrieval [6-8]. In construction, semantic region detection methods are typically used to locate one particular type of site objects, such as labours, plants, materials, using learning-based or non-learning-based methods [9]. The site objects are normally detected based on their image features. The detection accuracy is dependent on how easy the object(s) can be identified through recognising those features. However, semantic region detection based on object detection has two disadvantages: (1) it is limited or impossible to apply the developed models for detecting other undefined objects, (2) the object detection approach is difficult to be generalised to other datasets. This is because the content captured by the site images (e.g., the site layouts and the event locations) is highly diversified from time to time throughout the project duration. To reuse the existing methods for other target objects or other datasets, significant time and effort are required either to modify the model when a non-learning-based method is used [10] or to re-train the model when a learning-method is used [9].

In this study, we developed a new semantic region detection approach based on a deep convolutional neural network (CNN). The deep CNN model was calibrated using transfer learning technique. In the training stage, the images and saliency matrices were used as input and output data of the network respectively. Saliency matrices were computed using the labelled ground-truth bounding boxes. In the detecting stage, semantic regions are extracted by a region selection algorithm based on the output saliency matrix. As the labelling process avoided specifying any target objects, the model can be also used for other image datasets without a modification or re-training process of the trained model. This proposed approach is proved to be effective for construction applications.

This paper is structured as follows: Section 2 reviews the existing literature of semantic region detection, Section 3 proposes the new semantic region detection method, Section 4 presents the case studies with the results of semantic region detection, Section 5 discusses the performance of the semantic region detection method and transfer learning technique, Section 6 concludes the work by discussing its contributions and limitations, Appendix A explains the concept of transfer learning technique. Appendix B, C, D describe the experiments and results of the selection of the pre-trained module, activation function and learning rate, respectively.

2. LITERATURE REVIEW

A review of the literature suggests that semantic region detection can be classified into object-based method and content-based method. Object-based semantic region detection is to locate the tight rectangle regions or mask regions which capture pre-determined target objects. On the other hand, content-based semantic region detection is to locate the regions containing important content derived from the image pixels without pre-determining any semantic information or any target objects. Notably, this classification is not new. [11] used the same methods (i.e., object-based vs. content-based) for classifying image retrieval applications. Since the aim of this research is to detect semantic regions, the literature was thus categorised into these two main streams, so as to shed light on the novelty of this research work.

2.1. Object-based semantic region detection and transfer learning

Object detection is one of the popular techniques in computer vision for construction image analysis. The existing object detection methods are classified into two categories: knowledge-based method and learning-based method.

Knowledge-based methods emphasise on the uses of the common features of target objects. For example, Hui, et al. [10] proposed an image processing framework based on colour, edge and shape features of a brick for detecting bricks in building façades. The limitation of knowledge-based methods is that these methods only apply to objects which have stable and distinct features in their appearance. For objects having dynamic and complex appearance, it is difficult to generalise any classification rules for segmenting them from construction site images. To overcome this limitation, learning-based methods become a viable solution.

Learning-based methods emphasise on the uses of training datasets for calibrating the object detection models. Objects can be detected by the trained model using images that depict the objects of the same type. The traditional learning procedure involves a feature selection process [9]. The features are extracted based on a training dataset. Then, the features are used for calibrating an object detection model. The commonly-used features are wavelet features [e.g., Haar-like feature [12]], Gabor feature [13,14], and statistical feature [e.g., histograms of oriented gradients (HOG) [12]]. Such methods were extensively used to detect site objects. For example, Memarzadeh, et al. [9] used HOG and colour features extracted from a dataset of 8,000 images to train a support vector machine (SVM) model for detecting construction equipment and workers. Cord and Chambon [15] used morphological descriptors extracted from a dataset of 6,875 images to train an AdaBoost defect detection model for detecting road defects. Cha, et al. [16] used Hough transform to extract the specified features based on a training set of 52 images to train a linear support vector machine (LSVM) model for detecting loosened bolts. The detection efficiency largely depends on the robustness and computational complexity of the selected features of the objects [17]. Thus, extra time and effort are required to modify the method and feature selection process if the target objects are changed. On the other hand, due to the advancement of computing power in recent years, deep learning methods are used for object detection without a feature selection process. Deep learning methods use a large training set for developing a relatively large network with more hidden layers and nodes, compared to the neural network models in machine learning (e.g., multilayer perceptron (MLP)). Former research shows that deep learning generally provides satisfactory results for construction applications. For example, Fang, et al. [18] used deep learning method for detecting non-hardhat-used instances based on a training dataset with 81,000 images. Fang, et al. [19] used deep learning method (Faster R-CNN model) to detect workers and heavy equipment based on a training set of 10,000 labelled objects. The limitation of training a deep neural network is that a large training dataset is often required as compared to the traditional learning methods.

Some researchers used transfer learning technique to reduce the size of the training dataset and improve learning performance. Transfer learning refers to developing and calibrating a neural network model in one task (e.g., Task A) and applying the model to a new task (Task B) [20]. Transfer learning technique often saves training data or improves learning performance in developing and calibrating the model in Task B. The detailed explanation of transfer learning is in Appendix A. In construction applications, Kolar, et al. [21] used deep learning method (CNN model) to detect guardrails. They deployed transfer learning technique to reduce the size of the training set (i.e., 4,000 images). Kim, et al. [3] used a similar method (RFCN model) to detect construction equipment using a training set with 2,920 images.

Inspite of the advantages of using transfer learning technique, object-based semantic region detection requires either decision rules or a ground-truth dataset for classifying different site objects. Significant time and effort will be spent to develop the decision rules and manually label the objects in the site images.

2.2. Content-based semantic region detection

Content-based semantic region detection identifies the interested regions by locating the most important content in images. Unlike the object-based methods, the semantic meaning of the content is not known. The importance of the content will be derived based on images themselves. The semantic regions are used for the applications of object detection, image classification, image summary and storage, and content-based image indexing and retrieval [22].

Saliency computation is often used for content-based methods [23]. In saliency computation, a saliency map highlights the salient pixels in an image by mimicking the perception of the human eye when viewing an image. For example, highly likely, a human being will capture the context with showy colours and high contrast. In 1998, Itti [24] proposed a classical framework to compute “saliency”. The method uses linear filters to extract color, intensity, and orientation features based on an input image. A saliency map, which has the same size as the original image, is then developed by linearly combining all the features extracted from the images. Many researchers have proposed modified methods, such as GBVS [25] method, to improve the computational efficiency of Itti’s method.

In saliency-based region detection methods, semantic regions are located by a search strategy based on the saliency map. For example, Chang, et al. [26] proposed a method to detect semantic objects by combining a saliency map with region proposals generated according to specified objectiveness. Less effort is therefore required to detect an object using content-based methods, since no target objects should be labelled in an image set. In addition, it is feasible to detect the site objects in one image set by a developed model that was developed based on another image set when using content-based methods [23].

In construction applications, the accuracy of detecting semantic regions using content-based methods based on saliency computation is lower than that using object-based methods. It is because the semantic regions detected by saliency-based methods may not be relevant to a construction project. In addition, the accuracy of detection results is dependent on the accuracy and clarity of the saliency map. Unfortunately, the accuracy and clarity of saliency maps generated from site images are typically low because the site's environment is extremely complex and dynamic.

2.3. Semantic region detection for construction site images

The contents of images captured on a construction site are very different throughout the project duration, and the contents of images captured from different sites are even more different. There are two reasons. First, different construction projects involve different work scopes and different resources such that the targeted objects may not be the same. Second, a construction site is so dynamic that the content of images may not share the same semantic information from time to time. For example, the concreters may work on concreting works at early stage and the electricians may work on electrical installation at later stage of a building project. Because of the diversity of the content captured by the site images, the object-based models calibrated for a set of images are difficult to be re-used to detect an object captured in other sets of images. Therefore, a content-based (i.e., saliency-based) method was hence proposed for detecting semantic regions of construction site images in this study.

On the other hand, the saliency computation was modified since the saliency maps of site images from traditional methods may not highlight the objects of interested to site managers. Recently, co-saliency computation approaches have been proposed by researchers [27]. These approaches compute co-saliency maps of a group of images. Each co-saliency map emphasises the frequently-co-existing objects in a set of images. Compared with the saliency map, the co-saliency map is computed based on a given image set. The co-saliency computational methods exclude foreground objects which are less frequently appeared in the given images. In construction, objects rarely existing on sites (such as vehicles of government officials, nearby traffic flows and buildings) are of less interest. Therefore, the idea of saliency of co-existing was borrowed to detect project-relevant semantic regions from construction site images in this study.

Inspired by the literature review, we proposed a novel learning-based method for detecting the project-relevant semantic regions from construction site images. The proposed method is based on a deep neural network which was established and calibrated using transfer learning technique. The coarse bounding boxes of project-relevant events which frequently appeared in site images are labelled as ground truth information. Saliency matrix which is computed based on the ground truth information is used as training data of the deep neural network. It is noteworthy that, based on the proposed approach, (1) the use of transfer learning technique and coarse bounding boxes will reduce effort in ground truth labelling and improve learning performance in model training; (2) the deep neural network which was tuned by the saliency matrix will exclude non-project-relevant regions (such as vehicles of government officials, nearby traffic flows and buildings) and highlight the project-relevant objects (such as labours, plants and materials).

3. METHODOLOGY

The flowchart of the proposed approach is shown in Figure 1. The proposed approach consists of three main stages: (1) data collection and labelling, (2) model development, and (3) model training and evaluation. During the data collection and labelling stage, four-level coarse bounding boxes are proposed to label construction events (i.e., activities) instead of the commonly used tight bounding boxes for construction objects. As a result, it is no longer required to establish a dataset for different object types. This will save much human effort in labelling the objects in images (The details are explained in Section 3.1). During the model development stage, saliency map computation was modified using the labelled bounding boxes. Then, the pre-trained module was selected, along with the learning rate, and activation function at the output layer according to the results of experiments (The details are explained in Section 3.2). During the model training and evaluation stage, the proposed model was trained using transfer learning technique and the detection results were evaluated using the proposed metrics (The details are explained in Section 3.3).

The idea of co-saliency is integrated into the proposed method (Figure 1), as indicated in the two steps highlighted in dotted lines: (Step 1) labelling event regions in image series which contain co-existing objects, and (Step 2) computing the saliency map of each image according to the labelled bounding boxes. Transfer learning is adopted, as indicated in the three steps highlighted in bolded lines: (Step 1) establishing a dataset, (Step 2) establishing a model with a pre-trained module, and (Step 3) initialising the pre-trained module with the trained parameters.

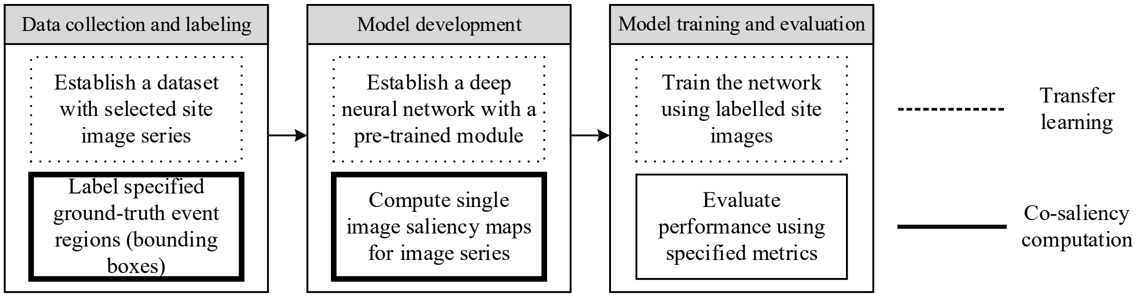

Data collection and labeling
Establish a dataset with selected site image series
Label specified ground-truth event regions (bounding boxes)
Model development
Establish a deep neural network with a pre-trained module
Compute single image saliency maps for image series
Model training and evaluation
Train the network using labelled site images
Evaluate performance using specified metrics
Transfer learning
Co-saliency computation

Figure 1. Overview of process of the proposed method

3.1. Establish training image dataset and label ground truth

Establishing the training dataset and labelling the ground truth are important steps for a learning process. The site images for model training can be selected from the site videos collected from different construction sites. The ground truth rectangle regions are manually labelled according to the intended semantic regions. These two steps are elaborated below.

(1) Image selection for forming the training dataset

Selecting suitable and relevant images for training a model is important to improve model accuracy. We suggest several guidelines to select the images. Firstly, site images for model training should be selected from site photo and video datasets of real construction projects. Secondly, the images and videos should be collected from several projects. Thirdly, training images should be extracted when different activities taking place in the field. Figure 2 shows some image samples selected to form the dataset.

