

Deep Neural Network based Attention Model for Structural Component Recognition

Sangeeth Dev Sarangi^a and Bappaditya Mandal^b

*Keele University, Newcastle-under-Lyme ST5 5BG, United Kingdom
sangeethdev1995@gmail.com, b.mandal@keele.ac.uk*

Keywords: Synchronous attention, dual attention network, structural component recognition.

Abstract: The recognition of structural components from images/videos is a highly complex task because of the appearance of huge components and their extended existence alongside, which are relatively small components. The latter is frequently overestimated or overlooked by existing methodologies. For the purpose of automating bridge visual inspection efficiently, this research examines and aids vision-based automated bridge component recognition. In this work, we propose a novel deep neural network-based attention model (DNNAM) architecture, which comprises synchronous dual attention modules (SDAM) and residual modules to recognise structural components. These modules help us to extract local discriminative features from structural component images and classify different categories of bridge components. These innovative modules are constructed at the contextual level of information encoding across spatial and channel dimensions. Experimental results and ablation studies on benchmarking bridge components and semantic augmented datasets show that our proposed architecture outperforms current state-of-the-art methodologies for structural component recognition.


1 INTRODUCTION AND BACKGROUND WORKS


Critical infrastructures like bridges are extremely important during any environmental disaster because the movement of people and vehicles across their constructions is made possible. As a result, inspecting bridges and other comparable structures might be considered a high-priority and mission-critical task. Manual examination of structural problems necessitates lengthy but essential decision-making periods, delaying assessment and damage control, management, mitigation and recovery actions. Computer vision and machine learning based concrete structural health inspection/monitoring bring great benefits such as better safety and security for humans, non-contact, at a (long) distance, rapid, cheap cost and labour, and low interference with the regular functioning of infrastructures.

The technique of locating and identifying distinctive sections of a structure using structural component recognition is anticipated to be a crucial first step in the automated inspection/management of civil infrastructure. Recognition of structural components

also offers significant supporting data for the automated vision-based damage assessment of civil constructions. Information on structural components can be utilised to improve the consistency of automated damage detection algorithms by removing damage like patterns on items other than the structural component of interest. Additionally, knowledge of structural components is necessary for the safety assessment of the entire structure because, according to the majority of current structural inspection guidelines, damage, and the structural components on which the damage appears are jointly evaluated to determine the safety rating (Spencer et al., 2019).

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 is often missed by the existing methodologies. In the background literature, various categories of the bridge components are exploited at the contextual level of information encoding across spatial as well as channel dimensions and this is achieved by deploying the attention mechanism in the model. Our research aims to develop novel contextual information in the deep convolutional neural network coupled with an attention model (DNNAM) for the automatic recognition of civil structural components on image/video data.

^a  <https://orcid.org/0000-0002-8427-031X>

^b  <https://orcid.org/0000-0001-8417-1410>

1.1 Structural Component Recognition using Deep Learning

Deep learning based approaches for recognising structural components have recently attracted a lot of attention. One of the main uses of convolutional neural networks (CNNs) is image classification, which involves estimating a single representative label from an input image. In order to accurately identify the region of interest, Yeum et al. (Yeum et al., 2019) classified candidate image patches of the welded joints of a highway sign truss construction using CNNs. Gao and Mosalam used CNNs (Gao and Mosalam, 2018) for classifying input photos into the relevant structural component and damage categories. Based on the outputs of the final convolutional layer, the authors approximated the localisation of the target item. Algorithms for object detection can also be used to identify structural elements. By automatically drawing bounding boxes around them, (Liang, 2019) employed the faster R-CNN technique to recognise and localise bridge components. Another effective method for addressing structural component recognition issues is semantic segmentation (Narazaki et al., 2017; Narazaki et al., 2018; Narazaki et al., 2020). Semantic segmentation algorithms produce label maps with the same resolutions as the input images rather than drawing bounding boxes or estimating approximate object locations from per-image labels. This is especially useful for precisely detecting, localising and classifying complex-shaped structural components (Spencer et al., 2019).

1.2 Attention Mechanism

In order to obtain cutting-edge performance and industry usable solutions, an attention mechanism with the CNN framework has been developed for extracting local discriminative features. Such networks are initially employed for sequential data analysis (Vaswani et al., 2017) as well as general image classification (Wang et al., 2017). Park et al. (Park et al., 2018) and Woo et al. (Woo et al., 2018) looked into how channel and spatial attention modules affected feature discrimination. Attention modules have been used for a variety of tasks, including object detection (Zhu et al., 2018; Zhou et al., 2020), multi-label classification (Guo et al., 2019), saliency prediction (Wang and Shen, 2017) and pedestrian attribute recognition (Tan et al., 2019). Long-range content-based interaction is used as the main primitive in this mechanism to get rid of convolution's poor scaling feature for wider receptive fields. Surprisingly, Cordonnier et al. (Cordonnier et al., 2019)

research showed that the self-attention block's operation is comparable to that of convolutional layers, with the potential for the same or higher performance (Bhattacharya et al., 2021).

