Journal of Information Systems Engineering and Management
2025, 10(19s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

ConvBAMnet: An Improved Approach for Camouflage and Occlusion Detection

Chaitali Mahajan[1], Ashish Jadhav[2]
[1]Department of Computer Engineering,
Ramrao Adik Institute of Technology, Nerul
D Y. Patil Deemed to be University, Mumbai,
Maharashtra, India, Pin: 400008
M.H.Saboo Siddik College of Engineering,
Byculla,Mumbai, India.
chaitalimahajan201@gmail.com
[2]Department of Computer Engineering,
Ramrao Adik Institute of Technology, Nerul
D Y. Patil Deemed to be University, Mumbai,
Maharashtra, India, Pin: 400008
Ashish.jadhav@rait.ac.in

ARTICLE INFO

Received: 23 Dec 2024

Revised: 31 Jan 2025

Accepted: 15 Feb 2025

ABSTRACT

Nowadays camouflage object detection is more challenging task as background and foreground objects are almost similar in color. Most of the time when the objects are fully or partially occluded along with camouflage then the problem is worst giving incorrect predictions. To address these issues, this paper proposes an algorithm ConvBAMnet which improves the feature extraction by introducing The Bottleneck Attention Module (BAM) layer. Convolutional layers, dense layers with dropout, bounding box regression, and attention processes work together to create a model that excels at identifying objects in difficult situations like obscured and camouflaged surroundings. BAM layers are specifically made to draw attention to specific channels and geographical locations within feature maps. The dataset considered over here is Moving camouflaged animal (MoCA) dataset. Two classes are considered for training purpose. Flat fish is highly camouflaged and Nile monitor is camouflaged as well as partially occluded. The problem of false and missed detection is improved giving better detection results by proposed method. The proposed model ConvBAMnet has been compared with ResNet 50 (Residual Neural Network 50) and VGG 16 (Visual Geometry Group 16) which were transfer learned on given dataset. The experimentation results indicate that proposed model has achieved an increase in accuracy compared to SOTA ResNet 50 and VGG 16.

Keywords: : camouflage object detection; occlusion; Bottleneck Attention Module; false and missed detection

1. Introduction

Over the past thirty years, numerous research groups have tackled the major problem of moving object recognition in a variety of difficult situations. There is a wide variety of moving objects, from waving trees to pedestrians. However, just a portion of these items need to be designated as moving. Things that belong to the so-called dynamic background, such as swaying trees, ocean waves, or escalators, must be identified as background. The primary obstacle in the detection of moving items is dealing with dynamic backgrounds, things that conceal, and occluded objects. [1-3].

A variety of features, including color, edges, texture, and so on, can be used to extract the moving subject in the foreground. The primary disadvantage of the trainable classifiers is that each frame takes longer to process computationally. Similarly, the requirement for a dedicated GPU (Graphical Processing Unit) has made the deployment of a deep learning method for this kind of problem too costly. [4,5].

The main challenge along with moving object detection is camouflage object detection(COD) whose objective is to locate things which are camouflaged and concealed within their environment. Finding items concealed within the scene that have a strong inherent resemblance to their environment in terms of hue, texture, form, and other characteristics is the goal of COD [6,7]. Furthermore, the usual difficulties for both general and salient object detection (such as background clutter, occlusion, and variations in illumination), COD faces an additional challenge, namely, how to precisely separate items from backgrounds that are comparable. This is because rarely COD situations have the perfect ratio of contrast to accentuate the main foreground object. [8]. It's difficult to obtain the target's edges because of the lack of contrast between the foreground and background. Also there are numerous interferences (noise, obstructions, shadows, etc.) in the complex environmental background that impact both the accuracy and speed of identification [9].

The unpredictable appearance and disappearance of the objects, false positives and negatives, as well as uncertainty in the data linkage provide problems to visual tracking. Objects in congested scenes with significant occlusion pose challenges for segmentation [10,11]. However, because of occlusion, object recognition is a difficult operation. Occlusion can occur when one object is obscured by another object of the same type (intra-class occlusion) or by an object of a different type (inter-class occlusion), which is caused by a fixed element [12]. There are different types of occlusion, from light to extreme occlusion. Static things like buildings and lampposts can obstruct target objects in the automobile environment. In situations when there is a lot of movement, such as in a crowd, dynamic objects like moving cars or other road users may self-occlude when portions of a pedestrian or cyclist overlap [13].

1.1 *Problem statement*

The existing research work have few issues, which are listed below:

- For detection of both camouflaged and occluded object there has been incorrect predictions in case of complex images.
- There is requirement of simplified approach to solve incorrect predictions in both cases.
- It's essential to increase the diversity of dataset, ultimately making the model robust and avoids overfitting.

This work addresses the issue occurred for detecting camouflage and occluded object. Main contribution of proposed work is as follows:

- Proposes an algorithm ConvBAMnet which includes BAM layer to enhance feature representation by considering both channel-wise and spatial-wise attention.
- To scale the number of channels in the attention layers, a new parameter called attention ratio has been included to optimized the unique demands of the task that is being offered.
- The Moving Camouflaged Animals (MoCA) dataset is initially in video format. Firstly, it has been converted into images then augmentation methods are employed to enlarge the dataset.
- Accuracy of proposed method is compared with VGG 16 and ResNet 50. Both SOTA algorithms are transfer learned on given dataset. ConvBAMnet increases the accuracy by 1 % and 3 % compared with VGG 16 and ResNet 50 respectively.

1.2 *Organization of paper*

This is how the remaining portion of the document is organized. The section that follows provides a succinct summary of relevant material. In section 3 proposed method along with Bottleneck Attention Module is explained. Section 4 includes experimental set up, dataset information and test findings. Last section concludes the findings.

2. Related Work

This section discusses contributed work on moving object detection, Camouflage and occlusion object detection.

2.1. *Moving object detection*

Dynamic background, changing illumination, and shadow are significant variables in complicated scenarios that cause traditional moving object detection algorithms to perform poorly. So, Ou, X. et al. proposed a moving entity recognition technique using Encoder-decoder structure in ResNet-18 as a solution to this issue in order to segment moving items from complicated scenes [14]. Wen et al. demonstrates a Faster R-CNN, a dual faster region-based convolutional neural network approach for identification of moving targets for synthetic aperture radar (SAR) for video. This new method can effectively suppress false alarms by combining the algorithmic detection of shadows in SAR images and Doppler energy [15]. The need for an effective algorithm to uncover hidden features grows as an image's attributes expand. The YOLOv3 model for multiple item detection on the KITTI and COCO datasets and the Convolution Neural Network (CNN) model for single object detection on the urban vehicle dataset are designed respectively. On traffic surveillance footage, things are monitored between simple Online Real Time Tracking (SORT) frames and YOLOv3 [16,17]. For drone detection also CNN is used for object detection and its tracking [18]. One of the primary limitations of the current object tracking technique is its time-consuming nature, particularly when dealing with videos that include a lot of data. The three parts of the authors' suggested technique—detection, tracking, and evaluation—are designed to address these problems. The detection phase includes noise reduction and foreground segmentation [19]. Edge detection comes under object detection and there are few papers which have worked on this. Current approaches typically rely significantly on the computation of several image attributes, which adds complexity and expense to the entire system. Convolutional Neural Networks (CNNs) that can predict edges based just on picture patches are trained [20]. The suggested method produces thin edge-maps that are visually appealing to humans and is applicable to any task involving edge detection without the need for earlier instruction or refinement [21].

