StructureNet: Deep Context Attention Learning for Structural Component Recognition

Akash Kaothalkar[1], Bappaditya Mandal[2] and Niladri. B. Puhan[1]
[1]*Indian Institute of Technology Bhubaneswar, India*
[2]*Keele University, Newcastle-under-Lyme, U.K.*
{*ak75, nbpuhan*}*@iitbbs.ac.in, b.mandal@keele.ac.uk*

Keywords: Class Contexts, Context Attention, Semantic Segmentation, Structural Component Recognition.

Abstract: Structural component recognition using images is a very challenging task due to the appearance of large components and their long continuation, existing jointly with very small components, the latter are often outcasted/missed by the existing methodologies. In this work, various categories of the bridge components are exploited at the contextual level information encoding across spatial as well as channel dimensions. Tensor decomposition is used to design a context attention framework that acquires crucial information across various dimensions by fusing the class contexts and 3-D attention map. Experimental results on benchmarking bridge component classification dataset show that our proposed architecture attains superior results as compared to the current state-of-the-art methodologies.

1 INTRODUCTION

Manual inspection of structural damages consumes long but crucial decision making time intervals, resulting in the delay of assessment and damage control/management/mitigation/recovery activities. The first step in the image/video based automatic damage assessment process is the detection or recognition (used interchangeably in this work) of the structural components, as damages can vary from structure to structure. For example, respective damages on columns and beams/slabs might be handled in different ways. Thus, if the captured image or video data can provide an initial assessment by recognizing the structural components (such as columns, beams, slabs, etc) automatically without actually going on the site of the damage, it can serve as a head-start for further inspection (Bhattacharya et al., 2021b). Required image or video data can be collected by using digital cameras, UAVs, or even satellite imaging.

Critical infrastructures like bridges play a very crucial role during any environmental disaster, as their structures are responsible for the movement of vehicles and people from one place to another. Thus, the inspection of bridges and similar structures can be treated as a high priority and mission critical task. Our aim in this work is to get some valuable information about the structural components without actually going on-site, in a non-intrusive manner, analyzing images/videos, captured at a distance, while consuming less time. Traditional methods use machine learning techniques which are mostly hand-engineered (Koch et al., 2014; Zhu and Brilakis, 2010). The datasets used for such works are either small in size or contain images of a single structure per image. Structural component recognition is also performed using 3-D point clouds (Golparvar-Fard et al., 2011a; Golparvar-Fard et al., 2011b; Lu et al., 2019), but these methods require setting up sensor networks near the structures, which can be a cumbersome, time-consuming and tedious process.

Recent benchmarks, such as (Narazaki et al., 2017; Yeum et al., 2019) make use of semantic segmentation for bridge structural component recognition considering pixel accuracy as the standard evaluation metric. These works make use of multi-scale convolutional neural networks (CNNs) using the existing architectures. Recent work (Narazaki et al., 2020) uses deep semantic segmentation models to recognise the bridge components. Other works (Gao and Mosalam, 2018; Liang, 2019; Miao et al., 2019) try to combine both defect and structural component segmentation procedures; however, they only consider binary classification with limited structural components. Thus, the works that are done in the field of structural components recognition using non-intrusive (at a distance) vision based methodologies are limited and the challenges are underestimated in the current literature. This is also evident by the reported low accuracy rates, such as shown in Table 1. Improvement in this field of bridge structural component recognition will strongly support automation in structural defects recognition/management and health monitoring (Bhattacharya et al., 2021a).

Recent works in semantic segmentation (Huang et al., 2019; Zhang et al., 2019) have exploited category features rather than global features used in the earlier works (Chen et al., 2018; Zhao et al., 2017). As each pixel belongs to a different category, exploiting class-level features gives improved performance. Non-local based self-attention (Wang et al., 2018) has also been a popular method for generating the class-level contexts. However, such methods rely on 2-D affinity matrix that can lose salient information along the channel dimensions. To encompass richer information, tensor decomposition theory (Kolda and Bader, 2009) exploits a 3-D attention map without losing information along the channel dimension.

