

Mask R-CNN for Moving Shadow Detection and Segmentation

Hend F. Bakr¹, Ahmed M. Hamad², and Khalid M. Amin³

Information Technology Department, Faculty of Computers and Information, Menoufia University, Egypt

{hend.farag@ci.menofia.edu.eg, ahmahit@ci.menofia.edu.eg, k.amin@ci.menofia.edu.eg}

Abstract—One of the primary tasks of completing and developing many computer vision applications is to identify and remove shadow regions. Most existing moving shadow detection methods depend on extracting hand-crafted features of object and shadow regions manually (for example the chromaticity, physical, or geometric properties). Shadow detection using handcrafted features is a challenging task due to different environmental conditions of the shadow such as camouflage and illumination irregularity problems that make these features inefficient to handle such problems. The proposed method uses Convolution Neural Networks (CNN) to automatically learn different distinctive features to model shadow under different environmental conditions. In this paper, the Mask Region Convolution Neural Network (Mask R-CNN) framework is evaluated and tested to automatically perform semantic segmentation in order to detect and classify shadow pixels from the entire video frame. To adapt Mask R-CNN for segmenting and detecting shadow regions, the most significant features are extracted from video frames in a supervised way using deep Residual Network (ResNet-101) architecture. Then, the Region proposal network (RPN) predicts regions of interest (ROI) and their classes that contain foreground objects. Finally, Fully Convolutional Network (FCN) generates a binary segmentation mask for each detected class in ROI. The proposed framework evaluated on common shadow detection datasets that have different environmental issues. Experimental results achieved high performance rates compared to several state-of-the-art methods in terms of average detection rate (96.81%), average discrimination rate (99.42%), and overall accuracy (98.09 %).

Keywords—Shadow detection; Deep learning; Object detection; Semantic segmentation; Mask R-CNN.

I. INTRODUCTION

Moving object detection is a basic step in most computer vision applications like video monitoring, autonomous driving, tracking applications, and counting applications [1, 2]. The object detection task includes defining the existence, determining the location coordinates, and recognizing the category of the moving object in the scene. Recognizing objects through its important features, locating objects to know their extent, and classifying objects to define their semantic class make the moving object detection step a challenging task [3]. One of the challenges of detecting moving objects is the appearance of shadow which misclassified as a part of the moving object. The shadow formed as a result of changing illumination conditions in different environments making the object detection to be a complex process [4]. The shadow takes the same motion and shape as the moving object which leads to distortion of object shape and size. Shadow degrades the performance of most computer vision applications [2] therefore, it is necessary to detect and remove the shadow to make these applications more robust and accurate.

In most of the existing literature, the process of detecting moving shadow includes extracting moving objects, extracting variant handcrafted features that enable to describe the shadow as texture, chromaticity, edges, or geometric features... etc., and then using a classification method to classify and integrate these features [4]. In most cases a post-processing step is needed to obtain the shadow and moving object regions [2]. There are common challenging conditions during the shadow detection process [5]. Some of these challenges have been addressed as follows: chromatic shadow (i.e. shadow regions do not have a clear specific photometric pattern and seem to be black areas); achromatic shadow (i.e. shadow regions seem to be free of colors); foreground–shadow camouflage and foreground–background camouflage (i.e. the foreground object parts have identical color, intensity, or texture to their corresponding backgrounds) [5]; non-textured surfaces (i.e. some regions don't have clear texture structure and seem to be flat regions); dark surfaces (i.e. some regions of the foreground and background have low brightness properties); overlapped regions of shadow and foreground objects; and multiple shadows (i.e. objects seem to have more than one casted shadow if there are more than one light source) [5]. To tackle the problems of detecting and removing shadow regions, many research studies have appeared. Previous methods relied on extracting distinctive features (e.g. color, texture, gradient, edges, and geometric features) to separate shadow from foreground objects [6]. These methods use a reference model of the background to classify these features as a shadow or an object.

The success of these methods depends on choosing the appropriate cues used for the detection [6]. The challenge of these methods is the way to select and extract the features manually to overcome different problems of the shadow environment. In addition, the selection of an effective background subtraction method for extracting moving objects is another challenge. Most background subtraction techniques suffer from many challenges including [7]; camouflage (i.e. the appearance of a moving object is similar to the background, which leads to false classification), bootstrapping (i.e. require more data free of moving objects to initialize the background model), dynamic background (e.g. waving branches of trees), and video noise [7, 8]. The problems of background modeling and moving object detection negatively affect the task of shadow detection, therefore, most of the recent methods depend on the source images to extract objects and detect shadows without the need for the background modeling.

Recently, deep learning-based techniques have shown impressive results for most tasks in computer vision including object detection and image classification [9] (as the automatically learned features have succeeded over the manually extracted features) [9]. Deep learning-based techniques are able to automatically learn massive data of various conditions and extract optimal features that provide a high performance of shadow detection and removal [10,11]. The convolution neural network (CNN) is the most popular feature extractor in computer vision tasks. CNN provides a significant development in the performance with the powerful deep extracted features [12]. Therefore, many researchers tend to use CNN to extract features for shadow detection and classification [13, 14]. Shadow detection methods based on CNN don't require a reference model for the background to detect shadow region.

This paper proposes to use Mask R-CNN deep learning framework [15] in the detection and segmentation of shadows. Using Mask R-CNN for shadow classification and segmentation achieves several improvements compared with previous literature methods:

1. Unlike most existing shadow detection methods that usually depend on extracting specific handcrafted features for shadow detection, Mask R-CNN Provides a powerful feature extractor for automatically shadow modeling with different distinctive features.
2. No need for background subtraction while Mask R-CNN performs accurate object detection from the raw input frame and thus overcomes many background subtraction issues such as (bootstrap, blurring, video noise).
3. Experimental results performed on CDnet dataset [16] and some public datasets (e.g. Highway, Caviar) under different environmental conditions of hard and soft shadows show that Mask R-CNN achieves average detection rate of 96.81%, average discrimination rate of 99.42%, and average performance accuracy of 98.09 % without additional post-processing comparing with most techniques in shadow detection literature.

The remaining parts of this paper include a review of the existing methods of shadow detection in section 2, an overview of the proposed Mask R-CNN framework in section 3, experimental results and discussions in section 4, and the conclusions in section 5.

II. RELATED WORK

Despite the existence of many studies of shadow detection since 1980, shadow detection remains avital research direction in most of the intelligent visual systems [17]. Prati et al. [4] labeled the methods of shadow detection according to the algorithm used in the detection process as deterministic algorithms or statistical algorithms. Sanin et al. [6] made taxonomy according to the distinctive shadow information used in the shadow detection process, such as chromatic features, geometrical features, and structural of texture features. According to recent advances in shadow detection methods presented in previous section, the existing shadow detection methods could be categorized to traditional, learning-based, and deep learning-based methods. Both traditional and learning-based methods are based on extracting handcrafted features from the current source image and/or corresponding background. The traditional methods and learning-based methods differ in the method of using and classifying the extracted features used in the process of shadow detection and removal. On the other side, deep learning-based methods provide schemes that can automatically extract and classify the most powerful features for the desired task.

Traditional methods rely on threshold schemes to distinguish between shadow and foreground features. Chromatic and texture features are important cues for identifying shadow areas. Accordingly, Cucchiara et al. [18] had employed hue, saturation, and value (HSV) color space to differentiate shadow pixels. The method of [19] exploited HSI color space along with the C1C2C3 color model. The method of [20] created a shadow model at the pixel level through imposing some constraints on chromaticity values, brightness, and RGB values.

When there is a color similarity between foreground and shadow regions (especially in gray-level frames), the reliance on color features alone became inefficient and not sufficient. Accordingly, most of the existing methods tend to use a mixture of features to detect shadows. The method presented in [21] utilized HSV color features and edge features along with irradiance information for modeling shadows. The methods presented in [22, 23] used color, texture, and gradient cues simultaneously for detecting shadow pixels. In [24, 25], shadow detected through fusing color and texture information using multiple descriptors for these cues. In [26], various features including color similarity, texture similarity, and optical reflection invariance are integrated based on shadow appearance. Wang et al. [27] designed several strategies using different illumination invariant descriptors for detecting shadows.

Learning-based methods depend on learning features for shadow classification to avoid the drawbacks of thresholding in the classification process. The learning-based methods use a classifier designed by the extracted features to separate shadow from moving objects. In [28], the shadow was formulated using the physical color features and the gaussian mixture model (GMM) was utilized to train and learn these features for shadow classification. In [29], the chromaticity and edge features extracted based on the background reference model, after that Support Vector Machines (SVM) utilized for shadow classification. In [30], different features based on the current video frame and the background model are extracted and then adopted the Partial Least Squares (PLS) with Logistic Discrimination (LD) for classifying shadows. Lin [31] proposed a scheme to extract features with different scales, next introduced a Multi-Layer Pooling Scheme (MLPS) for integrating and reducing the dimensionality of extracted features, finally, random forest method employed to classify shadow pixels. The method of [32] used the extreme learning machine (ELM) for learning shadow. Learning-based methods mainly depend on the manually extracted features and most of them are restricted to certain scenes, either indoor or outdoor video sequences.

Deep learning-based techniques support major steps in various problems of computer vision such as object detection and segmentation by providing schemes that can automatically extract the most powerful distinctive features [12]. For shadow detection from a still image, the method of [13] has extracted the features by using deep convolution neural networks (ConvNets) at the pixel level. Vicente et al. [14] introduced an active labeling method for a single image to obtain a large scale dataset then used this dataset by the semantic aware patch level CNN model to get a prior shadow map that refined based on patches of image semantics. Zheng et al. [33] introduced a distraction-aware network for learning and integrating pixels for semantic shadow segmentation. Lee et al. [10] proposed a deep learning method based on ConvNets for detecting moving shadow, they used the input image along with the corresponding background reference as inputs to the ConvNets model. Kim et al. [11] used VGG Net-16 a pre-trained convolution neural network to segment moving shadow. The method transferred the source image from RGB to HSV model then used the value, saturation channels with the value ratio of input image over the background image as the inputs to CNN model for learning shadow.