2.2 *Camouflage object detection*

Finding objects that are hidden in their surroundings is the goal of COD. COD is acknowledged to be an extremely difficult assignment because of elements including poor lighting, shadows, compactness, and significant likeness to the surrounding environment. Numerous writers looked into COD's performance and sorted the issue with improving feature extraction as well as segmentation [22-24]. For COD, Xu et al. proposed a novel multi-scale guided refining model [25]. Few COD methods have based on context based as well as fusion of context [26-28] also some have base of frequency approach [29].X.Xu et al. proposed a brand-new COD boundary guiding network that uses a two-step methodology for localization and refinement [30].

2.3 *Occlusion detection*

Since genuine object outlines and occlusion borders are usually not distinguished, segmenting heavily overlapping objects is difficult. Ke. L et al. proposed Dual Convolutional Network, in which the uppermost proposed added layer identifies the occluded objects (occluder), and partially occluded instances (occludee) are inferred by the bottom proposed layer [31]. To solve the issue of Object blockage boundary detection and class imbalance during training of occluded images authors [32] suggested Attention Loss function and a unified multi-task deep object occlusion boundary detect.

Considering CNN as a base, most of the authors tried to overcome occlusion and tracking issue [33]. Liu Y et al. suggest a more precise edge detector that makes use of deeper convolutional features [34]. A deep convolutional neural network was trained using few components of the optical flow field that are vertical and horizontal to detect occlusion edges in pictures and videos [35]. Fu, Huan, et al. have created a unique method based on CNNs and conditional random fields to comprehend occlude borders [36]. Li, Huaer, et al. Analysed Occluded Face Recognition with Convolutional Neural Networks [37]. Yuan, Yue, et al. [38] proposed an updated approach for occlusion detection where sturdy Faster R-CNN improved the capacity to recognise small objects and occlusions by substituting RoI Aligns for the harsh quantization that RoI Pooling imposed [39]. Wang, Chaohui, et al. suggested two distinct methods for detecting occlusion boundaries in video sequences through better contextual information exploration [40].

2.4 *Research Gap*

None of the previous research activities employed camouflage and occlusion object detection with inclusion of BAM layer. While numerous algorithms have been developed for detecting camouflage and occluded objects, they often rely on complex architectures. In contrast, the proposed ConvBAMnetmodel adopts a simpler approach to address these challenges in detection.

3. Materials and Methods

3.1 *Proposed approach*

The proposed methodology incorporates the Bottleneck Attention Module (BAM) layer into the architecture. The flow of proposed work is as shown in fig 1.

- Input Block: The input image, which is 224x224x3 (width x height x channels), is represented by this block.
- Convolutional Block 1: This block includes a BAM layer that applies both spatial and channel attention after the first Conv2D layer with 64 filters. The BAM layer is composed of two Conv2D layers for the channel attention component and another Conv2D layer for the spatial attention part.
- BAM Layer: The Bottleneck Attention Module layer, defined by the BAM class, improves feature representation by taking into account both channel-wise and spatial-wise attention. Channel attention and spatial attention are its two different forms of attention systems. The channel attention sub-module records channel-wise dependencies and lowers the dimensionality of the input feature maps. The spatial attention sub-module captures spatial relationships within feature maps. The proposed method applies both channel and spatial attention mechanisms and returns the multiplication of elements feature map input with the weights for attention. Two convolutional layers are followed by BAM layers, to enhance feature representation. After every convolutional layer, max-pooling is performed to down-sample the feature maps. Flattening and dropout layers are used for regularization and dimensionality reduction.
- MaxPooling Block 1: This block uses a MaxPooling2D layer with a pool size of 2x2 to downsample the feature maps.
- Convolutional Block 2: This block contains the second Conv2D layer with 128 filters, followed by another BAM layer for spatial and channel attention, similar to the structure mentioned in Convolutional Block 1.
- MaxPooling Block 2: The feature maps are down sampled once more by a second MaxPooling2D layer.
- Flatten Block: The multidimensional feature maps are compressed into a vector of one dimension by this block.
- Dropout Block: To avoid overfitting, this block applies dropout at a rate of 0.5.
- Fully Connected Block for Bounding Box Coordinates: Three Dense layers with 128, 64, and 32 units each make up this block. The fourth and final Dense layer predicts the bounding box coordinates. Bounding box coordinates are predicted using three dense layers activated by ReLu. The output layer consists of four units handles output bounding box coordinates using a linear activation function.

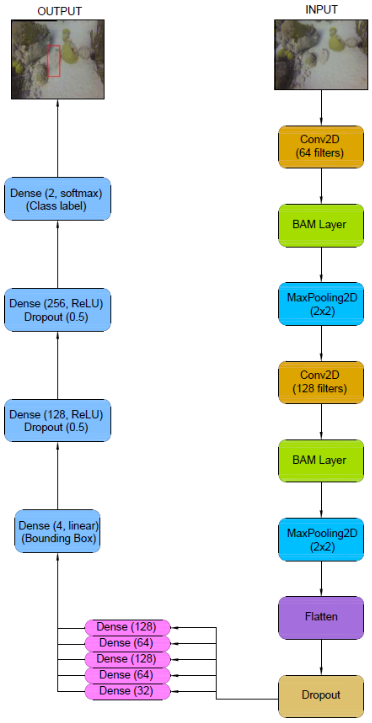
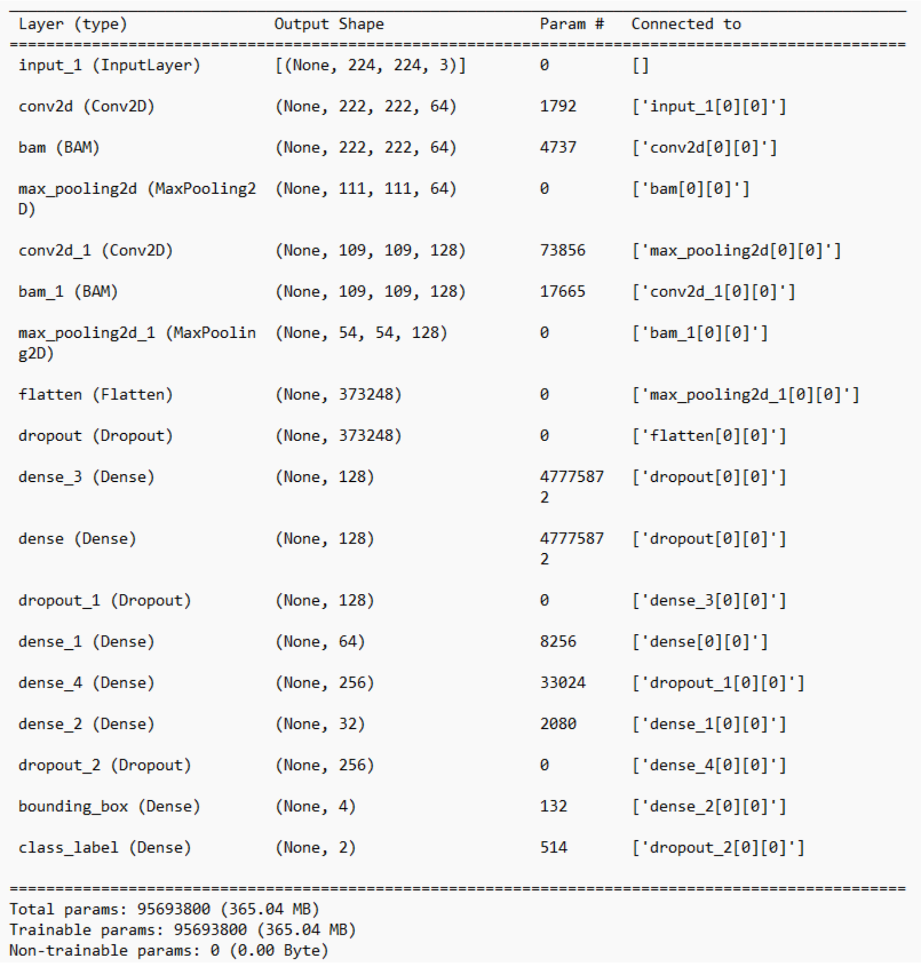


