

Kernelized dynamic convolution routing in spatial and channel interaction for attentive concrete defect recognition

Gaurab Bhattacharya^a, N. B. Puhan^{a,*} and Bappaditya Mandal^b

^aSchool of Electrical Sciences, Indian Institute of Technology, Bhubaneswar, Pin: 752050, India

^bSchool of Computing and Mathematics, Keele University, Newcastle ST5 5BG, United Kingdom.

ARTICLE INFO

Keywords:

Kernel salient feature encoder
spatial-channel attention
concrete structural defect
convolutional neural network
multi-target multi-class classification

ABSTRACT

Image/video based defect recognition is a crucial task in the automation of visual inspection of concrete structures. Although some progress has been made to automatically recognize the defects in concrete structural images, significant challenges still exist. In this work, we propose a deep convolutional neural network architecture that embeds novel spatial-channel interaction based concurrent attention (SCA) for multi-target, multi-class recognition of concrete defects. SCA module stems from the novel kernel salient feature (KSF) encoder that captures higher-order features with robust discriminative representation of concrete defects. KSF encoder incorporates kernelized convolution followed by dynamic routing operation to be used as the primary building block. The proposed CSDNet architecture is able to apportion higher weightage in the defective regions while suppressing the large background area (healthy region), thereby improving the recognition performance of the overlapping defects in concrete structures. Experimental results and ablation study on three large benchmark datasets show the consistent superiority of our proposed network as compared to the current state-of-the-art methodologies. This end-to-end trainable architecture can be augmented with unmanned aerial vehicles (UAVs) to monitor the health of massive infrastructures to leverage the high concrete defect recognition performance.

1. Introduction

Understanding of the structural stability, risk assessment and planning have become a pivotal issue to ensure safety and well-being of concrete infrastructures and associated human lives. Automatic visual inspection of concrete structures necessitates the development of next-generation efficient solutions for defect recognition. However, several real-world artifacts vitiate the performance of conventional convolutional neural network (CNN) architectures for concrete defect recognition [18]. This includes, firstly, the presence of artifacts such as poster remains, marking, graffiti, shadows, potholes, etc. and wide variations in surface texture and color change of the defect appearance, resulting in the deterioration of the recognition performance. Secondly, the presence of overlapping defect classes (such as a spallation is often accompanied by exposed bar) further exacerbates the problems of multi-target defect recognition task. Thirdly, the variations in aspect ratio, scale, resolution and defect appearance also lead to the degradation in performance. Thereby, innovative vision based solutions are required to be augmented with UAVs to perform superior monitoring of structural defects to ensure safety and risk assessment.


In recent years, convolutional neural network (CNN) has shown unprecedented development to recognize salient patterns on images. Several state-of-the-art methodologies [16, 25, 27, 10, 12] have shown exemplary performance on various computer vision challenges. Inspired from the development, multiple research initiatives have been undertaken

in [13, 14, 17, 31] to obtain data-driven solutions for structural health monitoring. However, [14, 17] dealt with cracks as the only defect subset, excluding other defect categories for structural damage classification. Several other works in this category used random structured forest [24], AlexNet and VGG models using SDNET-2018 [6] and CSSC datasets [33]. Similarly, the authors in [13, 31] considered non overlapping multi-class defects which do not address the real-world issue of overlapping structural defects. The literature search also indicates that these methods do not address the complicated nature of structural health monitoring problem involving overlapping defect classes (such as spallation leading to exposed bar, which often leads to/co-exist with corrosion). Recently, [4, 18] analyzed overlapping multi-class defects in CODEBRIM dataset using attention augmented CNN and reinforcement learning, respectively.

The traditional feature selection mechanism of CNNs can be augmented with visual attention to highlight relevant local discriminative regions [11, 19, 30, 32]. The recent research focuses on large-scale image classification using residual attention [30], channel and spatial attention [19, 32] and self-attention [3]. The use of attention mechanism for classification of concrete structural defects, however, is limited. Also, most of these methods usually consider one type of attention mechanism, such as residual attention [4, 30], channel attention [11, 35] or self attention [3] or they consider multiple attention modules without enhancing features extracted using *Conv* layers, such as [19, 32].

In this paper, we propose CSDNet: a deep attention network which concurrently consolidates spatial-channel relationship for overlapping multi-class concrete defect recognition. The primary ingredient in CSDNet is the novel kernel salient feature (KSF) encoder which helps to address wide

*Corresponding author

 gb14@iitbbs.ac.in (G. Bhattacharya); nbpuhan@iitbbs.ac.in (N.B. Puhan); b.mandal@keele.ac.uk (B. Mandal)

ORCID(s): 0000-0003-0244-2390 (G. Bhattacharya);
0000-0001-5932-1579 (N.B. Puhan); 0000-0001-8417-1410 (B. Mandal)

variations in scale and area of appearance using the dynamic routing strategy between *Conv* layers. In KSF encoder, we have proposed the use of *kernelized convolution* which approximates the complex defect features with more discriminative information. Moreover, the dynamic convolution routing enables the network to obtain viewpoint-equivariant information for aggregating variations in defect appearance. In KSF encoder, we have incorporated *kernelized convolution* operation to approximate the complex feature aggregation. Furthermore, we propose the novel spatial-channel attention (SCA) module to concurrently consolidate discriminative cues across the channels and spatial planes by highlighting regions of interest. The SCA modules provide focused attention on defective regions and enable the network to address the appearance and surface texture variations for visually similar defect instances. Contrary to [4, 30, 35, 3], SCA module uses multiple attention masks to enhance the performance. Also, contrary to multi-attention networks [19, 32], SCA module uses KSF encoders to enhance the features extracted using *Conv* layers with kernelized convolution and dynamic routing strategy. Another contribution in the architecture is the self-attention mask (SAM) in the transition layers which helps to extract minute local patterns for overlapping defect recognition.

