The Visual Computer (2023) 39:2347–2363
https://doi.org/10.1007/s00371-022-02445-x

ORIGINAL ARTICLE

A paced multi-stage block-wise approach for object detection in thermal images

Shreyas Bhat Kera[1] · **Anand Tadepalli**[1] · **J. Jennifer Ranjani**[1]

Accepted: 20 February 2022 / Published online: 7 April 2022

Abstract
The growing advocacy of thermal imagery in applications, such as autonomous vehicles, surveillance, and COVID-19 detection, necessitates accurate object detection frameworks for the thermal domain. Conventional methods could fall short, especially in situations with poor lighting, for instance, detection during night-time. In this paper, we propose a paced multi-stage block-wise framework for effectively detecting objects from thermal images. Our approach utilizes the pre-existing knowledge of deep neural network-based object detectors trained on large-scale natural image data to enhance performance in the thermal domain constructively. The employed, multi-stage approach drives our model to achieve higher accuracies. And the introduction of the pace parameter during domain adaption enables efficient training. Our experimental results demonstrate that the framework outperforms previous benchmarks on the FLIR ADAS dataset on the person, bicycle, and car categories. We have also illustrated further analysis of the framework, such as the effect of its components on accuracy and training efficiency, its generalizability to other thermal datasets, and its superior performance on night-time images in contrast to state-of-the-art RGB object detectors.

Keywords Object detection · Thermal images · Pace · Multi-stage · Domain adaptation · Transfer learning · EfficientDet

1 Introduction

With the increasing popularity of artificial intelligence and machine learning in recent years, more techniques are being developed for object detection. However, the majority of interest has been focused on object detection in the visible spectrum [2] for applications such as surveillance and self-driving vehicles. Although very effective with the current state-of-the-art technologies, its limitations arise in situations like difficult lighting conditions, camouflaging colours, and environmental occlusions. Attention has turned towards thermal imaging with the infrared (IR) spectrum to solve the problems in the RGB images, and thus, the thermal sensors are seeing an industry boom [19]. Another vital advantage that thermal imaging has over RGB format is the added property of privacy protection. People captured in the visible spectrum are readily identifiable, which is not the case in thermal images [25].

The recent COVID-19 pandemic may cause the global thermal sensing market to reach 6.7 billion dollars within the next four years, according to [18]. This increase in thermal cameras also increases the importance of a robust object detection methodology on thermal images, especially when considering the vital class of objects in pedestrian detection applications, such as persons, cars, and bicycles. The need for both an accurate and efficient framework for training such models is a challenging task since the pursuit for increased accuracy often leads to decreased efficiency and vice versa. We aimed to create a framework that learns object detection in IR images as efficiently as possible while attaining increased accuracy.

Another challenging aspect of object detection in thermal images is the limited number of object detection algorithms in the IR domain compared to its RGB counterparts. This

This work was submitted when J. Jennifer Ranjani was associated with the Institute.

✉ J. Jennifer Ranjani
j.jenniferranjani@yahoo.co.in

Shreyas Bhat Kera
f20181119@pilani.bits-pilani.ac.in

Anand Tadepalli
f20181117@pilani.bitspilani.ac.in

[1] Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani Campus, Pilani 333 031, India

might be due to factors such as a limited number of datasets and the markedly smaller size of these datasets compared to their RGB counterparts [5,7,10,13,14,16]. For example, the popular RGB dataset, Common Objects in Context (COCO) [29] contains over 200,000 labelled images, while the FLIR system's Advanced Driver Assistance Systems dataset (FLIR ADAS) [20] consists of 10,000 labelled thermal images only. We address this challenge by utilizing state-of-the-art detectors with rich visible information and applying it to the IR domain. Some researchers have worked with multi-spectral datasets [23], wherein the detectors make use of aligned RGB-thermal pairs and can benefit from the advantages of both domains. However, this may not always be practical since applications like self-driving cars, surveillance, etc., do not have access to cameras that capture the required synchronized multi-modal inputs. Thus, focusing only on thermal imagery reduces the computational burden, and it facilitates the usage of our method more suitable for daily application.

Specific aspects of thermal images in object detection have been largely unexplored, such as the generalizability of models to other thermal datasets, as well as a quantifiable approach to the effect that occlusion has on thermal object detection. We have addressed these challenges also in our work. In this paper, we propose a novel framework for object detection in thermal images, namely the paced multi-stage block-wise framework, for object detection on thermal images. We have introduced the usage of multiple stages of training to accurately adapt a state-of-the-art detector to the thermal domain while also leveraging information from the visible domain. We have also introduced the concept of pace in the block-wise domain adaptation, which improves efficiency without a significant accuracy drop. Using this, to the best of our knowledge, we report the highest accuracy on the FLIR ADAS dataset on the classes of persons, cars, and bicycles for any existing thermal object detection framework. Also, we have demonstrated several inferences, such as how the detector retained information in the visible domain while adapting to the thermal, the framework's efficacy when dealing with night-time images, the effectiveness on other thermal datasets, and the performance of our detector when objects are occluded.