Fig. 1. Block diagram of proposed ConvBAMnet model

- Fully Connected Block for Class Labels: This block consists of two sets of Dense layers with 128 and 256 units, respectively, interleaved with dropout layers. The last Dense layer uses a softmax activation function and two units to forecast the class labels. Three dense layers with ReLu activation are used for class label prediction. Applying dropout regularisation comes after the initial dense layer. The output layer has 2 units (for binary classification) with a method called softmax activation to get output probabilities of class.
- Output Blocks: The model has two output blocks: "bounding_box" for predicting bounding box coordinates and "class_label" for predicting class labels.

Table 1. Proposed model ConvBAMnet summery



```
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]        0         []

 conv2d (Conv2D)             (None, 222, 222, 64)         1792      ['input_1[0][0]']

 bam (BAM)                   (None, 222, 222, 64)         4737      ['conv2d[0][0]']

 max_pooling2d (MaxPooling2  (None, 111, 111, 64)         0         ['bam[0][0]']
 D)

 conv2d_1 (Conv2D)           (None, 109, 109, 128)        73856     ['max_pooling2d[0][0]']

 bam_1 (BAM)                 (None, 109, 109, 128)        17665     ['conv2d_1[0][0]']

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 128)          0         ['bam_1[0][0]']
 g2D)

 flatten (Flatten)           (None, 373248)               0         ['max_pooling2d_1[0][0]']

 dropout (Dropout)           (None, 373248)               0         ['flatten[0][0]']

 dense_3 (Dense)             (None, 128)                  4777587   ['dropout[0][0]']
                                                          2

 dense (Dense)               (None, 128)                  4777587   ['dropout[0][0]']
                                                          2

 dropout_1 (Dropout)         (None, 128)                  0         ['dense_3[0][0]']

 dense_1 (Dense)             (None, 64)                   8256      ['dense[0][0]']

 dense_4 (Dense)             (None, 256)                  33024     ['dropout_1[0][0]']

 dense_2 (Dense)             (None, 32)                   2080      ['dense_1[0][0]']

 dropout_2 (Dropout)         (None, 256)                  0         ['dense_4[0][0]']

 bounding_box (Dense)        (None, 4)                    132       ['dense_2[0][0]']

 class_label (Dense)         (None, 2)                    514       ['dropout_2[0][0]']

==================================================================================================
Total params: 95693800 (365.04 MB)
Trainable params: 95693800 (365.04 MB)
Non-trainable params: 0 (0.00 Byte)
```

Table 1 illustrates the proposed model summery, showing the different layers and their connections. The BAM layers are integrated after each convolutional layer to enhance feature representation, capturing both channel-wise and spatial-wise attention. The model outputs both bounding box predictions (coordinates) and class label predictions (probabilities) as the final results.

3.2 *BAM attention module*

Utilising channel dependencies is a crucial step towards enhancing CNN model performance. Researchers employed attention blocks, or the BottleNeck Attention Module (BAM), to enhance the functionality of state-of-the-art models at a small computational cost [41-42]. Within the suggested framework, the BAM is a special layer made for successfully capture both spatial and channel-wise attention. Channel attention and spatial attention are its two components.

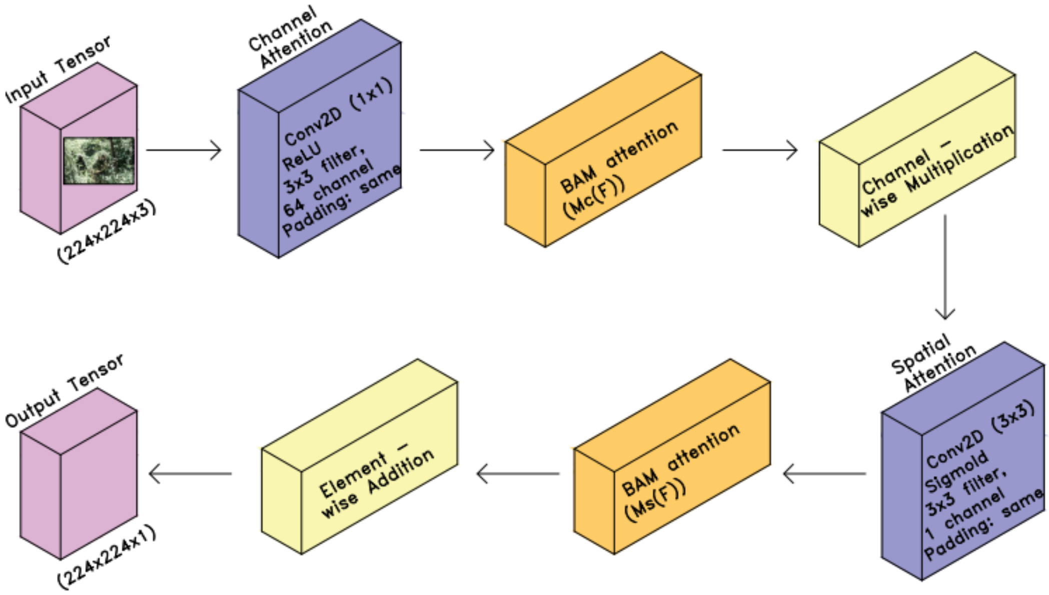


Fig. 2. Block diagram of BAM

- Channel Attention

Two Conv2D layers are used in the channel attention section. The first layer activates a ReLu after reducing the number of channels by a predetermined reduction ratio instance (16). The channel attention map is produced by the second layer, which sigmoid activates to bring the number of channels back to their initial state.