Although CNN outperforms in image classification and object detection but is not flawless. Despite the CNN efficiency in determining the class of a target, but CNN can't define its location [34]. CNN is used only for detecting one object in the image at a time and gives poor performance in case of the existence of multiple objects in the visual region (due to the interference of objects) [34]. Therefore, moving shadow detection-based CNN methods [10, 11] need to perform background subtraction process then divide the source image into windows to feed CNN.

Region-based convolutional neural networks (R-CNNs) are developed for applying deep models to detect and classify multiple objects from the input image [34]. R-CNN starts by employing a selective search approach [35] for extracting multiple region proposals where each proposal is a bounding box (Bbox) around the interesting boundary of an object in the image. Then, each region proposal is fed into CNN to generate output features. Finally, a set of Support Vector Machine (SVM) classifiers use these features to determine the class of each object within each region proposal [34]. Fast Region-based Convolutional Neural Network (Fast R-CNN) [36] improves the performance of R-CNN [34] by extracting features from the whole image instead of extracting features on each region proposal. The region proposals extracted through selective search then applied to ROI pooling scheme along with the CNN output features to classify and detect ROIs in the image. For more improvement, faster R-CNN [37] used a region proposal network (RPN) rather than the selective search utilized in Fast R-CNN. The use of RPN decreases the generated number of proposal regions while achieving accurate and fast object detection. Mask R-CNN [15] has the same framework as Faster R-CNN but introduced a fully convolutional network (FCN) [38] for locating objects on the pixel level by generating a binary mask for each object. Mask R-CNN improved accuracy of detecting objects by ROI alignment scheme (Fig.1). The proposed method exploits the efficiency of Mask R-CNN [15] in object localization, classification, and segmentation at a pixel level to detect and classify shadows. Table 1 summarizes the comparison of advantages and disadvantages of the previous methods and the proposed method for shadow detection.

Table 1. Comparison of the proposed framework and previous methods for shadow detection.

Category	Methods	Advantages	Disadvantages
Traditional techniques	HSV [18]	<ul style="list-style-type: none"> - Simple to implement - Less processing time - Achieve acceptable results with few features - Multiple features can be integrated to improve performance 	<ul style="list-style-type: none"> - Inefficient for most shadow conditions - Require prior knowledge about scene type - Have large misclassifications due to sensitivity to thresholds of classifying shadow - Need post processing to refine results - Mainly depend on background model for extracting used features.
	HSI with C1C2C3 [19]		
	RGB [20]		
	HSV with edge features [21]		
	Gradient and color [22,23]		
	Color and texture fusion [24,25]		
Optical reflection invariance [26]			
Illumination invariant cues [27]			
Learning based techniques	GMM [28]	<ul style="list-style-type: none"> - Good performance than traditional methods by learning different features - Don't need massive data for training - Less computational complexity than deep learning 	<ul style="list-style-type: none"> - Need to extract ideal handcrafted features - Require prior knowledge of the scene environment - Constrained to specific environment sense - Depend on background reference to extract features of shadow
	SVM [29]		
	PLS with LD [30]		
	MLPS [31]		
	EML [32]		
Deep learning based techniques	Patch level CNN [14,33]	<ul style="list-style-type: none"> - Better performance than traditional and learning based methods - Can handle different conditions of shadow by learning optimal features save effort in selecting and extracting features manually by convolution to extract features automatically 	<ul style="list-style-type: none"> - Can detect only one object at a time therefore divide image to small patches - Poor performance in case of the existence of multiple objects in the visual region. - Need background subtraction for object detection [10,11]. - Computationally expensive and need massive data for training
	ConvNets [13,10]		
	VGG Net-16 [11]		
	Mask R-CNN (proposed)		

III. THE PROPOSED FRAMEWORK

Mask R-CNN is a recent powerful, simple, and flexible model designed for instance segmentation (i.e. giving different labels for a single class) [15]. Mask R-CNN is an extension of the Faster R-CNN model [37] which is designed for object detection. Mask R-CNN adds an extra stage for generating binary segmentation masks for the detected objects as shown in Fig.1.

In this paper, Mask R-CNN is adapted and evaluated for moving shadow detection. Mask R-CNN [15] consists of two principal phases extended from Faster R-CNN. The first phase includes extracting features based on CNN then generating a candidate bounding box using Region Proposal Network (RPN) [37]. The second phase shares the extracted features to classify and segment objects for generating class labels, bounding box offset, and binary mask image for each instance in each Region of Interest (ROI). Figure 2 shows the architecture of the proposed framework for detecting and segmenting shadow at the pixel level. First, CNN is applied to the input frame for extracting features. ResNet-101 [39] is selected as a convolutional backbone for feature extraction in the proposed framework and for bounding box classification and regression. Then, the extracted feature maps are presented to a Region Proposal Network (RPN) which generates various bounding boxes according to the objectness

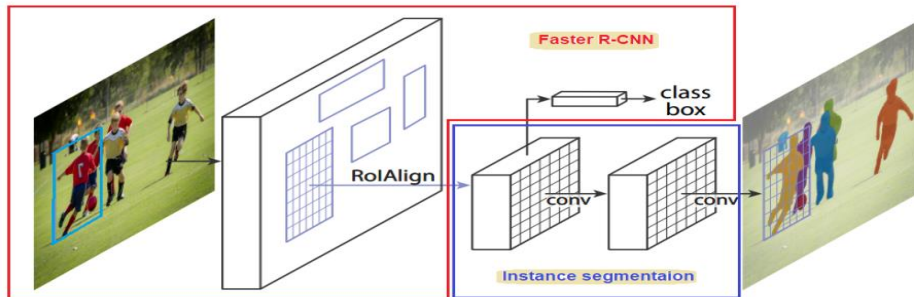


Fig.1. Mask R-CNN framework composed of Faster R-CNN along with FCN for object segmentation of each ROI [15].

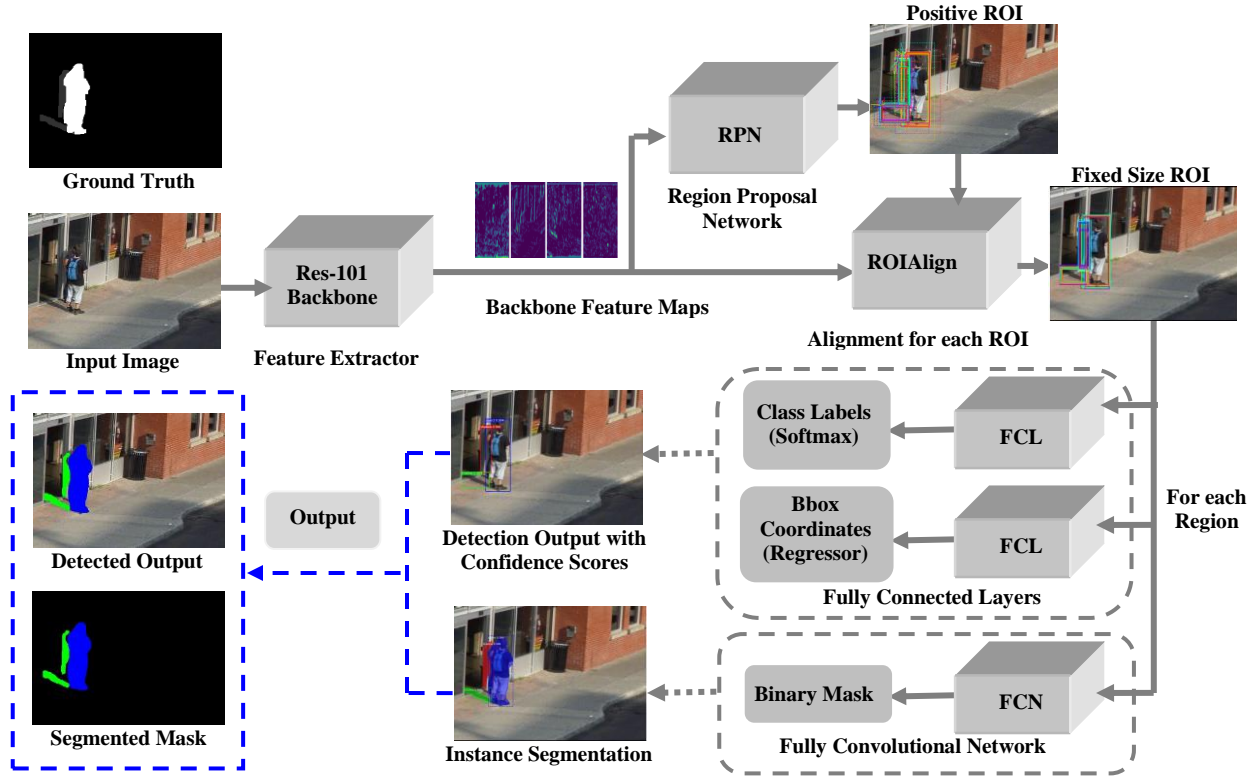


Fig.2. The Architecture of Mask R-CNN Framework for shadow detection and segmentation.

(i.e. the existence or absence of the object in candidate regions). The positive ROIs which contain moving objects including shadows are passed to the ROIAAlign phase for preserving accurate spatial adjustment for each ROI. The following step is to pass the resulted ROIs from the ROIAAlign phase to Fully Connected Layers (FCL) which produces the class labels of the objects in each specific ROI in addition to the coordinates of the bounding box for this ROI with its confidence score. Finally, in parallel with generating the bounding boxes and classes, a small FCN [38] is applied to the produced ROI's (generated from the ROIAAlign phase) to get a binary segmentation mask for each object in ROI.