Below our major contributions are summarized:

- We propose CSDNet architecture which embeds KSF encoder as the primary ingredient for nonlinear feature extraction, SCA module for attention mechanism along with self attention mask (SAM) encoding highly localized features to improve the recognition performance of the multi-target multi-class concrete overlapping defects, alleviating the problems of variations due to image acquisition and presence of unwanted inclusions/artifacts.
- We propose SCA module that unifies the benefit of novel attention network and *kernelized convolution* to resolve multiple challenges. Firstly, SCA module separately investigates spatial and channel information to highlight discriminative features to distinguish variations in surface textures and unwanted inclusions. Secondly, the use of KSF encoders inside SCA module captures variation cues in scale, area of appearance and orientation of the defect images.
- We demonstrate the superiority of our CSDNet architecture on three large benchmark concrete defect datasets by outperforming recognition performance of the state-of-the-art methods.

The rest of this paper is organized as follows: Section 2 illustrates the proposed CSDNet architecture, KSF encoder and SCA module, Section 3 presents the performance of CSDNet and comparisons with state-of-the-art methods for three datasets, Section 4 performs an extensive analysis and ablation study to investigate the impact of individual modules. Finally, Section 5 concludes the paper.

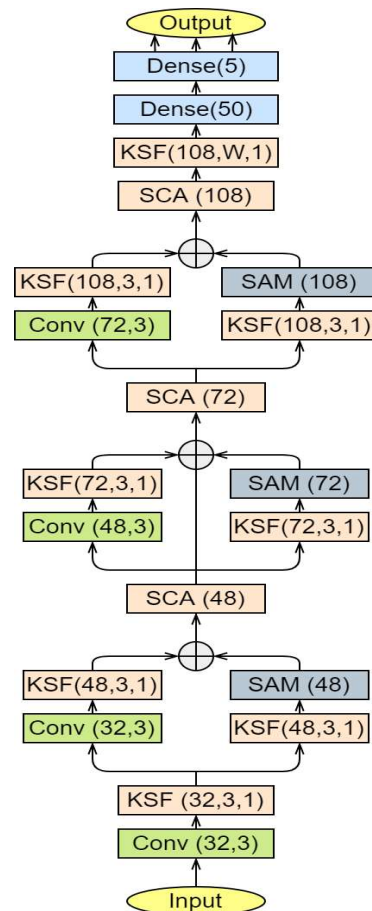


Figure 1: Block diagram of CSDNet. Here, Conv(A,B) represents convolution operation using A filters of size (B,B) and Dense(C) represents the dense operation with C nodes. Parameter specifications of KSF and SAM are given in the module description.

2. Proposed Architecture

Our proposed CSDNet architecture stems from the novel kernel salient feature (KSF) encoder and spatial-channel attention (SCA) modules with designed combination of KSF encoders and self-attention mask (SAM) in transition layers. The proposed CSDNet architecture is given in Figure 1.

2.1. Kernel Salient Feature Encoder

Feature maps generated with higher order non-linearity during convolution operation incorporate more discrimination ability than linear classifiers with point-wise non-linearity added with ReLU activation [5]. Wang *et al.* in [29] proved that careful selection of kernel functions generates patch-wise non-linearity and hence enables the network to obtain more expressible visual features for classification. To encode crucial information from multi-target images with large variations in scale, illumination and resolution, we propose kernel salient feature encoder (KSF) with its novel feature description strategy. KSF encoders encapsulate the variations of local features by leveraging the dynamic convolution routing with *kernelized convolution* operation to provide generalized representation of overlapping defect fea-

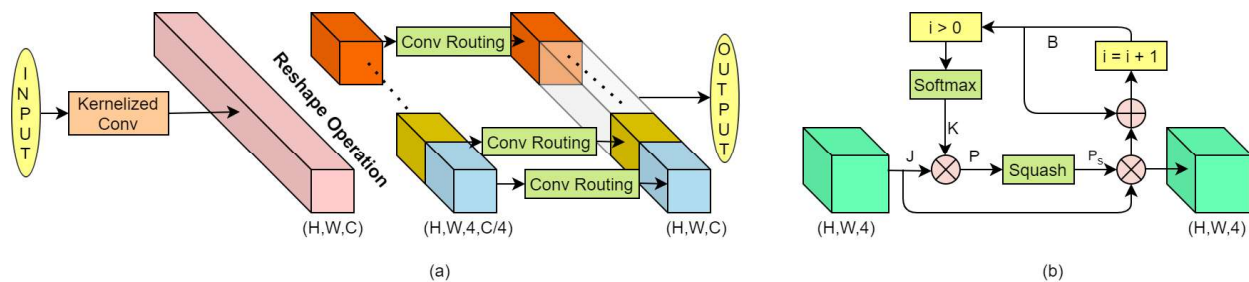


Figure 2: Proposed Kernel Salient Feature (KSF) encoder in (a) with the dynamic convolution routing mechanism in (b). In (a), the kernelized convolution operation is performed using gaussian and polynomial kernel and C channels. In (b), operation on one block after reshaping is portrayed. In Figure 1, the KSF encoders are represented as $KSF(A,B,C)$ representing the channels, kernel size and strides of A , (B,B) and C , respectively.

tures.

Let us represent the input to the KSF as $\phi(H_{in}, W_{in}, C_{in})$, where (H_{in}, W_{in}) represent the spatial dimension of the input and C_{in} is the number of channels. In Figures 1 and 3, the KSF encoders are represented as $KSF(C, k, s)$, which describes the kernelized convolution operation with C number of filters with spatial dimension (k, k) and stride s . This kernelized convolution operation results in the intermediate output $I(H, W, C)$, such that,

$$I_{i,j,k} = \sum_p \sum_q \sum_r \psi(\phi(i-p, j-q, r)) \psi(F_k(p, q, r)). \quad (1)$$

Here F_k and ψ denote the k_{th} filter and the kernel function, respectively.