It consists of two convolutional layers; the first one performs 1x1 convolutions with a reduction ratio of (channels // reduction_ratio) to minimise the number of channels. The computational complexity is decreased in this stage. To provide non-linearity, the first convolutional layer's output is subsequently passed via ReLu activation function. By executing 1x1 convolutions with a sigmoid activation function, the second convolutional layer returns the original number of channels. The relevance of each channel is indicated by the channel-wise attention weights that are generated by this layer and fall between [0, 1].

- Mc (F) (Channel-wise Feature Map)

Mc (F) represents the feature map after applying channel-wise attention. It is obtained by performing a convolution operation on the input feature map with a 1x1 filter followed by a ReLu activation function. This operation aims to recalibrate the importance of different channels in the feature map based on learned attention weights. Mc(F) highlights informative channels while suppressing less relevant ones.

$$Mc(F) = \{(f_{1x1}(\frac{C}{r})\ ReLu\ (f_{3x3}(\frac{C}{r}))\}P \qquad (1)$$

Where,

Mc (F) is Channel-wise Feature Map

f_{1x1} is Convolution of 1 x1

f_{3x3} is Convolution of 3 x3

c is Channels

r is attention ratio

P is padding: same

- Spatial Attention

The spatial attention map is produced by the spatial attention component using a single Conv2D layer with a kernel size of three and a sigmoid activation. The significance of each spatial position in the feature maps is indicated by the spatial attention weights that are generated by this layer.

- Ms (F) (Spatial-wise Feature Map)

Ms (F) represents the feature map after applying spatial-wise attention. It is obtained by convolving the input feature map with a 3x3 filter followed by a sigmoid activation function. This operation captures spatial dependencies within the feature map to focus on relevant regions of interest. Ms (F) generates spatial attention weights for each spatial location, emphasizing informative regions while suppressing less relevant ones.

$$Ms(F) = \{(f_{3x3}(r, d)\ \sigma\ (f_{3x3}((r,d)\}P \qquad (2)$$

Where,

Ms(F) is Spatial wise feature map

f_{3x3} is Convolution of 3 x3

r is attention ratio

d is dilation rate (here its 2)

σ is sigmoid activation function

P is padding: same

- Multiplicative Integration

The feature maps are multiplied element-wise by both sets of attention weights following the computation of channel and spatial attention weights.

The attention map for channel and the spatial attention map are multiplied by the input to create the BAM layer's output. This aids in the model's ability to reduce unnecessary regions and channels and concentrate on pertinent ones.

The BAM layer combines both the Mc(F) and Ms(F) feature maps through element-wise multiplication, which helps the model focus on relevant channels and spatial locations while suppressing irrelevant ones. Finally, the element-wise addition step aggregates the information from the Mc(F) and Ms(F) feature maps to produce the output tensor, which retains spatial information while enhancing the representation of relevant features.

In the proposed model, attention processes are incorporated into the feature representation by adding BAM layers after the first and second convolutional layers. By collecting both spatial and channel-wise attention, the usage of BAMs can improve the model's performance and improve its ability to locate and identify objects in images.

3.3 *Contribution of BAM for occluded and camouflaged image detection:*

- To scale the number of channels in the attention layers, let's introduce a new parameter called attention_ratio. This parameter enables the attention mechanism's capacity to be adjusted, and its performance can be optimized to meet the unique demands of the task that is being offered.
- The number of channels in the attention layers can be managed by varying the attention_ratio parameter, which makes it possible to balance computational efficiency and model capacity. The best setup for the suggested job and dataset can be found by experimenting with different attention_ratio settings. This will help in following way:

In general, BAM aids CNN-based object detection models in adaptively focusing on significant characteristics and suppressing distracting or irrelevant data. This can result in increased resilience and accuracy of detection, particularly in difficult real-world situations like occlusion or camouflage.

4. Results and Discussion

4.1. *Setup for experimentation*

The suggested research develops a customised Convolutional Neural Network (CNN) model to address the difficulties in object detection tasks presented by obstructed and camouflaged images.

Python 3.10.12 has been used to create the models in this investigation. Tensorflow is an open-source software library that is widely used as a foundation for creating and implementing machine learning models. The study also makes use of High-level neural network Keras, a high-level neural network API designed to run on top of Tensorflow. This project was developed using Google Colab, a cloud-based platform that offers computational resource access.

4.2. *Dataset*

Moving Camouflaged Animals is the dataset that was used in this research study's experiments (MoCA). This dataset was compiled from a variety of sources, most notably videos, and then transformed into images. Here for experimentation two classes were considered, Flatfish which is highly camouflaged having 288 images and Nile_monitor which having occlusion with 600 images [44].

As dataset is having total images as 888, augmentation is employed. It defined a number of parameters for data augmentation methods that can be applied to the Keras Image Data Generator class. By using these methods, different versions of image data have been produced, which can strengthen model's resilience. The dataset images are increased 2 to 3 times than that of original images. A summary of each parameter is provided below:

- Rotation_range=20: During the data augmentation procedure, the images will be randomly rotated by a range of degrees (in a clockwise direction). Images in this instance will rotate by up to 20 degrees. This aids in the model's rotational invariance.
- Range_width_shift =0.1: and Range_height_shift =0.1: The array of the input image width and height that will be randomly shifted is defined by these parameters. The percentages of the overall width and height are represented by the values.
- Shear_range=0.2: The range of the shear angle is defined by this parameter in degrees. Images will be distorted in this case by up to 20% (0.2) of their original size.
- Zoom_range=0.1: In this case, the images will be enlarged by up to 10%.
- Horizontal_flip=True:The process of flipping the images horizontally is done at random, so it might or might not happen.
- Fill_mode="nearest":The nearest neighbour interpolation method is used to fill in the new pixels at the new pixel location when images are rotated or shifted. Although this technique keeps the image's sharp edges, the texture may become less smooth.

By expanding the amount and diversity of the training dataset from 888 images to 2500 images with data augmentation. This approach can help to prevent overfitting and enhance the model's generalisation performance. The model learns to become more resilient and invariant to these variances by being exposed to different transformations of the input data during training, which eventually improves performance on unknown data.

4.3. *Evaluation parameters*

Recall, precision, F1 Score, and accuracy were taken into account when assessing the model [43]. The ratio of accurate positives to positives that the classifier predicted is known as precision(P). Recall (R)is the percentage of gratifying information that is accurately identified as favourable among all favourable data points. It's sometimes referred to as true positive rate or hit rate. Essentially, harmonic is the F1 score mean of P and R. Accuracy have computed by utilising the confusion matrix. The existing and anticipated the classification system's classes are displayed in the confusion matrix. Confidence score is one way to calculate an evaluation standard is to use a confidence score. This confidence score, which is expressed as a percentage, indicates the likelihood that the image was accurately identified by the algorithm [45].