A. Feature Extraction Based on CNN

In the first phase of Mask R-CNN [15], a feature extractor based on CNN is required for extracting high-level deep features from the source frame. Selecting the feature extractor is an important task due to the following factors: (1) The architecture of CNN; (2) The number of network parameters; (3) Layers type. These factors have direct effects on the learning and predication speed, utilized memory capacity, and the effectiveness of the model [40].

ResNet [39] are deep residual learning models to facilitate the training of heavily deeper networks to overcome the quick degradation of accuracy (with increasing error rates) as a result of the increasing network depth. By residual learning high-level deep features are obtained from deeper CNN models which are required for detecting and segmenting shadow robustly. ResNet [39] can handle up to 152 layers deep network by training the residual mapping functions rather than training the desired signal directly. Figure 3 (a) shows the core building block of residual network where ResNet depends on a skip links (or shortcut connections) to map the input from the earlier layer to the succeeding layers without any adjustment of the input. Skip links enable the network to go deeper without causing problems of accuracy degradation with increasing depth [39].

In the proposed framework, the deeper ResNet-101 [39] network is selected to extract features for modeling shadow. Figure 3 (b) shows the general architecture of ResNet-101 with parameter layers. The convolution and pooling occur in five stages with stride of two generating five feature maps at various scales. The ResNet-101 layer [39] has four blocks with each includes [3, 4, 23, 3] stacks of 3 layers, respectively. The three layers stack is convolutions layers of 1×1 , 3×3 , and 1×1 . The first 1×1 layer is capable of reducing dimensions; the last 1×1 layer is

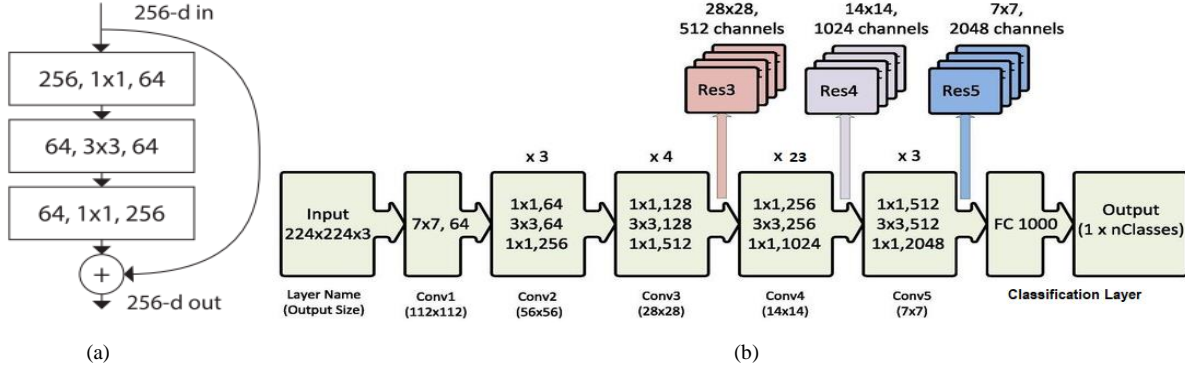


Fig.3. Example of general architecture for the residual network. (a) The building block of residual learning; (b) Architecture of ResNet-101 with parameter layers [39].

responsible for restoring (increasing) dimensions and the 3×3 layer acts as a bottleneck with smaller dimensions for input and output. Finally, a global average pooling with a 1000-path fully-connected layer using Softmax function [41] is conducted for the classification. During the experimentation of this framework, the input frames are enlarged to keep its spatial information as possible during the convolutions and pooling operations to extract high-level features. After that, these input frames are fed to the CNN feature extractor. The classification layers from the ResNet-101 network are excluded to be used as a feature extractor. The recommended layers presented in [15] are used for ResNet-101 where the last convolution layer of the fifth stage (called Res 5) is used to generate the feature map that has been shared for detecting ROI, classification, and segmentation.

B. Object Detection Based on RPN

Region Proposal Network (RPN) [37] applies a light binary classifier on various sets of predefined anchors (Bounding boxes) over the whole feature map produced from CNN and then determines the objectness score that defines if there is a foreground object on the candidate Bbox or not. Figure 4 (a) illustrates that RPN is initialized by moving a sliding window over the extracted CNN feature image. For each window, a set of anchors with predefined scales and aspect ratio are centered at the sliding window. These candidate anchors are checked to detect if there is an object or not (defining objectness score). The candidate Bbox that has a foreground object is considered a positive region of interest (ROI) if it has the highest value of Intersection-over-Union (IOU) (IOU is the intersection area between the predicted ROI and its ground truth ROI divided by the union area of the two regions), or if it has an IOU probability higher than 0.7 (set empirically with a relatively high value to detect only the ROI with high confidence). The negative Bbox (which does not cover any foreground object and considered to be background region) has IOU less than 0.3 (set empirically). Bbox that has IOU between 0.3 and 0.7 is excluded from the classification.

RPN applies Non-Maximum Suppression (NMS) [37] scheme to eliminate redundant and overlapping ROI proposals according to their IOU score where the Bboxes with low scores are eliminated and Bboxes with a high score of the objectness (i.e. positive Bbox) are qualified to the classification stage. After that, each candidate Bbox is mapped to low dimensional vector and then fed to two siblings Fully Connected Layers (FCL). The first layer is the regression layer that produces $4N$ representing the coordinates of N anchor boxes. The second layer is the classification layer that produces $2N$ probability scores to determine the presence or absence of the foreground object at each proposal. Usually, a positive Bbox does not completely cover the entire object. So, RPN regressor improves the surrounding Bbox by shifting and resizing this Bbox to the nearest correct boundaries of objects. Figure 4 (b) shows detection example using RPN on the selected CDnet dataset [16]. The detected positive ROIs produced by RPN are fed to the next stage for Bbox regression and foreground object label classification. The proposed framework has three categories which are foreground object class, shadow class, in addition to the background class.

C. ROI Align and Classification

The positive ROIs extracted from RPN along with the backbone feature map are used to map each ROI from the original image to its corresponding feature patches in the backbone feature map to be fed later into an ROI pooling layer for the classification step and Bbox regression [36, 37]. The ROI pooling layer scale the blocks of each ROI feature map that has a variable size into small predefined size (e.g. 7×7) to pass into a Fully Connected Layer (FCL).

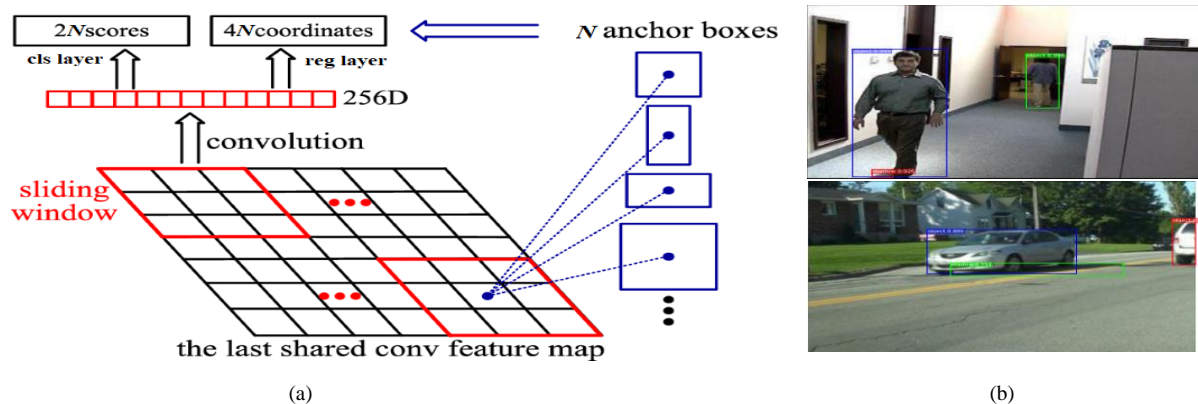


Fig.4. (a) A description of Region Proposal Network [37]; (b) Detection example using RPN proposals.

The scaling is performed by: (1) Dividing the ROI feature map into a number of grids equal to the dimensions of the output map and often the size of the grids is identical; (2) Finding the maximum or average value on each grid then put these values to the output map.

During dividing ROI into grids, ROI pooling performs quantization (i.e. rounds the floating coordinates to integer values for convenience) when the coordinates of ROI are floating points. This quantization leads to loss of data and misalignment with the actual ROI feature map. Figure 5 (a) shows an example of how ROI pooling has this data loss that may occur from floating-point division. Data loss resulting from quantization occurs twice, the first loss occurs because of the coordinate mapping of the input image to the coordinates of the whole feature map. According to the example shown in Fig.5 (a) the input image in addition to the ROI coordinates is scaled by 32. Floor function may be then performed to adapt the coordinates of ROI to remove the fractional part due to the scaling process (ROI should have size of 20 instead of 20.78). This flooring process causes the coordinate on the feature map to miss some portions from the original input frame. The second loss of data occurs when coordinates on the feature map are quantized at the ROI pooling layer. Flooring function may be used again to remove the fractional part due to the quantization process. This floor function results in the loss of some portion of the ROI feature map in addition to the loss of spatial resolution.