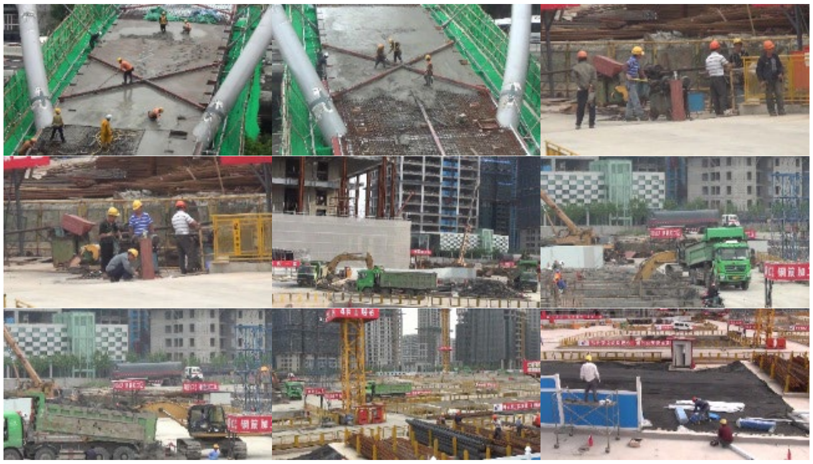


Figure 2. Image samples forming the dataset for model training

(2) Labelling the ground truth by defining the semantic regions in site images

In the proposed method, four-level coarse bounding boxes are used to label the event regions instead of using tight bounding boxes for specific targets/objects. As a result, it is not necessary to establish a dataset for each type of object.

Table 1 summarises the levels and definitions of semantic regions in site images. Level 1 semantic region indicates primary activity-active region. These regions capture the workers and machines that are working on site. Level 2 semantic region indicates secondary activity-active region. These regions capture the workers and machines that are currently not working. Level 3 semantic region indicates materials and tools which are relevant to the project. Level 4 semantic region indicates the main work components in an image such as a building or a foundation site.

Table 1. Summary of semantic regions manually labelled as ground truth information

Type	Name	Description
Level 1	Primary activity-active region	Coarse event regions capturing the activities of currently working workers and machines.
Level 2	Secondary activity-active region	Coarse Regions capturing the workers and machines that are onsite but not working currently.
Level 3	Materials and tools region	Regions of relevant materials and tools.
Level 4	Main component region	Regions of main work components.

The ground truth of semantic regions is labelled in each training image using bounding boxes. Figure 3 shows some examples of semantic regions being labelled (Blue: Level 1, Cyan: Level 2, Green: Level 3, Red: Level 4). Notably, some of the regions highlighted in our images are the activities “the workers using the equipment” and “construction products were built”. The reason that most event regions are workers is that most construction activities are labour-intensive. It is emphasised that the event regions are labelled with bounding boxes and not workers. The proposed method will save effort in labelling ground truth information for the learning process.

As shown in the starting image of the third row in Figure 3, only one event region is labelled using the proposed method. In contrast, the classic object detection method requires the labelling of all four workers (i.e., labelling 4 workers separately). This simple example indicates that the proposed method will save much of the total effort and time. In our study, the authors labelled events in 1000 images in less than 1 hour.

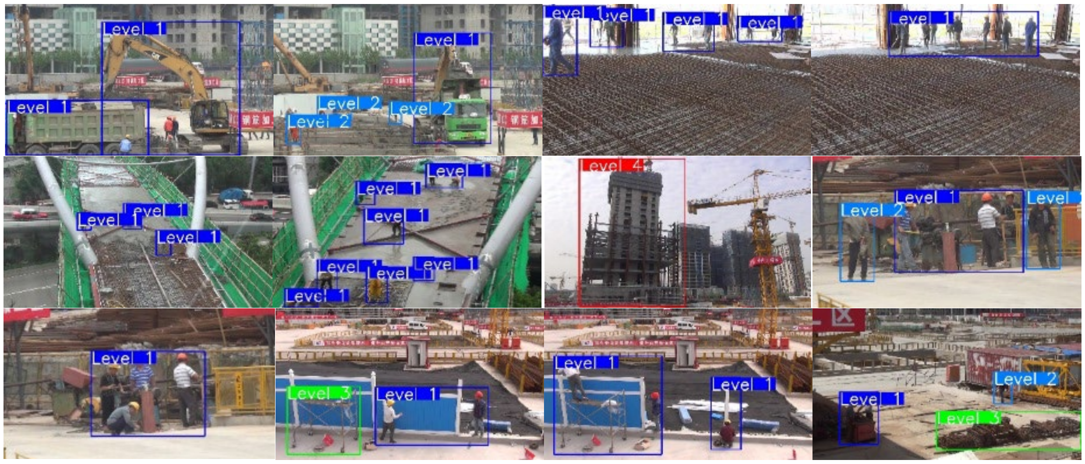


Figure 3. Examples of labelling semantic regions

3.2. Establish and train a deep neural network for semantic region detection

(1) Process for establishing the deep neural network

The proposed process for establishing a deep neural network to detect the semantic regions are shown in Figure 4. The input of the network is two-dimensional images, which is resized to a predefined size as the inputs of the pre-trained module. The pre-trained module contains the *convolutional layers* from a trained state-of-the-art network. The output of the pre-trained module is flattened and then passed to *fully connected layers* which use a rectified linear unit (ReLU) as the activation function. The differences between the convolutional layers and fully connected layers are given in Table 2. The output of the fully connected layers is transformed by an activation function f(x). The final output is a two-dimensional matrix (i.e., saliency matrix) in which each element indicates the saliency value corresponded to the image grid. In the proposed process for establishing a deep neural network, the pre-trained module and f(x) are chosen according to the training process.

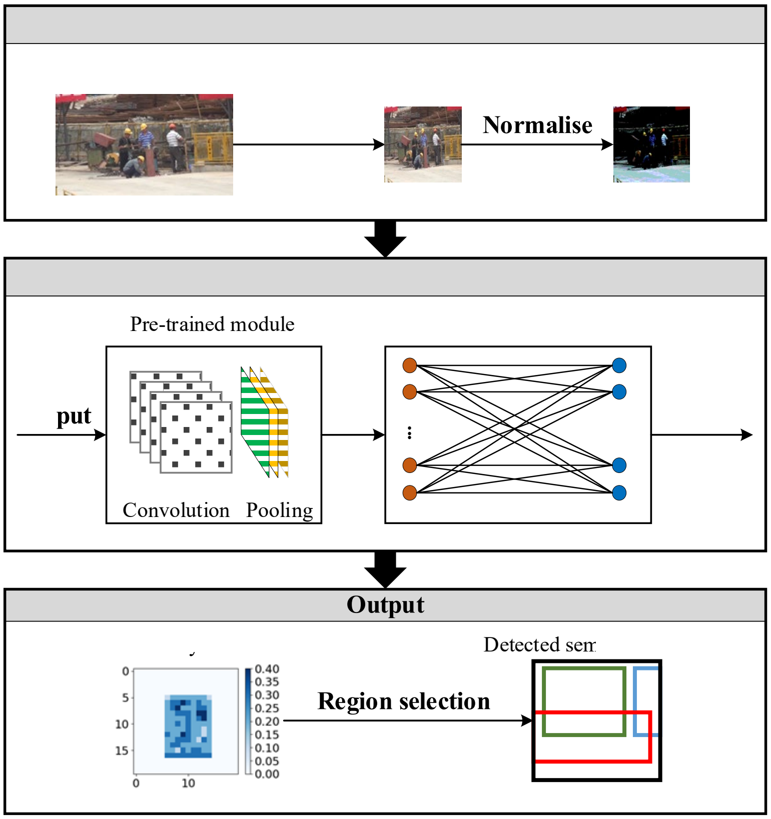


Figure 4. Proposed process for establishing deep neural network to detect semantic regions

Table 2. The difference between the convolutional layers and fully connected layers

	Convolutional layers	Fully connected layers (dense layers)
Input	A set of feature maps	One-dimensional vector
Output	A set of feature maps	One-dimensional vector
Advantages	The value of a node in a convolutional layer is computed based on the features in a small receptive field of the previous layer.	The value of a node in a fully connected layer is learned based on all input features of the previous layer.
Disadvantages	Convolutional layers are more efficient in terms of computing and model complexity because the features in one map share the same parameters.	Fully connected layers contain a massive set of parameters. It is computationally expensive.

The specific structure of the pre-trained module was determined according to the experiments. The experimental results are given in Appendix B. The results show that *nasnet* model and *pnasnet* model are the recommended model structures. The performance and structures of the pre-trained modules from *nasnet* and *pnasnet* are similar, except that the structure from *pnasnet* is quite simple and computationally efficient with only a small decrease in precision. The specific layers of the selected pre-trained modules are listed in Table B2 and [28].

In order to select a proper activation function f(x), three commonly used activation functions were tested in Appendix C: *softmax* (Eq.17), *softplus-like* (Eq.18) and *hyperbolic tangent* (*tanh*) (Eq.19) functions. The results show that the *tanh* function provides more stable converging behaviour during model training. Therefore, the *tanh* function was selected as the activation function in this research study.

(2) Saliency matrix generation and loss optimisation in the training stage

At the training stage, saliency matrix is generated by downsampling the saliency map of an image. The flowchart in Figure 5 was used to compute saliency maps in this study. Firstly, the event regions that capture frequently co-existing events and objects were manually labelled in image series. Then, saliency maps were computed according to the labelled regions. In the resulting saliency maps of image series, the co-existing objects are highlighted.

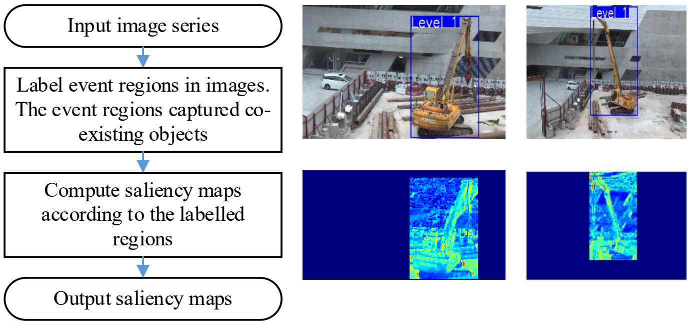

Input image series
Label event regions in images. The event regions captured co-existing objects
Compute saliency maps according to the labelled regions
Output saliency maps
Level_1
Level_1

Figure 5. Co-saliency computation based on labelled event regions in images

The saliency value (S_i^k) of the i^{th} pixel in a labelled region R_k is defined as per Eq. (1). The saliency values S_i^{global} and S_i^{local} are computed from the global and local contrast as per Eq. (2) and Eq. (3) respectively.

Notably, the parameters β and γ are constant; c_i and c_j are the colour value of the i^{th} and j^{h} pixel respectively; N_i is the neighborhood of the i^{th} pixel; k_i and k_j are the weights for the i^{th} and j^{h} pixel respectively.

$$S_i^k = \beta S_i^{global} + (1-\beta) S_i^{local} \tag{1}$$

$$\begin{gathered} S_i^{global} = \sum_j d_{i,j} \\ d_{i,j} = \begin{cases} \gamma \mathrm{dist}\left(c_i, c_j\right) & i \in R_k \wedge j \in \neg R_k \\ (1-\gamma)\mathrm{dist}\left(c_i, c_j\right) & i \in R_k \wedge j \in R_k \\ 0 & i \in \neg R_k \end{cases} \end{gathered} \tag{2}$$

$$S_i^{local} = \sum_{j \in N_i} \left(c_j k_j + c_i k_i\right) \tag{3}$$

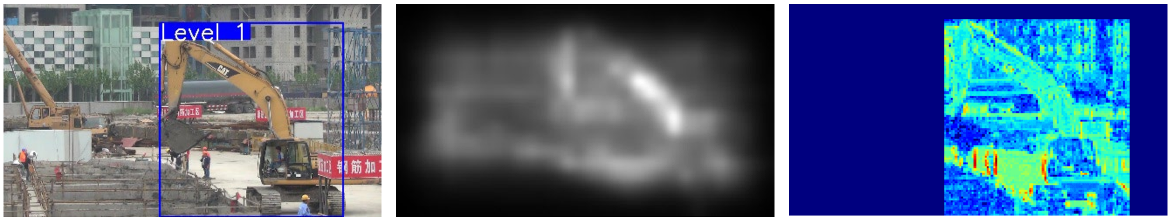
Then, the values in saliency map are rescaled into a range of [0, 1] by normalisation operation (Eq. 4) and mapped by quantification using a set of bounding values $\{b_n\}_{n=1}^{N+1}$ (Eq. 5). If a pixel is in different bounding boxes, the saliency value is the maximum value of all parent bounding boxes (Eq. 6). w_k is the weight of the bounding box R_k.

$$S_i^k = \frac{S_i^k - \min\left(S_i^k\right)}{\max\left(S_i^k\right) - \min\left(S_i^k\right)} \tag{4}$$

$$S_i^k = b_n,\ S_i^k \in (b_n, b_{n+1}] \tag{5}$$

$$S_i = \max\left(\left\{w_k S_i^k\right\}_{k=1}^{k=L}\right) \tag{6}$$

For example, given the images and bounding box in Figure 6(a), instead of generating the classic saliency map as shown in Figure 6(b), the saliency map based on the labelled bounding box was generated using the modified method as shown in Figure 6(c).