4.4. *Model performance and comparison*

The table displays the proposed model’s performance along with VGG 16 and RESNET 50 in terms of F1 score, recall, accuracy, and precision.

Table 2: Evaluation parameter comparison of proposed method with VGG 16 and RESNET 50

Model	Class	Precision	Recall	F1-Score	Accuracy
RESNET50	Flat_fish	0.88	0.99	0.93	0.95
	Nile_monitor	0.99	0.94	0.97	
	Avg.	0.96	0.96	0.96	
	Flat_fish	0.97	0.94	0.95	

VGG16	Nile_monitor	0.97	0.99	0.98	0.97
	Avg.	0.97	0.96	0.97	
Proposed model ConvBAMnet	Flat_fish	0.99	0.95	0.97	0.98
	Nile_monitor	0.98	0.99	0.99	
	Avg.	0.98	0.97	0.98	

In case of flat fish and Nile_monitor individually P, R and F1 Score has highest value for proposed model compared with transfer learned models.

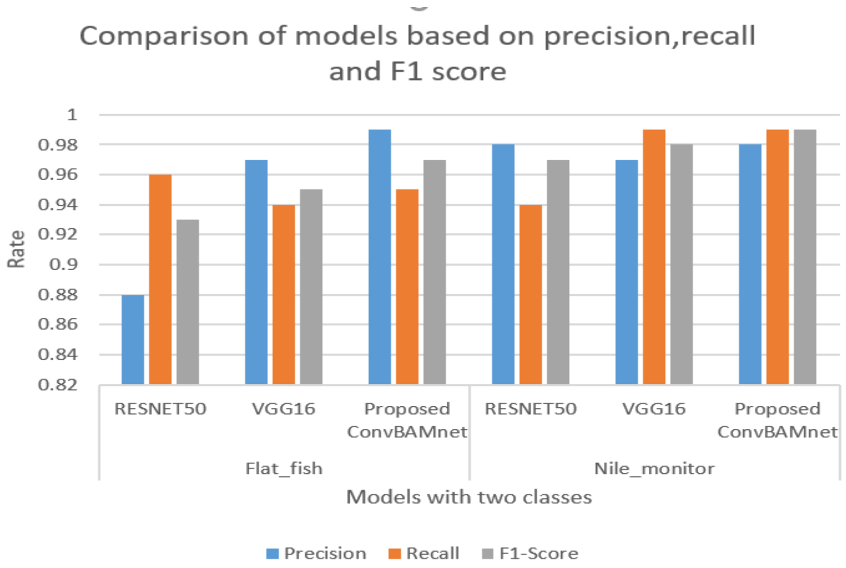


Fig.3. Comparison of models based on precision, recall and F1 score

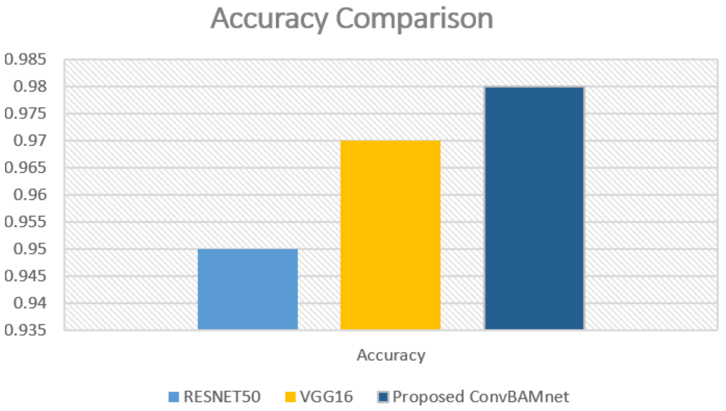


Fig.4. Accuracy comparison of proposed model with VGG 16 and RESNET 50

The ConvBAMnet model is compared with RESNET 50 and VGG 16. Both models have been transfer learned on given dataset. Computational complexity is high in case of transfer learning model [43]. The proposed methodology increases the accuracy by 1% compared with VGG 16 model and 3% compared with RESNET 50.

4.5 Comparison with State-of-Art

The proposed ConvBAMnet model is compared with State-of –art models with parameter complexity, Fβ score (The β value has been considered as 0.3) and loss. Though parameter complexity is more compared to few models, proposed model excels in Fβ score and loss.

Table 3. Comparison of proposed model with State-of-Art [46] methods

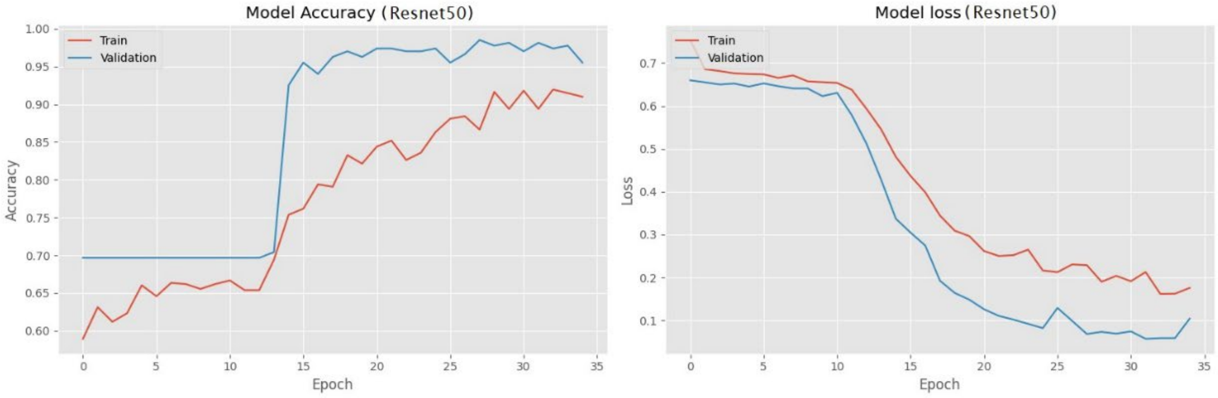
Model [46]	**Backbone**	**Parameters**	**F_β Score**	**Loss**
EGNet	ResNet-50	111.693M	0.136	0.035
BASnet	ResNet-50	87.060M	0.173	0.042
CPD	ResNet-50	47.851M	0.152	0.041
PraNet	ResNet-50	32.547M	0.296	0.030
SINet	ResNet-50	48.947M	0.256	0.028
SINetV2	ResNet-50	26.976M	0.229	0.031
PNSNet	ResNet-50	26. 874M	0.084	0.035
RCRNet	ResNet-50	53.790M	0.159	0.033
MG	VGG CNN	4.766M	0.195	0.067
SLT NT	PVTv2-B5	82.303M	0.331	0.027
SLT ST	PVTv2-B5	82. 383M	0.328	0.027
ZoomNeXt	PVTv2-B5	84.776M	0.497	0.010
Proposed Method	**CNN**	**95.693M**	**0.9**	**0.010**

4.6 Loss function

Proposed model is utilising the "categorical_crossentropy" loss for the "class_label" head, which works well for multi-class classification workloads. The cross-entropy between the actual class labels and the anticipated class probabilities is computed using this loss function. The "mean_squared_error" loss, which calculates the mean squared variation between the anticipated and actual bounding box coordinates, is used by the proposed model for the "bounding_box" head. Regression problems where you wish to minimise the difference between the predicted and ground truth values are a good fit for this loss.