Although this quantization loss does not affect the classification and Bbox regression because they are invariant to soft translations, but it has a bad effect on predicting accurate pixel masks because of the need for robust spatial localization. Mask R-CNN [15] employed ROIAlign layer to preserve the spatial localization of features without losing data. ROIAlign uses a bilinear interpolation [42] instead of the quantization to smoothly convert ROI feature maps that have different sizes into fixed feature vectors. Bilinear interpolation [42] keeps the floating coordinates of the ROI feature map and enables the pixel's value of the floating-point coordinates to be calculated.

Figure 5 (b) describes the general process for ROIAlign. The ROI Align is performed through the ROI pooling layer by dividing the ROI feature map into $n \times n$ grid. Each grid has a set of units and it doesn't perform quantization to the boundary of each grid unit. For each grid unit, the unit is divided equally by four regions and then the center pixel values for these four regions in the grid unit are determined. These centers already have floating points.

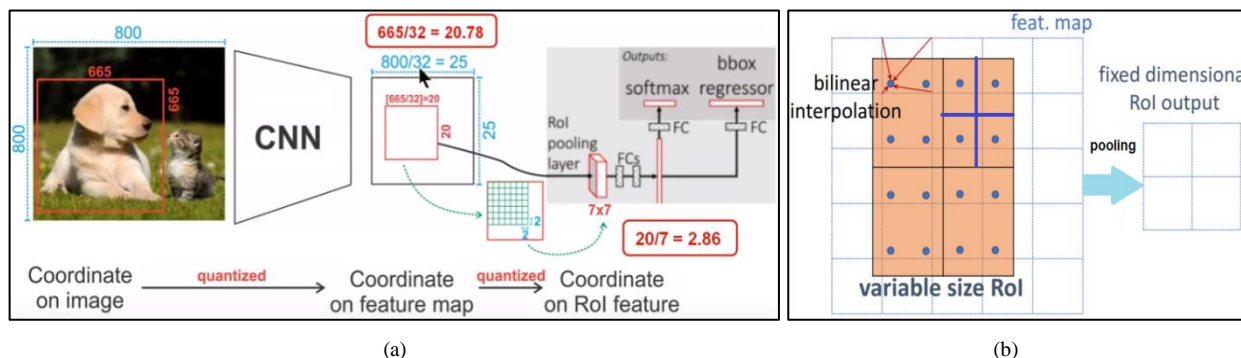


Fig.5. (a) Example of ROI pooling [36]; (b) Description of ROI Align [15].

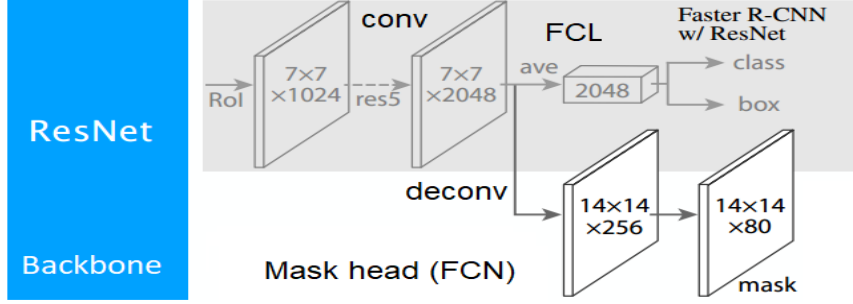


Fig.6. The head architecture for classification, Bbox regression, and mask prediction in Mask R-CNN with ResNet backbone [15].

Therefore, bilinear interpolation [42] is used to calculate its value from the neighboring grid locations in the feature map. After that, Max-pooling is performed for each grid unit on these four sampled locations. This step generates fixed-sized ROI feature maps while maintaining their spatial localization without losing image data. These maps are then mapped to fixed feature vectors for ROI classification and Bbox regression by FCL (Fig.6).

Figure 6 shows the head structure of Mask R-CNN for classification, Bbox offset, and the mask prediction. As shown in Fig.6, FCL takes the fixed feature vectors then applies a Softmax function [41] to get the probabilities for classifying each ROI's target into classes (e.g. shadow and moving object). Furthermore, FCL performs bounding box regression offsets to refine per-class bounding boxes relative to the actual ROI and this produces four values for each ROI Bbox ((x_c, y_c) coordinates of center point, width, and height of Bbox).

D. Binary Mask Prediction

Along with generating classes and Bbox offsets, Mask R-CNN also produces a binary mask that represents the spatial structure of object pixels by adding Fully Convolutional Network (FCN) [38] for segmenting each positive ROI. After the ROI align and pooling, the ROI feature map is applied to two more convolution layers for building the mask. The mask encodes the spatial appearance of the input object. For each positive ROI aligned by the ROIAlign layer, FCN [38] predicts 2D mask which enables each convolution layer in the mask prediction network to preserve specific $m \times m$ spatial structure for each object without dropping this ROI into fixed feature vectors as in FCL for label classification and Bbox offsets. As shown in Fig.6, FCN takes the pooled feature map of each ROI and performs the ROI up-sampling by using transpose convolution layer to convert the small-sized feature map to the same size as the input ROI. After that, one more Convolution layer followed by a binary classifier such as sigmoid function [39] is applied to produce the probability of pixel belonging to the foreground class or background. As each ROI has certainly at most one class (e.g. shadow or object) so FCN is trained as a binary classifier (i.e. FCN needs to learn to map input pixels to 1 or 0 according to the probability of each pixel where 1 refers to the presence of a foreground class and 0 will label the background class).

E. Loss Function

Mask R-CNN applies a multi loss function [15, 37] during the learning to evaluate the model and ensure its fitting to unseen data. This loss function is computed as a weighted total sum of various losses during the training at every phase of the model on each proposal ROI [36]. This weighted loss defined as:

$$\text{Loss} = L_{\text{Class}} + L_{\text{Bbox}} + L_{\text{Mask}} \quad (1)$$

Where L_{Class} (The loss of classification) shows the convergence of the predictions to the true class. L_{Class} combines the classification loss during the training of RPN and Mask R-CNN heads. L_{Bbox} (The loss of bounding box) shows how well the model localizes objects and it combines the Bbox localization loss during the training of RBN and Mask R-CNN heads. L_{Class} and L_{Bbox} losses are computed as follows:

$$L_{\text{class}}(p, u) = -\log p_u \quad (2)$$

$$L_{\text{Bbox}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} [L_1^{\text{smooth}}(t_i^u - v_i)], \quad \text{where } L_1^{\text{smooth}}(x) = \begin{cases} 0.5 x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (3)$$

The term p_u represent the predicted probability of ground truth class u for each positive Bbox. t_i^u, v_i represent the predicted Bbox for class u and ground truth Bbox, respectively. The term i represent Bbox coordinate offset (center coordinate (x, y) , width, and height). The mask head has a dimensional output of Km^2 as it generates k binary masks of size $m \times m$ for each one class of the K classes. L_{Mask} (The loss of segmenting the predicted mask) is

computed as the average binary cross-entropy for the predicted masks associated with the ground truth and it is computed as follows:

$$L_{\text{Mask}} = -\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [g_{ij} \log p_{ij}^k + (1 - g_{ij}) \log(1 - p_{ij}^k)] \quad (4)$$

where g_{ij} is the class label probability of a pixel (i, j) in the ground truth mask for ROI of size $m \times m$, p_{ij} is the predicted label of the same pixel in the mask generated for class k .

IV. EXPERIMENTAL RESULTS

A. Preparing the Dataset

The proposed framework is tested and evaluated using CDnet¹ [16] dataset along with the public Highway and Caviar video scenes² of moving shadow. The description details of these datasets are provided in Table 2 [5, 16]. CDnet dataset includes four video scenes having different environmental conditions with both soft and strong cast shadow. The dataset frames are annotated as pixel-wise semantic segmentation where each pixel in the image is labeled with a class. These classes are shadow, moving object, in addition to the background. For training Mask R-CNN the dataset images are converted into COCO format [44] and saved as JSON file. As shown in Fig.7(c), the JSON format for object detection and segmentation is a collection of attributes: "images" contains information about the image, "categories" contains information about the classes and their unique identifier, and "annotation" contains a list of segmentation points (x, y) pixel coordinates for the shape of each object in the image. The model mainly concentrates on (x, y) coordinates of each ROI. This paper introduces a simple method to convert the ground truth dataset from labeled images to JSON file format. The first step to convert the ground truth dataset from labeled images to JSON file format is to divide the dataset into three different datasets: training, validation, and testing with a ratio of 60%, 20%, and 20%, respectively. There is only one JSON file for each one of the three datasets. The next step is to create the JSON file for every image in the ground truth of each dataset through finding the external boundary of each ROI. Then, coordinates (x, y) for each boundary are listed as a vector of pixels and each coordinate is characterized with an ID and class label. Finally, this data are written in .JSON file as shown in Fig.7.

B. Training and Testing the Proposed Framework

Training and testing are conducted using the source codes of matterport Mask R-CNN³ implemented with Python 3, TensorFlow, and Keras. The proposed framework is trained online on Google Colab⁴ [43] with free GPU and memory resources. The testing and the evaluation values of the proposed framework have resulted from epoch 50 of the trained model which takes around 5.5 hours to train on 8610 images of the training dataset. The framework produces bounding boxes containing the segmented masks for each class in the image as shown in Fig.8.

