

# Shadow detection from images using fuzzy logic and PCPerturNet

Jyotismita Chaki

School of Information Technology and Engineering,  
Vellore Institute of Technology, Vellore, India

## Correspondence

Jyotismita Chaki, School of Information Technology  
and Engineering, Vellore Institute of Technology,  
Vellore Campus, Tiruvalam Rd, Katpadi, Vellore,  
Tamil Nadu 632014, India.  
Email: [jyotismita.c@gmail.com](mailto:jyotismita.c@gmail.com)

## Abstract

Shadow detection is a challenging and essential task for interpreting the scene. Regardless of encouraging findings from current Deep Learning (DL) approaches used for shadow detection, the methods are also dealing with inconsistent situations where the visual representation of non-shadow and shadow regions is equivalent. In this article, a DL based approach is introduced for image pixel-level shadow detection. The proposed CNN-based approach, pattern conserver convolutional neural network (PCPerturNet) profits from a new design where shadow features are defined utilizing an effective skip-connection mapping arrangement. To make PCPerturNet robust from the change in brightness and contrast, several perturbed instances are generated by using a fuzzy-logic based method to train the system. Also, five types of augmentations are applied to images during training to make the system robust from the change in scale, orientation and flip. PCPerturNet derives and conserves shadow patterns in manifold layers and uses those layers progressively in several units to produce the shadow mask. The output of the proposed method is tested on two freely accessible databases and one self-created database where the accuracy rate obtained is 96.4%, 96.8%, and 89.4% which indicates that the proposed method outperforms the other shadow detection approaches used in the literature.

## 1 | INTRODUCTION

Shadow is a natural radiance effect, induced by the obstruction of light by other objects, arising in variations in intensity and colour of local areas. Shadow is an unavoidable occurrence in maximum images, and so grappling with the shadow is an inevitable phase in many image processing assignments. Irrespective of the role where shadow areas are either examined or ignored, shadow holds some useful information about the structure and properties of the scene components and their connection with one another. Using this type of information could have a major influence on the efficacy and consistency of any image processing technique in the existence of shadows. If detected and perceived accurately, shadows may provide details about the orientation of the light and scene layout. Although in some situations, shadows offer valuable information, like the relative position of the source element, they create problems in machine vision applications like object detection, segmentation, and counting objects. For example, in many cases, the object detection system detects shadows together with

people in a video sequence (rather than detecting just the person) [1, 2]. As a consequence, the approach is unsuccessful to detect a particular individual in video sequence when that person's shadow is connected to other objects in the scene. In addition to this, if there are several objects in a scene (like some people close to one another, or several vehicles on a road), the shadow of one can obstruct other object parts and decrease the efficiency of the detection system [3]. Needless to say, shadows may have also been a source of unpredictability. Shadow areas appear to have alike characteristics to obstructions. This may misguide object recognition techniques. The shadow of trees, clouds, man-created structures (such as walkways) on the roads etc. may make it increasingly challenging to detect roads in satellite images [4–6] and it affects the accuracy and the quality of such application results. Object recognition, edge extraction, change direction, image matching etc. needs shadow-free satellite images. Therefore, it is essential to detect shadow from these type of images before further processing.

For all the steps being taken throughout many years to solve the issue of image shadow detection, it remains an ongoing and

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *IET Image Processing* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

very motivating research topic. One explanation of this is that there is almost no background knowledge about the source of light or the barriers on the scene. Shadow characteristics differ from one image to another depending on the composition of the scene, leaving researchers, in several cases, empty-handed. Early shadow detection procedures are typically based on graphical models that deal with primarily light chromaticity or lighting invariant hypotheses and utilize hand-crafted attributes such as illumination indicators, hue, and others. Also, most of the shadow detection networks redesigned to detect local patches. The association between various patches of the image is often segmented as images are decomposed into patches. The patch segmentation is typically centred on the sliding pane, irrespective of the conceptual meaning of the image. Resultantly, certain possible relationships within the original image are removed by the features that are derived from patches. In order to create a finished shadow area detection result, more optimization for local patch results is required. The network can only learn the internal relationships between various parts of the images by utilizing the entire original image as the network input. With the advent of deep convolutionary neural networks (CNN), researchers are using these networks in recent years to conduct several machine vision tasks. The issue of shadow detection perhaps can be now best handled utilizing modern deep learning (DL) techniques.

Inspired by the above findings, and in order to avoid the reasoning procedures, a DL approach is proposed for the detection of the entire shadow regions automatically. The original image is the input of the proposed network, and the output is the probability distribution representation for the shadow regions. The proposed approach profits from pattern conserver CNN trained by image perturbation (PCPerturNet). PCPerturNet can detect “self and cast shadow” (henceforth will be referred to as “shadow”). The shadow can be divided into penumbra (soft) and umbra (hard) region. From the penumbra region, the texture of the background information can be retained. But as the umbra region is too dark, so it is very difficult to differentiate the shadow part from the dark objects present. PCPerturNet intends not only to detect global and local shadow perspectives but also to conserve these patterns with the purpose that they may not be lost in this dense and deep framework. As a consequence, this model’s well-trained weights, which have previously collected characteristics of shadow, will reliably determine pixels of shadow in any unobserved images. To deal with variants in shadow area characteristics, the use of perturbed image instances is proposed. Perturbed instances are slightly altered forms of the original image that varies in brightness and contrast which may be unnoticeable to the human visual system. Perturbed instances have an important contribution to the advancement of more reliable deep neural networks and researchers have shed light on the process of precisely understanding the different features of these network inputs [7, 8]. Inspired by the abovementioned evidence, perturbed instances are used to train the system along with the original training images. Thus, PCPerturNet is trained with more varied instances, which induce learning different types of shadow fea-

tures as opposed to the training with only the original training images.

Here a framework is built to produce a range of perturbed instances for the shadow detection model (PCPerturNet) by using fuzzy rules. These freshly created image instances are modified forms of the original images. Before train the system, the perturbed instances are developed in the pre-processing phase of the shadow detection model. Consequently, the training phase of PCPerturNet is pretty straightforward. The proposed perturbed data creator may be utilized for other programs with minimal data of training.

The major two contributions of this article are as described in the following:

1. Development of a new perturbed instance creator by varying the brightness and contrast of the original image which is based on fuzzy logic that will allow the proposed network to work better than other shadow detection procedures reported in the literature by using publicly accessible datasets. The proposed model is resilient to different factors such as variations in light and faint grey low-contrast shadows in the environment.
2. Proposing a novel CNN network (PCPerturNet) that specifically captures local and global shadow features in an image through several down-sampling and up-sampling units. PCPerturNet incorporates precise mapping in its down-sampling units by using the proposed encoder pattern conserver and convolution mapping to maintain acquired shadow patterns through processing. The proposed decoder pattern conserver is the branch of up-sampling units that helps to reconstruct the linguistic shadow information.

The organization of the remaining part of this article is as follows: a summary of current techniques for shadow detection research is provided in Section 2. The proposed approach is explained in detail in Section 3, including the method for creating perturbed instances using fuzzy logic. The experimental findings are described in Section 4. Section 5 includes the performance comparison of the proposed method with other recent techniques for shadow detection used in the literature. Lastly, Section 6 concludes the study along with the future research direction.

## 2 | RELATED WORKS

To date, some studies have been proposed to detect shadows in a single image. Previous works established physical models focused on histogram and illumination-based principles [4–6]. Such observations can only perform well on high-resolution images like satellite images while performing poorly on complicated low-resolution images. The main reason behind the poor performance of this method when applied to low resolution image is that dark and light objects may be misclassified as shadow and non-shadow area respectively. Also, it is difficult to detect deep green grass or tree leaves from shadow.