In case of proposed work, it is advantageous to use mean squared error loss for the regression component (bounding box prediction) and categorical loss due to cross-entropy for the classification part (class label prediction), as this enables the model to efficiently learn and optimise both aspects of the predicted item detection task. This study compares proposed model with transferred learned models: VGG 16 and Resnet50.

The following graphs shows Model's accuracy and loss verses epoch.



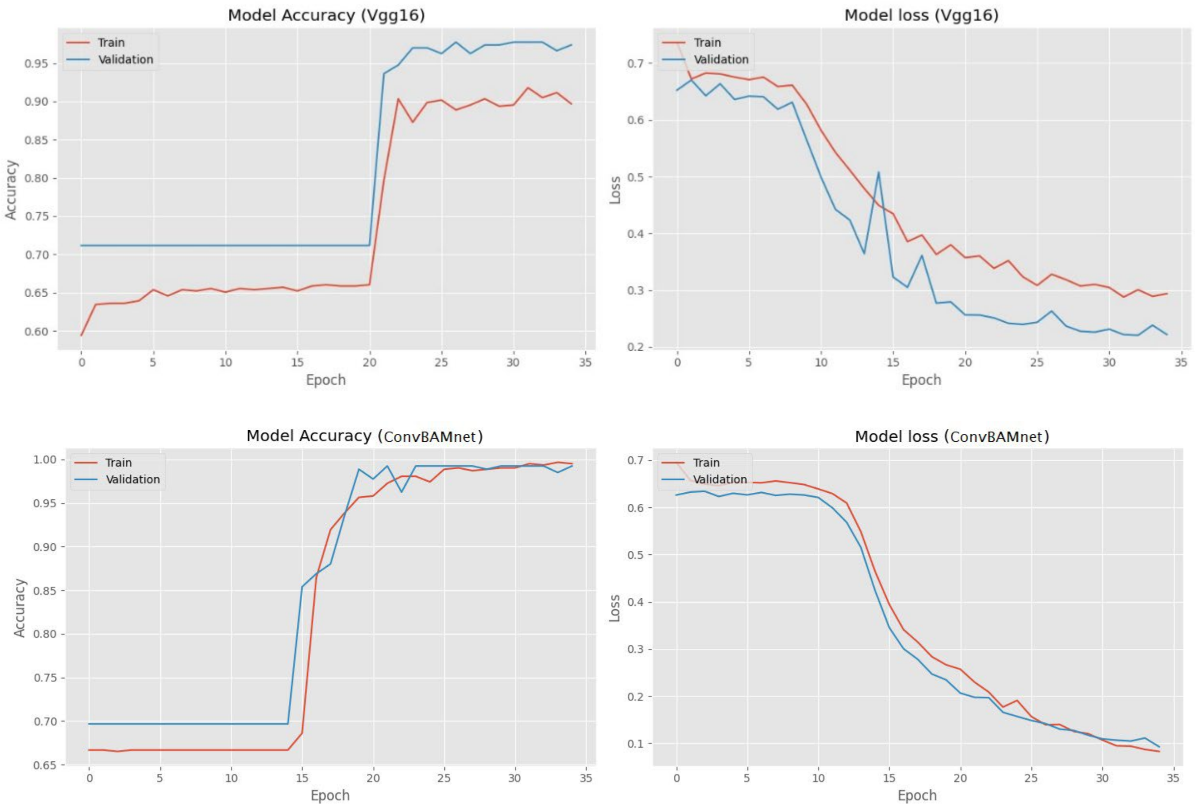


Fig. 5. Validation, training accuracy and loss of the models verses epoch

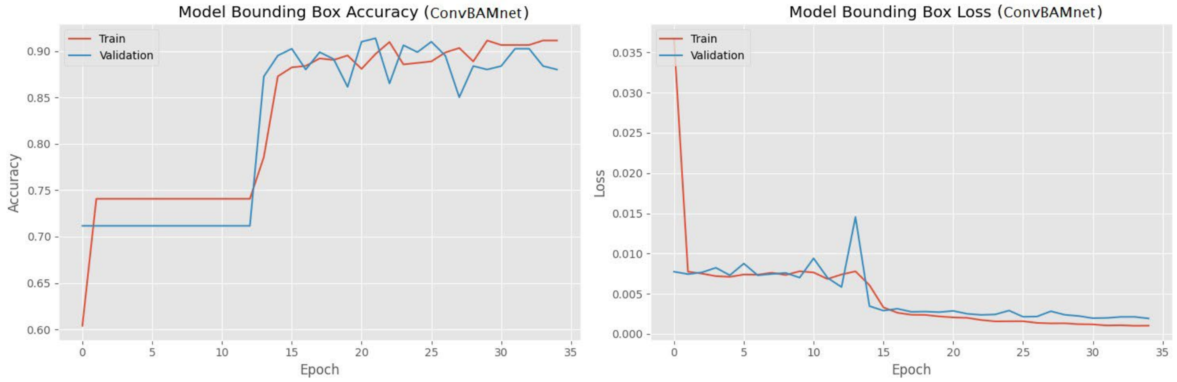


Fig.6. Bounding box loss and accuracy of the proposed model ConvBAMnet

The proposed ConvBAMnet shows the least signs of overfitting. Its validation accuracy consistently matches or slightly exceeds its training accuracy, indicating excellent generalization. In contrast, both VGG16 and ResNet50 show a more pronounced gap between training and validation accuracy, with validation accuracy consistently higher. This suggests proposed ConvBAMnet is learning more robust features that generalize well to unseen data.

Proposed method exhibits the most stable learning curve after the initial breakthrough. Its accuracy increases smoothly and consistently. ResNet50 shows more fluctuations in validation accuracy, while VGG16 has a more abrupt jump followed by relative stability. The smooth, gradual improvement of ConvBAMnet suggests a more reliable and predictable learning process.

Also the proposed model ConvBAMnet and ResNet50 show earlier breakthroughs (around epoch 15) compared to VGG16 (around epoch 20). This suggests that ConvBAMnet learns critical features more quickly, potentially requiring less training time to achieve high performance. The smooth, stable improvement of ConvBAMnet after the initial jump suggests a more robust learning process, less susceptible to local optima or dataset peculiarities that might cause the fluctuations seen in ResNet50's performance.