In more recent research, the StructureNet framework by Kaothalkar et al. (Kaothalkar et al., 2022), makes a contribution to the recognition of structural components by putting forth a novel architecture that combines class contexts and inter-category interactions discovered through the creation of a 3-D attention map. Contextual data is taken into account from a categorical perspective in class contexts, which is an aggregation of characteristics belonging to that class (Zhang et al., 2019). However, it lacks focus on specific portions of the structural components that might be crucial information for their recognition.

2 PROPOSED METHODOLOGY

The proposed DNNAM architecture consists of synchronous dual attention modules (SDAM) and residual modules, which together aid in the extraction of crucial discriminative characteristics from various scales to enhance the performance of both multi-target multi-class and single-class classification. Fig. 1 shows the proposed architecture.

2.1 Residual Module

Deep convolutional neural networks have made significant improvements to image categorization challenges. To address the vanishing gradient issue with a more complex architecture, ResNet (He et al., 2016) adds skip connections from the previous layers. Fig. 2 (a) shows the architectural components of our residual module. We stacked 3 convolution layers together and used skipped connection technique to establish an additional link between the input and output tensor. In our DNNAM model, we undertake feature extraction using filters of various kernel sizes employing a large number of residual blocks, ensuring a deeper network with the capacity to capture a wide range of structural component features. The synchronous dual attention module is sandwiched between residual modules to improve receptivity and the possibility of receiving salient local discriminative information.

2.2 Synchronous Dual Attention Module

The synchronous dual attention module fuses crucial attention operations, including self, spatial, and chan-

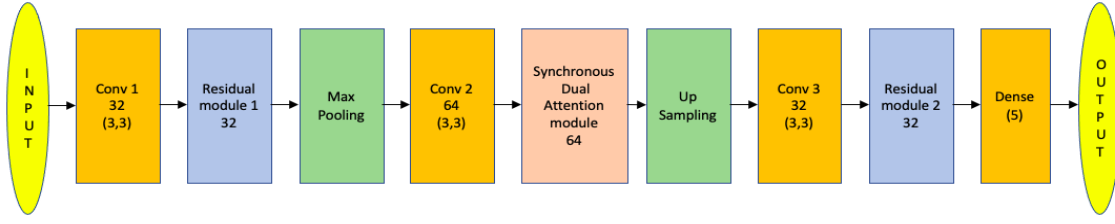


Figure 1: Proposed DNNAM Architecture model for structural component recognition. Here Conv block represents convolutional operation with the first number representing the number of filters and the next two numbers giving the filter dimension for each channel. Dense represents the dense layer, where the first number gives the number of nodes. The proposed synchronous dual attention module is composed of a batch of multi-feature attention module and a parallel excitation module. The number denoted in the synchronous dual attention module and residual module represents filter size.

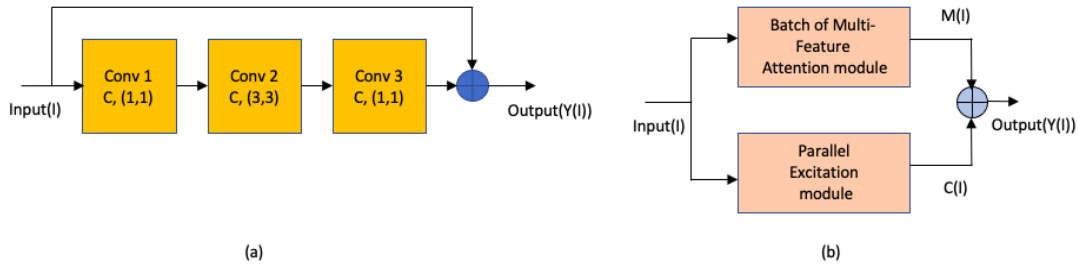


Figure 2: (a) Proposed residual module for DNNAM architecture. Here Conv block represents convolutional operation with the first number representing the number of filters and the next two numbers giving the filter dimension for each channel. (b) Proposed Synchronous dual attention module for DNNAM architecture is composed of a batch of multi-feature attention module and a parallel excitation module.

nel attention synchronously and aggregates their results to highlight discriminative features for multiple target structural component classes. The block diagram shown in Fig. 2 (b) is comprised of two modules: a batch of multi-feature attention module and a parallel excitation module. The batch of multi-feature attention module (BMFA) is created to encode several representations of extremely localised features, allowing the network to pick up on even the smallest component classes. The parallel excitation module (PEM) is used to synchronously highlight the significant aspects and lessen the impact of weak or unimportant features as it encodes the spatial and channel information for artefacts. To increase the impact of the synchronous dual attention module, the outputs of these two attention modules are fused together.