After the *kernelized convolution*, we perform dynamic convolution routing operation that enables aggregation of viewpoint-equivariant information to encapsulate variations in the defect appearance [21]. This routing operation enables each kernelized convolution block to predict the next layer outcome with squash function. Here, the highly localized regions estimate the pose, orientation and precise location of defects during the iteration. In higher levels, these part information can be aggregated to obtain the whole information about the defects, helping the network to investigate individual overlapping instances.

Figure 2 (a) and (b) describe the block diagram of KSF and the dynamic convolution routing, respectively. Before the routing operation, we reshape the response from the *kernelized convolution* I to consider four consecutive spatial planes at a time to enclose similar localized information by eliminating redundancy.

$$I(H, W, C) \rightarrow \text{Reshape} \rightarrow J(H, W, 4, C/4). \quad (2)$$

Unlike [21], we predict one output tensor for each such block, since our experiments reflect that the prediction for multiple tensors for each block does not benefit the performance; however it takes a toll in memory requirement and training time. For the routing operation, two parameters are initialized, namely i to be 0 as the loop counter and the B_m logit values for each block as zero, $m \in [1, C/4]$. Then, we compute the corresponding coefficient value K_m from B_m using the softmax function. For the first iteration (i.e. when

$i = 0$), K_m value is ignored.

$$K_m = \text{softmax}(B_m). \quad (3)$$

These coefficient values are then element-wise multiplied with the corresponding input blocks to obtain the prediction P . This prediction value P passes through the squash function, as in (4) to obtain the probability of occurrence of an entity by limiting the vector length from 0 to 1.

$$P_s = \frac{\|P\|^2}{1 + \|P\|^2} \frac{P}{\|P\|}. \quad (4)$$

Followed by (4), the logit value B_m is updated for all $m \in [1, C/4]$ and the loop counter i is increased by one. After performing the routing operations for $i = 3$ times, we obtain the output feature map from the KSF encoder Out with dimension $I(H, W, C)$ after concatenating $(C/4)$ blocks each of dimension $(H, W, 4)$.

$$\begin{aligned} B_m &= P_s * J_m + B_{m-1}, \\ i &= i + 1, \\ Out_m &= P_s * J_m. \end{aligned} \quad (5)$$

Here, J_m and Out_m represent the m_{th} part of reshaped output and the KSF encoder output, respectively each having dimension $(H,W,4)$, where $m \in [1, C/4]$. Both the input (J) and output (Out) can be written as follows:

$$\begin{aligned} J &= \text{Concat}(J_1, J_2, J_3, \dots, J_{C/4}), \\ Out &= \text{Concat}(Out_1, Out_2, Out_3, \dots, Out_{C/4}). \end{aligned} \quad (6)$$

The KSF encoder helps to encode large variations in scale, translation and rotation in our CSDNet architecture. The features extracted from non-linear kernelized convolutions encoding multi-target information help to generate salient image descriptors for subsequent SCA modules.

2.2. Spatial-Channel Attention Module

During the investigation of concrete structural defects, we encounter variations in surface texture, unwanted inclusions, defect pose, orientation, area of appearance, scale, etc. To alleviate these challenging variations, spatial and channel information are explored for attention modeling in the image classification task [19, 32]. In our work, we propose spatial-channel attention module (SCA) to simultaneously alleviate

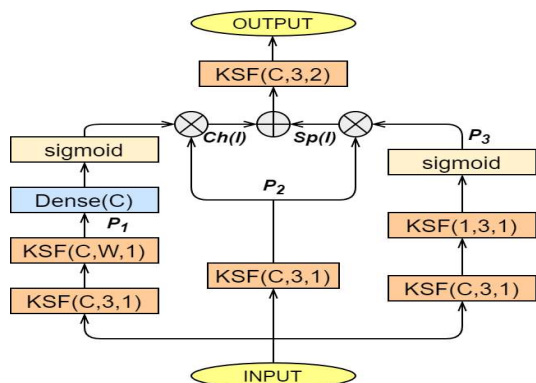


Figure 3: Proposed SCA module. Here $KSF(A,B,C)$ represents convolution operation using A filters with kernel size (B,B) and stride as C . In CSDNet, SCA modules are denoted as $SCA(X)$, where X represents the number of channels for convolution operation.

both the challenging defect appearance variation and multi-target defect recognition in concrete images. The block diagram of SCA module is depicted in Figure 3.

In SCA, we perform channel attention by incorporating the squeezing operation which embeds global channel information. Unlike [11, 19, 32] which uses global average and max pooling for squeezing operation, we propose the use of KSF encoders to extract global descriptors considering the $filter_{size}$ to be same as the spatial dimension of the input tensor $I(W, W, C)$ with a novel strategic configuration. Our experiments in ablation study (Section 4) demonstrate the benefit of using KSF instead of pooling for global channel information embedding.

$$I(W, W, C) \longrightarrow KSF(C_{out}, 3, 1) \longrightarrow KSF(C_{out}, W, 1) \longrightarrow P_1, \quad (7)$$

$$I(W, W, C) \longrightarrow KSF(C_{out}, 3, 1) \longrightarrow P_2.$$

Here C_{out} values are given in Figure 1. Interaction between P_1 and the subsequent dense layer enables the network to obtain the channel weights and these weights upon sigmoid activation, get multiplied with P_2 to highlight crucial channel information $Ch(I)$.

$$Ch(I) = \text{sigmoid}(W * P_1) * P_2. \quad (8)$$

Another part of SCA module performs spatial attention by squeezing across channels to obtain global spatial embedding which undergoes spatial excitation by sigmoid activation to generate spatial attention map P_3 . Salient spatial regions of P_2 are then highlighted by multiplying with P_3 , obtaining crucial spatial information $Sp(I)$.