The proposed *StructureNet* framework contributes towards structural component recognition by proposing a novel architecture that fuses class contexts and inter-category relations obtained through designing a 3-D attention map. Class contexts consider the contextual information from a categorical perspective, which is an accumulation of features belonging to that class (Zhang et al., 2019). The attention map captures long-range dependencies and its fusion with class contexts generates a modified feature map comprising class-level relations as well as class pixels aggregations. The interaction between pixel representations and class-level predictions provides a better scope for generating crucial features as they can be exposed to class distribution across the dataset. The datasets available in the field of structural component recognition are limited and thus Bridge Component Classification Dataset is collected from the authors of (Narazaki et al., 2017) and the results are compared with the relevant benchmarking methods.

2 ARCHITECTURE FOR STRUCTURAL COMPONENT RECOGNITION

To exploit various class-level features of structural components, the proposed StructureNet consists of a backbone architecture along with attention modules to extract feature maps. Soft predictions are computed from the backbone architecture, which are amalgamated to generate the class contexts and attention maps. At the later stage of the architecture, both are fused to generate prediction ready semantically rich modified feature maps. The model architecture is shown in Fig. 1. Across the model description, 1×1 *conv* refers to the sequence of $conv \rightarrow batchnorm \rightarrow ReLU$, except in last prediction layer.

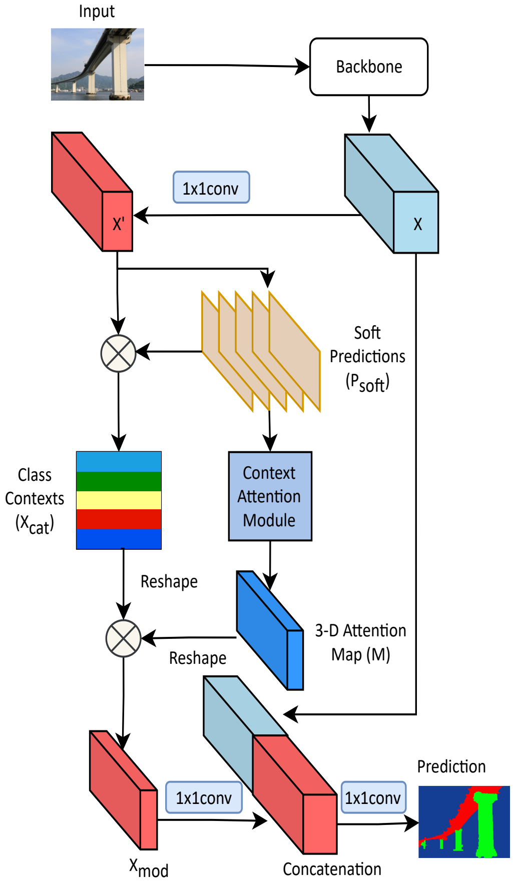


Figure 1: Illustration of StructureNet. The architecture follows, (i) the image is given as input to the backbone architecture ; (ii) soft predictions are computed; (iii) class contexts and attention maps are generated; (iv) fusion of both class contexts and 3-D attention map; (v) concatenation and final prediction.

Backbone: The model uses ResNet-50 (He et al., 2016) architecture with an output stride of 8 following PSPNet (Zhao et al., 2017), where classification and last two pooling layers are removed and the dilation rate of the convolution of the last two stages are set to 2 and 4, respectively.

The model architecture works on fusion of two major aspects (i) generating *class contexts* and (ii) generating a 3-D attention map using the *context attention module*. They are described in the following subsections.

2.1 Generating Class Contexts

The purpose of generating class contexts is the interaction of class-level features with the globally trained features (Zhang et al., 2019). To generate class contexts, we make a soft prediction from the output feature map $X \; \varepsilon \; \mathcal{R}^{C\times H\times W}$. Soft predictions represented as $P_{soft} \; \varepsilon \; \mathcal{R}^{N\times H\times W}$ are initial predictions that are made on the output features of the backbone architecture. Here, N is the number of classes and C, H, and W are channel, height, and width dimensions, respectively. The channel dimension of X is reduced from C to C' through 1×1 *conv* to save the computations and is represented by X'. Applying 1×1 *conv* with N filters on X' generates the soft predictions. Class contexts are represented by $X_{cat} \; \varepsilon \; \mathcal{R}^{N\times C'}$ and generated by the dot product of X' and P_{soft}. It is preceded by required reshape operation and followed by normalization through $softmax$ operation. *Class contexts*, X_{cat} is obtained by:

$$X_{cat} = softmax(X^{'} \cdot P_{soft}) \tag{1}$$

The feature maps, $X \; \varepsilon \; \mathcal{R}^{C\times H\times W}$ are learned over the entire dataset. Since the class context of each of the categories interacts with the respective features across the dataset, it broadens the learning capacity of the model.