2.2.1 Batch of Multi-Feature Attention Module

We use a batch of multi-feature attention module to combine several representations of the highly localised parallel feature extraction process in order to encode relevant information from visually identical concrete structural components. To distinguish between non-bridge, columns, beams and slabs, other

structural, and other non-structural components, this module serves to encapsulate highly localised feature selection mechanisms. To ensure that the most significant aspects are attended to, the attention actions in this proposed module are repeated several times. To produce parallel non-linear projections in feature space, each attention module uses three dense layers to conduct synchronous computations. Here, the input (I), is taken into account along with its corresponding height, width and the number of channels. Then, the outputs T2 and T3 are multiplied elementally, a *SoftMax* function is used to create the attention mask, and T1 is multiplied with the attention mask to emphasise the critical features. The identity mapping is then carried out by the addition of an input tensor to the output. The output of the BMFA module, which aggregates attentive features from several representations, is produced by adding all three of the outputs generated by the attention operations att1, att2 and att3. Fig. 3 shows the batch of multi-feature attention module.

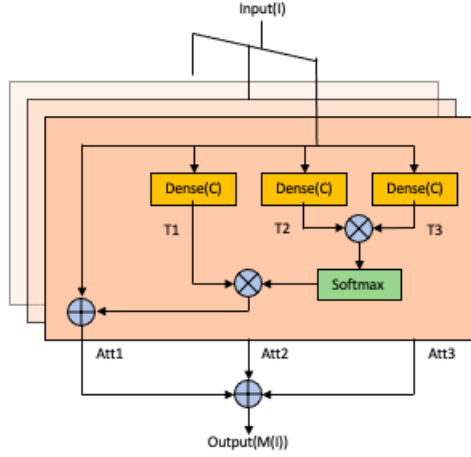


Figure 3: The Batch of multi-feature attention module is a combination of 3 self-attention layers. In each layer, the Input(I), together with its matching height, width, and channel count, are taken into consideration. The attention mask is then created using the SoftMax function, the outputs T2 and T3 are multiplied elementally, and T1 is multiplied with the attention mask to highlight the important features. After that, the identity mapping is completed by adding an input tensor to the output.

2.2.2 Parallel Excitation Module

In order to synchronously encode salient spatial and channel information, the convolution layer captures local spatial features across all of the channels (He et al., 2016; Hu et al., 2018). We must selectively draw attention to and suppress other aspects while emphasising the channel wise discriminative structural component features. The parallel excitation module analyses the key spatial and channel information individually to address these issues and enhance performance. This module includes a function that squeezes the input tensor’s spatial plane using global average pooling before stimulating it channel wise to get channel information. The module can automatically contain the global channel description thanks to the channel squeezing operation, which provides statistics for the entire image on a channel by channel basis. The following dense layers use non-linear adaptive re-calibration to extract discriminative channels with important features while also utilising contextual channel information. In order to create the channel attention feature map Ch(I) as illustrated in Fig. 4, the output of two dense layers is activated using the sigmoid function and multiplied with the input (I). This is carried out to emphasise the characteristics required for channel specific identification.

Similar to how the first portion of the parallel excitation module squeezes the channels, the second half of the module uses convolution blocks to capture the

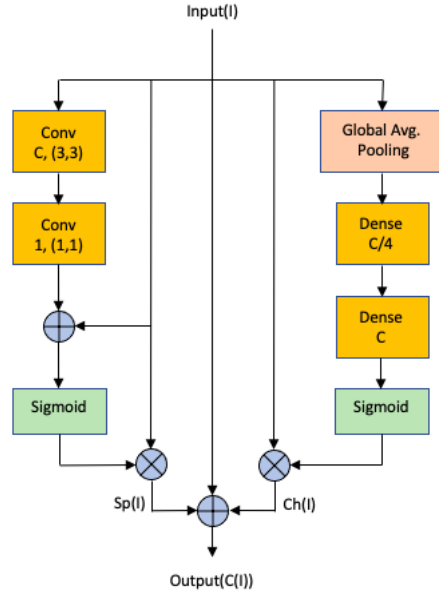


Figure 4: The Parallel Excitation Module examines the important Spatial, Sp(I) and Channel, Ch(I) Information Separately. The first half of the module uses convolution blocks to capture the common spatial features present in all channels. The second half of the module includes a function that squeezes the input tensor’s spatial plane using global average pooling before stimulating it channel-wise to get channel information.