Later, pixel and edge details were investigated to detect shadows. For instance, Tian et al. [9] used four shadow properties (point-wise product of daylight spectral power distribution with sRGB CMFs, the point-wise product of skylight spectral power distribution with CMFs, linear sRGB surface pixel values of illuminated by daylight (in non-shadow regions) and skylight (in shadow areas)) along with the edge characteristics of non-shadow and shadow regions. Four different datasets such as Zhu, Lalonde, Guo and a self-created dataset and commonly-used precision, recall, and F-measure are used to check the efficiency of the method. However, the drawback of this approach is that it cannot accommodate some conditions related to difficult scenes and lighting, like overexposure areas, noisy areas, and dense textured areas. Zhang et al. [10] used the ratio edge property (the ratio of the pixel intensities between two adjacent pixels) to detect shadow pixels in the image. The ratio edge distribution is analysed and a significance test is done to detect the shadow. Overall accuracy is used as the performance measure. Kang et al. [11] suggested a shadow detection algorithm focused on an extended random walker (ERW) by incorporating both shadow properties and spatial similarities between adjacent pixels. The support vector machine is used to attain an initial detection map, with the classified non-shadow and shadow pixels. The proposed method is compared with other shadow detection method like normalized saturation-value difference index and self-adaptive feature selection and overall accuracy is used to check the superiority of the proposed method. Then, instead of utilizing pixel-level indications individually, the region level indications were studied. For instance, Vicente et al. [12] used the least-squares kernel support vector machine (LSSVM) for the separation of non-shadow and shadow regions. The leave-one-out confidence values is calculated for every training sample and is compared with the ground truth shadow annotation to obtain the leave-one-out error rate. The proposed method is compared with unary SVM and convolutional nets and balanced error rate is used to check the superiority of the proposed method. Yuan et al. [13] used a local contour based method to detect the shadow region from the image. Gao et al. [14] and McFeely et al. [15] used the spatio-temporal moving shadow detection (STMSD) technique and colour and texture based methods respectively to detect shadow regions. Alterations in the texture of the current and neighbouring video frames and background are obtained using the watershed method, and each sub-area is categorized in terms of texture to achieve the required moving shadow and target area. Some researchers used transform domain techniques to detect shadow region. Nagarathinam et al. [16] used spatial wavelet transform (SWT) and Zernike moment (ZM) based method to detect moving shadow. The main drawback of this method is, SWT cannot capture edges properly and provides limited information along vertical, horizontal and diagonal direction. The major drawback of ZM is, it is not invariant to scale changes. All of the above approaches are focused on hand-crafted technologies that are not adequate discriminative in noisy and complicated scenes.

Shadow detection approaches focused on DL have recently become very common because of its performance. [17]. Initially,

researchers viewed CNN as an effective feature extractor and produced substantial changes in efficiency for strong deep features. Shen et al. [18] used a structured convolution neural network to capture the local shadow structure information. Non-local measures are used to address shadow retrieval as an optimization issue. The efficiency of this method is compared with four different methods like boosted decision tree-binary conditional random field model, boosted decision tree-conditional random field model, unary pairwise, CNN-conditional random field model. Percentage of accuracy is used as the performance measure. Later, convolution neural networks were suggested as a consequence of the creation of fully convoluted networks (FCNs). For instance, Vicente et al. [19] introduced a semantic-aware stacked CNN model that combines two neural networks: image level FCN and a patch-based CNN. During training, the output of FCN as well as the corresponding RGB image input is utilized to train the patch-CNN to extract the semantic shadow. The efficiency of this method is compared with two different methods like Convnets+Conditional Random Field and LookOP+Markov Random Field. Balanced error rate is used as the performance measure. Jiang et al. [20] proposed skip connection-based Injection networks to detect shadow. Different weights are assigned for shadow and non-shadow parts for efficiently detection of the shadow part from the image. The efficiency of this method is compared with six different methods like boosted decision tree-binary conditional random field model, unary-pairwise, CNN-conditional random field model, stacked CNN, stacked CNN-LinearOpt and stacked cGAN. Percentage of accuracy is used as the performance measure. Zheng et al. [21] suggest a distraction-aware shadow detection network (DSDNet) by specifically learning and incorporating the meanings of visual distraction areas into an end-to-end system. Distraction-aware shadow (DS) module is used to acquire distraction-aware, discriminatory features for shadow detection by directly forecasting false positives and false negatives. The efficiency of this method is compared with six different methods like ADNet, bidirectional feature pyramid network with recurrent attention residual module, direction aware spatial context, stacked conditional generative adversarial networks, stacked cGAN and stackedCNN. Balanced error rate is used as the performance measure. Tang et al. [22] suggested an end-to-end SDRNet shadow detection and removal network. The proposed network is based on an encoder-decoder structure. Encoder downsamples the features and provides semantic and global shadow information. While decoder upsamples the features that combines low-level detail with high-level semantic information. Balanced error rate is used as the performance measure.

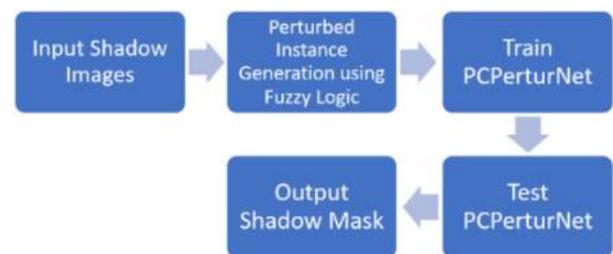
More recently, pattern knowledge has been examined. Zhu et al. [23] introduce a network to detect shadows by investigating and integrating the regional patterns in dense and deep layers and the local patterns in shallow layers of the deep CNN. First, they create a recurrent attention residual (RAR) framework to merge the patterns in two neighbouring convolution layers and use an attention map to pick the residual and then optimize the pattern attributes. Second, they create a bi-directional feature pyramid network (BFPN) to merge shadow patterns across

**TABLE 1** Pros and cons of different DL based methods mentioned in the literature

Method	Pros	Cons
Fully convolutional neural network	Required features are automatically calculated	This approach is restricted from accumulating only local attributes from tiny regions. Thus, they are not able to extract features of shadow from the adjacent regions that might be meaningful
Injection network	Computationally simple	The optimal depth of an injection network can vary from one shadow dataset to another, reliant on the difficulty of the task and the number of labelled data obtainable for training
DSDNet	Distraction semantics (i.e., shape, pattern, objects) are considered which is vigorous for detecting shadow on images with complex backgrounds	Some weak shadow images (where shadows have a very comparable brightness to the background) or some very dark background images (where the shadows are almost blended into the background) cannot be detected
SDRNet	Both shadow detection and removal can be done by using a single network	Multiscale and global feature method increases the complexity of the overall network
BFPN-RAR	The residual learning technique improves the feature refinement by learning the residual of input features	Fail in some extreme cases, such as the soft shadows and shadows with tiny details
scGAN	By using the tunable sensitivity parameter, the model can be efficiently trained	Suffers from the drawback of its inflexibility
ARGAN	Can handle shadows with complex scenes and is flexible to incorporate adequate unsupervised shadow images to train a model	Once this type of model is trained, it cannot quickly satisfy the need of a new requirement

various convolution layers. In [24–26], generative adversarial networks (GANs) were adopted to collect pattern details. Wang et al. [24] proposed a stacked conditional GAN model to learn shadow detection and removal together. Here the authors proposed a stacked joint learning paradigm that consists of two conditional GANs. The second GAN is stacked upon the first one. The generator and discriminator of the first GAN are conditioned on the input shadow image. The discriminator of the second GAN differentiates the concatenation of outputs from the generator of the first and second GAN. Balanced error rate is used as the performance measure. Nguyen et al. [25] proposed an scGAN model where a sensitivity parameter was added for the generator to monitor the intensity of the shadow detector. Balanced error rate is used as the performance measure. Ding et al. [26] suggested an attentive recurrent adversarial generative network (ARGAN) for shadow detection. A shadow attention detector is used in the generator to generate a shadow attention map. Long- and short-term memory is used to ensure that the detected shadow regions are accurate. The efficiency of this method is compared with four different methods like stacked conditional generative adversarial networks, Direction aware spatial context, ADNet, and bidirectional feature pyramid network with recurrent attention residual module. Balanced error rate is used as the performance measure. But the unpredictability of GAN still restricts its applications. Though, these pattern-based methodologies still suffer from images with complicated backgrounds, as they utilize the pattern to reduce potential variances between ground truth and predictions, which try to placate the most common cases whereas neglecting difficult cases.

Although the aforementioned techniques have reported some good performance, they are still suffering from an absence of precise global and local shadow patterns in their masks. Table 1 shows the pros and cons of different DL based methods discussed here.