2.2 Context Attention (CA) Module

According to tensor decomposition theory (Kolda and Bader, 2009; Chen et al., 2020), any high-rank tensor can be represented as the combination of rank-1 tensors. Non-local self-attention models can lead to the loss of information in channel dimension as they work with a 2-D affinity matrix. Since context features contain both spatial as well as channel information, working with a 3-D attention map without losing channel dimension seems to be more accurate.

To solve the issue with non-local self attention-based methodology, we treat soft predictions as a high-rank problem and refine them using tensor decomposition. From the previous works, it can be observed that the context prediction is a high-rank problem (Huang et al., 2019). Low rank tensors are synthesized together to generate a high-rank 3-D attention map without salient losing information in any dimension. Soft predictions, P_{soft} are given as input to this module and rank-1 tensors are generated across each dimensions. These rank-1 tensors are generated across each dimension by applying a sequence of $Globalpool \rightarrow 1\times 1$ *conv* $\rightarrow$ *sigmoid* across each dimension i.e. category (N), height (H) and width (W). For each dimension, m rank-1 tensors are generated, where m is selected to be the number of classes. This selection for m enables each class to learn about all the other classes and create an robust attention map.

All these rank-1 tensors are synthesized together to get attention map $M \; \varepsilon \; \mathcal{R}^{N\times H\times W}$. For instance, $p_{n1} \; \varepsilon \; \mathcal{R}^{N\times 1\times 1}, p_{h1} \; \varepsilon \; \mathcal{R}^{1\times H\times 1}$ and $p_{w1} \; \varepsilon \; \mathcal{R}^{1\times 1\times W}$ will be synthesized to create an auxiliary attention map, $\mathcal{M}_1$. All these auxiliary attention maps are linearly scaled and added, where the scaling parameter, α is a trainable parameter. Thus, the final output is 3-D attention map M given by (2) and (3). The context attention module is illustrated in Fig. 2.

$$\mathcal{M}_i = p_{ni} \cdot p_{hi} \cdot p_{wi} \tag{2}$$

$$M = \sum_{i=1}^{r} \alpha_i \mathcal{M}_i \tag{3}$$

In our architecture, the CA module goes ahead by applying tensor decomposition theory to soft predictions and fusing the attention map generated with the class level contexts rather than only with pixel representations. The proposed work also differentiates by the selection of the rank variable r. Choosing the value of r equal to the number of classes can be thought of as dedicating one attention map per class and therefore helps the model architecture to generate inter-category relationships.

2.3 Fusion of Attention map with Class Contexts

The final stage of the network fuses the generated class contexts (X_{sec}) and attention map (M) to create more semantically rich pixel representations (X_{mod}). Dot product between the two across the class dimension (N) yields $X_{mod} \; \varepsilon \; \mathcal{R}^{C'\times H\times W}$, given by:

$$X_{mod} = \{X_{cat} \; \cdot M\}_{Across\ the\ dimension\ N} \cdot \tag{4}$$

The architecture utilizes 3-D attention map which is obtained by passing P_{soft} through *context attention module* yielding $M \; \varepsilon \; \mathcal{R}^{N\times H\times W}$. Finally, the dot product between class contexts (X_{sec}) and attention map (M) yields the modified feature maps (X_{mod}).

Class contexts learn the relationship between pixels and category representation, while the attention map makes every class learn about all the other classes thus generating inter-category relationships. Thus, the fusion operation yields pixel representations rich with category interrelations as well as contextually affluent features.