common spatial features present in all channels. To avoid losing important contextual features after convolution, we inserted a skip connection with the input before proceeding to the sigmoid activation function. The recovered features are spatially excited, and the output is then multiplied by the input tensor to highlight the crucial spatial data Sp(I). In contrast to (Woo et al., 2018), where the spatial attention is carried out via average and max pooling operation, the global channel features are squeezed to extract salient spatial information to provide spatial statistics by decreasing the input through its channel dimension. Instead of employing 1×1 convolution directly for the aggregation of spatial information, another 3×3 convolution block is placed before it in order to aid in efficient feature extraction. Finally in this case, along with spatial, Sp(I) and channel, Ch(I) information the input is also added utilising a skip connection to prevent the loss of crucial discriminative information and to alleviate the vanishing gradient problems as shown in Fig. 4. Finally, we add the outputs of the BMFA module, M(I) and PEM module, C(I) to obtain the output of the SDAM module as shown in Fig. 2 (b).

3 Experimental Results and Discussions

3.1 Bridge Component Classification Dataset

We have evaluated our algorithms and compared them with the existing methods on the benchmarking dataset for bridge component classification provided by the authors (Narazaki et al., 2017; Narazaki et al., 2020; Narazaki et al., 2018), obtained for academic research and algorithmic evaluation comparison purposes. This dataset includes 1,563 bridge photos in a total of 320×320 pixel dimensions, out of which 1329 images are used for training and the remaining 234 images are used for testing. Each image is classified into one of five classes: Non-bridge, Columns (including piers), Beams and Slabs, Other Structural (trusses, arches, cables, abutments, extraordinary braces, amazing bearings, etc.), and Other Non-structural (fences, poles, etc.).

3.2 Implementation Details

In our DNNAM model, we employ a max pooling method that summarises the average and most activated presences of several features. To filter noisy activations in a lower layer of a convolution network, pooling abstracts activations in a receptive field into a single representative value. Spatial information within a receptive field is lost during pooling, even though it aids classification by maintaining only robust activations in upper layers. This information may be crucial for the exact localisation needed for semantic segmentation (Noh et al., 2015). We use unpooling layers in our model, which reverse the pooling process and reconstruct the initial size of activations, to address this issue. This unpooling method is especially helpful for re-creating the input object’s structure.

The batch size for training methods is 16. Following earlier research (Narazaki et al., 2017), the dataset has had random cropping, random flipping, and random rotation applied in addition to the centre crop. Weighted Binary cross entropy loss is used for training. Binary cross entropy is used to compare each of the projected probabilities to the actual class output, which can only be either 0 or 1. The score that penalises the probabilities based on how far they are from the predicted value is then calculated. This shows the degree to which the value resembles the actual. The number of classes in the dataset, in this case, 5 is used as the rank value. The setting for the learning rate, α is 0.001. In order to improve the DNNAM

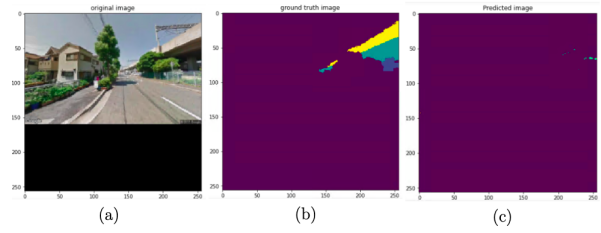


Figure 5: (a) Real image, (b) ground truth image and (c) Predicted image with more than 80% pixel accuracy. Simply put, a non-bridge component class comprised more than 80% of the original image in this situation. As a result, more than 80% of the pixels have been accurately identified. This example is meant to demonstrate that high pixel accuracy does not always imply superior segmentation skills.

model, the Adam optimizer is used for optimization strategy with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The models are trained using the Bridge component classification dataset over 100 epochs. The experiments are carried out on a system with an Intel(R) Core(TM) i7-9700 processor, 32 GB of RAM, and an NVIDIA GeForce RTX-2080 8GB GPU card utilising the Python Keras API and TensorFlow backend.

3.3 Performance Metrics

For comparison with the previous benchmarking methods, we are using the following performance matrices

3.3.1 Pixel accuracy

The percentage of accurate pixel class prediction compared to the ground truth is measured as pixel accuracy (PA) over the test set. It is the proportion of correctly classified pixels in our image. Now we can take into account a situation that we encountered throughout the project’s first stages to reveal the problems associated with this metric. Fig. 5 (a) and (b) show the real image and the ground truth image, respectively, that was given to the model. The model is attempting to recognise or segment structural components in the bridge image. Fig. 5 (c) depicts the prediction with more than 80% accuracy. That means, in this case, more than 80% of the original image belonged to one specific class (non-bridge component class). Therefore, more than 80% of the pixels are identified correctly, but the remaining 20% are inaccurate if the model assigns all pixels to that class. Because of this, even though our accuracy is great, the model is failing to accurately predict or identify the structural components of the image.