**FIGURE 1** The overall block diagram of the proposed approach

A new shadow detection network (PCPerturNet) is introduced here, which profits from a new CNN framework trained on reproductive perturbed instances, contributing to improved results on two publicly accessible and one self-created dataset. Before the system is trained, ten perturbed images are generated from each dataset image using a fuzzy rule-based method. During training, five types of augmentation are applied to make the system invariant to scale, orientation, and flipping. As a consequence, the network can easily be used in many other image segmentation applications.

### 3 | PROPOSED APPROACH

This section details the proposed approach to detect shadow from the image. Figure 1 shows the overall block diagram of the proposed approach.

First, ten perturbed instances are created from every original image by using the fuzzy logic approach to deal with the variants in the shadow area characteristic. These perturbed images are generated by varying the brightness and contrast of the original image to make the system robust from handling

different dark or light images. Then, the proposed pattern conserving convolutional neural network (PCPerturNet) is trained using those perturbed instances. The proposed architecture can be used to detect local as well as global shadow from the image and it can conserve the patterns generated through the up-sampling and down-sampling units of the deep framework. The conservation of the pattern is needed so that they may not be lost in the dense and deep framework. After training, the test images are fed to PCPerturNet to generate the shadow mask. The details of every step are mentioned in the following sections.

### 3.1 | Generating perturbed instances using fuzzy logic

Literature shows that shadows may be detected by utilizing HSV colour space as the shadow alters the background light substantially, whereas the hue or chromaticity varies only moderately. The proposed fuzzy approach utilizes HSV colour space in which only the V channel is augmented by retaining chromatic details such as Hue (H) and Saturation (S) to produce perturbed examples from the original dataset images. This method is used here precisely to enhance low bright and contrast image. Augmentation of the V channel is done under the guidance of the augmented parameters  $E_1$  and  $E_2$ . This augmentation will convert the current intensity value  $P_C$  to the augmented intensity value  $P_E$ .

The first step in the proposed approach is to transform the RGB image of size  $M \times N$  to HSV and then to compute the histogram  $b(P)$  where the  $P \in V$ .  $b(P)$  signifies the number of image pixels with the intensity value  $P$ . The proposed approach utilizes two augmentation parameters  $E_1$  and  $E_2$ , which regulates the degree to which the intensity value  $P$  needs to be augmented. The first augmentation parameter  $E_1$  is the average intensity value of the image and can be computed using Equation (1) from the histogram:

$$E_1 = \frac{\sum_P P b(P)}{\sum_P b(P)} \quad (1)$$

The parameter  $E_1$  divides the histogram  $b(P)$  into two groups  $C_1$  and  $C_2$  representing pixel values in the form  $[0, E_1 - 1]$  and  $[E_1, 255]$  respectively. The augmentation of the V channel is based on two fuzzy membership values of  $\mu_1$  and  $\mu_2$ , determined for the pixel groups  $C_1$  and  $C_2$ . Parameter  $E_1$  plays a major role in the calculation of fuzzy membership values,  $\mu_1$  and  $\mu_2$ .

The second augmentation parameter  $E_2$  specifies the augmentation strength to measure the augmented  $P_E$  intensity values for both groups  $C_1$  and  $C_2$ . Parameter  $E_2$  specifies the augmentation level to which the values of intensity  $P$  can be altered depending on the quantities of membership  $\mu_1$  and  $\mu_2$ . The value of  $E_2$  can be determined experimentally by the degree to which augmentation is needed. From the experimental analysis, 10 different values for  $E_2$  ranges from 150 to  $-40$  at the sep-

aration of 20 are used to create 10 different perturbed instances from a single dataset image. As a consequence, every training image will have 10 perturbed instances varying in the colour of the light source and intensity in the image. With these 10 perturbed images, the original dataset image is also fed to the system during training.

The value of  $\mu_1$  for group  $C_1$  is based on the difference between  $P$  and  $E_1$ . This can be expressed by using the representation mentioned in Equation (2). This means that, if the difference between  $P$  and  $E_1$  is large then the augmentation of the pixel intensity will be small.

$$((P - E_1) : \text{Large}) \Rightarrow (\text{Intensity}_{\text{Augmentation}} : \text{Small}) \quad (2)$$

The aforementioned rule suggests that the intensity value of the pixels near to  $E_1$  should be augmented higher, while the values farther away from  $E_1$  will be augmented less. The pixel values in between would be augmented respectively. The mathematical description in Equation (3) is used to implement the above fuzzy rule.

$$\mu_1(P) = \frac{E_1 - P}{E_1} \quad (3)$$

where  $P \in C_1$ . When the membership value for  $P$  is attained, the augmented intensity value of  $P_E$  for group  $C_1$  can be determined using Equation (4).

$$P_E = P + \mu_1(P) E_2 \quad (4)$$

$\mu_1(P)$  specifies the amount of the augmentation parameter  $E_2$  to be applied to  $P$  to get the augmented  $P_E$  value.

The fuzzy membership value  $\mu_2$  for group  $C_2$  is based on the idea of how far the intensity value  $P$  is from the maximum value of  $L$  (for 8-bit image  $L = 255$ ). The fuzzy rule for group  $C_2$  can be expressed using Equation (5). This means that, if the difference between  $P$  and  $L$  is large then the augmentation of the pixel intensity will be large.

$$((P - L) : \text{Large}) \Rightarrow (\text{Intensity}_{\text{Augmentation}} : \text{Large}) \quad (5)$$

The above rule suggests that the values of the pixels closest to  $L$  should be augmented less, whereas the values farther away from  $L$  will be augmented higher. The pixel values will be distributed correspondingly. The mathematical expression of Equation (6) can be used to apply the above fuzzy rule.

$$\mu_2(P) = \frac{L - P}{L} \quad (6)$$

where  $P \in C_2$ . Once the membership value for  $P$  is attained, the augmented intensity value  $P_E$  for group  $C_2$  can be calculated by using Equation (7).