To match with the shape of the backbone generated feature maps, the channel dimension of X_{mod} is changed back from C' to C through 1×1 *conv*. The value of C' is chosen to be 512 during implementation. Finally, we concatenate X and X_{mod} and refine it by 1×1 *conv* to get the final predictions.

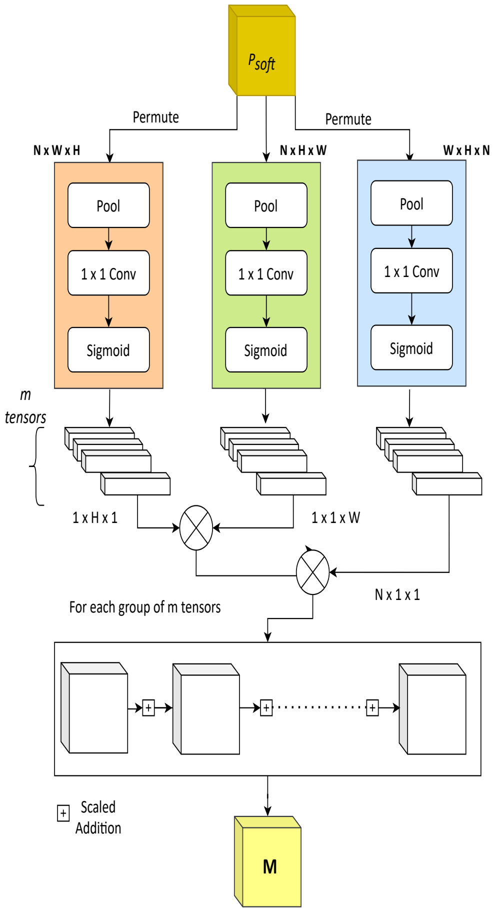


Figure 2: The architecture of context attention module. P_{soft} (soft predictions) are input to the module, which performs low rank factorization on each of the dimensions (H, W, N) r times. The output of the module is a 3-D attention map utilizing all the information contained in soft predictions.

3 EXPERIMENTS

3.1 Bridge Component Classification Dataset

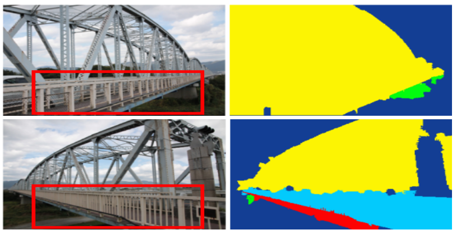
This dataset contains a total (train+test) of 1,563 bridge images (Narazaki et al., 2017) obtained for research and comparison evaluation purposes. The test set has 234 images. The pixel-wise labeling belongs to 5 classes: Non-bridge, Columns, Beams and Slabs, Other Structural, and Other Non-structural. The images in the dataset have dimensions of 320 × 320 pixels. The challenges that follow along with this dataset are: (i) inconsistency in labeled ground truths. As shown in Fig. 3, two images with similar class (inside red box) are labelled differently. (ii) occlusion of small structures due to larger ones and (iii) position of camera viewpoints relative to the structure also plays a major role. Thus, accurate segmentation over such a dataset is a challenging process.

3.2 Benchmarking Methods

We compare the proposed StructureNet with the works of (Narazaki et al., 2017) which comprises of multi-scale CNNs and some relevant existing architectures such as ResNet (He et al., 2016). The work mentions two types of results, i.e., with scene information and without scene information. Both the results are considered for comparison study. Another work by (Yeum et al., 2019) has exploited Bridge Component Classification Dataset with the use of FCNs (Long et al., 2015) which is also taken into consideration. Three different architectures (FCN45, SegNet45, and Seg45-S) are tested over Bridge Component Classification Dataset with three different configurations (Naive, Parallel, and Sequential) in a recent work (Narazaki et al., 2020) by the same authors.