The example in Fig. 5 is intended to show that excellent segmentation skills are not always implied by great pixel accuracy. When our classes are severely

out of balance, one or more classes dominate the picture while other classes make up a very minor fraction of it. Unfortunately, this can't be disregarded because it can be seen in many real world data sets. As a result, we offer substitute metrics that are more effective in addressing this problem.

3.3.2 Mean Intersection-Over-Union

One of the most used metrics in semantic segmentation is the intersection-over-union (IOU), often known as the Jaccard Index. The IOU is an exceedingly effective metric that is relatively simple to use. The IOU can be defined as the area of union between the predicted segmentation and the ground truth divided by the area of overlap between the predicted segmentation and the ground truth. The mean IOU (mIOU) of the picture is determined for binary or multi-class segmentation by averaging the IOU of each class, where IOU is given by equation 1 and is calculated across each semantic class and then averaged.

$$IOU = \frac{TP}{TP + FN + FP} \quad (1)$$

Where TP , FN , and FP stand for true positives, false negatives and false positives, respectively. These terms are produced by comparing the actual labels with those that were predicted.

Now, using the same example as pixel accuracy, let's try to see why this metric is superior. For simplicity, let's assume that all structural elements belong to the same class. Let's compare the anticipated segmentation to the actual or ground truths. At first, we determine the IOU for the structural component. We consider the image's overall area to be 100 pixels, which focuses on the overlap of the structural components first. To check for overlapping component pixels, we can make the predicted segmentation on Fig. 5 (c) appear to be moved directly over the ground truth on Fig. 5 (b). There are 0 overlapping structural component pixels because the model does not identify any pixels as structural components. The pixels from both images that were classified as structural components are included in the union, but not the overlapped or intersected pixels. That is significantly less than the 80% pixel accuracy we predicted. It is evident that it gives a far more realistic image of how well our segmentation worked, though.

3.4 Comparison with Benchmarks

The performance of the suggested architecture compared to other benchmarks is summarised in Table 1. The results from earlier techniques (Narazaki et al., 2017; Yeum et al., 2019) are expressed in terms of

Table 1: Comparison with Pre-existing Benchmarks on the Bridge component classification dataset. We are considering mIOU as more important than PA, Perhaps because of a lack of high-quality ground creation, mIOU is better than PA on this dataset.

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 ⁴	57.46	89.08
DNNAM	65.94	82.85

¹(Narazaki et al., 2017) ²(Narazaki et al., 2020) ³(Yeum et al., 2019) ⁴(Kaothalkar et al., 2022)

mean IOU (mIOU), and a comparison is made with the most recent work by (Kaothalkar et al., 2022) and (Narazaki et al., 2020), which takes into account both mIOU and Pixel Accuracy (PA). Naive (N), Parallel (P), and Sequential (S) models with various configurations are also compared in the (Narazaki et al., 2020) paper.

The convergence curves produced during the DNNAM network's training are shown in Fig. 6. As time goes on, we could see that both training and testing, accuracy and mean IOU are steadily rising while loss is decreasing. Our proposed DNNAM model achieves a mean IOU of 65.94% with pixel wise accuracy of 82.85%. Thus, when compared to all of the previous research, our model surpasses them in terms of mean IOU and also outperforms (Narazaki et al., 2017; Narazaki et al., 2020; Yeum et al., 2019) in terms of pixel wise accuracy for the prior work.

The inconsistent labelling of a few ground truths on this dataset, also described in (Kaothalkar et al., 2022), is a problem for performance saturation on testing data. The average processing time of our developed model is 0.1001 seconds. Fig. 7 shows the segmentation results of our proposed DNNAM model on the Bridge Component Classification test set. To create the attention maps that help explain the proposed network's decision-making process, sample images from the dataset are applied to the DNNAM architecture are shown in Fig. 8. These attention maps assist the network to focus on these regions automatically by emphasising higher weightage on the