$$I(W, W, C) \longrightarrow KSF(C_{out}, 3, 1) \longrightarrow KSF(1, 1, 1) \longrightarrow \text{sigmoid} \longrightarrow P_3, \quad (9)$$

$$Sp(I) = P_3 * P_2.$$

Here $KSF(1, 1, 1)$ operates as similar to 1×1 convolution operation, however the presence of *kernelized* convolution and dynamic routing result in a more expressible global spatial description. Finally, the channel and spatial attention responses $Ch(I)$ and $Sp(I)$ are combined to provide the output

of SCA,

$$Out(I) = Ch(I) + Sp(I). \quad (10)$$

Advantages of SCA module are as follows; firstly, SCA module addresses the variations in pose, orientation, surface texture and presence of unwanted inclusions by concurrently examining the channel and spatial information across the tensor while suppressing the redundant information. Secondly, we use the novel KSF block as the primary primitive to extract spatial and channel features. Hence, overlapping defect classes can be jointly encoded using the complex feature aggregation in conjunction with attention mechanism and kernelized feature encoding operation.

2.3. CSDNet Configuration

In this subsection, we describe the CSDNet configuration which addresses the relevant challenges and considers the use of *kernelized* salient feature encoding operation to obtain a viewpoint-equivariant feature extraction mechanism with interleaved SCA module for discriminative feature selection. The use of multiple SCA modules enable gradual fine-tuning of features for complex defect images, as shown in Figure 1.

The transition blocks are designed incorporating two parallel paths to aggregate fine-grained features for subsequent attention operation. For this, we use one *Conv* layer followed by a KSF block in one path, and one KSF block followed by a self-attention attention mask (SAM) in the other. SAM blocks encode highly localized minute features following these steps: at first, three concurrent dense operations are performed considering input $I(W, W, C)$. Then the outcome of these dense operations T_1 and T_2 are multiplied to generate self-attention mask after the softmax operation. This mask highlights the discriminative regions on the third branch output T_3 with the identity mapping from input, obtaining the output $SAM(I)$.

$$I(W, W, C) \rightarrow \text{Dense}(C) \rightarrow T_i(W, W, C) \quad \forall i \in [1, 3],$$

$$SAM(I) = T_3 * \text{softmax}(T_1 * T_2) + I(W, W, C). \quad (11)$$

In order to obtain global channel description before dense layers in CSDNet, we propose the use of KSF keeping the convolution filter dimension as same as the input tensor spatial dimension, as done for channel attention. Moreover, to reduce the spatial dimension throughout the network for finer feature extraction, KSF blocks are used with stride as 2 with filter size of 3×3 in *Conv* layers.

3. Experimental Results

In this section, we analyze the performance of CSDNet architecture on four large concrete structure defect datasets: CODEBRIM [18], SDNET-2018 [6], Concrete crack defect [36] and Concrete Structure Spalling and Crack database (CSSC) [33], where CODEBRIM contains overlapping five-class defect images and the other three datasets contain crack and spalling defects. In all experiments, we have demonstrated the higher performance by our novel architectures with comparisons drawn with the state-of-the-art methods

Table 1

Multi-target recognition results on CODEBRIM dataset for the proposed CSDNet. Here four input image dimensions (96, 128, 160, 192) are considered with mini-batch sizes of 16 and 32. All experiments are conducted incorporating the gaussian and the polynomial kernels.

Input Image Size	Gaussian Kernel						Polynomial Kernel					
	Batch size: 16			Batch size: 32			Batch size: 16			Batch size: 32		
	Train acc.	Validation acc.	Test acc.	Train acc.	Validation acc.	Test acc.	Train acc.	Validation acc.	Test acc.	Train acc.	Validation acc.	Test acc.
96	99.98	90.89	87.18	99.94	89.19	86.86	99.97	90.48	86.49	99.95	88.98	85.28
128	99.98	91.42	87.34	99.93	90.74	87.02	99.98	91.16	87.18	99.92	90.26	86.71
160	99.94	90.13	86.71	99.89	91.05	86.55	99.94	89.27	85.91	99.91	87.59	85.12
192	99.92	89.25	85.76	99.87	88.28	85.44	99.93	89.06	85.28	99.88	86.74	84.65

Table 2

Comparison of the recognition performance (%) of state-of-the-art results with the proposed CSDNet on CODEBRIM dataset.

Architecture	Multi-target accuracy		Parameters in million
	Best validation	Best val-test	
AlexNet [16]	63.05	66.98	57.02
VGG-A [25]	64.93	70.45	128.79
VGG-D [25]	64.00	70.61	134.28
T-CNN [1]	64.30	67.93	58.60
Densenet-121 [12]	65.56	70.77	11.50
WRN-28-4 [34]	52.51	57.19	5.84
ENAS-1 [20]	65.47	70.78	3.41
ENAS-2 [20]	64.53	68.91	2.71
ENAS-3 [20]	64.38	68.75	1.70
MetaQNN-1 [2]	66.02	68.56	4.53
MetaQNN-2 [2]	65.20	67.45	1.22
MetaQNN-3 [2]	64.93	72.19	2.88
AlexNet* [16]	70.26	68.46	57.02
VGG-A* [25]	76.49	74.82	128.79
VGG-D* [25]	77.52	75.21	134.28
ResNet-50* [10]	77.68	76.79	25.6
Densenet-121* [12]	79.72	78.84	11.50
SE-ResNet-50 [11]	72.86	70.71	28.13
CBAM [32]	80.63	78.63	11.78
ResNeSt [35]	75.92	73.46	27.50
MDAL [4]	86.15	84.29	10.43
CSDNet with polynomial kernel	91.16	87.18	2.11
CSDNet with gaussian kernel	91.42	87.34	2.11

* Denotes ImageNet-pretrained.

following the original implementation protocol.