$$P_E = P + (\mu_2(P) \times E_2) \quad (7)$$

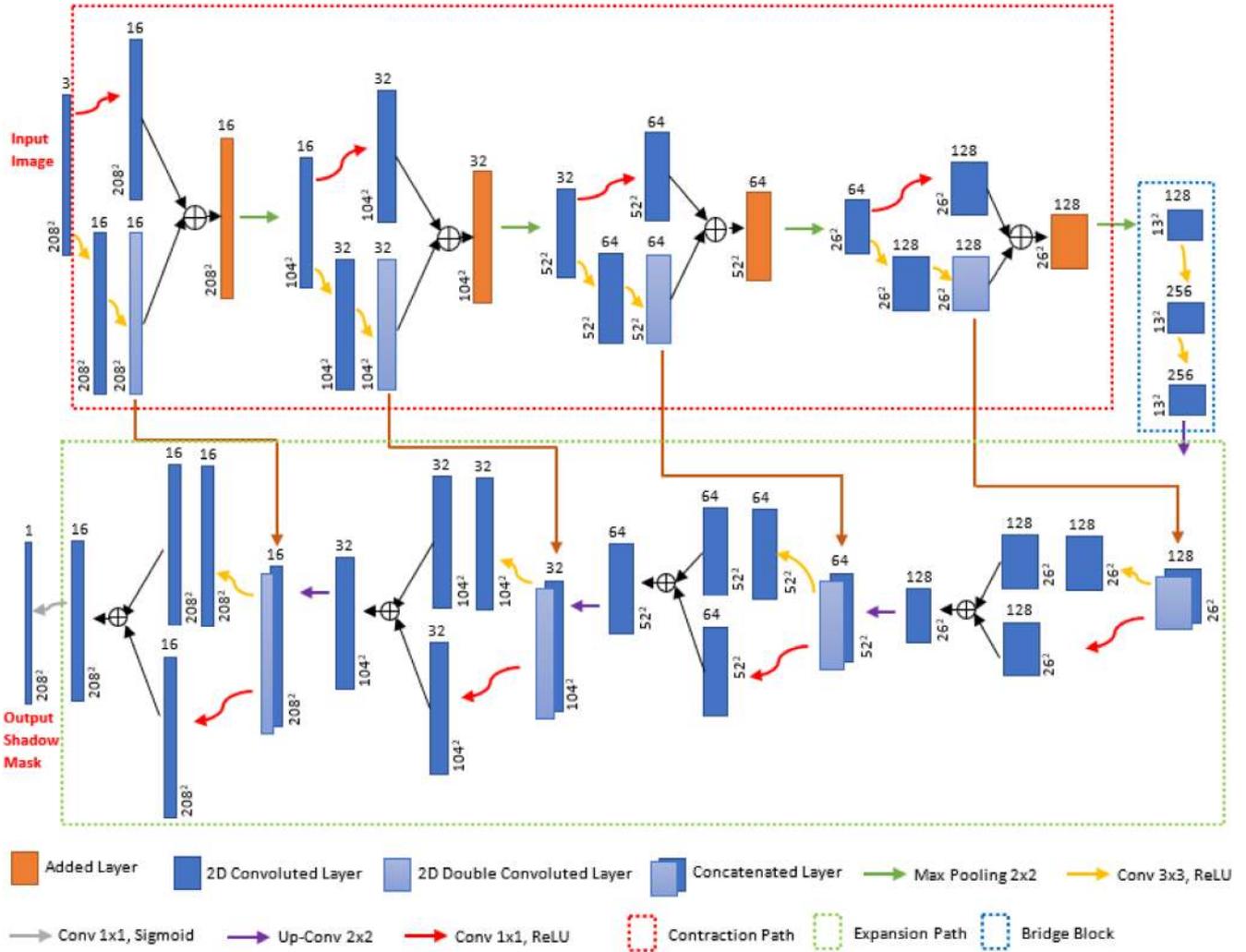


FIGURE 2 A pictorial summary of the proposed PCPerturNet model

$\mu_2(P)$  decides what amount of the intensity value  $P$  and augmented parameter  $E_2$  has to be used to get the augmented value  $P_E$ .

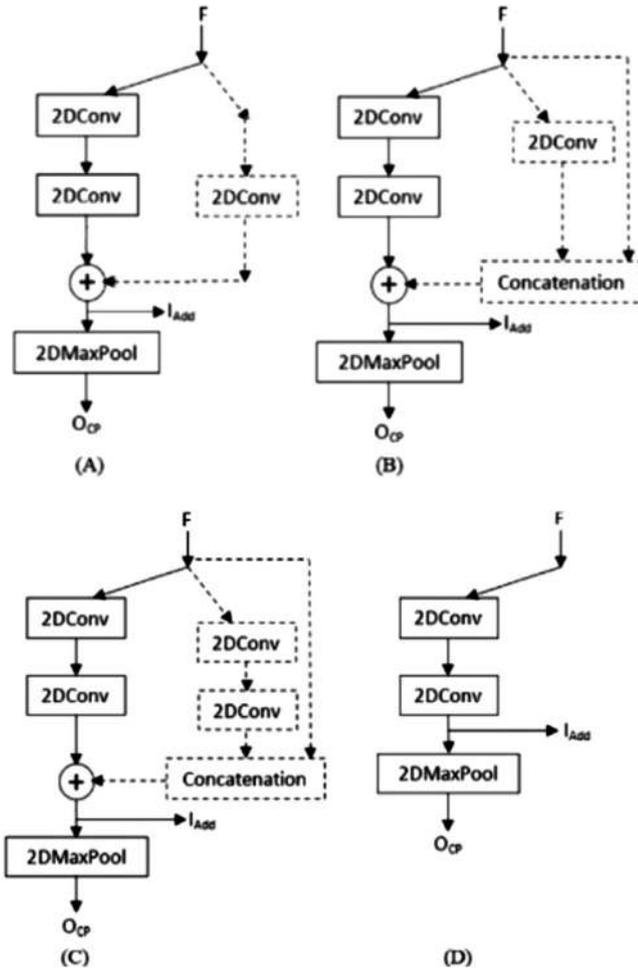
Replacing the old  $P$  values of the  $V$  channel with the augmented  $P_E$  values would allow the  $V$  channel to be expanded, resulting in augmented  $V_E$  channel. This augmented achromatic details  $V_E$  can be coupled with the saved chromatic data (Hue and Saturation channels) to produce an augmented  $HSV_E$  the image that is ultimately transformed to an augmented  $RGB_E$  image. Thus, perturbed instances are created from every training image using fuzzy logic.

### 3.2 | PCPerturNet

Appropriate image shadow detection includes a pixel-level learning system that is able of retrieving both rough and fine image information. Here, PCPerturNet is proposed for shadow detection, motivated by U-Net and Res-Net. The proposed architecture consists of two primary pathways:

contraction and expansion. The contraction path is generally important for the downsampling of the input image’s linguistic attributes into a manifold attribute map. This is the encoder path of the architecture. These deep features are the most significant shadow properties in the image. To detect the shadow from these dense features, a different path is needed to up-sample these attributes and slowly produce an outcome. The size of the output image is the same as the input. This is achieved through the expansion path, which is the network’s decoder path. The paths of the PCPerturNet are linked with each other utilizing a bridge unit through a long skip connection. Residual units are included in both contraction and expansion paths, which incorporate short skip connections (inside the unit) alongside established long skip connections. Short skip connections make for quicker convergence while training and require dense models to be learned. Figure 2 illustrates a pictorial summary of the proposed PCPerturNet model.

As seen in Figure 2, the contraction path comprises of four units and is called as the contraction path (CP) unit. As the input



**FIGURE 3** (A) Proposed CP unit with novel convolution mapping, (B) CP unit with convolution-input mapping, (C, D) CP unit with convolution-convolution-input and with no skip connection

image goes through several consecutive CP units, the depth of the image will rise and the spatial dimension will reduce. Such dense attributes provide high-level global image details compared to the shallow feature. There are four units in the expansion path to decode the global and local features. The usage of such units, defined as expansion path (EP) units, is a crucial aspect in producing a resultant map of the same spatial scale as the input. EP unit reduces the depth of the output and raises its spatial size. The architecture of CP and EP is discussed in detail in the following section.

### 3.2.1 | Contraction path unit

Figure 3(A) displays the description of the proposed CP unit. Every CP unit consists of two divisions. The outcome of the two divisions is connected. The left division is liable for retrieving the necessary attributes from the input image. This division is called an attribute extractor division and is represented using a solid line in Figure 3. The right division includes a new skip connection. This is responsible for conserving input

patterns. This division is called an encoder pattern conserver (EPC) division and is denoted using the dotted line in Figure 3. The proposed EPC skip connection of Figure 3(A) is not identical to the skip connection's identity mapping utilized in the literature.

The suggested EPC branch pattern adds valuable characteristics to the shadow detection network. First, it passes comparatively low-level input features to output to maintain the key features of the image and prevent losing them in subsequent convolution layers. It also allows the model to retain the entire details learned from the preceding layers. Second, it removes the issue of gradient vanishing in dense models by enabling the backpropagation of the gradient flow. As a consequence, convergence in dense models is rendered easily.  $1 \times 1$  kernel is used in the convolution layer of the EPC branch. The inspiration of using a  $1 \times 1$  kernel to input is that by using this the most appropriate and essential attributes are selected from the input and called a convolution mapping.

There are two outputs for the CP unit. A unit of CP, seen in Figure 3(A) receives the attribute map ( $F$ ) from its preceding unit and produces two outcomes:  $O_{CP}$  and  $I_{ADD}$ . Here, the size of the stride of the CP unit is fixed to 1 and the pool in the max-pooling layer to 2. In the convolution layers, choosing the stride size of more than 1 can decrease the scale of the outcome quickly. Instantly reducing the scale of the attribute maps could remove the learning patterns of the network. The same logic applies to the size of the pool. The depth of  $O_{CP}$  is increased to double the depth of  $F$ , whereas the spatial dimension is decreased to half of  $F$ . The outcome,  $I_{ADD}$ , is utilized as input for the EP unit. As a consequence, the CP unit collects the characteristics of its input by integrating the properties of both the attribute extractor division and the pattern conserver division.