ResNet-101 network [39] is used as backbone architecture for feature extraction in the proposed framework. To avoid the model overfitting and enhance the performance of the model [43], the transfer learning from the pre-trained Mask R-CNN model on COCO dataset [44] is used to initialize the weights of the CNN feature extractor and to reuse most parts of the pretrained model with modifying its final output layers to cope with three resulted classes (i.e. shadow, object, and background). This paper follows the open source implementation³ of Mask R-CNN and performs parameters fine tuning to fit the pre-trained model with the selected shadow dataset. The input frames of the network have 3 channels (i.e. R, G, and B) and are resized to 512×512 . For training Mask R-CNN, the optimal parameters fine-tuning is determined experimentally. To find the optimal weights of the model to reduce the error between the predicted and desired output, the model is optimized by the stochastic gradient descent (SGD) [45] algorithm with the following hyper-parameters; an initial LEARNING_RATE of 0.001, LEARNING_MOMENTUM of 0.9, and WEIGHT_DECAY of 0.0001. Momentum is used to reduce the fluctuations in weight variations over continuous iteration that work on waiting after the weight is updated. The weight in the next time is updated using a weight decay amount. This speeds the training gradually, with a minimized risk of variation [45]. The learning rate is low to allow the loss value to be reduced gradually in a linear form to reach the optimal training output.

¹ <http://www.changedetection.net/>




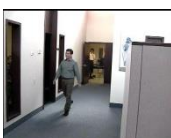


² <http://arma.sourceforge.net/shadows/>

³ https://github.com/matterport/Mask_RCNN/

⁴ <http://colab.research.google.com/>

The USE_MINI_MASK hyper-parameter which is employed for enhancing the training speed and saving loading memory by resizing the extracted masks to a smaller size has a large effect on mask prediction performance. This paper focuses on detecting and segmenting accurate masks having spatial structure as the desired output masks, therefore the USE_MINI_MASK hyper-parameter is set to False to disable the masks from being smaller than the desired output. The remaining parameters of Mask R-CNN training are described in Table 3. The proposed model is trained for 50 epochs and the loss function is evaluated for each epoch during the learning process for both the training data and the validation data.

Table 2. The description information of CDnet¹ dataset and some public² video scene.

Dataset	Video Type	Video length	Labeled Frames	Frame size	Object type	Detail description
 Bungalow	Outdoor	1700	1400	240×360	Vehicle	<ul style="list-style-type: none"> – Strong chromatic shadow – Large shadow and objects – Medium noise level – Non textured surfaces – Have self-shadows – Foreground-shadow camouflage
 People in shade	Outdoor	1200	1050	244×380	People	<ul style="list-style-type: none"> – Strong and overlapped shadow – Small shadow and medium objects – High noise level – Foreground-background camouflage – Dark and non-textured surfaces
 BusStation	Outdoor	1250	950	360×240	People	<ul style="list-style-type: none"> – Strong shadow – Medium shadow and objects – High noise level – Multiple objects – Static shadow in surface
 Cubicle	Indoor	7400	6300	352×240	People	<ul style="list-style-type: none"> – Strong and weak shadow – Medium and large shadow – Medium objects – High noise level – Foreground-background camouflage – Non textured surfaces
 Caviar	Indoor	2360	1113	384×288	People	<ul style="list-style-type: none"> – Weak shadow – Small objects and Medium shadow – Foreground-background camouflage – Achromatic shadow – Non textured surfaces
 Highway 1	Outdoor	440	10	240×320	Vehicle	<ul style="list-style-type: none"> – Large and strong shadow – Small and medium objects – Foreground-background camouflage – Foreground-shadow camouflage – Dark surface with static shadow

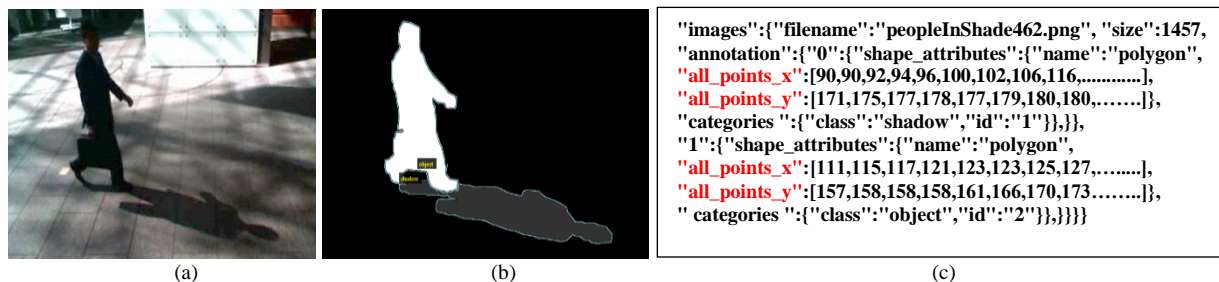


Fig.7. Ground truth images to JSON file Conversion; (a) Input image; (b) External boundary of each region; (c) JSON file format.

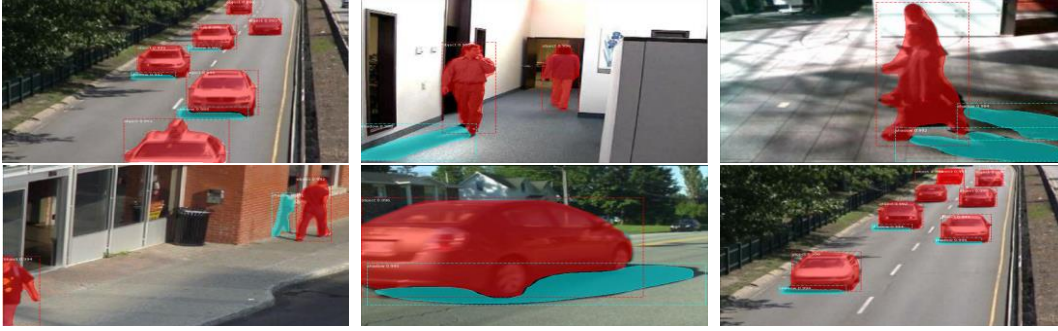


Fig.8. Detection outputs of some examples in various scenes (transparent green for shadow and transparent red for object).

Table 3. Tuning the hyper-parameters for training Mask R-CNN³ [15].

Hyper-parameters	Value	Definition
BACKBONE	resnet101	The backbone network architecture
BACKBONE_STRIDES	[4, 8, 16, 32, 64]	The stride of each layer of Res-101 backbone
GPU_COUNT	1	Number of used GPU
IMAGES_PER_GPU	3	Number of images to train per one GPU
BATCH_SIZE	3	Total number of images to process by number of GPU_COUNT
STEPS_PER_EPOCH	100	Number of training steps for epoch
VALIDATION_STEPS	150	Number of validation steps run at the end of each training epoch
NUM_CLASSES	3	Number of classes (background, object, and shadow)
BBOX_STD_DEV, RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]	Standard deviation of Bbox refinement for RPN and final prediction
DETECTION_MAX_INSTANCES	100	Maximum number of final detected ROI
DETECTION_MIN_CONFIDENCE	0.9	Minimum confidence threshold to accept the final detected ROI
DETECTION_NMS_THRESHOLD	0.3	Threshold of Non-maximum suppression for detection
IMAGE_CHANNEL_COUNT	3	Number of channels per image (e.g. RGB)
IMAGE_MAX_DIM,IMAGE_MIN_DIM	512,512	Define input image resizing (use padding with zero)
LOSS_WEIGHTS (Loss balancing parameters)	"rpn_class_loss": 1	The loss weight to classify presence/absence of any object by RPN
	"rpn_bbox_loss": 1	The Bbox localization loss of the RPN
	"mrcnn_class_loss": 1	Used to tackle incorrect classification of objects present in ROI
	"mrcnn_bbox_loss": 1	This loss assigned for Bbox localization of the detected class
	"mrcnn_mask_loss": 1	This loss determined to masks defined for each object in ROI
POOL_SIZE	7	Pooled ROI size
MASK_POOL_SIZE	14	Pooled mask head size
MAX_GT_INSTANCES	50	The highest number of ground truth objects to detect in one image
PRE_NMS_LIMIT	6000	Number of ROIs kept before non-maximum suppression
POST_NMS_ROIS_INFERENCE	1000	Number of ROIs kept after NMS in inference (testing)
POST_NMS_ROIS_TRAINING	2000	Number of ROIs kept after NMS in training
TRAIN_ROIS_PER_IMAGE	512	Number of ROIs per image applied to the classifier and mask head
ROI_POSITIVE_RATIO	0.33	Ratio of positive ROIs used for training the classifier and mask head
RPN_ANCHOR RATIOS	[0.5, 1, 2]	Ratios of Bbox (e.g. width/height) at each cell in feature map
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)	Length of square Bbox side in pixels (Anchor scales)
RPN_ANCHOR_STRIDE	1	The amount of movement that the anchor slide over the feature map
RPN_NMS_THRESHOLD	0.7	Non-max suppression cutoff to filter proposal detected by RPN
RPN_TRAIN_ANCHORS_PER_IMAGE	256	Number of Bbox anchors per image used to train RPN
USE_MINI_MASK	False	Enable to resize ROI masks to smaller size to reduce memory load

The learning of the proposed model is performed through the training dataset, while it uses the validation dataset to calibrate the training efficiency and reduce network overfitting [11]. If the efficiency of the model on the training dataset is high but remains unchanged or decreases when the model is trained on the validation dataset (in other words, if the training loss is reduced but the validation loss has not changed or increased), then this means that the model fall into the problem of overfitting [43]. To monitor the performance of the proposed model, the loss values are obtained during the training and validation in each epoch. Figure 9 shows the loss outcomes during the training

of the proposed framework. As shown in Fig.9, at the increasing of the epoch, both the training and validation loss values are going close to 0 with the convergence of both in output value which is considered as an indication of the accuracy of the proposed framework.

In order to assess the efficiency of the proposed framework, the confusion matrix [43] on the testing dataset is provided in Table 4. As shown in the confusion matrix, the values from left to right in first row represent true-positive (TP) percent of detecting shadow correctly and false-positive (FP) percent of misclassifying shadow as object. The second row represents false-negative percent (FN) of misclassifying object as shadow and true-negative percent (TN) of detecting object correctly.