3.1. Implementation setup

For the implementation of CSDNet, we have used Keras API with Tensorflow 1.14.0 at the backend. For all three datasets, we have used stochastic gradient descent optimizer with initial learning rate 0.001, momentum of 0.9 for 200 epochs during training. For all experiments, we use the gaussian kernel with variance = 1 and the polynomial kernel with degree = 3, bias = 1. The training is performed using Margin loss [22]. During dynamic routing, number of routing iterations is kept as three. We train the network on a system with 16 GB RAM on Intel Core i7 processor powered by GeForce RTX-2070 8 GB GPU card.

3.2. Performance on CODEBRIM dataset

Concrete DEfect BRidge IMage (CODEBRIM) dataset is presently the most complex state-of-the-art overlapping defect image dataset containing five defect classes: crack, spallation, efflorescence, exposed bars and corrosion [18],

obtained for non-commercial research and educational purpose. This dataset was constructed by investigating 30 unique bridges with varying weather condition, surface texture and degree of damage with image acquisition procedure involving variations in resolution, illumination, scale and aspect ratio. This results in the generation of 5354 defect images and 2506 background images, where number of images containing crack, spallation, efflorescence, exposed bars and corrosion are 2507, 1898, 833, 1507 and 1559, respectively.

For evaluation, we follow the original implementation protocol of choosing training, validation and test images [18] considering the classification to be correct if existence of all the defect classes are correctly recognized. To observe the best possible image spatial dimension, mini-batch size and kernel functions, we evaluate the performance of CSDNet with four different image dimensions, two batch sizes and two kernel functions. The experimental results are noted in Table 1. From this, we observe that the CSDNet architecture gives best performance with input image dimension of $128 \times 128 \times 3$ and batch size of 16 which uses the gaussian kernel at KSF encoders. Hence, we have considered this image dimension, kernel and batch size for all the future experiments.

The comparison of the proposed CSDNet with the state-of-the-art methods are reported in Table 2. This table includes the comparison with traditional CNN architectures (such as AlexNet [16], VGG [25], *etc*), visual-attention based CNN architectures (such as SE-ResNet-50 [11] and ResNeSt [35]) and reinforcement learning methods (such as ENAS [20] and Meta-QNN [2]). Here, we observe that except the newly proposed MDAL network [4], CSDNet architecture demonstrates significant performance improvement (87.34% test accuracy compared to 78.84% by ImageNet-pretrained DenseNet-121 [12]). CSDNet outperforms the recently proposed MDAL architecture by more than 3% improvement in test accuracy with $5\times$ lesser parameters. Our proposed CSDNet takes approximately 52.39 seconds of average training time in each epoch using mini-batch size of 16 on CODEBRIM dataset. For testing, each image from CODEBRIM dataset takes 0.173 ms. Unlike [4], CSDNet extracts attentive features in channel and spatial axes using KSF and SCA modules to aggregate multi-scale information without explicitly exploring features in multiple scales, thereby reducing the number of parameters significantly. Moreover, rather than exploring fine-grained localized features across multi-

Table 3

Comparison of crack defect recognition accuracy (%) of CSDNet with the state-of-the-art methods for concrete bridge deck, wall and pavement on SDNET-2018 dataset.

Model Description	Bridge image result			Wall image result			Pavement image result		
	Train accuracy	Validation accuracy	Test accuracy	Train accuracy	Validation accuracy	Test accuracy	Train accuracy	Validation accuracy	Test accuracy
Alexnet [6]	98.25	94.43	91.86	97.52	90.26	87.88	98.48	97.15	95.22
VGG-16 [26]	98.55	86.45	85.19	97.25	88.24	84.29	99.14	89.79	88.56
VGG-16 [†] [26]	96.35	90.15	87.76	94.55	91.24	86.29	97.59	94.37	89.33
Alexnet* [6]	98.78	95.84	92.07	98.34	92.59	90.16	99.06	97.54	95.85
VGG-16* [26]	94.22	90.25	88.59	93.89	91.86	87.46	97.58	93.45	92.13
Fine-tuned VGG-16 ^{†*} [26]	98.59	94.36	92.79	97.28	93.88	91.48	99.12	97.59	96.78
ResNet-50* [10]	98.49	95.88	93.15	97.96	95.08	92.36	99.15	98.11	97.28
Densenet-121* [12]	98.85	96.03	93.58	98.12	97.49	93.19	99.46	98.27	97.59
SE-ResNet-50 [11]	98.96	96.25	94.18	98.36	97.58	93.79	99.32	98.29	97.36
CBAM [32]	99.16	97.35	94.13	98.56	97.84	94.03	99.19	97.89	97.46
ResNeSt [35]	99.03	96.32	93.96	98.41	97.46	94.22	99.19	98.35	97.61
MDAL network [4]	99.91	98.56	94.35	98.79	98.12	93.76	99.94	98.92	98.26
CSDNet with polynomial kernel	99.93	98.74	94.61	98.91	98.23	94.15	99.95	98.97	98.29
CSDNet with gaussian kernel	99.93	98.81	94.76	98.92	98.36	94.63	99.96	99.12	98.34

[†] Denotes image augmentation.

* Denotes ImageNet-pretrained.

Table 4

Single-class defect recognition accuracy (%) on CODEBRIM.

Type of defect	Accuracy
Crack	91.86
Spallation	88.92
Efflorescence	89.57
Exposed bars	96.15
Corrosion	87.23

ple scales [4], CSDNet explores localized information across channel and spatial domains to improve classification performance.