The performance of a classification model in predicting the two classes "flatfish" and "nile_monitor" is represented by this confusion matrix.

The number of accurate predictions for each class is represented by the diagonal elements (77 and 185), whereas the number of inaccurate or misclassified predictions is represented by the off-diagonal values (4 and 1).

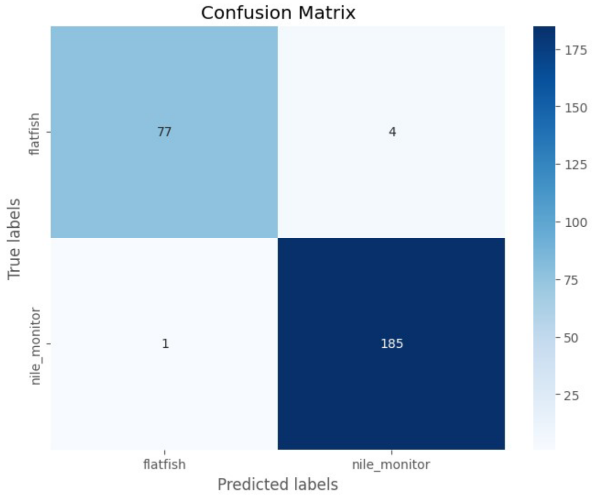


Fig.7. Confusion matrix of proposed method

The mean of all the precision scores for each threshold is used to get the confidence score. Few images from both the classes along with confidence score is as in below diagram.

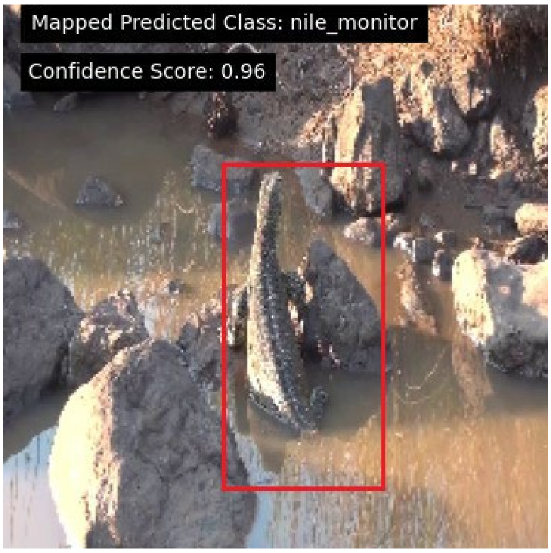


Fig.8.Confidence score of Nile_monitor test 1

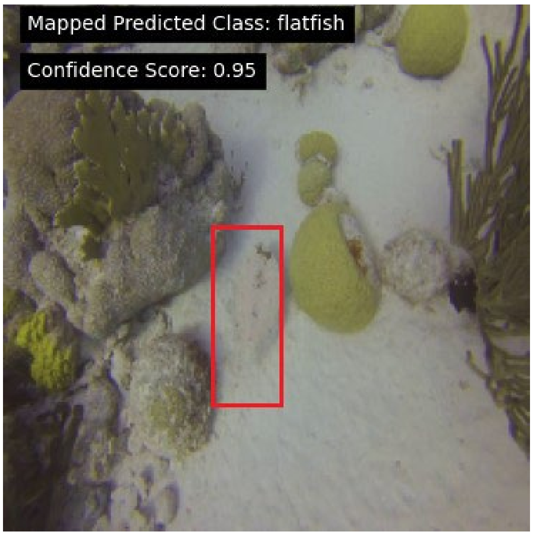


Fig. 9. Confidence score of flatfish test 1

5. Conclusion

One of the challenges for camouflage and occluded object detection is to detect the object accurately. To achieve the goal, this paper proposes the ConvBAMnet model. The model incorporates the BAM layer into the architecture. More discriminative characteristics can be learned by the model from the input data by adding BAM layers after the convolutional layers. Better object detection performance results from this, particularly in complex occlusion and camouflaged objects. More discriminative characteristics can be learned by the model from the input data by adding BAM layers after the convolutional layers. Better object detection performance results from this. The proposed model is compared with VGG 16 and Resnet50 (transfer learned) model. For both classes proposed ConvBAMnet model outperforms than other models. The proposed model has accuracy of 98% which is higher by 1% and 3% compared with transfer learned VGG 16 and Resnet50.The suggested approach will be tested in the future on several datasets and expanded for multiple object detection.

Acknowledgments

For this research work, no funding was obtained.

References

[1] Chapel MN, Bouwmans T. Moving objects detection with a moving camera: A comprehensive review. Computer science review. 2020 Nov 1;38:100310.

[2] Wang P, Wu J, Fang A, Zhu Z, Wang C, Ren S. Fusion representation learning for foreground moving object detection. Digital Signal Processing. 2023 Jun 30;138:104046.

[3] Chen Z, Khemmar R, Decoux B, Atahouet A, Ertaud JY. Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility. In2019 Eighth International Conference on Emerging Security Technologies (EST) 2019 Jul 22 (pp. 1-6). IEEE.

[4] Nallasivam M, Senniappan V. Moving human target detection and tracking in video frames. Studies in Informatics and Control. 2021 Mar 1;30(1):119-29.

[5] Zhao X, Wang G, He Z, Jiang H. A survey of moving object detection methods: A practical perspective. Neurocomputing. 2022 Sep 7;503:28-48.

[6] Li X, Li H, Zhou H, Yu M, Chen D, Li S, Zhang J. Camouflaged object detection with counterfactual intervention. Neurocomputing. 2023 Oct 7;553:126530.

[7] Liu Y, Zhang K, Zhao Y, Chen H, Liu Q. Bi-RRNet: Bi-level recurrent refinement network for camouflaged object detection. Pattern Recognition. 2023 Jul 1;139:109514.

[8] Zhuge M, Lu X, Guo Y, Cai Z, Chen S. CubeNet: X-shape connection for camouflaged object detection. Pattern Recognition. 2022 Jul 1;127:108644.

[9] Bi H, Zhang C, Wang K, Tong J, Zheng F. Rethinking camouflaged object detection: Models and datasets. IEEE transactions on circuits and systems for video technology. 2021 Nov 2;32(9):5708-24.

[10] Ong J, Vo BT, Vo BN, Kim DY, Nordholm S. A bayesian filter for multi-view 3d multi-object tracking with occlusion handling. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020 Oct 28;44(5):2246-63.

[11] Dong W, Roy P, Peng C, Isler V. Ellipse R-CNN: Learning to infer elliptical object from clustering and occlusion. IEEE Transactions on Image Processing. 2021 Jan 20;30:2193-206.

[12] Saleh K, Szénási S, Vámossy Z. Occlusion handling in generic object detection: A review. In2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI) 2021 Jan 21 (pp. 000477-000484). IEEE.