Three other variations of the CP units are also attempted as seen in Figure 3. Figure 3(B) and (C) use the convolution-input and convolution-convolution-input mapping respectively instead of the only convolution mapping. Figure 3(D) displays a CP unit without a skip connection. Here just a branch of the attribute extractor. The design of the convolution-input and convolution-convolution-input mapping is complex than the convolution mapping, as the input is concatenated to the convolution mapping. However, the convolution mapping provides the best results in reality as seen in Table 1.

The proposed convolution mapping is having two significant consequences on the detection of shadows. First, this allows the model to remember what it has learned in the preceding layers and to retain those patterns. Second, this enables the flow of the gradient across the units of the network. The probable explanation of this better efficiency in convolution mapping relative to convolution-input and convolution-convolution-input mapping may be the reason that, in convolution mapping, the network selects essential input features rather than concatenating input to convolution output as in convolution-input and convolution-convolution-input mapping. The concluding row of Table 2 reveals that without using the EPC division the efficiency of the system greatly impairs.

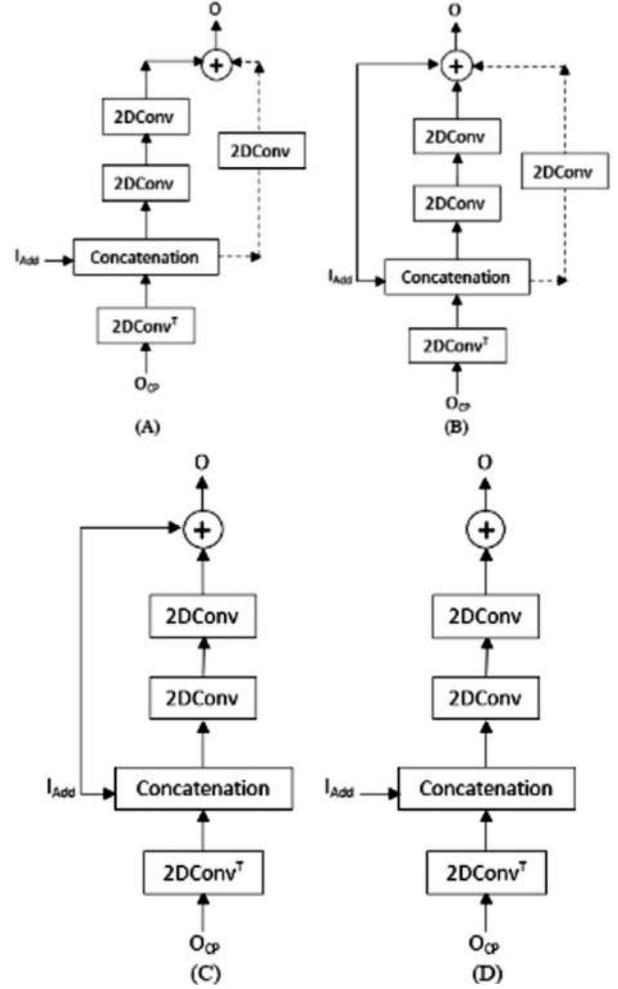
**TABLE 2** The training performance of different CP units on SBU and ISTD datasets

Method	SBU training set				ISTD training set			
	ACC (in %)		Number of epochs needed		ACC (in %)		Number of epochs needed	
	O	O + P	O	O + P	O	O + P	O	O + P
Convolution mapping (Figure 3A)	92.4	96.2	9	20	90.1	93.6	5	14
Convolution-input mapping (Figure 3B)	87.3	92.1	14	25	84.3	88.2	11	22
Convolution- convolution-input mapping (Figure 3C)	86.8	91.8	17	27	84.9	88.5	13	25
No EPC branch (Figure 3D)	78.2	84.5	6	15	72.2	78.9	4	12

### 3.2.2 | Expansion path unit

Figure 4(A) demonstrates the components of the proposed expansion path (EP) unit. Each EP unit consists of two divisions. The outcomes of these two divisions are combined to generate the final output.

The input of the EP unit is added with a transposed convolution layer to up-sample the data. The up-sampled data are then concatenated with the attributes of the respective  $I_{ADD}$ . Thereafter, two consecutive convolution layers are added in the left branch to reconstruct linguistic information and represented using the solid line in Figure 4. The right division is a new skip connection, which is responsible for conserving the patterns of the up-sampled data. It is named as the decoder pattern conserver (DPC) branch and is represented using the dotted line in Figure 4. An addition is then utilized to sum the performance of the DPC and the second convolution layer of the left division. Here the strides are set to 2 and 1 for the transposed convolution layer and the convolution layer respectively. Choosing the size of the transposed convolution layers stride more than 2 raises the attribute maps too fast and prohibits the network model from retrieving shadow patterns properly. For the transposed convolution, if the stride is 1 no up-sampling is carried out, and thus the transposed convolution should be greater than or equivalent to 2. The spatial dimension of  $O$  is raised to double whereas the depth of  $O$  is decreased to half the depth of the input to the EP unit.  $O$  is utilized as input for the subsequent EP unit. As a consequence, the EP unit upsamples the attribute map utilizing transposed convolution layers. This utilizes learned attributes from preceding layers and DPC output to retrieve shadow features and combine these features into the result produced.  $1 \times 1$  kernel is used in the convolution layer of the DPC branch to choosing the most appropriate features from the feature map.

**FIGURE 4** (A) Proposed EP unit with novel skip connection, (B, C, D) Other EP units with and without skip connection

The other three variants of the EP units are examined as shown in Figure 4 (B, C, and D respectively). The numerical findings referring to each type are listed in Table 2. These outcomes indicate that the efficiency of the shadow detection method increases with the skip connection of the EP unit (Figure 4(A)).

As the CP unit with the convolution mapping in Figure 3(A) and EP unit as seen in Figure 4(A) correspond to the finest results among all the structures; these unit structures are selected to be implemented into the proposed network model. From now, the recorded numerical outcomes are derived utilizing the PCPerturNet with the abovementioned CP and EP units.

The  $3 \times 3$  kernel size is used for both the convolution and transposed convolution layers of the CP's attribute extractor division and the EP's left branch. The rectified linear unit (ReLU) activation function is used for all convolution layers accompanied by the layer of batch normalization. All addition units are accompanied by the ReLU feature to maximize the network's non-linearity and boost the overall efficiency. In the bridge unit there is a dropout layer with a 0.20 drop rate. Research showed that the dropout layer protects the network model from overfitting by increasing the regularization feature and growing the generalization error of the network model.

### 3.3 | Training and testing details of PCPerturNet

Every single image transmitted to the training network is resized from its original size to  $208 \times 208$ . Here five forms of geometric augmentation are used for all the images in the training phase. These changes are horizontal flipping, vertical flipping, rotation, scale-up, and scale-down, and in each epoch are arbitrarily chosen to be off or on in a combination of 1 to 5. The orientation range is set to  $[0-360^\circ]$  at the separation of  $5^\circ$ , the scale-up factor range is set to  $[1.0-2.0]$  at the separation of 0.1 and scale-down factor is set to  $[1.0-0.5]$  at the separation of 0.1.

The loss function used to reduce the difference between the original shadow ( $S_{True}$ ) and the resultant shadow ( $S_{Pred}$ ) map is the soft dice loss ( $SDL$ ).  $SDL$  is defined by using Equation (8).

$$SDL(S_{True}, S_{Pred}) = 1 - \frac{2 \sum_{\text{pixels}} S_{True} S_{Pred}}{\sum_{\text{pixels}} S_{True}^2 + \sum_{\text{pixels}} S_{Pred}^2} \quad (8)$$

This value is repeated and summed over all pixels. This loss function is utilized to explicitly use the predicted probabilities instead of thresholding and translating them to a binary mask. So far as the performance of the neural network is concerned, the numerator is dealing with specific activations between the predictor and the true mask, while the denominator is dealing with the number of activations in each mask individually. This has the effect of normalizing the loss by the scale of the target mask so that  $SDL$  does not fail to learn from groups with a less spatial presence in the image.

The preliminary learning rate of the model is assigned to  $10^{-3}$ , the optimizer utilized for this experiment is Adam optimizer and the number of Epoch is set to 30. During training, the model is analysed on a validated dataset after each epoch. If the performance of the model on the validation dataset starts to decline (e.g. the loss starts to increase or the accuracy starts to decrease), the training phase is stopped. The model is then considered to have good generalization efficiency.