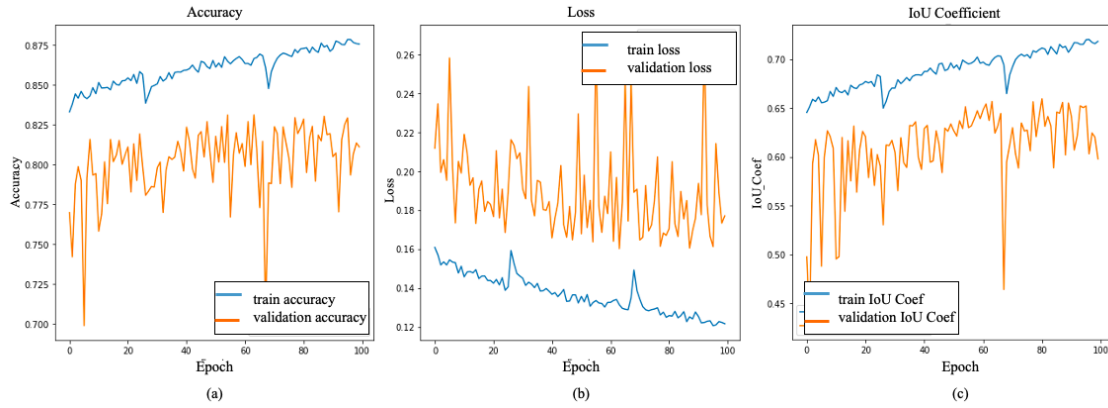


Figure 6: Performance curves generated during the training of the DNNAM network. Here blue and orange curves represent training data and validation data accuracy over the epochs, respectively. (a) Accuracy (b) Loss and (c) IOU coefficient.

structural component regions that help to extract robust discriminatory features for their classification.

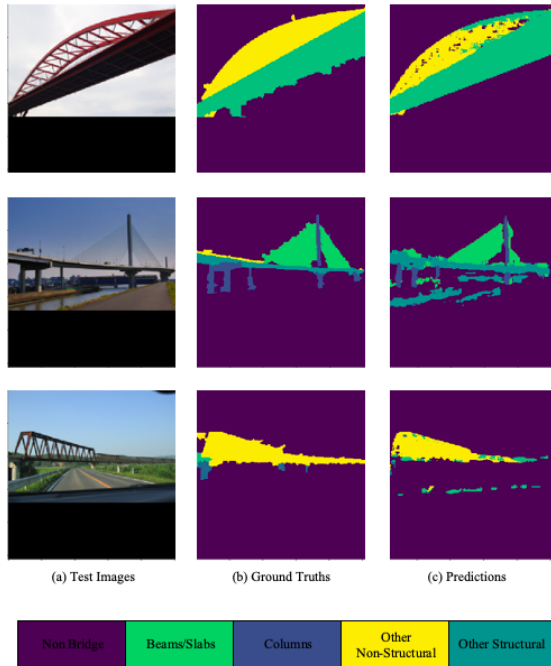


Figure 7: Segmentation results of our proposed DNNAM model. Our proposed DNNAM model yields a mean IOU of 65.94% with pixel wise accuracy of 82.85%.

The results are presented in terms of pixel accuracy on the ResNet23 model (mIOU score is taken from (Narazaki et al., 2020)), with the naive component classifier (CPNT - N) and component classifier with scene information (CPNT - Scene) being proposed in the first benchmark on the dataset by (Narazaki et al., 2017). The bridge component classification dataset uses the benchmark from another

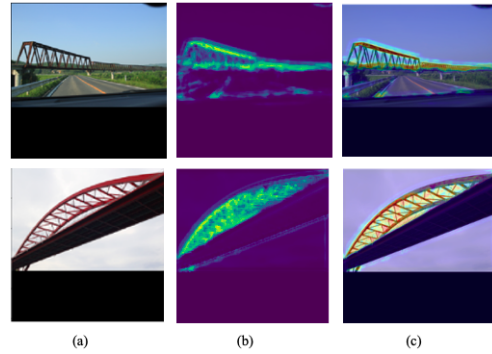


Figure 8: Attention maps obtained from the proposed DNNAM network for sample images from the Bridge component classification dataset. (a) Original images are followed by their respective (b) attention maps and (c) heatmaps are placed side-by-side.

study (Yeum et al., 2019). The majority of the results come from the various approaches reported by (Narazaki et al., 2020), among which FCN45-N reports the best mIOU of 57.0% and the best pixel accuracy of 84.1%. If we observe the comparison with pre-existing benchmark's Table 1, we could see most of Narazaki's work yielded an average of 55% in terms of Mean IOU values. Narazaki's earlier work is outperformed by recent work by (Kaothalkar et al., 2022), which has the mIOU (57.46%) and best pixel accuracy (89.08%) values. Mean IOU is considered the best performance metric and our proposed DNNAM model got mIOU (65.94%). Thus our proposed model DNNAM, in terms of mIOU performs 8.48% better than the previously established highest values. Also, we could notice that all the benchmarking works are generating pixel accuracy greater than 80% and our model also keeps that margin in the experimental results with 82.85% PA.

As we explained in the previous section perfor-

Table 2: Assessment of DNNAM on Semantic Augmented Make3D dataset when compared with the baseline model.