(a)
Input image and bounding box

(b)
Classic saliency map

(c)
Modified saliency map

Figure 6. Examples of saliency computation

A saliency matrix is used as the output of the deep network. When training the deep neural network model, the loss value is minimised during the iterations. In this study, the loss value is computed as the average of the squared difference between the predicted outputs (i.e., output saliency matrix $\widetilde{Y}=\{\tilde{y}_k, 1\leq k\leq N\}$) and the desired outputs (i.e., ground truth saliency matrix $Y=\{y_k\}, 1\leq k\leq N$) as Eq.(7). Error backpropagation [29] is used to update the variables during the iterations. In particular, the loss of a prediction, which is computed using Eq.(7), is distributed backward throughout each layer of the network. The model with a smaller loss value will have a better performance.

$$\text{Loss function: } J=\frac{1}{N}\sum\nolimits_k\left(y_k-\tilde{y}_k\right)^2 \tag{7}$$

(3) Semantic region selection method

In this study, semantic regions are bounded by rectangles in an image. Four indicators are proposed so as to describe the morphology of a rectangle region. The four indicators are (i) saliency score, (ii) region scale, (iii) saliency centrality, and (iv) saliency saturation. The definitions of these indicators are shown in Eqs. (8)–(12). Saliency score indicates the number of saliency content in a rectangle region. Region scale indicates the proportion of a rectangle region in an image. Saliency centrality indicates the degree of saliency centralised in a rectangle region. Saliency saturation indicates the state of a rectangle region is saturated with saliency content.

$$\text{Saliency score: } SS(R)=\sum_{x\in R} S(x)\text{, R is a rectangular region} \tag{8}$$

$$\text{Region scale: } RS(R)=\frac{RA(R)}{A}\text{, A is the area of image} \tag{9}$$

$$\begin{gathered} RA(R)=w\times h, \\ \text{w is width of region candidate, h is height of region candidate} \end{gathered} \tag{10}$$

$$\begin{gathered} \text{Saliency centrality: } SC(R)=\frac{SS(R_{1/4})}{SS(R)}/\frac{RA(R_{1/4})}{RA(R)}, \\ R_{1/4}\text{ is the } 1/4 \text{ rectangle region at the center of Region R} \end{gathered} \tag{11}$$

$$\text{Saliency saturation: } ST(R)=\frac{SS(R)}{RA(R)} \tag{12}$$

The rectangular region proposals are selected according to the four metrics. Figure 7 shows the region proposals selection algorithm. This algorithm has one stem and two branches. The stem can be used to obtain Figure 8(a), and the left and right branches can be used to obtain Figure 8(b) and Figure 8(c) respectively. Figure 8(a) shows the region proposals for semantic region detection. Figure 8(b) shows the region proposals for object detection. Figure 8(c) shows the region proposals for cropping an input image to a down-sized thumbnail image. The parameters T_1, T_2 and T_3 are manually selected thresholds used to reduce the number of region proposals.

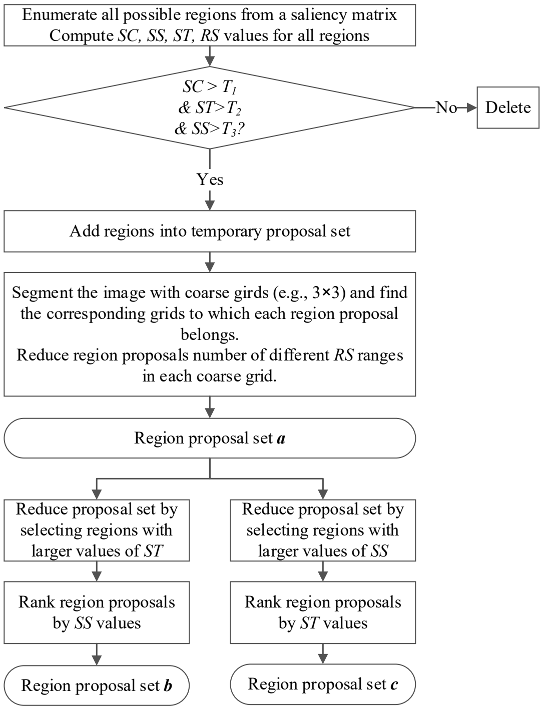

Enumerate all possible regions from a saliency matrix
Compute SC, SS, ST, RS values for all regions
SC > T1 & ST>T2 & SS>T3?
No
Delete
Yes
Add regions into temporary proposal set
Segment the image with coarse girds (e.g., 3×3) and find the corresponding grids to which each region proposal belongs.
Reduce region proposals number of different RS ranges in each coarse grid.
Region proposal set a
Reduce proposal set by selecting regions with larger values of ST
Reduce proposal set by selecting regions with larger values of SS
Rank region proposals by SS values
Rank region proposals by ST values
Region proposal set b
Region proposal set c

Figure 7. Algorithm of region proposal selection

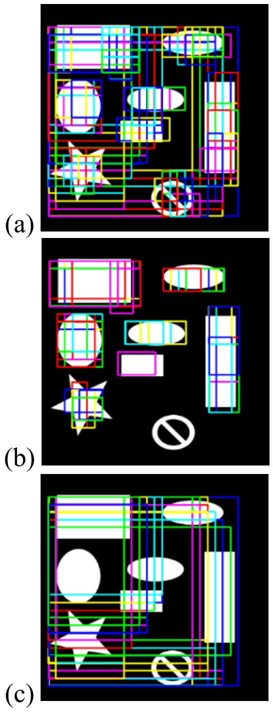
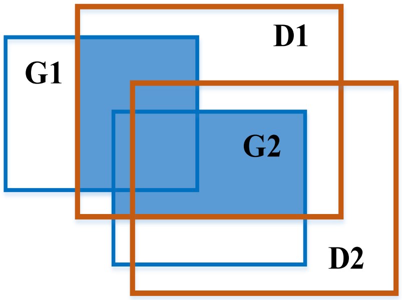


Figure 8. Examples of region proposal selection

3.3. Performance evaluation of the method for semantic region detection

The accuracy of the results of semantic region detection is evaluated based on the intersection of a detected semantic region and ground-truth bounding boxes (Figure 9).



Detection: D1, D2
Ground truth: G1, G2
D1∩(G1∪G2): Intersection of a detection and ground truth based on all ground truth bounding boxes
D1∪(G1∪G2): Union of a detection and ground truth
G1∪G2: Union of ground truth

Figure 9. Relationship of detections and ground truth bounding boxes

Intersection over union (IoU) is used to measure the accuracy of object instance detection (Eq.(13)). It is defined as the ratio of the intersection over the union of a detection and one single ground truth box in a site image, e.g., IoU of $D1=(D1\cap G1)/(D1\cup G1)$. In the object detection research domain, the threshold of IoU for a detected result is often set as 0.5. Normally, an $IoU>0.5$ is considered a good prediction. Notably, another popular metric to evaluate region detection is the percentage of overlap (PoO) between the detected region and the ground truth within the bounding box. PoO is computed by $PoO_G= (D\cap G)/G$ or $PoO_D=(D\cap G)/D$. In contrast with the PoO, the $IoU==(D\cap G)/(D\cup G)$ not only emphases on the overlapping but also the ratio of the overlap both in the detected region (D) and ground-truth bounding box (G). The value of IoU changes when the size of either the ground-truth bounding box or the corresponding detection result changes, while PoO_G or PoO_D might not change in certain situations. For instance, the value of PoO_G remains unchanged given the change of D and an unchanged overlap value; the value of PoO_D remains unchanged given the change of G and an unchanged overlap value. As such, the IoU is recommended when measuring the accuracy of the detection.

One pixel may belong to different ground truth bounding boxes. Intersection over ground truth union (IoU_G) in Eq.(14) is used to measure the precision of semantic region detection, e.g., the shaded section in Figure 9 is the IoU_G of $D1=[D1\cap(G1\cup G2)]/(G1\cup G2)$.

$$IoU=\frac{\text{Intersection of a detection and a single ground truth box}}{\text{Union of a detection and a single ground truth box}} \quad (13)$$

$$IoU_G=\frac{\text{Intersection of a detection and ground truth boxes}}{\text{Union of ground truth boxes}} \quad (14)$$

Intersection over ground truth union (IoU_G) is used to determine whether a detection is correct for semantic region detection. If the value of IoU_G exceeds a defined threshold, the detection is a true positive (TP), otherwise, false positive (FP). Region detection precision P is a ratio calculated by dividing the number of true positives by the number of detections including both true positive and false positive. To reduce the bias of using different thresholds, precision-threshold curve (P~t_i) and average precision (AP) are used as the evaluation metrics. Noted that AP is computed by Eq.(15). The maximum value of AP is 1.

$$\mathrm{AP}=\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{P}\left(\mathrm{IoU_G}\geq \mathrm{t_i}\right) \tag{15}$$

Average recall (AR) metrics are used to assess the completeness of detecting the semantic region [30]. They are a group of metrics given different number of detections per image [31]. That means, for instance, AR can be calculated for a group of images having the number of detections of 1, 3, 10, 20, and 50 per image (Eq. 16). As mentioned, IoU can be calculated based on ground truth and detection. For a particular ground truth bounding box in an image, the IoU associated for all detections in the image can be calculated. If the maximum of IoU (i.e., $\mathrm{IoU_{Max}}$) exceeds 0.75, this ground truth bounding box is regarded as a true positive (TP), otherwise, false negative (FN). The AR is a ratio calculated by dividing the number of true positives by the number of detections. The maximum value of AR is 1.

$$\mathrm{AR}=\left\{\mathrm{AR}^{\mathrm{N_D}=1},\mathrm{AR}^{\mathrm{N_D}=3},\mathrm{AR}^{\mathrm{N_D}=10},\mathrm{AR}^{\mathrm{N_D}=20},\mathrm{AR}^{\mathrm{N_D}=50}\right\} \tag{16}$$

4. PRACTICAL CASE STUDY

In this section, a case study using the proposed method is illustrated. Two image sets were collected. The first dataset consists of 1,109 site images captured by a camera (HD 1080p) from three building projects as shown in Figure 2. The second dataset consists of 176 images of excavators which were randomly collected from the Internet.

Three experiments were conducted for evaluating the performance of the proposed method. The objective of the 1st experiment is designed to evaluate the performance of detecting semantic region for site images using the proposed method based on labelled image dataset. The objective of the 2nd experiment is designed to evaluate the performance of detecting an excavator (semantic region) in site images using the proposed method based on randomly searched site images. The objective of the 3rd experiment is to evaluate the performance of detecting a semantic region detection using the proposed method based on modified saliency in contrast with the existing methods based on saliency.

The computations were performed by a computer with the configuration of a CUDA GPU (GeForce GTX 1050, 4G, compute capability: 6.1) with an i7-7700 CPU and 8 GB RAM.