To achieve a mask for the identified shadow part for every test image, it is resized to  $208 \times 208$ . It is then transmitted to PCPerturNet and, as an output, the mask or probability map is produced.

### 3.4 | Performance evaluation

Shadow detection is a segmentation of pixel-level and a performance measure is needed to assess output accurately. The way for assessing system performance is the overall accuracy ( $ACC$ ). Accuracy is computed by using Equation (9).

$$ACC = \frac{TP + TN}{TN + TP + FP + FN} \quad (9)$$

where  $TN$ ,  $TP$ ,  $FP$ , and  $FN$  are the number of non-shadow pixels appropriately recognized as non-shadow, shadow pixels appropriately identified as a shadow, non-shadow pixels inaccurately identified as a shadow, and shadow pixels wrongly recognized as non-shadow, respectively. Also, two error rates:  $ER1$  (shadow pixel incorrectly identified as non-shadow) and  $ER2$  (non-shadow pixels wrongly recognized as shadow) are calculated by using Equations (10) and (11). Three more performance measure matrices are added to check the efficiency of the model: precision ( $PR$ ), recall ( $RE$ ) and paired t-test ( $PT$ ) and are calculated by using Equations (12)–(14). The motivation to choose  $PT$  as one of the performance measures is that  $PT$  helps to measure the difference between two images (in this case, the target mask and the obtained output mask) with a unique condition. By using  $PT$ , the exactness of the shadow mask obtained from the proposed approach can be measured very efficiently.

$$ER1 = \frac{FN}{FN + TP} \quad (10)$$

$$ER2 = \frac{FP}{FP + TP} \quad (11)$$

$$PR = 1 - ER2 \quad (12)$$

$$RE = 1 - ER1 \quad (13)$$

$$PT = \text{Target\_Mask} - \text{Obtained\_Mask} \quad (14)$$



**FIGURE 5** Sample shadow images from SBU (first two) and ISTD (last two) dataset

## 4 | EXPERIMENTATIONS AND RESULTS

For training of PCPerturNet, Intel Core i7 9th Generation processor, 12 GB DDR4 RAM, NVIDIA GeForce GTX 1600 Ti Graphics (6GB), is used which minimizes the time of training. The implementation of the proposed approach is done utilizing TensorFlow.

### 4.1 | Datasets

The details of the used datasets to check the performance of the system are mentioned in this section.

#### 4.1.1 | The Stony Brook University (SBU) dataset [27]

This is probably the biggest freely accessible shadow detection dataset, comprising of 4085 and 638 images as a train and a test set. The images in this dataset contain a wide variety of scenes, such as ocean, town, highways, forest, parks, water, automobiles, wildlife, and buildings. It also comprises various image types such as landscape, aerial, selfies, close range, etc. A shadow-mask of every training and test image is contained within the SBU dataset.

#### 4.1.2 | The image shadow triplets (ISTD) dataset [28]

This data set comprises of a shadow, a shadow-free image, and a shadow mask to fulfil the requirement of learning. It comprises 1870 image triplets under 135 different scenarios, in which 1330 is allocated for train, while 540 for the test. Here, various shapes of shadows are constructed by various objects, like boards, umbrellas, twigs, persons etc.

Some images from SBU (first two) and ISTD (last two) datasets are shown in Figure 5.

#### 4.1.3 | Self-created dataset

The effectiveness of the proposed approach is tested on a self-created dataset comprises of two hundred shadow images that I captured utilizing Nikon Digital SLR 5200 camera.



**FIGURE 6** Sample images from the self-captured dataset



**FIGURE 7** Perturbed instances generated from the original image; (first row) Original image, (second and third row) Generated perturbed instances by varying the value of  $E_2$  from 150 to  $-40$  at a separation of 20

The captured image size is  $4496 \times 3000$  pixels in JPEG format. This includes images of statue, utensils, furniture, and plants. Along with this the dataset images contains some challenging combination of image shadows like overlapping shadow and multi object shadow. To raise the speed of the computation, the images are rescaled to  $208 \times 208$  pixels. Some sample images are shown in Figure 6.

#### 4.1.4 | The expanded dataset created from the original dataset

The synthetic perturbed instances are generated from 4085 SBU images and added them with the original SBU training images and thus a total of 44,935 number of images are there in the expanded training image set. As a consequence, 40,850 perturbed images are added to the original training image set. For the ISTD training set, 20,570 is the total number of training images, after including the perturbed instances.

Figure 7 shows the original dataset image and 10 perturbed instances created from the original dataset image by varying the value of  $E_2$ .

## 4.2 | Performance evaluation

The proposed shadow detection methodology is trained on the two expanded datasets: SBU and ISTD. First, PCPerturNet is tested on the SBU, ISTD, and self-created test sets after

**TABLE 3** The training performance of different EP units on SBU and ISTD datasets

Method	SBU training set				ISTD training set			
	ACC (in %)		Number of epochs needed		ACC (in %)		Number of epochs needed	
	O	O+P	O	O+P	O	O+P	O	O+P
DPC (Figure 4A)	92.8	95.9	10	20	91.4	94.1	6	15
DPC (Figure 4B)	89.1	93.5	16	29	85.7	89.5	12	22
DPC (Figure 4C)	87.3	92.4	13	24	80.1	83.3	10	20
No DPC branch (Figure 4D)	77.6	83.1	7	14	73.1	77.3	5	11

**TABLE 4** The test performance of PCPerturNet

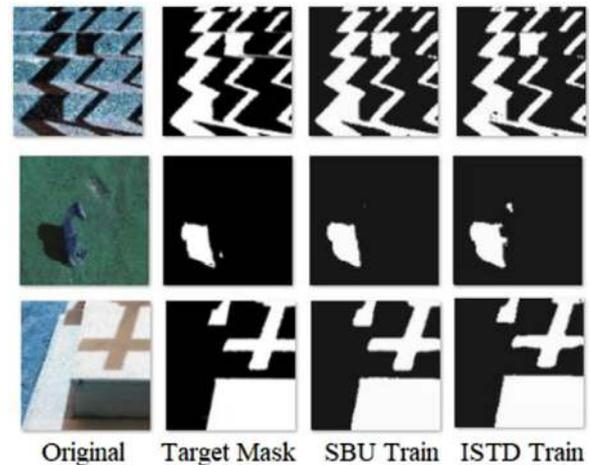
Test set	Performance metric	SBU training set		ISTD training set	
		O	O + P	O	O + P
SBU test set	<i>ACC</i> (in %)	93.1	<b>96.4</b>	92.6	94.7
	<i>ER1</i> (in %)	4.1	<b>0.5</b>	1.5	1.1
	<i>ER2</i> (in %)	<b>2.8</b>	3.1	5.9	4.2
	<i>PR</i> (in %)	<b>97.2</b>	96.9	94.1	95.8
	<i>RE</i> (in %)	95.9	<b>99.5</b>	98.5	98.9
	<i>PT</i>	0.21	<b>0.17</b>	0.25	0.22
ISTD test set	<i>ACC</i> (in %)	91.8	93.4	94.3	<b>96.8</b>
	<i>ER1</i> (in %)	0.7	1.3	0.8	<b>0.2</b>
	<i>ER2</i> (in %)	7.5	5.3	4.9	<b>3</b>
	<i>PR</i> (in %)	92.5	94.7	95.1	<b>97</b>
	<i>RE</i> (in %)	99.3	98.7	99.2	<b>99.8</b>
	<i>PT</i>	0.23	0.22	0.21	<b>0.18</b>
Self-created test data	<i>ACC</i> (in %)	85.2	<b>89.4</b>	83.6	88.1
	<i>ER1</i> (in %)	2.5	<b>1.6</b>	6.2	5.2
	<i>ER2</i> (in %)	12.3	<b>9</b>	10.2	6.7
	<i>PR</i> (in %)	87.7	<b>91</b>	89.8	93.3
	<i>RE</i> (in %)	97.5	<b>98.4</b>	93.8	94.8
	<i>PT</i>	0.34	<b>0.24</b>	0.37	0.23

training using the expanded SBU and ISTD training set individually. Because the size of self-created dataset is relatively small, PCPerturNet is not trained on it and is only utilized for testing purposes.