[13] Gilroy S, Jones E, Glavin M. Overcoming occlusion in the automotive environment—A review. IEEE Transactions on Intelligent Transportation Systems. 2019 Dec 9;22(1):23-35.

[14] Ou X, Yan P, Zhang Y, Tu B, Zhang G, Wu J, Li W. Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes. IEEE Access. 2019 Jul 30;7:108152-60.

[15] Wen L, Ding J, Loffeld O. Video SAR moving target detection using dual faster R-CNN. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2021 Feb 25;14:2984-94.

[16] Gong M, Shu Y. Real-time detection and motion recognition of human moving objects based on deep learning and multi-scale feature fusion in video. IEEE Access. 2020 Feb 3;8:25811-22.

[17] Aradhya HR. Object detection and tracking using deep learning and artificial intelligence for video surveillance applications. International Journal of Advanced Computer Science and Applications. 2019;10(12).

[18] Rohan A, Rabah M, Kim SH. Convolutional neural network-based real-time object detection and tracking for parrot AR drone 2. IEEE access. 2019 May 27;7:69575-84.

[19] Mahalingam T, Subramoniam M. A robust single and multiple moving object detection, tracking and classification. Applied Computing and Informatics. 2021 Jan 4;17(1):2-18.

[20] Wang R. Edge detection using convolutional neural network. InAdvances in Neural Networks–ISNN 2016: 13th International Symposium on Neural Networks, ISNN 2016, St. Petersburg, Russia, July 6-8, 2016, Proceedings 13 2016 (pp. 12-20). Springer International Publishing.

[21] Poma XS, Riba E, Sappa A. Dense extreme inception network: Towards a robust cnn model for edge detection. InProceedings of the IEEE/CVF winter conference on applications of computer vision 2020 (pp. 1923-1932).

[22] Liu Y, Li H, Cheng J, Chen X. MSCAF-Net: a general framework for camouflaged object detection via learning multi-scale context-aware features. IEEE Transactions on Circuits and Systems for Video Technology. 2023 Feb 15.

[23] Lv Y, Zhang J, Dai Y, Li A, Barnes N, Fan DP. Towards deeper understanding of camouflaged object detection. IEEE Transactions on Circuits and Systems for Video Technology. 2023 Jan 5.

[24] Le TN, Nguyen TV, Nie Z, Tran MT, Sugimoto A. Anabranch network for camouflaged object segmentation. Computer vision and image understanding. 2019 Jul 1;184:45-56.

[25] Xu X, Chen S, Lv X, Wang J, Hu X. Guided multi-scale refinement network for camouflaged object detection. Multimedia Tools and Applications. 2023 Feb;82(4):5785-801.

[26] Sun Y, Chen G, Zhou T, Zhang Y, Liu N. Context-aware cross-level fusion network for camouflaged object detection. arXiv preprint arXiv:2105.12555. 2021 May 26.

[27] Dong B, Zhuge M, Wang Y, Bi H, Chen G. Accurate camouflaged object detection via mixture convolution and interactive fusion. arXiv preprint arXiv:2101.05687. 2021 Jan 14.

[28] Ji GP, Fan DP, Chou YC, Dai D, Liniger A, Van Gool L. Deep gradient learning for efficient camouflaged object detection. Machine Intelligence Research. 2023 Feb;20(1):92-108.

[29] Lin J, Tan X, Xu K, Ma L, Lau RW. Frequency-aware camouflaged object detection. ACM Transactions on Multimedia Computing, Communications and Applications. 2023 Mar 23;19(2):1-6.

[30] Xu X, Zhu M, Yu J, Chen S, Hu X, Yang Y. Boundary guidance network for camouflage object detection. Image and Vision Computing. 2021 Oct 1;114:104283.

[31] Ke L, Tai YW, Tang CK. Deep occlusion-aware instance segmentation with overlapping bilayers. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2021 (pp. 4019-4028).

[32] Wang G, Wang X, Li FW, Liang X. Doobnet: Deep object occlusion boundary detection from an image. InComputer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14 2019 (pp. 686-702). Springer International Publishing.

[33] Aslan MF, Durdu A, Sabanci K, Mutluer MA. CNN and HOG based comparison study for complete occlusion handling in human tracking. Measurement. 2020 Jul 1;158:107704.

[34] Liu Y, Cheng MM, Hu X, Wang K, Bai X. Richer convolutional features for edge detection. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 3000-3009).

[35] Sarkar S, Venugopalan V, Reddy K, Ryde J, Jaitly N, Giering M. Deep learning for automated occlusion edge detection in RGB-D frames. Journal of Signal Processing Systems. 2017 Aug;88:205-17.

[36] Fu H, Wang C, Tao D, Black MJ. Occlusion boundary detection via deep exploration of context. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 241-250).

[37] Li H, Alghowinem S, Caldwell S, Gedeon T. Interpretation of occluded face detection using convolutional neural network. In2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES) 2019 Apr 25 (pp. 000165-000170). IEEE.

[38] Yuan Y, Chu J, Leng L, Miao J, Kim BG. A scale-adaptive object-tracking algorithm with occlusion detection. EURASIP Journal on Image and Video Processing. 2020 Dec;2020:1-5.

[39] Zhou T, Li Z, Zhang C. Enhance the recognition ability to occlusions and small objects with robust faster R-CNN. International Journal of Machine Learning and Cybernetics. 2019 Nov;10:3155-66.

[40] Wang C, Fu H, Tao D, Black MJ. Occlusion boundary: A formal definition & its detection via deep exploration of context. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020 Nov 19;44(5):2641-56.

[41]Mahum R, Irtaza A, Javed A, Mahmoud HA, Hassan H. DeepDet: YAMNet with BottleNeck Attention Module (BAM) TTS synthesis detection. EURASIP Journal on Audio, Speech, and Music Processing. 2024 Apr 1;2024(1):18.

[42] Zia A, Mahum R, Ahmad N, Awais M, Alshamrani AM. Eye diseases detection using deep learning with BAM attention module. Multimedia Tools and Applications. 2023 Dec 27:1-24.

[43]Panthakkan A, Anzar SM, Al Mansoori S, Mansoor W, Al Ahmad H. A systematic comparison of transfer learning models for COVID-19 prediction. Intelligent Decision Technologies. 2022 Jan 1;16(3):557-74.

[44]Lamdouar H, Yang C, Xie W, Zisserman A. Betrayed by motion: Camouflaged object discovery via motion segmentation. InProceedings of the Asian conference on computer vision 2020.

[45] Mandal S, Mones SM, Das A, Balas VE, Shaw RN, Ghosh A. Single shot detection for detecting real-time flying objects for unmanned aerial vehicle. InArtificial intelligence for future generation robotics 2021 Jan 1 (pp. 37-53). Elsevier.

[46] Pang, Y., Zhao, X., Xiang, T.Z., Zhang, L. and Lu, H., 2024. Zoomnext: A unified collaborative pyramid network for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*