To further analyze the network's ability to understand individual classes, we conduct another two experiments: firstly, we investigate the recognition accuracy of CSDNet for individual five defect classes in CODEBRIM dataset; secondly, we examine the ability of CSDNet to detect lesser than five classes correctly. From the results in Table 4, we observe that exposed bars can be more accurately recognized whereas corrosion stain has the lowest tendency to get classified correctly. The result in Table 5 illustrates that the network can correctly classify up to at least three classes with very high accuracy, however, performance drops while considering four or more number of defect classes.

3.3. Performance on SDNET-2018 Dataset

SDNET-2018 dataset [6], obtained under Attribution 4.0 International licensing, contains 8484 images of crack defects and 47608 background images captured from concrete bridge deck, wall and pavement surfaces, each having dimension $256 \times 256 \times 3$. These images were collected by generating patches from 230 image samples with variations in scale, crack widths, shadows and background noise. The experimental results in Table 3 depict the benefit of proposed CSDNet architecture over existing methods for all three types of structures, where CSDNet achieves test accuracies (in %) of 94.76, 94.63 and 98.34 for bridge deck, wall and pavement images, respectively, using the gaussian kernel inside KSF block, compared to 94.35, 93.76 and 98.26, respectively, us-

Table 5

Multi-target recognition ability on CODEBRIM dataset.

Number of classes correctly classified	Test accuracy
At least one	100
At least two	98.92
At least three	98.09
At least four	91.74

Table 6

Comparison of the performance of the proposed CSDNet with state-of-the-art methods in terms of recognition accuracy (%) on concrete crack image dataset.

Model name	Training accuracy (%)	validation accuracy (%)	Testing accuracy (%)
Deep CNN with adaptive threshold [9]	99.75	99.16	98.70
AlexNet* [16]	95.40	94.85	94.15
VGG-16* [25]	96.15	95.90	94.25
ResNet-50* [8]	98.40	98.00	97.80
DenseNet-121* [12]	99.20	98.40	98.25
SE-ResNet-50 [11]	99.75	99.70	99.60
CBAM [32]	99.81	99.47	99.12
ResNeSt [35]	99.80	99.65	99.55
MDAL network [4]	99.99	99.84	99.81
CSDNet with polynomial kernel	99.99	99.88	99.83
CSDNet with gaussian kernel	99.99	99.89	99.85

* Denotes ImageNet-pretrained.

ing MDAL network [4].

3.4. Performance on Concrete Crack Image Dataset

Concrete Crack Image dataset contains 20000 crack images and 20000 background images with dimension $227 \times 227 \times 3$ having prominent crack defects and less background clutter and unwanted inclusions [36], obtained under a Creative Commons Attribution 4.0 International license. Following the implementation protocol in [4, 8, 9], we have considered 32000 training, 4000 validation and 4000 test images while using equal number of crack and non-crack images for each subset. In comparison with state-of-the-art methods in Table 6, we can observe that our CSDNet outperforms all the existing methods by obtaining 99.85% and 99.83% recognition accuracy for polynomial and gaussian kernels, respectively, compared to 99.81% by MDAL network [4].

3.5. Performance on CSSC Dataset

The CSSC dataset [33] is composed of 15,000 crack images and 19,924 spallation images of dimension $130 \times 130 \times 3$.

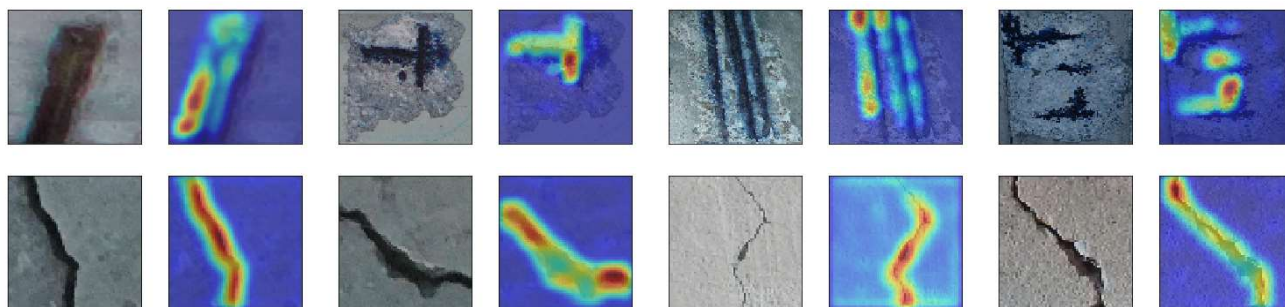


Figure 4: Generated attention maps from three datasets using CSDNet architecture. In attention maps, red color indicates highest attention whereas blue color represents lowest attention. First row (CODEBRIM), from left to right: (a) exposed iron bar with heavy spallation, (b) spallation with corroded bar, (c) exposed corroded bar, (d) Corroded bar with efflorescence and spallation. Second row from left to right: (e)-(f) images from Concrete Crack Image dataset, (g)-(h) images from SDNET-2018 dataset.

3, with training data consists of 24,941 images and test data of 9,983 images. Following the implementation protocol in [33], we have reshaped the images by $128 \times 128 \times 3$ to be used by our proposed method. For the comparison in Table 7, we have considered the state-of-the-art method as considered for other datasets. From these results, we observe that the multi-attention feature extraction and the routing mechanism results in improved performance, outperforming all the state-of-the-art methods. The proposed CSDNet obtains 96.53% and 96.65% test accuracy for polynomial and gaussian kernels, respectively, compared to 95.31% by MDAL network [4].

3.6. Performance on CIFAR-10 and CIFAR-100 datasets

The CIFAR-10 and CIFAR-100 datasets consist of 60,000 images of dimension $28 \times 28 \times 3$ having 10 and 100 classes, respectively. To validate the performance of our proposed CSDNet architecture for these standard classification datasets, we have compared our results with several state-of-the-art methods, such as ResNet with ELU activation [23], DenseNet [12], Wide residual network [34], Efficient Net [28] and the transformer-based methods such vision transformer [7] and big transformer [15]. From the results in Table 8, we observe that our method gives comparable performance to the recently proposed ViT [7], although it outperforms all other baselines by a good margin.