The following tables depict the performance of PCPerturNet. Here ‘O’ is the ‘Original’ dataset and ‘O + P’ is the ‘Original + Perturbed’ dataset. Table 2 shows the training performance of different CP units on SBU and ISTD datasets.

Table 3 shows the training performance of different EP units on SBU and ISTD datasets.

Table 4 demonstrates the assessment of empirical tests for the proposed approach (PCPerturNet) over the SBU, ISTD, and



**FIGURE 8** SBU test sample shadow mask output; (column 1) Original test image, (column 2) target mask to be generated from input test image, (columns 3 and 4) output mask when trained with expanded SBU and ISTD trained dataset

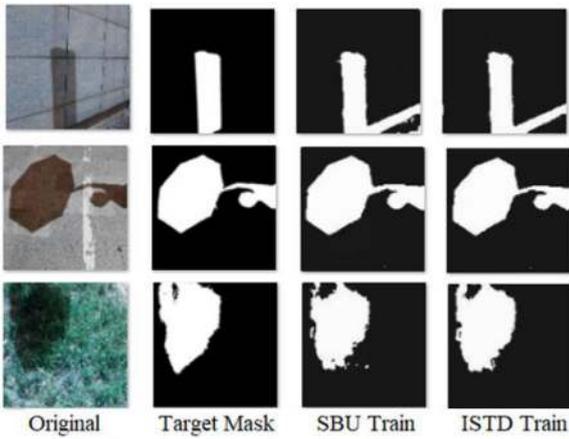
self-created test sample while trained using the SBU and ISTD training sets.

From Table 4 it is clear that when the system is trained with a greater number of training samples then the system is delivering better performance. When the system delivers a high *ER1*, it inclines to recognize more shadow as non-shadow pixels. Also, when the system delivers a relatively high *ER2*, it inclines to recognize more non-shadow as shadow pixels. The vital fact here is that steadiness between these errors should tend to better performance. While testing the system for every type of test sample *ER2* is greater than *ER1*. It implies that more non-shadow samples are recognized as a shadow than vice versa.

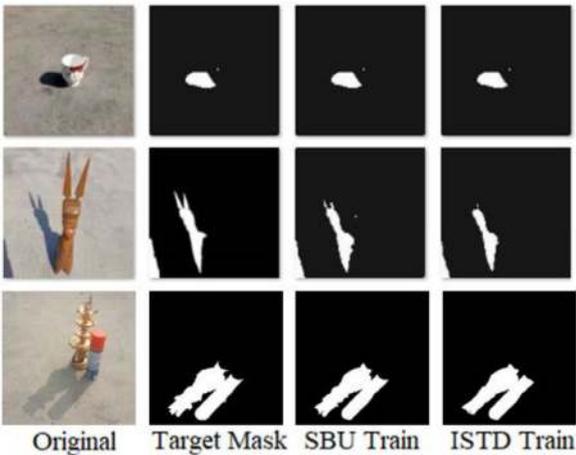
Figure 8 shows some detected shadow mask from SBU test images when trained with expanded SBU and ISTD training dataset.

Figure 9 shows some detected shadow mask from ISTD test images when trained with expanded SBU and ISTD training dataset.

Figure 10 shows some detected shadow mask from self-created test images when trained with expanded SBU and ISTD training dataset.



**FIGURE 9** ISTD test sample shadow mask output; (column 1) Original test image, (column 2) target mask to be generated from input test image, (columns 3 and 4) output mask when trained with expanded SBU and ISTD trained dataset



**FIGURE 10** Self-created test sample shadow mask output; (column 1) Original test image, (column 2) target mask to be generated from input test image, (columns 3 and 4) output mask when trained with expanded SBU and ISTD trained dataset

## 5 | PERFORMANCE COMPARISON OF THE PROPOSED METHOD WITH PREVIOUS METHODS USED BY THE RESEARCHERS FOR SHADOW DETECTION

In this section the performance of PCPerturNet is compared with the previous methods used for shadow detection by the researchers to check the efficiency of PCPerturNet. Individually the performance of the different methods is tested by trained the system using original and expanded SBU and ISTD training set.

Table 5 depicts the comparison of the performance of different shadow detection methods used by previous researchers. Here the system is trained with original (O) and expanded (O + P) SBU training set and tested by using SBU, ISTD, and self-created test set.

**TABLE 5** Comparison of the performance of PCPerturNet with the shadow detection methods used in the literature

Test set	Method	ACC (in %)		ER1 (in %)		ER2 (in %)	
		O	O + P	O	O + P	O	O + P
SBU	Plain U-Net	75.9	80.1	7.2	8.2	16.8	11.7
	IN [20]	79.4	83.5	7.2	5.9	13.3	10.6
	ST-CNN [19]	81.6	84.7	7.3	5.8	11.1	9.5
	ST-CGAN [24]	90.4	93.7	2.9	5.7	6.7	0.6
	scGAN [25]	89.6	92.7	6.2	0.3	4.1	6.8
	LSSVM [12]	80.7	84.3	9.6	3.1	9.6	12.6
	ERW [11]	75.2	80.1	9.9	10.3	14.8	9.5
	DSDNet [21]	87.6	91.5	3.7	2.5	8.9	5.9
	ARGAN [26]	90.5	92.8	8.5	5.2	0.9	2
	SDRNet [22]	88.4	91.7	2.3	3.1	9.3	5.2
	Histogram [5]	61.5	64.3	11.7	6.3	24.1	24
	SWT + ZM [16]	65.1	69.7	13.4	13.2	21.5	17.1
PCPerturNet	93.1	96.4	4.1	0.5	2.8	3.1	
ISTD	ARGAN [26]	86.5	88.2	4.1	4.9	9.5	6.8
	PCPerturNet	91.8	93.4	0.7	1.3	7.5	5.3
Self-created	ARGAN [26]	80.4	82.6	9.1	7.2	10.5	10.2
	PCPerturNet	85.2	89.4	2.5	1.6	12.3	9

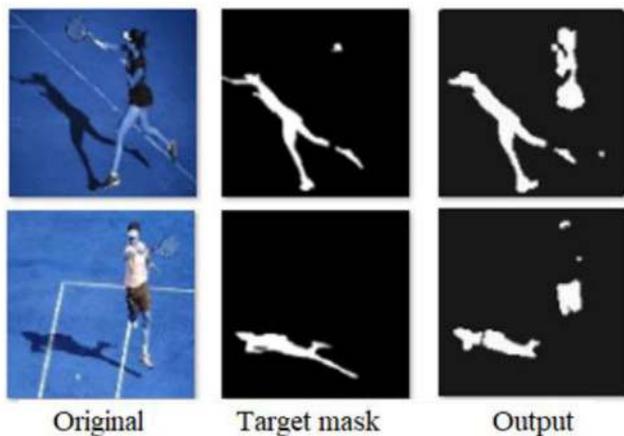
\*IN: Injection network, ST-CNN: STacked convolutional neural network, ST-CGAN: STacked conditional generative adversarial network, LSSVM: Least-squares support vector machine, ERW: Extended random walker, DSDNet: Distraction-aware shadow detection network, ARGAN: Attentive recurrent generative adversarial network, SWT + ZM: Stationary wavelet transform + Zernike moment.

It is clear from Table 4 that the proposed method (PCPerturNet) outperforms other strategies in ACC and ER1 while using the SBU training and test data set. The most possible cause of getting weaker ER2 is that the PCPerturNet is sensitive to the object's shadow placed on the object portion or part of some object part having very similar characteristics as of shadow. This condition has been found in a variety of instances which is responsible for low accuracy results, as depicted in Figure 11.

Another parameter is used to check the performance efficiency of the proposed method: computation cost. In this study execution time is considered as the computational cost. Execution time depends on various factors like configuration of the system, training image size, training image resolution, the network architecture etc. Table 6 depicts the comparison of the execution time of different shadow detection methods used by previous researchers. Here the system is trained with original (O) and expanded (O + P) training set of SBU and ISTD shadow datasets.