4.1. Label site images and train deep neural network for semantic region detection

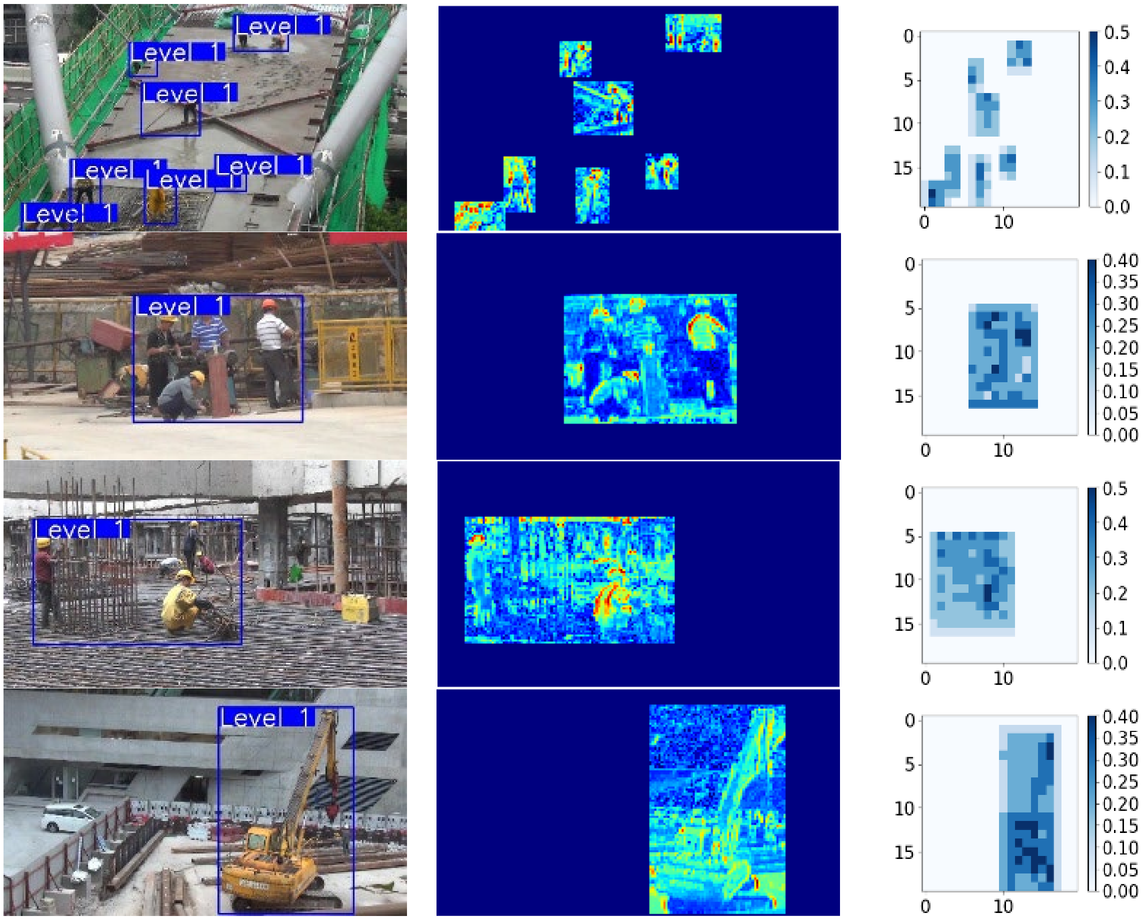
A deep neural network model consists of one pre-trained module using both *nasnet* and *tanh* activation functions at the output layer is developed. The model is trained using the learning rate with a maximum value of 0.0003.

Table 3 listed the numbers of labelled instances of semantic regions in the site images.

Table 3. Summary of ground truth information in the training set (containing 1109 images)

Level	Number of instances	Number of images
1	1,323	866
2	756	398
3	168	148
4	228	225

At the training stage, the saliency matrix for the output layer is computed based on the saliency map. Each element in the saliency matrix is the summation of the saliency values of pixels with the same coordinates in the grid. Figure 10 shows the examples of saliency map and saliency matrix for training images.



(a) (b) (c)

Figure 10. Saliency map and saliency matrix examples for training images. (a) labelled bounding boxes; (b) saliency map with the size of the original image; (c) saliency matrix computed from image grids

The loss of a prediction, which is computed using Eq.(7), is distributed backward throughout each layer of the network. The loss value is the *average of the squared differences* between the output of the network and the saliency matrix during the training process. The loss value during the training process is shown in Figure 11. The loss value was gradually decreased as iterations increased. The speed of loss reduction is high at the beginning and steady at the end of the iterations.

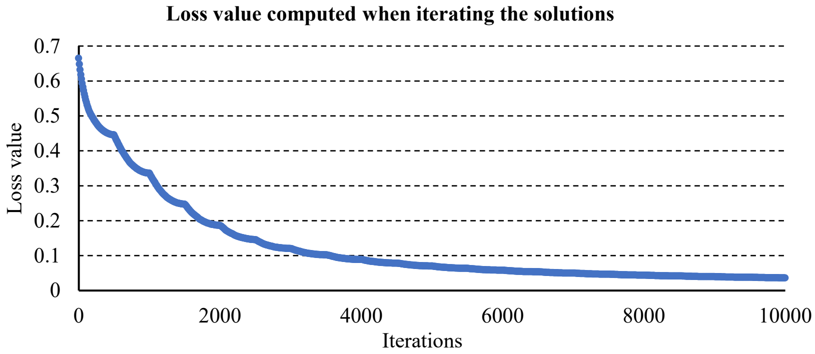


Figure 11. Loss value as per computing time during training

To improve the learning performance of the deep neural network, we should select a proper (i) pre-trained modules, (ii) activation function at the output layer, and (iii) learning rate for the training stage. The experiments for testing the learning performance based on different pre-trained modules, activation functions, and learning rates are given in Appendices B, C, and D, respectively.

4.2. Performance evaluation of semantic region detection for site images based on labelled image dataset

To evaluate the performance of detecting semantic region using the trained model, 222 images from labelled image dataset are used. Notably, these images were not used for training. **Error! Reference source not found.** shows some examples of the detected semantic regions based on the calculated saliency matrices.

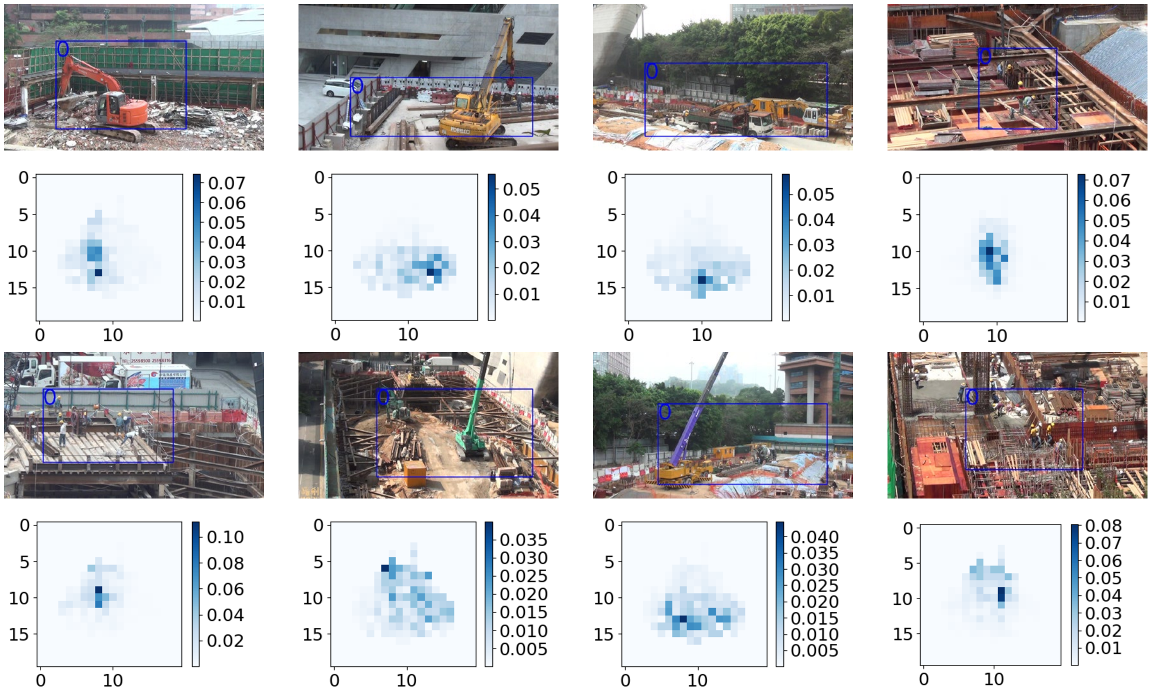


Figure 12. Examples of semantic regions detected using the proposed method

The experiment results show that 50 region proposals were selected as per the right branch in Figure 7. The 50 regions have the largest *ST* values. The detection results of 222 images were compared with the ground truth annotations which were labelled by the authors.

Figure 13 shows the distribution histogram of $\mathrm{IoU_G}$ of all detections (the number of all detections is 222×50). The result shows that most of the detected semantic regions have large $\mathrm{IoU_G}$ values, which means large areas of ground-truth bounding boxes are detected in those regions.

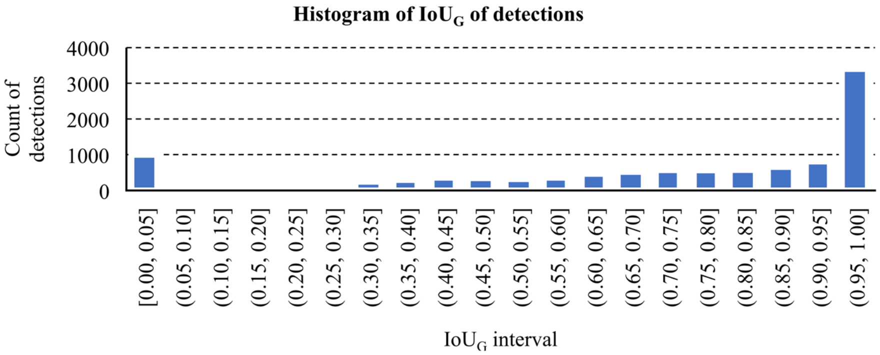


Figure 13. IoU_G distribution of all detections using labelled 222 images

Figure 14 shows the precision-threshold curve. The precision is computed when different thresholds of IoU_G are selected. When a small threshold is selected, the value of precision is high. Average precision (AP), which equals to the area under the precision-threshold curve, is 0.683. Noted that the maximum value of AP is 1. A larger value of AP indicates a more accurate detection result. Therefore, the resulting semantic regions are closer to the ground-truth bounding boxes.

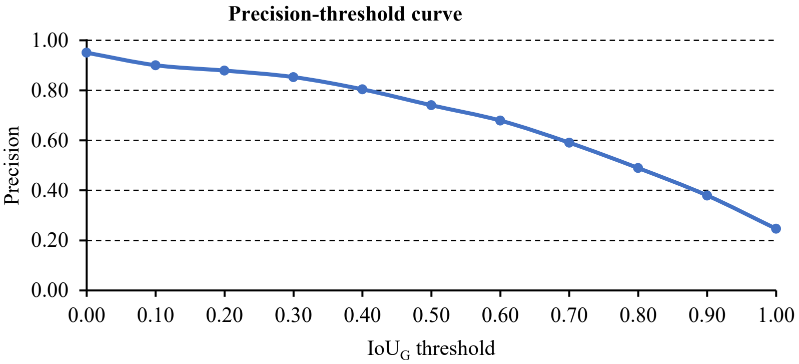


Figure 14. Precision of all detections using labelled 222 images when different thresholds of IoU_G are selected

Table 4 shows the average recalls (AR) given the numbers of detections ($N_D \in \{1,3,10,20,50\}$). AR closer to 1 indicates more accurate semantic region detection results. The thresholds for IoU to determine the true positives are set as 0.75, 0.85 and 0.95. Large thresholds are used to ensure most of the semantic information is detected.

Therefore, the AR is smaller when a large threshold is chosen, and the AR is larger when more detections are obtained.

Table 4. Average recall given different detections based on 222 labelled images

Threshold	$AR^{N_D=1}$	$AR^{N_D=3}$	$AR^{N_D=10}$	$AR^{N_D=20}$	$AR^{N_D=50}$
0.75	0.897	0.942	0.974	0.994	1.000
0.85	0.877	0.920	0.956	0.986	1.000
0.95	0.863	0.898	0.942	0.964	0.988

AP (0.683) and AR (Table 4) show that the detections using the proposed method successfully captured a large amount of semantic information from the site images. Therefore, the proposed method has a great potential in semantic region detections for construction applications.

4.3. Experiments of using the trained network for detecting excavators from images randomly collected from the Internet

To validate the model performance for another image set, the proposed method is used for detecting the object in images which are randomly collected from the Internet. 176 free images showing excavators available on the internet were collected. 215 instances of excavators in these images were labelled by the authors. The model, which was trained by the site images (i.e., the one mentioned in previous subsections), was used for detecting the excavators in these 176 images.

50 region proposals were selected as per the left branch in Figure 7. The 50 regions have the largest *SS* values. Figure 15 shows the distribution histograms of $\mathrm{IoU}_{\mathrm{Max}}$ with 50 detections per image. The result shows that most of the resulting semantic regions having $\mathrm{IoU}_{\mathrm{Max}}$values larger than 0.4, which is an acceptable value for object detection [32].