4. Analysis and Discussions

4.1. Attention maps

Sample images from three concrete defect datasets are used to generate attention maps by revisiting the response provided by the final KSF encoder to visualize the defect feature localization capability of CSDNet. These attention maps enable visual illustration of feature selection mechanism which consolidate crucial discriminative features from the images, thereby highlighting the relevant regions while diminishing the regions with redundant information.

In Figure 4, we observe that the CSDNet architecture localizes defect regions within the image by aggregating robust

Table 7

Comparison of the performance of the proposed CSDNet with state-of-the-art methods in terms of recognition accuracy (%) on CSSC dataset.

Model name	Training accuracy(%)	Testing accuracy (%)
AlexNet [16]	95.75	87.96
VGG-16 [25]	95.86	89.42
ResNet-50 [8]	96.89	92.45
DenseNet-121 [12]	98.55	93.15
SE-ResNet-50 [11]	99.25	95.10
CBAM [32]	98.95	94.73
ResNeSt [35]	99.10	95.05
MDAL network [4]	99.25	95.31
CSDNet with polynomial kernel	99.45	96.53
CSDNet with gaussian kernel	99.40	96.65

Table 8

Comparison of the performance of the proposed CSDNet with state-of-the-art methods on CIFAR-10 and CIFAR-100 datasets.

Model Name	Test accuracy (%)	
	CIFAR-10	CIFAR-100
ResNet + ELU [23]	94.40	73.51
DenseNet-121 [12]	96.54	82.82
WRN [34]	96.11	81.15
SENet [11]	97.88	84.59
EfficientNet-B7 [28]	98.79	91.72
BiT-M [15]	98.91	92.17
ViT-L [7]	99.42	93.92
CSDNet with polynomial kernel	99.39	93.46
CSDNet with gaussian kernel	99.45	93.95

features using KSF and SCA modules. It is able to apportion higher weightage to the defective regions while suppressing the large background area (healthy region) of the image plane, thereby improving the recognition rates for this challenging task with reduced number of network parameters (as shown in Table 2). Multi-target overlapping defects from CODEBRIM dataset are localized using CSDNet in Figure 4 (a)-(d). CSDNet also accurately localizes the crack regions from concrete crack defect and SDNET-2018 dataset images in Figure 4 (e)-(h) and (i)-(l), respectively.

Table 9

Ablation study on KSF and SCA modules in the CSDNet architecture on CODEBRIM and concrete crack image dataset (Recognition accuracy in %).

Model Description	CODEBRIM Dataset			Concrete Crack Image Dataset		
	Training accuracy	Validation accuracy	Testing accuracy	Training accuracy	Validation accuracy	Testing accuracy
Using linear kernel in KSF	97.86	88.42	84.81	99.75	99.62	99.50
Replacing KSF with <i>Conv</i> layers	96.15	85.24	83.86	99.24	98.61	96.25
Only channel attention in SCA	97.49	87.86	84.49	98.95	98.19	98.05
Only spatial attention in SCA	97.36	88.09	84.65	98.85	98.26	97.92
1 SCA + 1 transition block	95.34	83.25	79.11	98.24	97.56	95.98
Using global average pooling	98.79	90.26	85.13	99.83	99.75	99.63
CSDNet with polynomial kernel	99.98	91.16	87.18	99.99	99.88	99.83
CSDNet with gaussian kernel	99.98	91.42	87.34	99.99	99.89	99.85

4.2. Ablation Study

To demonstrate the significance of the individual modules, we conduct an extensive ablation study by replacing or removing several blocks for all three datasets. The experimental results are noted in Tables 9 and 10. Firstly, to understand the impact of nonlinear kernels in convolution operation, we replace the nonlinear kernels with traditional *Conv* layers, i.e. using linear kernels. Results of this experiment reveal a downfall in recognition performance due to the absence of non-linearity incorporation with convolution, which would enable CSDNet to recognize complex visual representations.

Secondly, we replace the KSF encoders with *Conv* layers to understand the impact of *kernelized* convolution in KSF. We expected the modified network to perform poorly due to the absence of dynamic routing operation with *kernelized* convolution to obtain multi-scale viewpoint-equivariant complex features from the defect images. The results in Tables 9 and 10 corroborate our assumption with performance degradation.

Thirdly, to analyze the impact of both the channel and spatial attention in SCA, we conduct separate experiments by keeping one of them at a time. As noted in Tables 9 and 10, we first analyze the impact of spatial attention by dropping the channel attention block, and vice versa. For both cases, the recognition performance degrades due to the lack of aggregation of both channel and spatial information.

Fourthly, we modify our network keeping only one SCA module and one transition block to understand the necessity of stacking multiple SCA modules for robust feature extraction. From Tables 9 and 10, we observe a significant degradation of recognition performance due to non-existence of finer features which results in high misclassification error for similar-looking overlapping defect classes.

Then, we replace the final KSF block of CSDNet and final KSF blocks inside the channel attention part of SCA modules with global average pooling to check the benefit of using KSF encoders for extracting global channel information. Experimental results in Tables 9 and 10 indicate that KSF blocks can lead to better performance by encoding global channel information.