The execution time varies depending on the complexity of the network architecture.

Figure 12 shows the comparison between shadow masks attained by using the PCPerturNet and different methods mentioned in the literature when trained with expanded SBU training set.



**FIGURE 11** Incorrect SBU test image mask output generated by PCPerturNet; (column 1) Original image, (column 2) target mask to be obtained from the system, (column 3) output generated from PCPerturNet

**TABLE 6** Comparison of the execution time of PCPerturNet with the shadow detection methods used in the literature

Method	Training time (in days)			
	SBU		ISTD	
	O	O + P	O	O + P
Plain U-Net, IN [20]	3.5	6.5	2	4.5
ST-CNN [19]	4	7.5	2.5	5.5
ST-CGAN [24], scGAN [25]	4.5	7.5	2.5	5.5
LSSVM [12], ERW [11]	3.5	6.5	2.5	4
DSDNet [21]	4.5	7	3	4.5
SDRNet [22]	4.5	6.5	2.5	4.5
Histogram [5], SWT + ZM [16]	3.5	7.5	2.5	4.5
ARGAN [26], PCPerturNet	4.5	8.5	3.5	6.5

Table 7 depicts the performance of an attentive recurrent generative adversarial network (ARGAN) as it outperforms PCPerturNet in ER2. Here the system is trained with original and expanded ISTD training set and tested by using SBU, ISTD, and self-created test set.

From Table 7 it is clear that ARGAN can better handle the non-shadow pixels, but the overall performance concerning

**TABLE 7** Performance of AR-GAN on SBU, ISTD and self-created test set

Test set	ACC (in %)		ER1 (in %)		ER2 (in %)	
	O	O + P	O	O + P	O	O + P
SBU	87.4	90.5	8	6	4.6	3.5
ISTD	91.5	93.2	4.7	4	3.8	2.8
Self-created	86.3	89.4	7.4	5.3	6.3	5.3



**FIGURE 12** Comparison of the output shadow mask of the proposed method (PCPerturNet) with the shadow detection methods mentioned in the literature

accuracy and  $ER1$  of PCPerturNet is good compared to other techniques used in the literature.

## 6 | CONCLUSIONS

A novel deep learning shadow detection process is proposed in this article to recognize image shadow pixels. The proposed PCPerturNet profits from a novel structure that distinguishes local and global shadow attributes utilizing sophisticated skip connection mapping. This network can conserve derived pattern structures in manifold layers. These patterns are eventually being utilized in multiple network units to produce final masks of shadow. The training phase of PCPerturNet is easy and can be specifically used or extended for other image segmentation purposes. The PCPerturNet is trained on several perturbed images created by fuzzy-logically dependent perturbed instance generators to make the system robust from the change in brightness and contrast. During training, five forms of augmentations are applied to images. With this variety of training images, the proposed network may obtain both rough and smooth shadow information in regions of shadow. This allows PCPerturNet tolerance to variations in the patterns of shadow, brightness, and contrast, orientation, scaling factor, and flipping which thus contributes to a greater generalization. As a consequence, PCPerturNet enhanced the literature shadow detection method. The proposed approach may easily be used in certain computer

vision systems that could be confused or deceived by the existence of shadows.

For the future, to increase the ER2 measure of the network, the design of the PCPerturNet may need to be updated to enhance the retrieval of shadow areas from derived attributes. This can include alterations to both the CP and EP units. Also, a module will be designed to recreate the image part after removing the shadow part from the image.

## REFERENCES

- Joshi, A.J., Papanikolopoulos, N.P.: Learning to detect moving shadows in dynamic environments. *IEEE Trans. Pattern Anal. Mach. Intell.* 30(11), 2055–2063 (2008)
- Benedek, C., Szirányi, T.: Bayesian foreground and shadow detection in uncertain frame rate surveillance videos. *IEEE Trans. Image Process.* 17(4), 608–621 (2008)
- Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. *Comput. Vision Image Understanding* 95(2), 238–259 (2004)
- Wang, T., et al.: Detection and removal of clouds and associated shadows in satellite imagery based on simulated radiance fields. *J. Geophys. Res.: Atmos.* 124(13), 7207–7225 (2019)
- Mostafa, Y., Abdelhafiz, A.: Accurate shadow detection from high-resolution satellite images. *IEEE Geosci. Remote Sens. Lett.* 14(4), 494–498 (2017)
- Sharma, D., Singhai, J.: An object-based shadow detection method for building delineation in high-resolution satellite images. *PFG–J. Photogramm. Remote Sens. Geoinf. Sci.* 87(3), 103–118 (2019)
- Yang, C.H.H., et al.: When causal intervention meets image masking and adversarial perturbation for deep neural networks. *arXiv preprint arXiv:1902.03380* (2019)
- Miller, D.J., Xiang, Z., Kesidis, G.: Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks. *Proc. IEEE* 108(3), 402–433 (2020)
- Tian, J., et al.: New spectrum ratio properties and features for shadow detection. *Pattern Recognit.* (2016), 51, 85–96
- Zhang, W., et al.: Moving cast shadows detection using ratio edge. *IEEE Trans. Multimedia* 9(6), 1202–1214 (2007)
- Kang, X., et al.: Extended random walker for shadow detection in very high resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* 56(2), 867–876 (2017)
- Vicente, T.F.Y., Hoai, M., Samaras, D.: Leave-one-out kernel optimization for shadow detection and removal. *IEEE Trans. Pattern Anal. Mach. Intell.* 40(3), 682–695 (2017)
- Yuan, X., Ebner, M., Wang, Z.: Single-image shadow detection and removal using local colour constancy computation. *IET Image Process.* 9(2), 118–126 (2014)
- Gao, J., Dai, J., Zhang, P.: Region-based moving shadow detection using watershed algorithm. In: 2016 International Symposium on Computer, Consumer and Control (IS3C). Xi'an, China, pp. 846–849 (2016)
- McFeely, R., Glavin, M., Jones, E.: Shadow identification for digital imagery using colour and texture cues. *IET Image Process.* 6(2), 148–159 (2012)
- Nagarathinam, K., Kathavarayan, R.S.: Moving shadow detection based on stationary wavelet transform and Zernike moments. *IET Comput. Vision* 12(6), 787–795 (2018)
- Ghazi, M.M., Yanikoglu, B., Aptoula, E.: Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing* 235, 228–235 (2017)
- Shen, L., Wee Chua, T., Leman, K.: Shadow optimization from structured deep edge detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, pp. 2067–2074 (2015)
- Vicente, T.F.Y., et al.: Large-scale training of shadow detectors with noisily-annotated shadow examples. In: *European Conference on Computer Vision*. Amsterdam, pp. 816–832 (2016)
- Jiang, X., Hu, Z., Ni, Y.: InNet: Learning to detect shadows with injection network. In: *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. Xi'an, China, pp. 1–6 (2018)
- Zheng, Q., et al.: Distraction-aware shadow detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, CA, pp. 5167–5176 (2019)
- Tang, J., et al.: SDRNet: An end-to-end shadow detection and removal network. *Signal Process.: Image Commun.* 84, 115832 (2020)
- Zhu, L., et al.: Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Munich, Germany, pp. 121–136 (2018)
- Wang, J., Li, X., Yang, J.: Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, pp. 1788–1797 (2018)
- Nguyen, V., et al.: Shadow detection with conditional generative adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. Venice, Italy, pp. 4520–4528 (2017)
- Ding, B., et al.: ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal. In: *Proceedings of the IEEE International Conference on Computer Vision*. Seoul, Korea (South), pp. 10213–10222 (2019)
- The Stony Brook University (SBU) dataset. Access 20 May 2019, [https://www3.cs.stonybrook.edu/~cvl/projects/shadow\\_noisy\\_label/index.html](https://www3.cs.stonybrook.edu/~cvl/projects/shadow_noisy_label/index.html)
- The Image Shadow Triplets (ISTD) dataset. Access 20 May 2019, <https://github.com/DeepInsight-PCALab/ST-CGAN>

**How to cite this article:** Chaki J. Shadow detection from images using fuzzy logic and PCPerturNet. *IET Image Process.* 2021;15:2384–2397.  
<https://doi.org/10.1049/ipr2.12221>