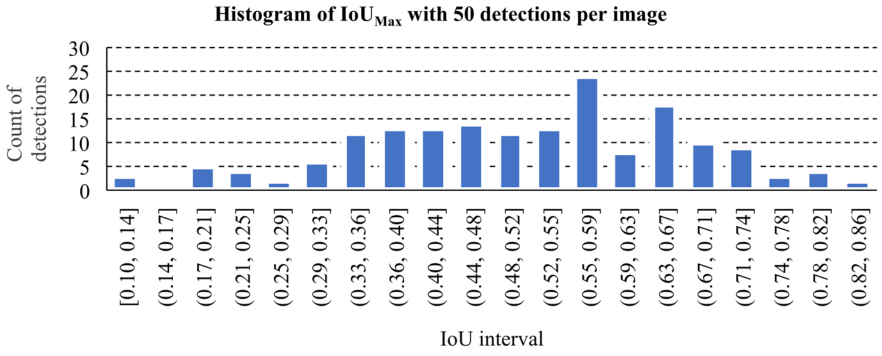


Figure 15. Distribution of IoU_{Max} of object region detections based on 176 randomly searched images

Figure 16 shows the precision-threshold curve for evaluating the results of object region detection. When a small threshold is selected, the value of precision is high. The average precision is 0.508. It is slightly smaller than the average precision as indicated in the last experiment (Section 4.2) since the model is now used for another dataset.

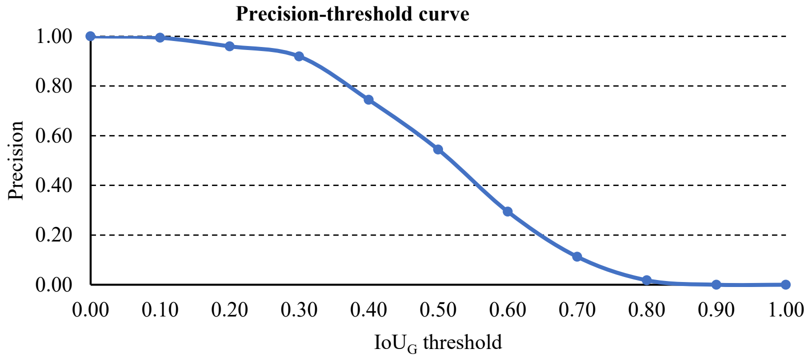


Figure 16. Precision of object region detections based on 176 randomly searched images

Table 5 shows the average recalls (AR) as per the numbers of detections ($N_D \in \{1,3,10,20,50\}$) for each images. The threshold of IoU is generally set as 0.5 in object detection [32]. In this experiment, multiple thresholds were used (i.e., 0.55, 0.65, 0.75, 0.85 and 0.95). Similar to the last experiment (Section 4.2), AR is smaller when a larger threshold is selected, and AR is larger when more detections are resulted.

Table 5. Average recall per image based on 176 randomly searched images

Threshold	$AR^{N_D=1}$	$AR^{N_D=3}$	$AR^{N_D=10}$	$AR^{N_D=20}$	$AR^{N_D=50}$
0.55	0.266	0.318	0.509	0.640	0.757
0.65	0.244	0.286	0.404	0.545	0.629
0.75	0.192	0.220	0.290	0.383	0.472
0.85	0.160	0.164	0.235	0.305	0.376
0.95	0.151	0.156	0.184	0.241	0.283

AP (0.508) and AR (Table 5) show that the proposed method has a great potential to detect object regions in site images without re-training a specific object detection model.

4.4. Comparison of the semantic regions detected using the proposed method and other available saliency-based methods

The proposed semantic region detection method is compared with semantic regions based on two saliency computation methods including GBVS [25] and ITTI [24,33]. Note that there are many existing methods for saliency computation. Their performances vary as per different testing images. GBVS and ITTI methods are chosen. It is because GBVS saliency computation method was recognised as one of the most commonly-researched methods [34,35], and ITTI saliency is one of the most comprehensive computation methods.

Figure 17 shows some examples for visual comparison of the detected semantic regions (five instances in each image) using the two saliency-based methods and the proposed method. As shown, the images cover a large part of the construction sites since they were captured at remote locations. However, the activities only take place in a relatively small region and some images captured the contents which are not construction sites (Image 5 and Image 6). In contrast with the proposed method, the two saliency-based methods extracted the semantic regions from saliency maps highlight the visually salient objects and the centralised regions. Some project-relevant regions are severely neglected in semantic regions detected using the two saliency-based methods. For instance, the activity regions in Images 1, 3, 7 and 8 are largely neglected, the object region in Image 2 is partially neglected, and the major site components in Images 4, 5, 6 and 7 are largely neglected.

The semantic regions detected by the proposed method and two saliency-based methods proved that the proposed method is more suitable for detecting the semantic regions compared with the classic-saliency-based method for construction applications.

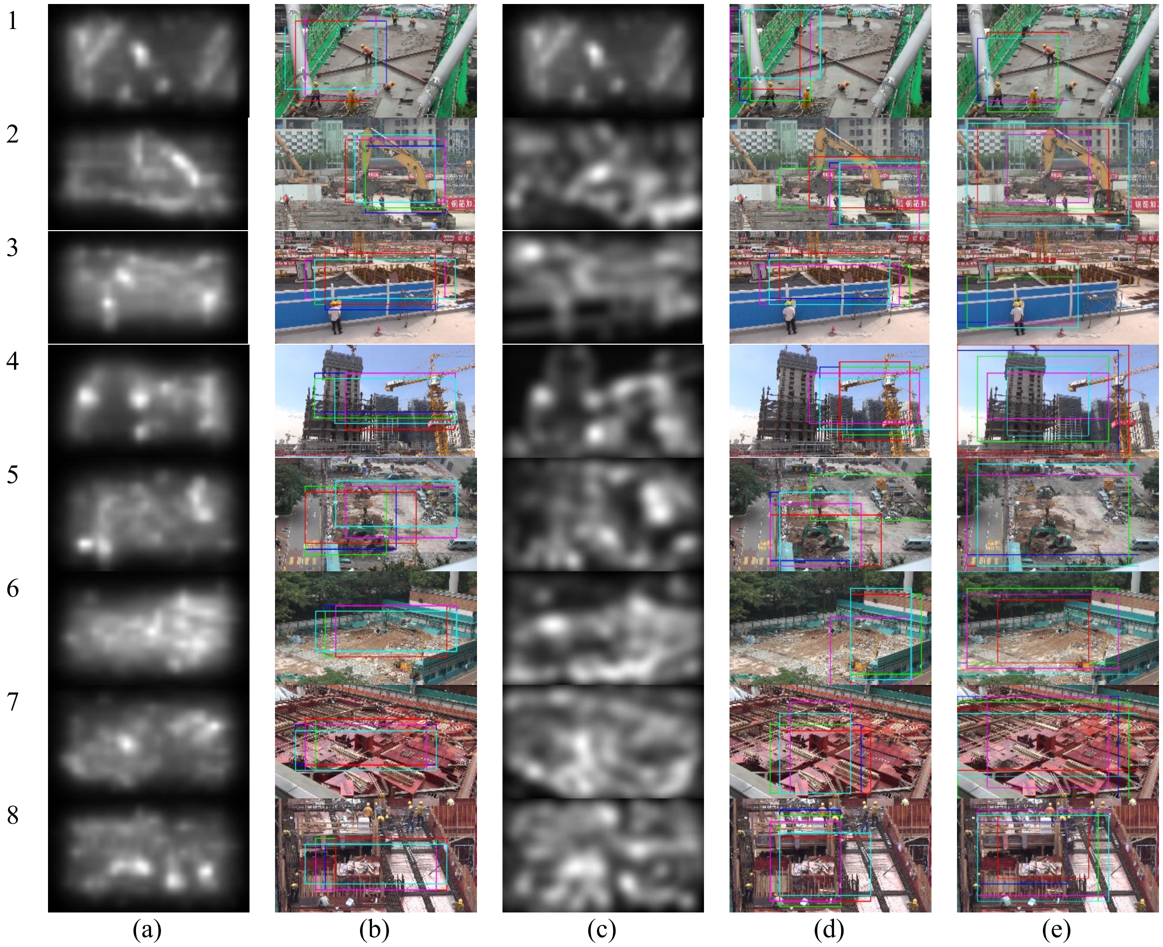


Figure 17. Comparisons of saliency-based region detection methods and the proposed method: (a) GBVS saliency map; (b) semantic regions detected using GBVS-based method; (c) ITTI saliency map; (d) semantic regions detected using ITTI-based method; (e) semantic regions detected using the proposed method.

As discussed above, the site images captured the content that is not relevant to construction activities, which are caused by the remote recording distance and fixed image aspect ratios (normally 16:9 and 4:3). Resulting semantic regions can be used to automatically crop such site images and videos for storing and retrieving the site images in an efficient way. That means, if the size of semantic regions is half of the original size, half of the data space will be saved. The results will be obvious when massive site images and videos dataset of a construction project with significantly long duration are stored.

5. Discussions of semantic region detection results and transfer learning performance

5.1. Discussions of the results in the first two experiments

The precision and recall are reported to evaluate the performance in the two experiments in Section 4.2 and 4.3. The results proved that the proposed method can be efficiently used to detect semantic regions in construction site images with an average precision of 0.683 and an average recall larger than 0.863. The trained model can be applied to detect object region from randomly searched images without re-training. The results of the object region detection give an average precision of 0.581 and an average recall larger than 0.509 (when the number of resulting semantic regions is more than 10).

Currently, there is no benchmark of precision for semantic region detection in the construction field. This proves the novelty of this research study. However, the authors compared the values for precision and recall with object detection methods shown in [36]. In [36], AP_{50} and AP_{75} were used to evaluate the results. AP_{50} and AP_{75} are average precisions (AP) when the thresholds of IoU are set as 0.50 and 0.75. [36] shows that the range of AP_{50} is from 0.441 to 0.611 for object detection based on the model of *Faster R-CNN-Resnet*, *YOLOv2*, *YOLOv3*, *SSD*, *DSSD* and *RetinaNet*. The value range of AP_{75} is much lower than AP_{50}. The range of AP_{75} is from 0.192 to 0.441 for object detection based on the models of *Faster R-CNN-Resnet*, *YOLOv2*, *YOLOv3*, *SSD*, *DSSD* and *RetinaNet*. In this research study, the average precision (AP) is given by the area under the precision-threshold curve. The definition of AP is close to AP_{50}. Compared with the benchmarks, our research study results of 0.683 and 0.581 are acceptable.

5.2. Evaluation and discussion the learning performance of transfer learning

Transfer learning has two advantages: (1) reducing data needed for training and (2) improving the learning performance. As such, we applied the transfer learning techniques in our study. The authors evaluated and discussed the performance of transfer learning based on the two aspects.

(1) Reducing data needed for training

Transfer learning reduces the required training data quantity. With transfer learning, a pre-trained model, trained on a large readily available dataset (on a completely different task, with the same input but different output) is reused. First, proper layers are determined to compute reusable features. The outputs of those layers are then used as input features to train a network that requires a smaller number of parameters. Therefore, the amount of training data is reduced. In general, it is difficult to determine the exact amount of data for a training model because the data quantity is case-specific and relevant to the task complexity. To our best knowledge, there is no such evaluation method yet in the research domains of construction management and computer science to determine the reduction in training data. For our research study, we found that transfer learning often required training sets of several thousand data items. Instead of measuring training dataset size, the authors, alternatively, measured the number of parameters. In this research study, the number of parameters decreased from 5.3 million to 4.6 million when using a pre-trained module based on *nasnet*.

(2) Improving the learning performance

Transfer learning will improve learning performance as shown in [37]. In [37], accuracy is chosen to evaluate the learning process. A larger accuracy value indicates better performance. The learning curve using transfer learning has a higher accuracy at the beginning of learning, a steep slope of the learning curve, and a high asymptotic accuracy value. In this research study, the loss of the output layer was used to evaluate the learning process. Notably, the loss values defined by Eq.(7). A small loss value indicates better performance. In contrast to the accuracy curve, the learning curve using transfer learning has a lower loss value at the beginning of learning, a steeper slope of the learning curve, or a lower asymptotic loss value.

We compared the loss values (i) using transfer learning and (ii) without using transfer learning using actual experiments for our proposed application. The results in Figure 18 and Figure 19 show that loss value, at the beginning with transfer learning (Figure 18) is smaller than the case without transfer learning (Figure 19). The results in Figure 19 show that the loss value has an increasing tendency without transfer learning (The increasing tendency often shows when we repeat the learning process). This is counter to our objective of minimising loss value. The results in Figure 18 and Figure 19 show that transfer learning performs better in the learning process.