Assessment	mIOU (%)	PA (%)
Make3D-S (Liu et al., 2010)		
Baseline ResNet-50	65.83	88.42
DNNAM	73.42	87.47

mance metrics, there are certain disadvantages to utilising the performance metric pixel accuracy, hence in our study, we are emphasising on the performance metric mean IOU. In the early stages of our experimental tests, we found that while running the model for a few additional epochs allowed us to attain pixel accuracy that was higher than that of the previous works, the model was unable to correctly anticipate or identify the structural elements of the image. Through our analysis and studies finally, we are able to establish that when the synchronous dual attention module and residual modules are combined as we proposed, it can capture long-range dependencies in the feature maps, improving the architecture’s efficiency and accuracy.

Assessment on Semantic Augmented Make3D Dataset: We evaluate the model’s performance on an additional dataset, the Semantic Augmented Make3D (Liu et al., 2010; Saxena et al., 2005; Saxena et al., 2008) dataset, acquired for research and comparison evaluation purposes. 400 training images and 134 evaluation images from 8 separate classes make up the Make3D-S dataset. Each image has an input resolution of 240×320 . This dataset is chosen since it contains outdoor images of various types of buildings and structures. The evaluation is summarised in Table 2, which demonstrates that our suggested DNNAM outperforms the backbone architecture (ResNet-50) for the Make3D-S dataset and can be used for the semantic segmentation task as well. Due to the inclusion of residual module and contextual level information encoding across spatial and channel dimensions, which provides more fine-tuned feature extraction and hence improves the metric values, results in Table 2 show superior results for the additional dataset as well.

4 Ablation Studies

To demonstrate the effectiveness of the fusion of synchronous dual attention modules (SDAM) and residual modules, a series of ablation study experiments are conducted. In the first case, the SDAM module is eliminated and observed the results without an attention mechanism. To achieve better outcomes in the second case, we exclusively use residual modules and

vary the number of residual modules as well. The results are summarised in Tables 3 and 4. It should be emphasised that when utilised separately, each module does not produce the best results; only when they are combined do they perform significantly better at making predictions.

4.1 Ablation Experiments on SDAM module

With a thorough ablation investigation, we assess many aspects of SDAM and report the findings on the structural component dataset (Narazaki et al., 2017; Narazaki et al., 2020; Narazaki et al., 2018; Kaothalkar et al., 2022). Firstly, we train the network by retaining just residual modules, i.e., by omitting the primary core synchronous dual attention module. The lack of an attentive feature extraction process in this case results in a performance drop. Due to substantial changes and the presence of overlapped structural components, which might have been adequately distinguished by utilising the complete attention mechanism, we can notice lower performance on the structural component dataset.

Secondly, we attempt to evaluate the significance of various attention processes included in the proposed architecture. The parallel excitation module (PEM) is taken out while all the remaining modules are kept in order to investigate this. Due to the lack of a spatial-channel attention mechanism to encode discrete structural components, experimental results in Table 3 show a decline in performance for recognizing the structural components. The same operation is then repeated while omitting all of the batch of multi-feature attention module (BMFA), demonstrating that the structural component dataset has lower accuracy because it lacks the highly localised feature selection that is necessary to distinguish between structural components that overlap and have similar appearances.

Thirdly, while leaving the other modules in place, we take off one of the parallel excitation module sub-channel networks and spatial attention components at a time to examine the effects of specific attention operations on recognition performance. Table 3 shows the structural component recognition performance by removing one of the attention sub-networks, which shows a reduced performance in both scenarios and further reinforces the need for using both channel and spatial attention components.

Finally, we double the quantity of SDAM modules to track any performance changes. Similar performance is shown by the testing results, but a single SDAM module near the core produced better re-

Table 3: Ablation experiments on synchronous dual attention module. This study demonstrates decreasing performances for each of these scenarios in both performance parameters (mIOU and PA), emphasising the importance of the proposed architecture design’s performance.

Model Description	mIOU(%)	PA(%)
DNNAM without SDAM	59.59	80.10
Residual Module + Only BMFA	63.35	81.63
Residual Module + Only PEM	51.08	80.35
Only Spatial attention in PEM	59.24	78.66
Only Channel attention in PEM	58.03	79.08
BMFA with only 2 layers	61.76	81.31
BMFA with 4 layers	61.41	79.68
DNNAM with 2 SDAM	48.07	74.73
Self attention replacing BMFA	61.06	80.61
Self attention replacing PEM	38.55	80.23
Self attention replacing SDAM	61.12	80.14
DNNAM	65.94	82.85

Table 4: Ablation experiments on the residual module. This study shows decreased performances for each of these scenarios in performance parameters (mIOU and PA), further highlighting the significance of the performance of the proposed architectural design.