The values of the confusion matrix in Table 4 show that, the object and shadow have been classified correctly with high accuracy. The false-negative rate is very small while the false-positive of classifying shadow as an object is relatively higher compared to the false-negative rate because the shadow in most cases of the datasets is spatially connected to the moving object and cast as a part of the object, thus it is difficult to separate the shadow from object without mistakes. Based on the confusion matrix outcomes, three other measures (detection, discrimination, and f-measure rates) are determined to evaluate the proposed framework quantitatively.

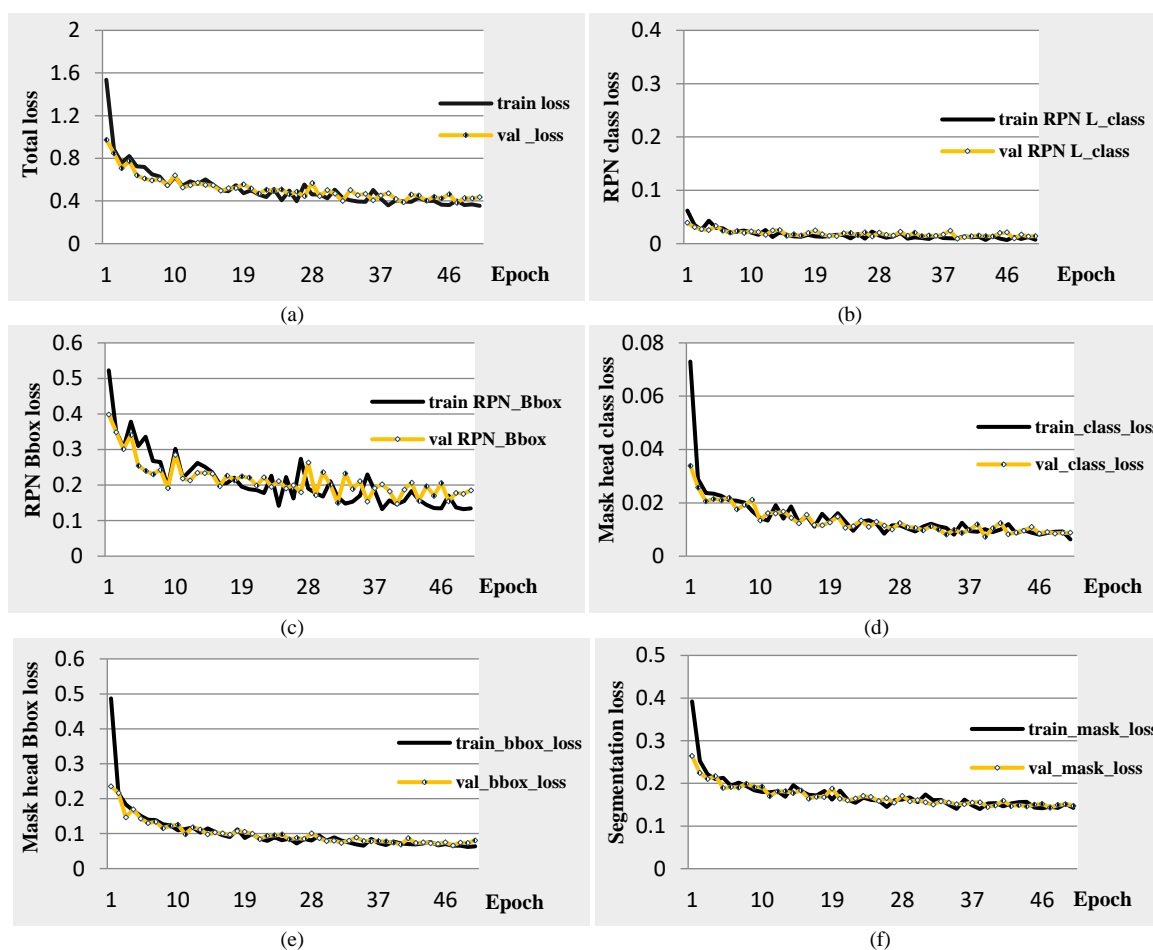


Fig.9. The training and validation loss per epoch. (a) Total training and validation loss; (b) RPN classification loss; (c) RPN Bbox localization loss; (d) Mask R-CNN classification loss; (e) Mask R-CNN Bbox localization loss; (f) Mask R-CNN segmentation loss.

Table 4. The confusion matrix of testing outcomes (unit %)

		Ground truth	
		Shadow (Positive)	Object (Negative)
Predicted	Shadow (Positive)	95.99	4.01
	Object (Negative)	0.53	99.47

C. Quantitative Evaluation

For validating the performance of the proposed framework, six dataset videos from CDnet and public datasets for indoor and outdoor scenes are selected. All the videos are real-world scenes with complex and normal situations of shadows. The ground truth of the videos is available. For quantitative evaluation, the common metrics: shadow detection rate (η) and shadow discrimination rate (ζ) are evaluated [4]. Also, the *F-measure* metric [23] is computed for validating the precision of the proposed framework. These metrics are determined as follows:

$$\eta = \frac{TP_{sh}}{TP_{sh} + FN_{sh}} \quad (5)$$

$$\zeta = \frac{TP_{obj}}{TP_{obj} + FN_{obj}} \quad (6)$$

$$F - measure = \frac{2 \times \eta \times \zeta}{\eta + \zeta} \quad (7)$$

Shadow detection rate (η) indicates the percentage of the output pixels which are relevant to the shadow class. Shadow discrimination rate (ζ) refers to the percentage of the output pixels which are relevant to the object class. Both TP_{sh} and TP_{obj} are the total numbers of true-positive pixels that are correctly classified as shadow or object, respectively. Both FN_{sh} and FN_{obj} are the total numbers of false-negative pixels that are falsely classified as shadow or object, respectively. The *F-measure* metric is the harmonic mean of shadow detection rate and shadow discrimination rate. The quantitative outcomes obtained from the proposed method are compared with several methods of the two categories of shadow detection methods (traditional and learning-based).

Table 5 and Fig.10 summarize the comparison of the quantitative results for the proposed method and six comparative methods: physical features based method (Phy [28]), large regions texture-based method (LR [6]), multiple local features fusion method (MFF [23]), detection using optical reflection invariant features (ORI [26]), shadow detection based on scaled-relation multi-layer pooling features (SMPF [31]), and detection based on illumination invariant features (IIF [27]).

As shown in Fig.10, Phy [28], MFF [23], and LR [6] have the lowest average shadow detection rate (less than 65%), these methods cannot differentiate between pixels of shadow and object in most situations as in outdoor scenes. These methods depend on a few handcrafted cues such as color or texture that cannot adapt to different conditions of video surrounding environment such as camouflage issues and illumination changes as in Bungalows and People in shade datasets. Also, their sensitivity to thresholds that used for shadow classification produces high misclassification pixels which lead to the degradation of the shadow detection performance. ORI [26], SMPF [31], and IIF [27] achieved a good average shadow detection rate between 86 % - 94 % .These methods restricted with different assumptions about the appearance of shadow. These methods rely on adapting multiple feature cues

Table 5. Shadow detection outcomes of the proposed framework and comparative methods on the selected datasets.

Video scene	Metric %	Phy [28]	LR[6]	MFF[23]	ORI[26]	SMPF[31]	IIF [27]	Proposed
Bungalows	η	15.79	70.23	3.57	85.02	91.00	77.3	93.72
	\mathcal{E}	89.79	64.16	94.57	78.08	97.80	85.94	98.11
	<i>f-measure</i>	26.85	67.06	6.87	81.4	94.27	81.39	95.86
People in shade	η	8.96	32.99	35.56	75.9	94.95	84.06	97.48
	\mathcal{E}	97.31	97.37	90.32	94.4	97.62	93.08	99.78
	<i>f-measure</i>	16.42	49.29	51.02	84.15	96.27	88.34	98.62
BusStation	η	60.15	36	63.49	65.18	94.23	87.64	98.02
	\mathcal{E}	94.3	94.33	89.72	95	99.64	89.09	99.85
	<i>f-measure</i>	73.45	52.11	74.36	77.31	96.86	88.36	98.92
Highway1	η	42.47	60.46	65.69	84.62	81.22	78.13	95.10
	\mathcal{E}	85.69	95.48	91.88	93.63	93.74	92.72	99.23
	<i>f-measure</i>	56.79	74.04	76.61	88.9	87.03	84.8	97.12
Cubicle	η	67.11	89.4	69.64	89.81	98.09	98.44	98.76
	\mathcal{E}	91.1	94.66	84.39	87.19	99.86	90.13	99.96
	<i>f-measure</i>	77.28	91.96	76.31	88.48	98.97	94.1	99.36
Caviar	η	40.69	73.23	92.55	94.91	—	95.71	97.75
	\mathcal{E}	86.68	98.83	84.88	94.55	—	77.84	99.60
	<i>f-measure</i>	55.38	84.13	88.55	94.73	—	85.86	98.66

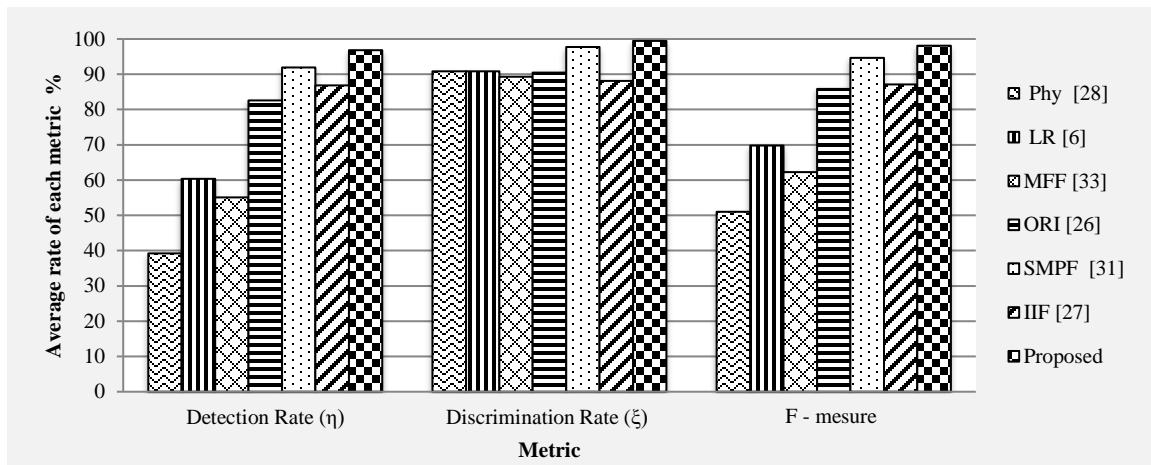


Fig.10. Average rate comparisons of proposed method with comparative methods in Table 5.

according to scene types and environmental condition for shadow detection. The proposed method achieved the highest average shadow detection rate (96.81%) in all dataset video scenes compared to the other existing methods.