3.3 Implementation Details

The ResNet-50 (He et al., 2016) backbone is pre-trained on the Bridge Component Classification Dataset for 200 epochs. A batch-size of 8 is used during both pre-training and training procedures. Following the previous work (Narazaki et al., 2017), data augmentation of random cropping, random flipping, and random rotation along with center crop are applied to the Bridge Component Classification Dataset. Class weights are calculated using median frequency balancing and a weighted cross-entropy loss is used for training. The value of rank m, is taken equal to the number of classes in the dataset, in this case, 5. The learning rate is set as 10^{-4} along with polynomial decay. For optimization, Adam optimizer is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The models are trained for 500 epochs on Bridge component classification dataset and Make3D-S and for 1000 epochs on Aerial imagery dataset. The experiments are implemented



(i) Images (ii) Ground Truths

Figure 3: Actual images and corresponding ground truths can be compared with respect to labels in images shown.

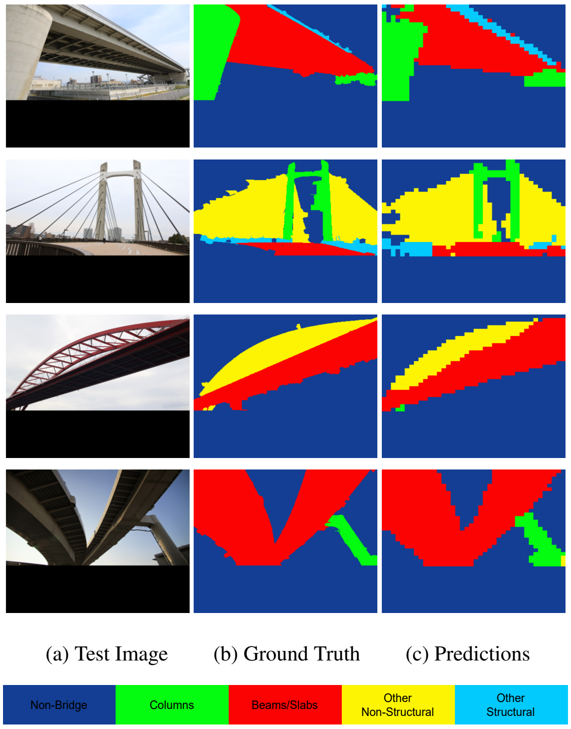

(a) Test Image (b) Ground Truth (c) Predictions

Figure 4: Segmentation results of StructureNet on Bridge Component Classification test set.

using Python keras api with Tensorflow backend on a system with Intel core i7 processor, 16 GB RAM, and NVIDIA GeForce RTX-2070 8GB GPU card.

3.4 Performance Metrics

For comparison with the previous benchmark methods, pixel accuracy (PA) over test set is evaluated which represents the percentage of correct pixel class prediction over the ground truth. We also calculate mean intersection over union (mIOU), where IOU given by (5) is calculated over each semantic class and then averaged.

$$IOU = \frac{TP}{TP+FN+FP} \quad (5)$$

where TP, FN and FP denote true positives, false negatives and false positives, respectively, which are obtained by comparing the ground truth labels and the predicted labels.

3.5 Results and Discussions

Comparison with Benchmarks: Table 1 summarizes the performance of the proposed StructureNet over the other existing benchmarks. Previous methods (Narazaki et al., 2017; Yeum et al., 2019) have presented the results in terms of pixel accuracy and the comparison is made with the latest work by Narazaki (Narazaki et al., 2020), where both pixel accuracy and mIOU are considered. StructureNet achieves pixel-wise accuracy of 89.08% with mean IOU of 57.46%. StructureNet thus performs better in terms of pixel accuracy as compared to existing works and outperforms (Narazaki et al., 2017) in terms of mIOU as well. As mentioned in Sec. 3.1, the inconsistent labeling of a few ground truths is an issue for performance saturation on testing data. For a 320 × 320 input image, the average processing time of StructureNet is 0.0567 seconds.

Table 1: Comparison with Benchmarks. Models with different configurations Naive (N), Parallel (P) and Sequential (S) are compared through pixel accuracy (PA) and mIOU.