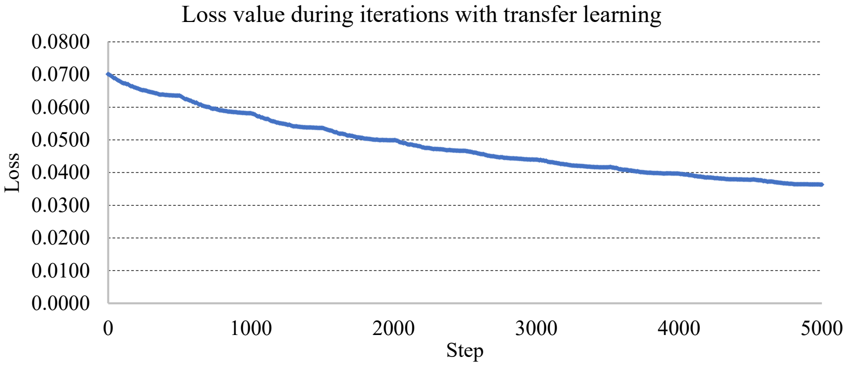


Figure 18. Loss performance with transfer learning

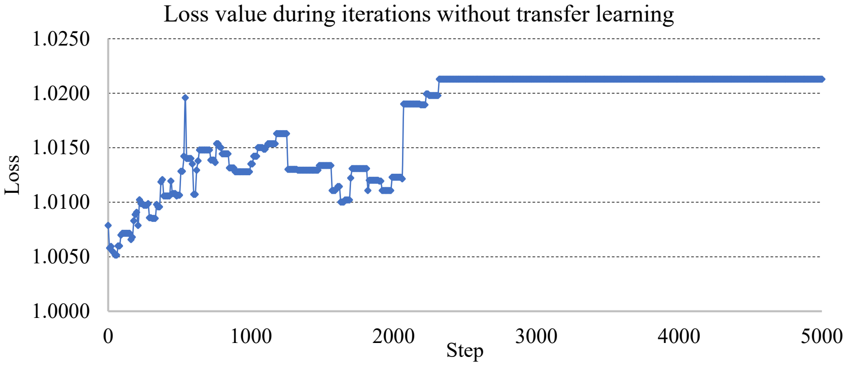


Figure 19. Loss performance without transfer learning

6. CONCLUSIONS

To improve the efficiency of storing, querying, summarising, and browsing of massive image data generated from construction sites, a novel semantic region detection method based on transfer learning and modified saliency computation was proposed. This method does not require to specify any target objects for detecting site objects.

Three experiments were conducted for evaluating the performance of the proposed method. The objective of the 1st experiment is designed to evaluate the performance of detecting semantic region for site images using the proposed method based on labelled image dataset. The objective of the 2nd experiment is designed to evaluate the performance of detecting an excavator (semantic region) in site images using the proposed method based on randomly searched site images. The objective of the 3rd experiment is to evaluate the performance of detecting a semantic region detection using the proposed method based on modified saliency in contrast with the existing methods based on saliency. The results proved that the proposed method can be efficiently used to detect semantic regions in construction site images with an average precision of 0.683 and an average recall larger than 0.863. The trained model can be applied to detect object regions from randomly searched images without re-training. The results of the object region detection give an average precision of 0.581 and an average recall larger than 0.509 (when the number of resulting semantic regions is more than 10). These numbers (average precision and average recall) may not be remarkable. However, they are still acceptable when compared with object detection precision and recall using state-of-the-art models. The experiment comparing the proposed method based on modified saliency with the methods based on classic saliency shows the semantic regions extracted using the proposed method captures more accurate project-relevant information while the classic-saliency-based methods highlight some objects which are not of interest to site managers.

The ***academic contributions*** of this research study listed below.

1. A **new** semantic region detection approach was proposed using construction site images. The semantic regions highlight the regions containing the most information in an image. This new method can be used to represent event (i.e., construction-activity) regions, and to crop the images to thumbnail size.
2. **New** coarse bounding boxes at four levels were proposed to label construction activities, instead of the tight bounding boxes which are commonly used for construction objects. This means, we are no longer required to establish a dataset for different object types. This saves much human effort in labelling objects in the images.
3. A **new** region selection algorithm was proposed. The algorithm can be used to obtain two types of proposed region: (i) region proposals for object detection and (ii) region proposals for cropping an input image.
4. The transfer learning technique is **reused** in this study. However, the authors selected the pre-trained module, along with the learning rate, and activation function of the output layer. The transfer learning technique improves the performance in model learning.
5. The saliency computation concept was **reused** in this study. However, the authors modified its computation using the labelled bounding boxes.

The ***practical contributions*** of this research study are: a semantic region labelling method for site images is proposed in construction domain, the trained model based on one image dataset can be also used for other site image datasets without re-training, and the detected results can be used to reduce image storage space and to add semantic region annotations for image retrieval.

Future research works focusing on increasing the accuracy of the proposed method by optimising the architecture of the deep neural network and the application of cropping site images using semantic regions are suggested.

ACKNOWLEDGEMENT

This research work is supported and funded by The Hong Kong Polytechnic University (Grant number: BE0S; Grant PI: Corresponding author; Grant title: Data-driven Resource Planning for Building Works in Hong Kong).

APPENDIX A: TRANSFER LEARNING TECHNIQUE

Transfer learning refers to developing and calibrating a neural network model in one task (e.g., Task A) and applying the model to a new task (Task B) [20]. This saves effort in developing the model from scratch in Task B. Users are only required to modify the existing Task A model to suit the requirements of Task B. For example, a model trained for text classification can be transferred to classifying images following the model tuning process, as shown in Figure A1.

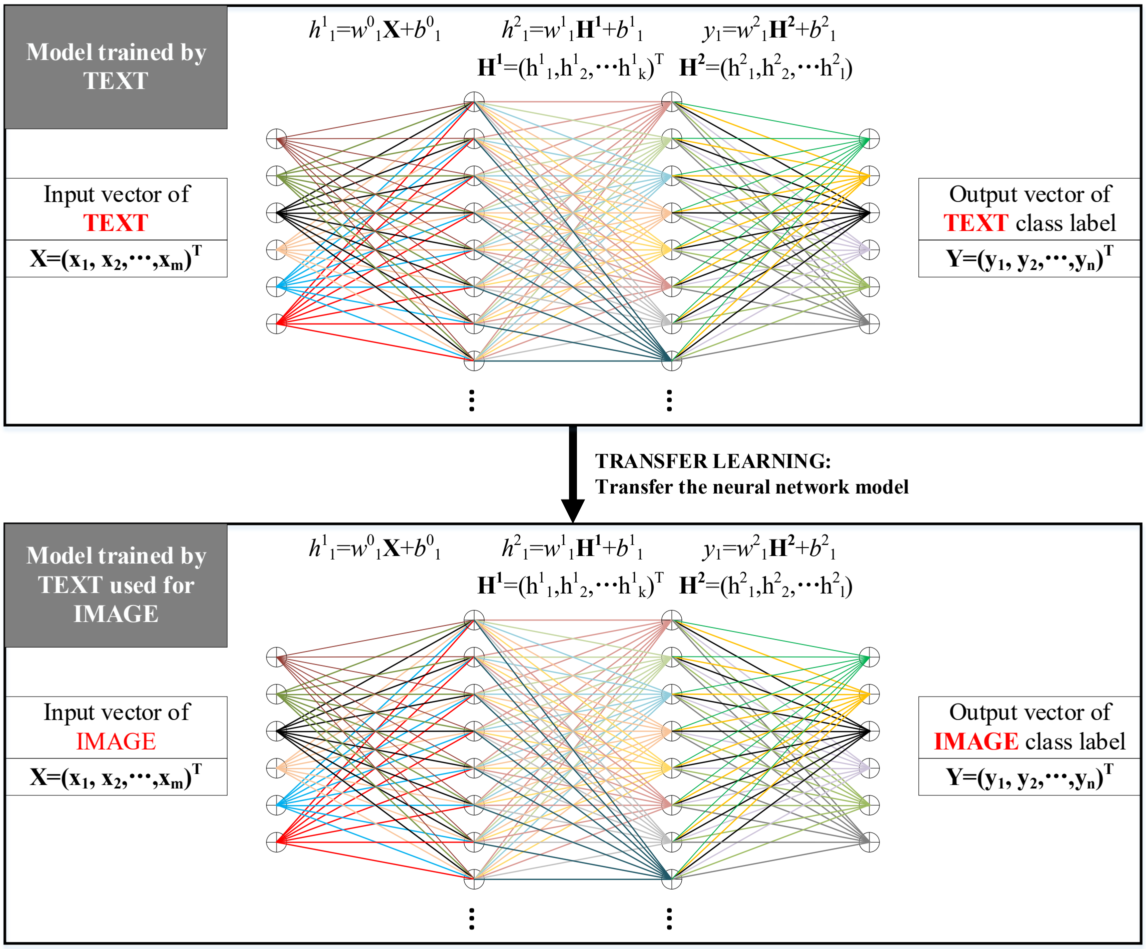

Model trained by TEXT
$h^1_1=w^0_1\mathbf{X}+b^0_1$
$h^2_1=w^1_1\mathbf{H}^1+b^1_1$
$y_1=w^2_1\mathbf{H}^2+b^2_1$
$\mathbf{H}^1=(h^1_1,h^1_2,\cdots h^1_k)^T$
$\mathbf{H}^2=(h^2_1,h^2_2,\cdots h^2_l)$
Input vector of TEXT
$\mathbf{X}=(x_1, x_2,\cdots,x_m)^T$
Output vector of TEXT class label
$\mathbf{Y}=(y_1, y_2,\cdots,y_n)^T$
TRANSFER LEARNING:
Transfer the neural network model
Model trained by TEXT used for IMAGE
$h^1_1=w^0_1\mathbf{X}+b^0_1$
$h^2_1=w^1_1\mathbf{H}^1+b^1_1$
$y_1=w^2_1\mathbf{H}^2+b^2_1$
$\mathbf{H}^1=(h^1_1,h^1_2,\cdots h^1_k)^T$
$\mathbf{H}^2=(h^2_1,h^2_2,\cdots h^2_l)$
Input vector of IMAGE
$\mathbf{X}=(x_1, x_2,\cdots,x_m)^T$
Output vector of IMAGE class label
$\mathbf{Y}=(y_1, y_2,\cdots,y_n)^T$

Figure A1: Concept of transfer learning

In this example, the neural networks comprise two hidden layers. The neural network model is fully connected. In the first task, the model was first used to classify text, the input is a vector of a text string (X); the output is a vector of text class labels (Y). This model was trained from scratch, which means all parameters (all *w* and *b*) are *randomly initialised* and updated to minimise training error (which is calculated from Y and then backpropagated to the hidden layers, H^1 and H^2). All parameters were trained using a TEXT dataset. In the second task, the hidden layers were reused to classify IMAGE. The parameters related to these layers (e.g., w^{0}_{1}, b^{0}_{1} and w^{1}_{1}, b^{1}_{1}) were initialised, and the trained values were used for TEXT classification. Other non-related parameters (e.g., w^{2}_{1}, b^{2}_{1}) are randomly initialised. Then, this model is re-trained using an IMAGE dataset. This process is known as *model tuning*.

Since many pre-trained models and the corresponding parameters are currently available online, two approaches are commonly used when using transfer learning (Figure A2): (i) develop a new model and *(ii) reuse a pre-trained model*. In transfer learning, the pre-training process consumes massive amounts of training time and computing power. Therefore, the reuse pre-trained models [Method (ii)] is more popular, since the pre-training process was done by other researchers.

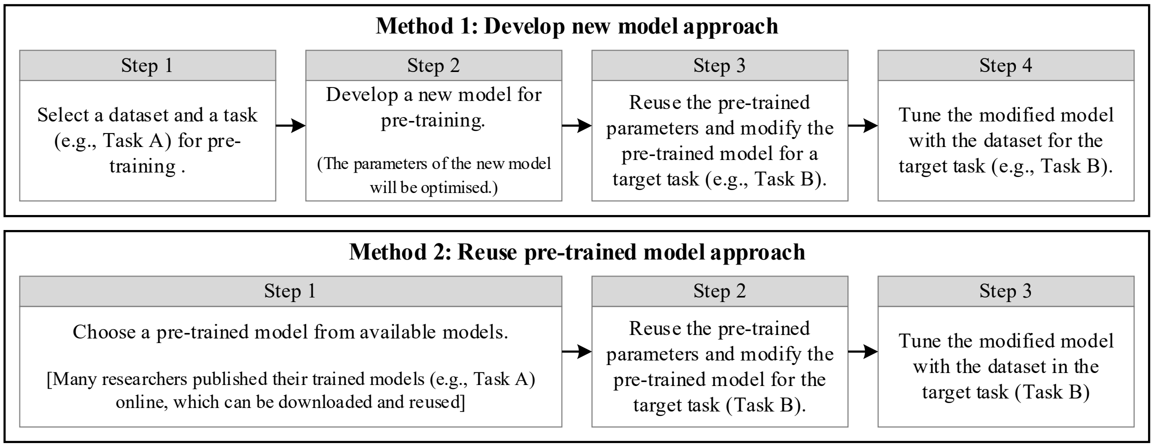

Method 1: Develop new model approach
Step 1
Select a dataset and a task (e.g., Task A) for pre-training .
Step 2
Develop a new model for pre-training.
(The parameters of the new model will be optimised.)
Step 3
Reuse the pre-trained parameters and modify the pre-trained model for a target task (e.g., Task B).
Step 4
Tune the modified model with the dataset for the target task (e.g., Task B).
Method 2: Reuse pre-trained model approach
Step 1
Choose a pre-trained model from available models.
[Many researchers published their trained models (e.g., Task A) online, which can be downloaded and reused]
Step 2
Reuse the pre-trained parameters and modify the pre-trained model for the target task (Task B).
Step 3
Tune the modified model with the dataset in the target task (Task B)

Figure A2: Develop new model vs reuse pre-trained model

In this research study, the authors reused a pre-trained model (*nasnet*) for image classification published by TensorFlow [38] to establish a new model producing a saliency matrix to extract semantic regions.