Although all comparison methods achieved acceptable values for the shadow discrimination rate with a value between 88% - 97%, the proposed method achieved the highest average discrimination rate of 99.42%, especially in datasets of People in shade, BusStation, Cubicle, and Caviar with a value close to 100%. Because the proposed model had learned through using a massive number of data samples that cover different conditions for the appearance of the shadow foreground objects in different environments (indoor and outdoor scenes). In addition, the proposed model relies on a deeper feature extractor such as ResNet-101 that can produce deeper high-level semantic features that have helped to improve the process of shadow detection and segmentation. The proposed model achieved the highest average accuracy (F-measure) of 98.09% compared to other methods.

According to Table 5 and Fig.10, it is concluded that the proposed framework outperforms several traditional and learning based methods in average detection rate, average discrimination rate, and average F-measure rate for all video scenes. Figure 11 summarizes the quantitative results of the proposed method and some comparable deep learning methods using Caviar dataset. The comparative methods include shadow detection based on: (1) Transfer learning from pre-trained AlexNet [46]; (2) Shadow detection utilizing 7 layers CNN [10] using source image with the background as an input to CNN; (3) Shadow detection exploiting CNN of VGG Net-16 [11] using background subtraction for foreground regions extraction.

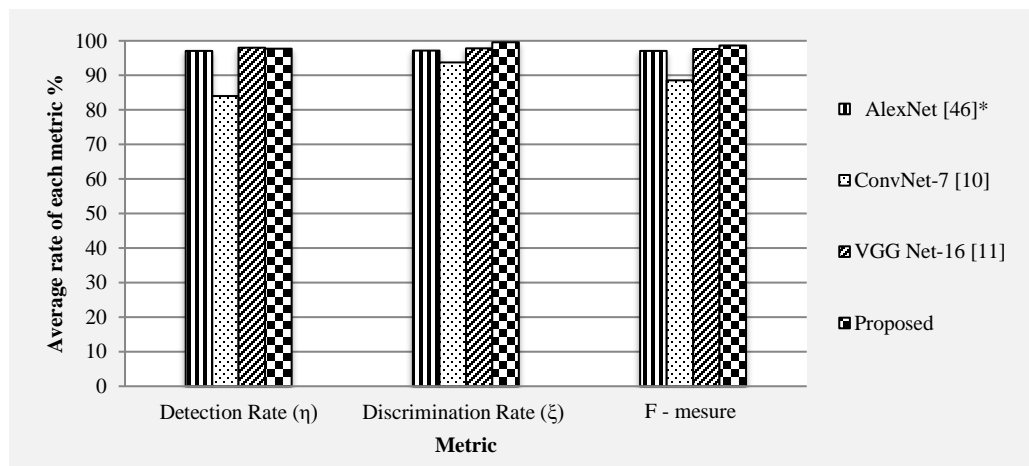


Fig.11. Average rate comparisons of proposed method with some comparable deep learning methods in Caviar dataset. (* The estimated values of [46] are reported in [11]).

As shown in Fig.11 deep learning-based methods have achieved a noticeable improvement in shadow detection accuracy compared to traditional methods. Although the methods that have deeper CNN [11, 46] achieve better accuracy compared to the shallow CNN of [10], the proposed shadow detection method using mask R-CNN [15] has superiority in shadow detection with accuracy of 98.66% for the following reasons: (1) The proposed framework depends on very deep ResNet-101 that provides strong high-level semantic features from the entire RGB input image rather than dividing the input image into small windows. These features have improved the segmentation and detection compared to ConvNets in [10, 11, and 46]; (2) The proposed framework is trained effectively by various challenging situations for shadow appearance in indoor and outdoor scenes. Moreover, one of the advantages of the proposed framework is the object detection without the need for a background subtraction technique and post-processing operations.

D. Qualitative Evaluation

Figure 12 shows a sample of the proposed model visual outcomes in different frames and scenes. The model is able to classify and separate accurately the shadow from the foreground object. To demonstrate the superiority and the efficiency of the proposed model, the qualitative comparison outcomes generated by the proposed method and several comparable methods on six diverse video scenes are shown in Fig.12. The first row (a) of Fig.12 is the input frames generated from the video scenes given in Table 2, second row (b) is the ground-truth used for evaluation process, the rows (c)-(g) are the segmentation results of different comparable methods, and the last rows (h) - (i) are the segmentation masks and the detected outputs resulted from the proposed model.

The results of Phy [28], LR [6], and MFF [23] appear to be unstable, and can't adapt to different illumination conditions that result in high misclassifications between shadow and objects as shown in all video scenes. ORI [26] and IIF [27] provide good results but still have little misclassifications in some complex shadow scenes. The proposed model provides accurate and detailed segmentation masks for object and shadow (e.g. Bungalows) because the proposed method learned with enough and various dataset samples. The proposed model visually outperforms several comparable methods in all video scenes under strong and different conditions of illumination.

V. CONCLUSION

In this work, the Mask R-CNN [15] deep learning framework is evaluated and tested to automatically perform semantic segmentation to detect and classify shadow pixels directly from the whole raw images without background modeling. Mask R-CNN extracts the most significant features from images in a supervised way using the strong ResNet-101[39] network that helped in enhancing the performance of shadow classification and segmentation. The model depends on the Region proposal network (RPN) [37] that detects the regions of interest that contain foreground objects and then generates a binary segmentation mask using a small FCN [38] that applied to each class detected in ROI. The proposed model is trained by transfer learning from the pre-trained Mask R-CNN model on the COCO dataset [44] with fine-tuning the hyper-parameters to fit the new shadow dataset quickly and effectively.

The proposed model is tested and evaluated quantitatively and qualitatively on a selected shadow dataset from a CDnet¹ benchmarks and a public shadow dataset². The experimental results are performed under different conditions of hard and soft shadows in order to prove that Mask R-CNN achieves high performance with an average detection rate of 96.81%, average discrimination rate of 99.42%, and average performance accuracy of 98.08% in comparing with several techniques in shadow detection literature. The results validate that with using better training datasets, Mask R-CNN will be a promising method for moving shadow detection and segmentation.

Mask R-CNN has many tasks that include localizing, classifying, and segmenting objects from the input image. Although the proposed framework achieves higher accuracy compared with other shadow detection methods, the computation time of the proposed framework is relatively high. Therefore, the proposed framework needs to be optimized for reducing computation time without affecting the accuracy of the performance. Reliance on GPU in the running of the proposed framework may contribute to decrease the processing time as it can process 5-8 frames per second.