5. Conclusion and Future Work

In this paper, we have addressed the challenges of recognizing overlapping concrete defects and proposed a novel deep architecture that embeds spatial and channel interactions. The benefit of our proposed CSDNet architecture is tri-fold, firstly, it uses novel KSF encoder which incorporates kernelized convolution with higher-order nonlinearity followed by dynamic routing operation for robust feature selection. Secondly, our proposed SCA module enables the network to perform spatial and channel interactions with minute localized feature selection to improve the recognition performance. Thirdly, the interleaved concurrent attention framework embedding both SAM and SCA modules is able to focus towards the defective regions (for both single and multi-target defect classes), while suppressing the large background (healthy) region. Experimental results and ablation study on three large benchmark datasets show the efficacy of our proposed architecture with 5× reduction in the number of network parameters as compared to the current state-of-the-art. In terms of future work, we hope to investigate the impact of our network for other kinds of concrete and steel structural defects.

References

- [1] Andriarczyk, V., Whelan, P.F., 2016. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters* 84, 63–69.
- [2] Baker, B., Gupta, O., Naik, N., Raskar, R., 2016. Designing neural network architectures using reinforcement learning, in: *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–18.
- [3] Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V., 2019. Attention augmented convolutional networks, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3286–3295.
- [4] Bhattacharya, G., Mandal, B., Puhan, N.B., 2020. Multi-deformation aware attention learning for concrete structural defect classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.
- [5] Cui, Y., Zhou, F., Wang, J., Liu, X., Lin, Y., Belongie, S., 2017. Kernel pooling for convolutional neural networks, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 2921–2930.
- [6] Dorafshan, S., Thomas, R.J., Maguire, M., 2018. Sdnet-2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in brief* 21, 1664–1668.
- [7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly,

Table 10

Ablation study on KSF and SCA modules in the CSDNet architecture on SDNET-2018 dataset (Recognition accuracy in %).

Model Description	Bridge image result			Wall image result			Pavement image result		
	Training accuracy	Validation accuracy	Testing accuracy	Training accuracy	Validation accuracy	Testing accuracy	Training accuracy	Validation accuracy	Testing accuracy
Using linear kernel in KSF	98.45	96.37	93.25	96.49	95.12	93.28	98.97	97.38	96.78
Replacing KSF with <i>Conv</i> layers	98.08	95.77	92.84	95.14	93.49	92.87	98.35	97.05	96.28
Only channel attention in SCA	98.36	96.26	93.15	96.91	94.35	92.97	98.48	97.28	96.59
Only spatial attention in SCA	98.29	96.09	93.11	97.05	94.87	93.17	98.77	97.14	96.84
1 SCA + 1 transition block	97.39	93.46	90.72	94.92	92.67	90.85	96.81	93.27	92.16
Using global average pooling	99.03	97.86	94.15	98.27	96.98	93.78	99.21	98.63	97.52
CSDNet with polynomial kernel	99.93	98.74	94.61	98.91	98.23	94.15	99.95	98.97	98.29
CSDNet with gaussian kernel	99.93	98.81	94.76	98.92	98.36	94.63	99.96	99.12	98.34

- S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 .
- [8] Dung, C.V., et al., 2019. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction* 99, 52–58.
- [9] Fan, R., Bocus, M.J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., Liu, M., 2019. Road crack detection using deep convolutional neural network and adaptive thresholding, in: *IEEE Intelligent Vehicles Symposium (IV)*, pp. 474–479.
- [10] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
- [11] Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141.
- [12] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708.
- [13] Kim, B., Cho, S., 2018. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. *Sensors* 18, 3452.
- [14] Kim, H., Ahn, E., Shin, M., Sim, S.H., 2019. Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring* 18, 725–738.
- [15] Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N., 2020. Big transfer (bit): General visual representation learning, in: *European conference on computer vision*, Springer. pp. 491–507.
- [16] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems (NIPS)*, pp. 1097–1105.
- [17] Li, Y., Li, H., Wang, H., 2018. Pixel-wise crack detection using deep local pattern predictor for robot application. *Sensors* 18, 3042.
- [18] Mundt, M., Majumder, S., Murali, S., Panetsos, P., Ramesh, V., 2019. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11196–11205.
- [19] Park, J., Woo, S., Lee, J.Y., Kweon, I.S., 2018. Bam: Bottleneck attention module. arXiv preprint arXiv:1807.06514 .
- [20] Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J., 2018. Efficient neural architecture search via parameter sharing, in: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4095–4104.
- [21] Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., Rodrigo, R., 2019. Deepcaps: Going deeper with capsule networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10717–10725.
- [22] Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules, in: *Advances in neural information processing systems (NIPS)*, pp. 3856–3866.
- [23] Shah, A., Kadam, E., Shah, H., Shinde, S., Shingade, S., 2016. Deep residual networks with exponential linear unit, in: *Proceedings of the Third International Symposium on Computer Vision and the Internet*, pp. 59–65.
- [24] Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems* 17, 3434–3445.
- [25] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .
- [26] Słoński, M., 2019. A comparison of deep convolutional neural networks for image-based detection of concrete surface cracks. *Computer Assisted Methods in Engineering and Science* 26, 105–112.
- [27] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.
- [28] Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: *International conference on machine learning*, PMLR. pp. 6105–6114.
- [29] Wang, C., Yang, J., Xie, L., Yuan, J., 2019. Kervolutional neural networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31–40.
- [30] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X., 2017. Residual attention network for image classification, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164.
- [31] Wang, N., Zhao, Q., Li, S., Zhao, X., Zhao, P., 2018. Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering* 33, 1073–1089.
- [32] Woo, S., Park, J., Lee, J.Y., So Kweon, I., 2018. Cbam: Convolutional block attention module, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19.
- [33] Yang, L., Li, B., Li, W., Liu, Z., Yang, G., Xiao, J., 2017. Deep concrete inspection using unmanned aerial vehicle towards cssc database, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 24–28.
- [34] Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. arXiv preprint arXiv:1605.07146 .
- [35] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., et al., 2020. Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 .
- [36] Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J., 2016. Road crack detection using deep convolutional neural network, in: *Proceedings of the IEEE international conference on image processing (ICIP)*, pp. 3708–3712.