APPENDIX B: SELECTION OF THE PRE-TRAINED MODELS

A suitable pre-trained module consisting of convolutional layers is used for semantic region detection in the proposed method. Five state-of-the-art neural networks were tested. They are *vgg16* [39], *inception-v4* [40], *resnet* [41], *nasnet* [28], and *pnasnet* [28]. The parameters of these five networks were trained on the ILSVRC-2012-CLS dataset [42] for image classification.

Table B1 summarises the input data size, the number of trainable variables, parameters, along with the performance (top-1, top-5 accuracy) for image classification based on ILSVRC-2012-CLS image dataset of the five networks [38].

Table B1. Summary of five deep neural networks (This table was produced according to the published data by Tensorflow team [38]))

Pre-trained models	*Vgg16*	*Resnet* (v1, 101)	*Inception* (v4)	*Nasnet* (mobile)	*Pnasnet* (mobile)
Input image size	224	224	299	224	224
Number of trainable variables	32	314	306	742	615
Number of trainable parameters (million)	138.36	44.55	46.01	5.29	5.07
Top-1 accuracy (%)	71.5	76.4	80.2	82.7	82.9
Top-5 accuracy (%)	89.8	92.9	95.2	96.2	96.2

The pre-trained module is used as a feature extractor to generate the feature maps with different sizes and depths. Five pre-trained modules were selected from the five deep neural networks in Table B2. These five networks are the state-of-the-art methods for image classification. The starting layers of the five pre-trained modules are the input layer of these five networks. The ending layers are selected according to several rules: (i) the size of an output feature map from the ending layer is closed to the defined size of the saliency matrix, (ii) the total number of the scalar elements in the feature maps from the ending layer is relatively close for the five pre-trained modules. According to these rules, the architectures of the tested networks using five pre-trained convolutional modules are shown in Table B2.

Table B2. Architecture of deep neural networks with different pre-trained modules

Deep neural networks	**Layer/Cell**	**Output shape**
Vgg16	Input	[None,224,224,3]
	Conv_Pool_Cell	[None,112,112,64]
	Conv_Pool_Cell	[None,56,56,128]
	Conv_Pool_Cell	[None,28,28,256]
	Conv_Pool_Cell	[None,14,14,512]
	Flatten_Cell	[None, 100352]
	Output	[None,400]
Resnet	Input	[None,224,224,3]
	Pad_Cell	[None,230,230,3]
	Conv_BN_Pool_Cell	[None,112,112,64]
	Block (3 units)	[None,28,28,256]
	Block (4 units)	[None,14,14,512]
	Flatten_Cell	[None, 100352]
	Output	[None,400]
Inception-v4	Input	[None,299,299,3]
	Stem_Cell	[None,35,35,384]
	4×Inception-A	[None,35,35,384]
	Reduction-A	[None,17,17,1024]
	6×Inception-B	[None,17,17, 1024]
	Inception-B	[None,17,17, 192]
	Flatten_Cell	[None, 55488]
	Output	[None,400]
Nasnet (mobile)	Input	[None,224,224,3]
	Stem_Cell	[None,56,56,44]
	Stem_Cell	[None,28,28,88]
	4×Normal_Cell	[None,28,28,264]
	Reduction_Cell	[None,14,14,352]
	4×Normal_Cell	[None,14,14,528]
	Flatten_Cell	[None,103488]
	Output	[None,400]
Pnasnet (mobile)	Input	[None,224,224,3]
	Stem_Cell	[None,56,56,65]
	Stem_Cell	[None,28,28,135]
	3×Normal_Cell	[None,28,28,270]
	Reduction_Cell	[None,14,14,540]
	2×Normal_Cell	[None,14,14,540]
	Flatten_Cell	[None, 105840]
	Output	[None,400]

The computation time and the converge curve of loss value are obtained using a fixed validation set during a training process as shown in Figures B1-B4. Two commonly-used activation functions *softmax* function, and *tanh* function are tested since their performances are unknown at this step.

Figure B1 and Figure B2 are the results from the experiment using *softmax* as the activation function at the output layer. Figure B3 and Figure B4 are the results based on the experiment using *tanh* as the activation function. In these two experiments, the learning rates are different since converging speeds of different activation functions are different.

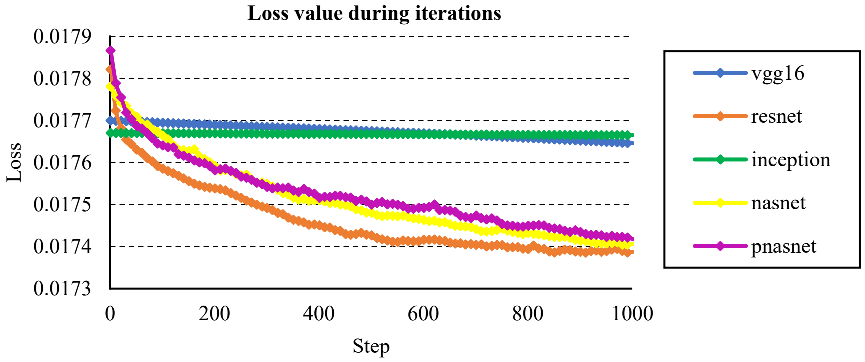


Figure B1. Loss value during iterations as per pre-trained modules (*softmax* function for output layer)

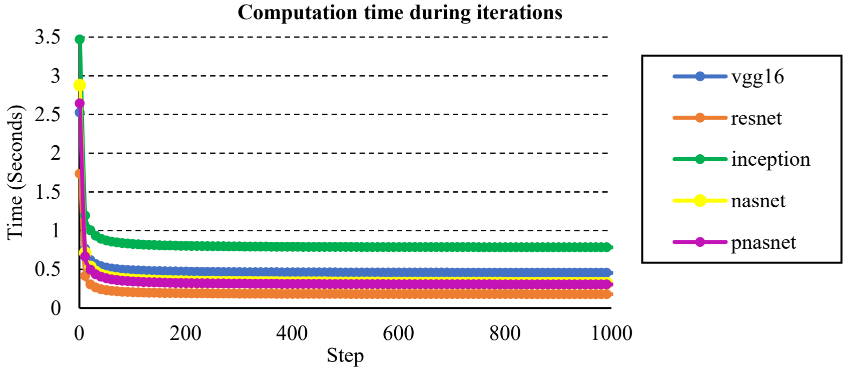


Figure B2. Computation time during iterations as per pre-trained modules (*softmax* function for output layer)

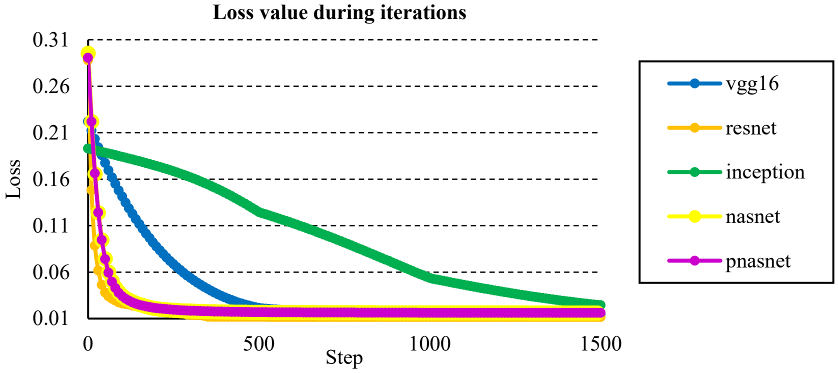


Figure B3. Loss value during iterations as per pre-trained modules (*tanh* function for output layer)

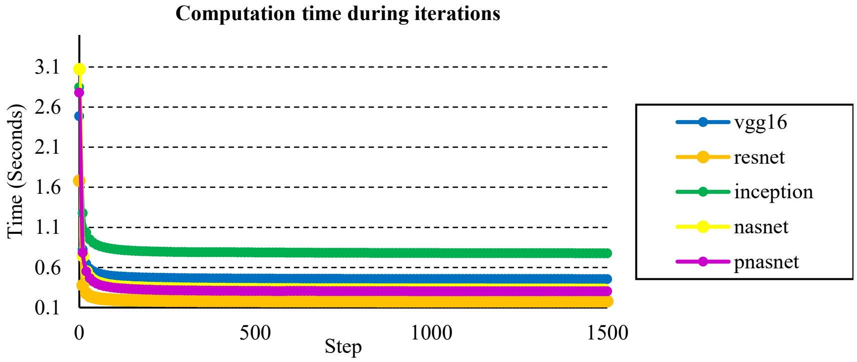


Figure B4. Computation time during iterations as per pre-trained modules (*tanh* function for output layer)

The results show that the loss and time performances using different pre-trained modules are different for tuning the model. In particular, Figure B1 and Figure B3 show that the converge speeds of *resnet*, *nasnet* and *pnasnet* are faster than *inception* and *vgg16* models; Figure B2 and Figure B4 shows that *inception* and *vgg16* models consume more time than *resnet*, *nasnet* and *pnasnet* models in each iteration.

The performance and structures of the pre-trained modules from *nasnet* and *pnasnet* are similar, except the structure from *pnasnet* is more simple and computationally efficient. The *nasnet* module is used for selecting the activation function (Appendix C) and learning rate (Appendix D).

APPENDIX C: SELECTION OF ACTIVATION FUNCTION FOR TRAINING

At the output layer of the deep neural network, an activation function is used to normalise the range of output values and control the loss gradient backpropagation process. In order to select a proper activation function, three activation functions were tested: *softmax*, *softplus-like* and *hyperbolic tangent* (*tanh*) function. The definitions of these three functions are listed below. The activation function is used to convert the value of an output node (x_i) to $f(x_i)$.

$$\textit{Softmax}\text{ function: } f(x_i)=\frac{e^{x_i}}{\sum_i e^{x_i}} \quad (17)$$

$$\textit{Softplus}\text{-like function: } f(x_i)=\ln(1+e^{x_i}) - \ln 2 \quad (18)$$

$$\text{Hyperbolic tangent (}\textit{tanh}\text{) function: } f(x_i)=\frac{2}{1+e^{-2x_i}}-1 \quad (19)$$

Figure C1 shows the graphs of the three activation function. Figure C2 shows the results of transforming a random signal (**x**) using the three activation functions respectively. Figure C3 is the converge curve of loss value based on the three functions and *resnet* module. Figure C4 is the converge curve of loss value based on the three functions and *nasnet* module.

These three functions are used to transform a vector ($\mathbf{x}=\{x_i|L\leq x_i\leq U\}$) to another vector ($\mathbf{y}=\{y_i|L'\leq y_i\leq U'\}$), Figure C1 and Figure C2 show that these three functions will compress the value range of the vector **x** ($L'<L$ and $U'<U$). Figure C3 and Figure C4 show that the loss value dereases at a relatively slow speed when *tanh* function is used. However, the converging value of *tanh* function is closed to the one of the other two functions. Thus, *tanh* function provides a more stable converging behaviour during model training. Therefore, *tanh* function is selected as the activation function in this research study.