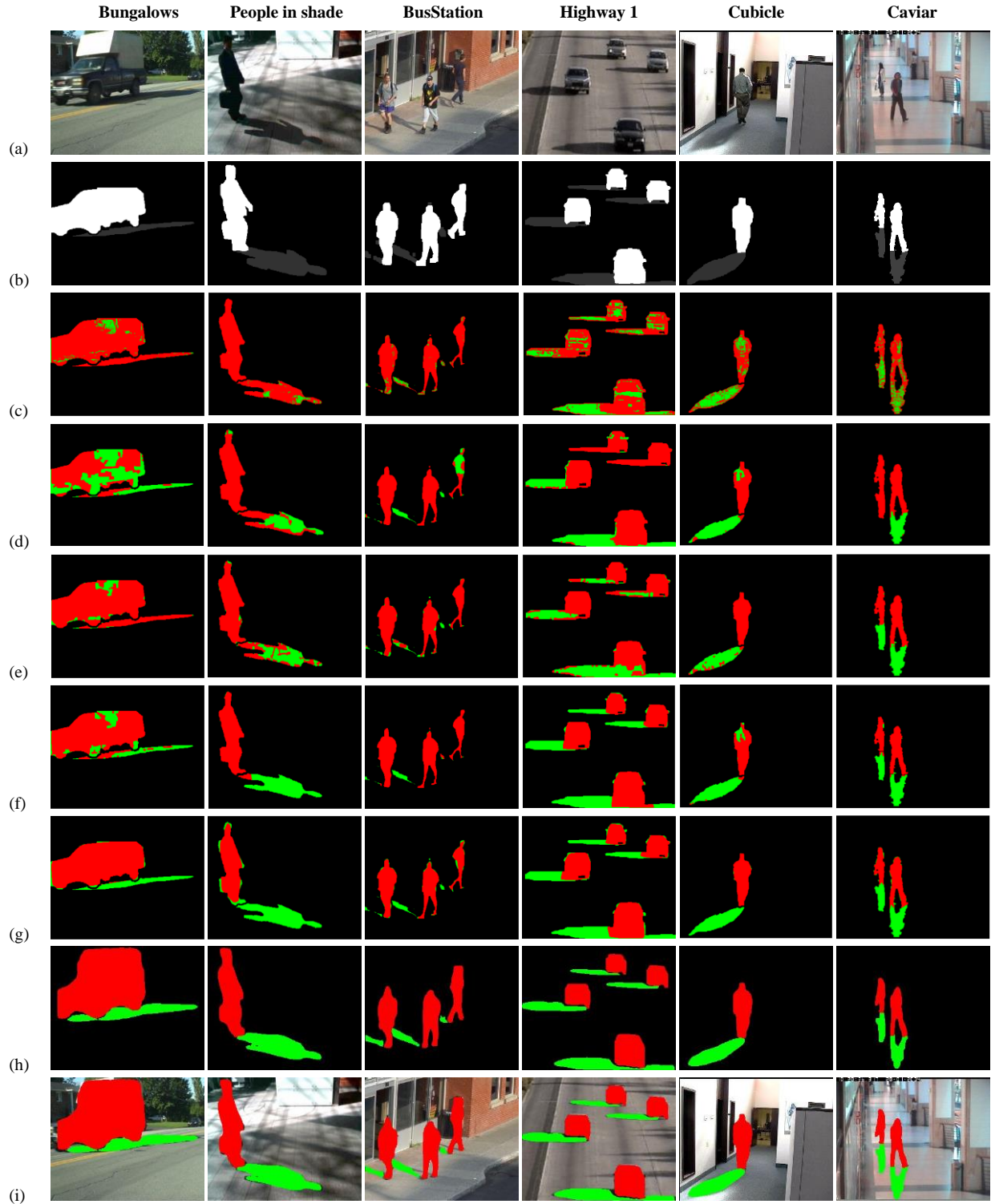


Fig.12. Visual results comparison (the shadow pixels labeled with green and object pixels labeled with red). (a) Input frames; (b) Ground truth masks; (c) Segmentation masks resulted from Phy [28]; (d) Segmentation masks resulted from LR [6]; (e) Segmentation masks resulted from MFF [23]; (f) Segmentation masks resulted from ORI [26]; (g) Segmentation masks resulted from IIF [27]; (h) Segmentation masks resulted from the proposed method; (i) Detected outputs of the proposed method.

REFERENCES

- [1] B. Johansson, J. Wiklund, P. Forssn, and G. Granlund, "Combining shadow detection and simulation for estimation of vehicle size and position," *Pattern Recognition Letters*, vol. 30, no. 8, pp. 751–759, 2009.
- [2] A. Amato, I. Huerta, M. G. Mozerov, F. X. Roca, and J. Gonzalez, "Moving cast shadows detection methods for video surveillance applications," In *Wide Area Surveillance*, Springer, Berlin, Heidelberg, pp. 23-47, 2014.
- [3] A. Sanin, C. Sanderson, and B. C. Lovell, "Improved shadow removal for robust person tracking in surveillance scenarios," in *International Conference on Pattern Recognition (ICPR)*, pp. 141–144, 2010.
- [4] A. Prati, I. Mikic, M. M. Trivedi and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 918-923, June 2003.
- [5] M. Russell, J. J. Zou and G. Fang, "An evaluation of moving shadow detection techniques," *Computational Visual Media*, vol. 2, no. 3, pp.195-217, 2016.
- [6] A. Sanin, C. Sanderson and B. C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern recognition*, vol. 45, no. 4, pp. 1684-1695, Apr 2012.
- [7] S.Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," *CVPR*, Providence, RI, pp. 1937-1944, 2011.
- [8] M. Piccardi, "Background subtraction techniques: a review," In *IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, vol. 4, pp. 3099 – 3104,2004.
- [9] M. Hayat, M. Bennamoun, and S. An, "Deep reconstruction models for image set classification," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 37, no. 4,pp. 713–727, April 2015.
- [10] J.T. Lee, K.T. Lim and Y. Chung," Moving shadow detection from background image and deep learning," In *Image and Video Technology*; Springer: Cham, Switzerland, pp. 299–306, 2016.
- [11] D.S. Kim, M. Arsalan and K.R. Park, "Convolutional Neural Network-Based Shadow Detection in Images Using Visible Light Camera Sensor," *Sensors*, 18, 960, 2018.
- [12] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, 2018.
- [13] S. H. Khan, M. Bennamoun, F. Sohel and R. Togneri, "Automatic Shadow Detection and Removal from a Single Image," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 431-446, 1 March 2016.
- [14] T.F.Y. Vicente, L. Hou, C.P. Yu, M. Hoai, and D. Samaras, "Large-scale training of shadow detectors with noisily-annotated shadow examples," In *European Conference on Computer Vision*; Springer: Cham, Switzerland, pp. 816–832, 2016.
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," In *Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969, 2017.
- [16] Y. Wang, P.M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, pp. 387–394, June 2014.
- [17] H. Liu, C. Yang, X. Shu and Q. Wang, "A new method of shadow detection based on edge information and HSV color information," *2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, Shenzhen, 2009, pp. 286-289, 2009.
- [18] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337-1342, Oct 2003.
- [19] B. Sun and S. Li, "Moving Cast Shadow Detection of Vehicle Using Combined Color Models," *Chinese Conference on Pattern Recognition (CCPR)*, Chongqing, pp. 1-5, Oct 2010.
- [20] A.Varghese and G. Sreelekha, "Sample-based integrated background subtraction and shadow detection," *IPSN Transactions on Computer Vision and Applications*, vol. 9, no. 25, 2017.
- [21] J.Dai, D. Han, and X. Zhao, "Effective moving shadow detection using statistical discriminant model," *Optik*, vol. 126, no. 24,pp. 5398-5406 ,2015.
- [22] W. Xie, J. Zhao, and Y. Hu, "Moving Shadow Detection Algorithm Using Multiple Features," *International Conference on Computer Science and Applications (CSA)*, Wuhan, pp. 95-98, Nov 2015.
- [23] B. Wang, Y. Yuan, Y. Zhao, and W. Zou, "Adaptive moving shadows detection using local neighboring information," In: *Asian Conference on Computer Vision*. Springer, Cham, pp. 521–535, 2016.
- [24] J. Dai, M. Qi, J. Wang, J. Dai, J. Kong, "Robust and accurate moving shadow detection based on multiple features fusion," *Optics and Laser Technology*, vol. 54, pp. 232-241, Dec 2013.
- [25] H. F. Bakr, A. M. Hamad, and K. M. Amin, "Detecting moving shadow using a fusion of local binary pattern and gabor features," *IEEE*, In *13th International Conference on Computer Engineering and Systems (ICCES)*, pp. 393-398, Dec 2018.
- [26] B. Wang and C.L. Chen, "Optical reflection invariant-based method for moving shadows removal," *Optical Engineering*, vol.57, no. 9, 2018.
- [27] B. Wang, Y. Zhao and C. L. P. Chen, "Moving cast shadows segmentation using illumination invariant feature," in *IEEE Transactions on Multimedia*, 2019.
- [28] J. Huang and C. Chen, "A physical approach to moving cast shadow detection," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, pp. 769-772, 2009.
- [29] A. J. Joshi and N. P. Papanikolopoulos, "Learning to Detect Moving Shadows in Dynamic Environments," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2055-2063, Nov. 2008.
- [30] J.Dai, D. Han and X. Zhao, "Effective moving shadow detection using statistical discriminant model," *Optik*, vol. 126, no. 24,pp. 5398-5406 ,2015.
- [31] C.W. Lin,"Moving cast shadow detection using scale-relation multi-layer pooling features," *Journal of Visual Communication and Image Representation*, vol. 55, no. 48, pp. 504-517,2018.

- [32] Y. Yi, J. Dai, C. Wang, J. Hou, H. Zhang, Y. Liu, and J. Gao, "An Effective Framework Using Spatial Correlation and Extreme Learning Machine for Moving Cast Shadow Detection," *Applied Sciences*, vol. 9, no.23, 5042, 2019.
- [33] Q. Zheng, X. Qiao, Y. Cao, and R.W. Lau, "Distraction-aware shadow detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5167-5176, 2019.
- [34] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, pp. 580-587, 2014.
- [35] J.R.R Uijlings, K.E.A. van de Sande, T. Gevers and A.W. Smeulders, "Selective Search for Object Recognition," *International journal of computer vision*, vol.104, no.2, pp. 154-171, 2013.
- [36] R. Girshick, "Fast R-CNN," In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015.
- [37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," In *Advances in neural information processing systems (NIPS)*, pp. 91-99, 2015.
- [38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431-3440, 2015.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.
- [40] H. A. Qadir, Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken, and I. Balasingham, "Polyp Detection and Segmentation using Mask R-CNN: Does a Deeper Feature Extractor CNN Always Perform Better?," In *13th International Symposium on Medical Information and Communication Technology (ISMICT)*, Oslo, Norway, pp. 1-6, 2019.
- [41] W. Liu, Y. Wen, M. Scut, Z. Yu, and M. Yang, "Large-margin Softmax loss for convolutional neural networks," In *Proceedings of the 33rd International Conference Machine Learning*, pp. 507-516, 2016.
- [42] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," In *Advances in neural information processing systems*, pp. 2017-2025, 2015.
- [43] E. Bisong, "Google Colaboratory," In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA, 2019.
- [44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," In *ECCV*, 2014.
- [45] T. Zhang, "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms," In *Proceedings of the 21st International Conference on Machine Learning*, Banff, AB, Canada, pp. 919-926, 4-8 July 2004.
- [46] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," In *Advances in Neural Information Processing Systems 25*; Curran Associates, Inc, New York, NY, USA, pp. 1097-1105, 2012.