Model Description	mIOU(%)	PA(%)
DNNAM without Residual Module	58.41	77.85
Only 1 Residual Module	56.86	78.41
Using 2 Residual Module	60.08	78.59
Using 4 Residual Module	62.30	79.52
DNNAM	65.94	82.85

sults with minimal running times. Table 3 shows decreased performances for each of these scenarios, further highlighting the significance of the proposed modules.

4.2 Ablation Experiments on Residual Module

We conduct a broad range of experiments on the residual module to assess the impact of various changes and the results are summarised in Table 4. Firstly, the proposed network is trained using only

the synchronous dual attention modules and not any residual modules, which use fewer parameters. However, as seen in Table 4, the classification performance suffers when the residual module is absent.

A deeper network results from the extraction of local features across all channels with the assistance of residual modules. To capture multi-scale feature representation, each residual block combines features from all prior responses. It demonstrates the effectiveness of a deeper network, such as a residual module, in obtaining reliable features for the identification of overlapping structural components. Moreover, these networks emphasise accurate spatial feature estimation.

Additionally, the network can alleviate the vanishing gradient problem with generalised performance owing to the identity mappings across the residual units. These traits of the residual module contribute to the structural component dataset’s increased performance. Furthermore, we tested altering the number of residual modules in the design to see how performance changed and discovered that keeping 3 will result in the optimal performance matrices with the least amount of running time. We obtain decreased results in Table 4 for each of these scenarios, further highlighting the significance of the proposed modules.

5 Conclusion and Future Work

In this work, we address the challenges involved in structural component recognition, a crucial step in the inspection process and management of civil infrastructures. To improve classification performance with fewer parameters, our proposed DNNAM architecture is built using synchronous dual attention modules and residual modules, which are used to extract robust salient discriminative features from multiple scales. Numerous experimental results and ablation studies on benchmarking datasets demonstrate the superiority of the proposed DNNAM architecture as compared to other current state-of-the-art approaches, particularly in terms of the performance metric mean IOU for multi-target and multi-class classification problems. The classification of additional kinds of structural components may benefit from the success of the proposed DNNAM architecture.

The structural component recognition studied in this research is a crucial building block for autonomous robot navigation in post-earthquake/natural or other calamity disaster affected areas. The proposed DNNAM system can be used in conjunction with unmanned aerial vehicles (UAVs) to quickly

identify structural elements and can accurately detect deterioration, anticipate how long a structure will last and monitor large concrete structures.

REFERENCES

- Bhattacharya, G., Puhan, N. B., and Mandal, B. (2021). Stand-alone composite attention network for concrete structural defect classification. *IEEE Transactions on Artificial Intelligence*, 3(2):265–274.
- Cordonnier, J.-B., Loukas, A., and Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*.
- 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.
- Guo, H., Zheng, K., Fan, X., Yu, H., and Wang, S. (2019). Visual attention consistency under image transforms for multi-label image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 729–739.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.
- Kaothalkar, A., Mandal, B., and Puhan, N. B. (2022). Structu-net: Deep context attention learning for structural component recognition. In Farinella, G. M., Radeva, P., and Bouatouch, K., editors, *Proceedings of the 17th International Joint Conference on Computer Vision*, pages 567–573. SCITEPRESS.
- 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.
- 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. *arXiv preprint arXiv:1805.06041*.
- Narazaki, Y., Hoskere, V., Hoang, T. A., and Spencer Jr, B. F. (2018). Automated vision-based bridge component extraction using multiscale convolutional neural networks. *arXiv preprint arXiv:1805.06042*.
- Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528.
- Park, J., Woo, S., Lee, J.-Y., and Kweon, I. S. (2018). Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*.
- Saxena, A., Chung, S., and Ng, A. (2005). Learning depth from single monocular images. *Advances in neural information processing systems*, 18.
- 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.
- Spencer, B. F., Hoskere, V., and Narazaki, Y. (2019). Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, 5(2):199–222.
- Tan, Z., Yang, Y., Wan, J., Hang, H., Guo, G., and Li, S. Z. (2019). Attention-based pedestrian attribute analysis. *IEEE transactions on image processing*, 28(12):6126–6140.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., and Tang, X. (2017). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.
- Wang, W. and Shen, J. (2017). Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378.
- Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.
- 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). Acfnnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807.
- Zhou, S., Wang, J., Zhang, J., Wang, L., Huang, D., Du, S., and Zheng, N. (2020). Hierarchical u-shape attention network for salient object detection. *IEEE Transactions on Image Processing*, 29:8417–8428.
- Zhu, Y., Zhao, C., Guo, H., Wang, J., Zhao, X., and Lu, H. (2018). Attention coupler: Fully convolutional attention coupling network for object detection. *IEEE Transactions on Image Processing*, 28(1):113–126.