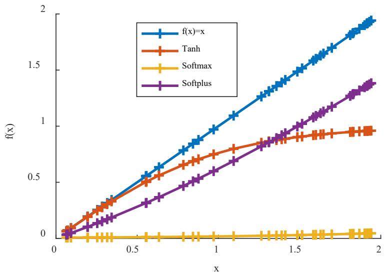


Figure C1. Graph of three activation functions

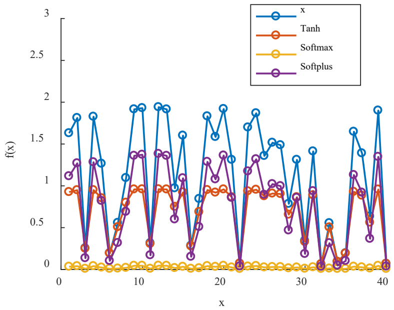


Figure C2. Example of transforming random vector (x) using three activation functions

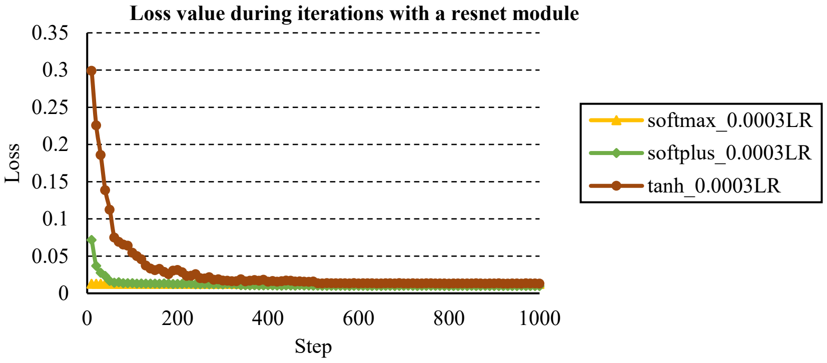


Figure C3. Loss value during iterations as per the three functions (a *resnet* module is used)

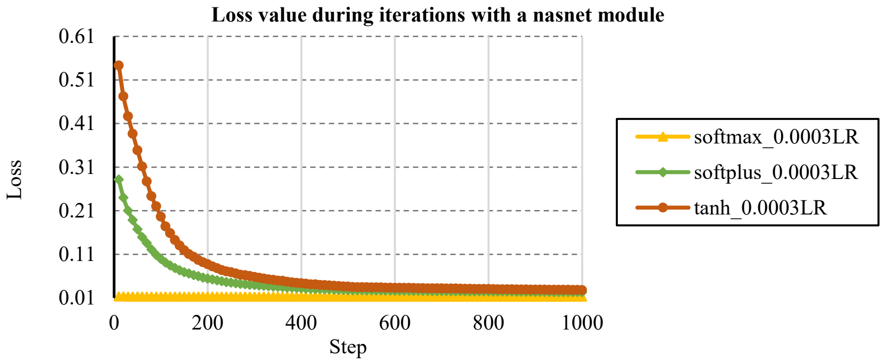


Figure C4. Loss value during iterations as per the three functions (a *nasnet* module is used)

APPENDIX D: SELECTION OF LEARNING RATE FOR TRAINING

When training the deep neural network, the converging speed of the solution is dependent on the learning rate. A high value of learning rate can decrease the loss value at a fast speed at the beginning of the training process but overshooting may happen at the end. A lower value of learning rate can be used but longer time is required to search the optimum solution which may be a local minimum. Previous studies show that adaptive learning rate is more efficient than a fixed one for training purpose [43].

In this research study, the learning rate is designed as a linear decay function in each cycle as per the iteration steps. Figure D1 shows the decay process. Figure D2 shows the loss value comparison of learning rates (i.e., with maximum values of 0.003, 0.0003, 0.0001, 0.00003) with the iterations of 1,000 steps when *tanh* function is used. The results show that a low learning rate may cause the loss value to converge with prolonged time. Thus, in consideration of the training time and loss value, the learning rate is chosen as 0.0003.

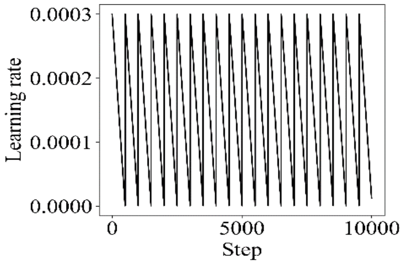


Figure D1. Learning rate during iterations

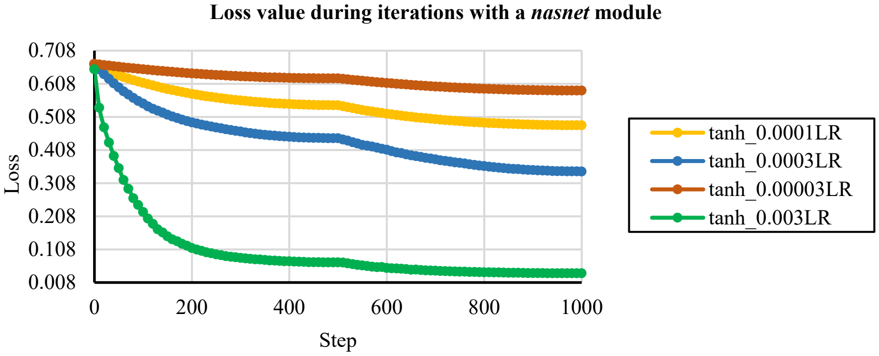


Figure D2. Loss values during iterations as per different learning rates (*tanh* function is used for output layer)

REFERENCES

[1] L. Chen, Y. Wang, Automatic key frame extraction in continuous videos from construction monitoring by using color, texture, and gradient features, Automation in Construction 81 (2017) 355-368. doi:10.1016/j.autcon.2017.04.004.

[2] Y.J. Cha, W. Choi, O. Buyukozturk, Deep learning-based crack damage detection using convolutional neural networks, Computer-Aided Civil and Infrastructure Engineering 32 (5) (2017) 361-378. doi:10.1111/mice.12263.

[3] H. Kim, H. Kim, Y.W. Hong, H. Byun, Detecting construction equipment using a region-based fully convolutional network and transfer learning, Journal of Computing in Civil Engineering, ASCE 32 (2) (2018) 04017082. doi:10.1061/(ASCE)CP.1943-5487.0000731.

[4] I. Brilakis, L. Soibelman, Comparison of manual and user-guided methodologies for the classification and retrieval of construction site images, Construction Research Congress 2005 (2005) 1-10. doi:10.1061/40754(183)126.

[5] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road damage detection and classification using deep neural networks with smartphone images, Computer-Aided Civil and Infrastructure Engineering 33 (12) (2018) 1127-1141. doi:10.1111/mice.12387.

[6] I. Brilakis, L. Soibelman, Y. Shinagawa, Material-based construction site image retrieval, Journal of Computing in Civil Engineering 19 (4) (2005) 341-355. doi:10.1061/(ASCE)0887-3801(2005)19:4(341).

[7] I.K. Brilakis, L. Soibelman, Shape-based retrieval of construction site photographs, Journal of Computing in Civil Engineering 22 (1) (2008) 14-20. doi:10.1061/(ASCE)0887-3801(2008)22:1(14).

[8] I. Brilakis, L. Soibelman, Multimodal image retrieval from construction databases and model-based systems, Journal of Construction Engineering and Management 132 (7) (2006) 777-785. doi:10.1061/(ASCE)0733-9364(2006)132:7(777).

[9] M. Memarzadeh, M. Golparvar-Fard, J.C. Niebles, Automated 2d detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors, Automation in Construction 32 (2013) 24-37. doi:10.1016/j.autcon.2012.12.002.

[10] L. Hui, M.W. Park, I. Brilakis, Automated brick counting for façade construction progress estimation, Journal of Computing in Civil Engineering, ASCE 29 (6) (2015) 04014091. doi:10.1061/(ASCE)CP.1943-5487.0000423.

[11] A. Sasithradevi, S.M.M. Roomi, G. Maragatham, Content based video retrieval via object based approach, TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 781-787. doi:10.1109/TENCON.2017.8227965.

[12] D. Neumann, T. Langner, F. Ulbrich, D. Spitta, D. Goehring, Online vehicle detection using haar-like, lbp and hog feature based image classifiers with stereo vision preselection, 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 773-778. doi:10.1109/IVS.2017.7995810.

[13] N. Nabizadeh, M. Kubat, Brain tumors detection and segmentation in mr images: Gabor wavelet vs. Statistical features, Computers & Electrical Engineering 45 (2015) 286-301. doi:10.1016/j.compeleceng.2015.02.007.

[14] E. Zalama, J. Gomez-Garcia-Bermejo, R. Medina, J. Llamas, Road crack detection using visual features extracted by gabor filters, Computer-Aided Civil and Infrastructure Engineering 29 (5) (2014) 342-358. doi:10.1111/mice.12042.

[15] A. Cord, S. Chambon, Automatic road defect detection by textural pattern recognition based on adaboost, Computer-Aided Civil and Infrastructure Engineering 27 (4) (2012) 244-259. doi:10.1111/j.1467-8667.2011.00736.x.

[16] Y. Cha, K. You, W. Choi, Vision-based detection of loosened bolts using the hough transform and support vector machines, Automation in Construction 71 (2016) 181-188. doi:10.1016/j.autcon.2016.06.008.

[17] S. Paisitkriangkrai, C. Shen, A. van den Hengel, Strengthening the effectiveness of pedestrian detection with spatially pooled features, Springer International Publishing, Cham, 2014, pp. 546-561.

[18] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T.M. Rose, W. An, Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, Automation in Construction 85 (2018) 1-9. doi:10.1016/j.autcon.2017.09.018.

[19] W. Fang, L. Ding, B. Zhong, P.E.D. Love, H. Luo, Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach, Advanced Engineering Informatics 37 (2018) 139-149. doi:10.1016/j.aei.2018.05.003.

[20] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, Cambridge, Massachusetts, 2016.

[21] Z. Kolar, H. Chen, X. Luo, Transfer learning and deep convolutional neural networks for safety guardrail detection in 2d images, Automation in Construction 89 (2018) 58-70. doi:10.1016/j.autcon.2018.01.003.

[22] M. Zhang, Y. Yang, H. Zhang, Y. Ji, N. Xie, H.T. Shen, Deep semantic indexing using convolutional localization network with region-based visual attention for image database, Springer International Publishing, Cham, 2017, pp. 261-272.

[23] J. Wang, H. Lu, X. Li, N. Tong, W. Liu, Saliency detection via background and foreground seed selection, Neurocomputing 152 (2015) 359-368. doi:10.1016/j.neucom.2014.10.056.

[24] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, Ieee Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1254-1259. doi:10.1109/34.730558.

[25] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, Advances in neural information processing systems, 2007, pp. 545-552.

[26] K.Y. Chang, T.L. Liu, H.T. Chen, S.H. Lai, Fusing generic objectness and visual saliency for salient object detection, 2011 International Conference on Computer Vision, 2011, pp. 914-921. doi:10.1109/ICCV.2011.6126333.

[27] D. Zhang, H. Fu, J. Han, A. Borji, X. Li, A review of co-saliency detection algorithms: Fundamentals, applications, and challenges, Acm Transactions on Intelligent Systems and Technology 9 (4) (2018). doi:10.1145/3158674.

[28] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, arXiv e-prints, 2017.

[29] R. Rojas, The backpropagation algorithm, Neural networks 149-182. doi:10.1007/978-3-642-61068-4_7.

[30] P.O. Pinheiro, R. Collobert, P. Dollár, Learning to segment object candidates, Advances in neural information processing systems, 2015, pp. 1990-1998.

[31] T.Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick, P. Dollár, Microsoft coco: Common objects in context, arXiv e-prints, 2014.

[32] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3150-3158. doi:10.1109/CVPR.2016.343.

[33] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis & Machine Intelligence (11) (1998) 1254-1259.

[34] T. Judd, F. Durand, A. Torralba, A benchmark of computational models of saliency to predict human fixations, 2012.

[35] J. Abeln, L. Fresz, S.A. Amirshahi, I.C. McManus, M. Koch, H. Kreysa, C. Redies, Preference for well-balanced saliency in details cropped from photographs, Frontiers in Human Neuroscience 9 (704) (2016). doi:10.3389/fnhum.2015.00704.

[36] Á. Casado-García, C. Domínguez, J. Heras, E. Mata, V. Pascual, The benefits of close-domain fine-tuning for table detection in document images, arXiv e-prints, 2019.

[37] L. Torrey, J. Shavlik, Transfer learning, Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques, IGI Global, Hershey, PA, USA, 2009.

[38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv e-prints, 2016.

[39] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv e-prints, 2014.

[40] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, Thirty-First AAAI Conference on Artificial Intelligence,2017.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv e-prints, 2015.

[42] N. Silberman, S. Guadarrama, Tensorflow-slim image classification model library, 2016.

[43] Y. Dauphin, H. De Vries, Y. Bengio, Equilibrated adaptive learning rates for non-convex optimization, Advances in neural information processing systems, 2015, pp. 1504-1512.