Benchmarking Works	**mIOU(%)**	**PA(%)**
CNPT - N[1]	50.8	80.3
CPNT - Scene [1]	-	82.4
FCN45 [2]	-	82.3
FCN45 - N [3]	57.0	84.1
FCN45- P [3]	56.9	84.1
FCN45- S [3]	56.6	83.9
SegNet45- N [3]	54.5	82.3
SegNet45 - P [3]	55.2	82.9
SegNet45 - S [3]	55.2	82.9
SegNet45-S - N [3]	55.8	83.1
SegNet45-S - P [3]	55.9	83.3
SegNet45-S - S [3]	55.4	82.7
StructureNet	**57.46**	**89.08**

[1](Narazaki et al., 2017) [2](Yeum et al., 2019) [3](Narazaki et al., 2020)

The first benchmark on the dataset by (Narazaki et al., 2017) proposed naive component classifier (CPNT - N) and component classifier with scene information classifier (component classifier with scene information - Scene), where the results are presented in terms of pixel accuracy on ResNet23 model (mIOU score is taken from (Narazaki et al., 2020)). The benchmark from other work (Yeum et al., 2019) is taken for Bridge component classification dataset and results for other dataset are excluded. All the other results are various methods proposed in (Narazaki et al., 2020) out of which FCN45-N reports the best mIOU of 57.0% and best pixel accuracy of 84.1%. StructureNet outperforms the best performing metric by 0.46% and 4.98%, respectively. The fusion of the 3-D attention map and class contexts captures long range dependencies in the feature maps and thus elevating the performance of the architecture, thereby resulting in better accuracy.

Assessment on Other Datasets: To assess the ability of StructureNet to generalise on the semantic segmen-

Table 2: Assessment of StructureNet on two other datasets when compared with the backbone model.

Assessment	mIOU (%)	PA (%)
Make3D-S (Liu et al., 2010)		
Baseline ResNet-50	65.83	88.42
StructureNet	**74.52**	**93.65**
Aerial Imagery (Humans in the loop, 2020)		
Baseline ResNet-50	51.56	68.07
StructureNet	**55.86**	**70.22**

tation task, we evaluate the performance of the model on two other datasets namely Semantic Augmented Make3D (Liu et al., 2010; Saxena et al., 2005; Saxena et al., 2008) dataset (referred as Make3D-S), obtained for research and comparison evaluation purposes and Aerial imagery dataset (Humans in the loop, 2020), obtained under CC0 1.0 Universal (CC0 1.0) licensing. We also compare the performance of these models with respect to the backbone ResNet-50 architecture and show that the StructureNet outperforms the backbone architecture. Table 2 summarizes the assessment and shows that for both the datasets, StructureNet performs better than the backbone architecture (ResNet-50).

Make3D-S consists of 400 training images and 134 evaluation images belonging to 8 different classes. The input resolution of each image is 240 × 320. This dataset was selected because it captures outdoor scenes consisting of structures like buildings of different varieties. Aerial imagery of Dubai captured by MBRSC satellites and annotated with pixel-wise semantic segmentation in 6 classes (Humans in the loop, 2020). The total volume of the dataset is 72 images grouped into 6 larger tiles. We have separated two tiles (18 images) for evaluation and augmented the remaining tiles for training purposes. The input resolution for this dataset was kept to be 224 × 224. The augmentation used are similar to those reported for Bridge component classification dataset. This dataset can be considered challenging due to presence of satellite images along with less training examples. It is evident from Table 2 that our proposed StructureNet outperforms the baseline works for semantic segmentation task as well.

Results in Table 2 show a significant jump from the baseline ResNet-50 architecture, reason being the incorporation of fusion of class contexts and context attention module, which added more fine-grained feature extraction and thus improving the metric numbers. The inter-category relationships generated by the context attention module adds a deeper insight in feature extraction.

Ablation Study: To show the efficacy of fusion of class contexts and 3-D attention map, we test the network individually only when one of the two is present. For the first case, we remove the context attention module and directly combine soft predictions and class contexts. For the second case, we apply the context attention module to the output feature map (X), removing the class context branch. The results are summarized in Table 3. It can be noted that individually each module does not yield the optimum results, it's only when they are fused there is a significant improvement in the prediction performance.

Table 3: Ablation study to show efficacy of fusion of attention map with class contexts.

Condition	mIOU (%)	PA (%)
Only class context	45.80	78.37
Only Context attention (CA) module	39.40	70.52
Fusing class context and CA module	**57.46**	**89.08**

4 CONCLUSION

In this work, we have proposed a new architecture StructureNet to address the challenging task of structural component recognition. The novel architecture fuses class contexts with an attention map generated through tensor decomposition encoding information across spatial as well as the channel dimensions. Class contexts are rich with the knowledge encoding feature maps correlating to various classes. The attention map captures long-range dependencies without any loss in the channel dimensions. Thus, the fusion operation generates an information enriched feature map comprising inter-category relations as well as category-feature interactions. Experimental results on multiple benchmarking datasets show the superiority of the proposed architecture as compared with the existing methods.

REFERENCES

Bhattacharya, G., Mandal, B., and Puhan, N. B. (2021a). Interleaved deep artifacts-aware attention mechanism for concrete structural defect classification. *IEEE Trans. Image Process.*, 30:6957–6969.

Bhattacharya, G., Mandal, B., and Puhan, N. B. (2021b). Multi-deformation aware attention learning for concrete structural defect classification. *IEEE Trans. Circuits Syst. Video Technol.*, 31(9):3707–3713.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of ECCV*, pages 801–818.

Chen, W., Zhu, X., Sun, R., He, J., Li, R., Shen, X., and Yu, B. (2020). Tensor low-rank reconstruction for semantic segmentation. In *European Conference on Computer Vision*, pages 52–69. Springer.

Gao, Y. and Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):748–768.

Golparvar-Fard, M., Bohn, J., Teizer, J., Savarese, S., and Peña-Mora, F. (2011a). Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. *Automation in construction*, 20(8):1143–1155.

Golparvar-Fard, M., Pena-Mora, F., and Savarese, S. (2011b). Monitoring changes of 3d building elements from unordered photo collections. In *2011 ICCV Workshops*, pages 249–256. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on CVPR*, pages 770–778.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612.

Koch, C., Paal, S. G., Rashidi, A., Zhu, Z., König, M., and Brilakis, I. (2014). Achievements and challenges in machine vision-based inspection of large concrete structures. *Advances in Structural Engineering*, 17(3):303–318.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering*, 34(5):415–430.

Liu, B., Gould, S., and Koller, D. (2010). Single image depth estimation from predicted semantic labels. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1253–1260. IEEE.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on CVPR*, pages 3431–3440.

Lu, R., Brilakis, I., and Middleton, C. R. (2019). Detection of structural components in point clouds of existing rc bridges. *Computer-Aided Civil and Infrastructure Engineering*, 34(3):191–212.

Miao, X., Wang, J., Wang, Z., Sui, Q., Gao, Y., and Jiang, P. (2019). Automatic recognition of highway tunnel defects based on an improved u-net model. *IEEE Sensors Journal*, 19(23):11413–11423.

Narazaki, Y., Hoskere, V., Hoang, T. A., Fujino, Y., Sakurai, A., and Spencer Jr, B. F. (2020). Vision-based automated bridge component recognition with high-level scene consistency. *Computer-Aided Civil and Infrastructure Engineering*, 35(5):465–482.

Narazaki, Y., Hoskere, V., Hoang, T. A., and Spencer, B. F. (2017). Vision-based automated bridge component recognition integrated with high-level scene understanding. *13th International Workshop on Advanced Smart Materials and Smart Structures Technology (ANCRiSST)*.

Humans in the loop (2020). Semantic segmentation of aerial imagery v1. data retrieved on June 01, 2021 from , https://www.kaggle.com/humansintheloop/semantic-segmentation-of-aerial-imagery.

Saxena, A., Chung, S. H., Ng, A. Y., et al. (2005). Learning depth from single monocular images. In *NIPS*, volume 18, pages 1–8.

Saxena, A., Sun, M., and Ng, A. Y. (2008). Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840.

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on CVPR*, pages 7794–7803.

Yeum, C. M., Choi, J., and Dyke, S. J. (2019). Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure. *Structural Health Monitoring*, 18(3):675–689.

Zhang, F., Chen, Y., Li, Z., Hong, Z., Liu, J., Ma, F., Han, J., and Ding, E. (2019). Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE ICCV*, pages 6798–6807.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE CVPR*, pages 2881–2890.

Zhu, Z. and Brilakis, I. (2010). Concrete column recognition in images and videos. *Journal of computing in civil engineering*, 24(6):478–487.