2 Related works

Researches have come up with solutions that bridge the gap between thermal and RGB detectors, which can be categorized into two types, those that use multispectral imagery and those that use solely thermal imagery.

2.1 Multispectral pedestrian detection

Several works such as [37,39,45,46] rely on visual images for detection; [37] proposed a novel backbone architecture for pedestrian detection based on the human visual system that can be applied to most of the existing architectures. In many cases combining RGB and thermal images has improved the accuracy of object detectors. For example, improved detection in difficult lighting conditions with a cross-modality learning comprised of a region reconstruction network (RRN) and multi-scale detection network (MDN) was used in [44]. [42] devised a fusion network of RGB and thermal image pairs and explored two different types of networks, early fusion, and late fusion. Illumination-aware faster R-CNN (IAF RCNN) [27] used FRCNN to perform multi-spectral pedestrian detection by leveraging a two-stage network to combine RBG and thermal image features. [28] also proposed the use of a fusion network by combining results from a multi-spectral proposal network (MPN) and a multi-spectral classification network (MCN) to perform pedestrian detection. A region feature alignment module along with weighted feature fusion was proposed by [48].

Pseudo-multi-modal thermal object detector (MMTOD) [12] consists of parallel ResNet branches for thermal and RGB images, respectively. These branches capture the features connecting the two spectra before being passed through a Faster-RCNN [36]. It performs inference on only thermal images as it does image-to-image translation and creates a pseudo-RGB image using CycleGAN [51], which acts as a second input.

Though the above detectors leverage the power of two-stage networks, they are preferred when learning features of the thermal and RGB inputs fall short in speed. These detectors have a high inference time because of their complex frameworks. A popularly used solution to speed up the inference time is to use a single-stage framework. For example, two single-shot detectors (SSDs) were adopted by [50] to fuse RGB and thermal features with gated fusion units (GFU).

Many works also use RGB-Depth images for object detection or pose estimation. In [39], transfer learning on a pre-trained deep CNN is utilized to provide a rich feature set, and it incorporates a depth channel according to distance from the object centre. In [45], a multistream input of flow, RGB, and depth combined with the contextual region of interest pooling layers that deal with contextual information for joint human detection and head pose estimation, is proposed. Utilization of 3D physical structure and colour information along with a multi-channel colour shape descriptor proposed in [46] works as a physical blob detector to detect humans. Occlusion is another factor that affects the performance in the realm of multi-object tracking, as observed in [49].

2.2 Pedestrian detection in thermal imagery

Initial attempts such as [24] use an adaptive fuzzy C-means clustering for segmentation and retrieval of candidate pedestrians, which were then classified with the CNNs. The resultant architecture was computationally less complex as compared to the sliding window framework. The work in [1] uses thermal position intensity histogram of oriented gradients (TPIHOG) and additive kernel SVM (AKSVM) for night-time only detection. The authors in [17] use a pixel-wise contextual attention network (PiCA-Net [30]) or R3-Net [11] to create saliency maps. Then, faster R-CNN [36] is trained for pedestrian detection using the original thermal image, replacing the last channel with the generated saliency map.

A few approaches leverage RGB images as a data augmentation by performing thermal to RGB image translation. For instance, several data pre-processing steps were applied by [22] to make thermal images look more similar to greyscale-converted RGB images, then a fine-tuning step was performed on a pre-trained SSD300 [31] detector. The common drawbacks of most of the methods mentioned above are the use of many complex pre-processing steps or handcrafted features, negatively impacting performance.

In [6], as thermal images contain lesser information as compared to RGB images (colour, texture), the paper attempts to capture as much information from the ResNet backbone as possible using a dual-pass fusion block (DFB) and a channel-wise enhance module (CEM) to retrieve information from every layer, combining each of the features with varying weightage.

More recently, the authors of [8] prioritized efficiency using a VGG network and made robust with a residual branch, which was used only during training thereby retaining the inference time performance. It was also improved using their proposed continuous fusion strategy. In [47], a dilation and deconvolution single-shot multibox detector (DDSSD) that improves SSD with feature fusion using dilation convolution and deconvolution modules for better performance on smaller objects was proposed.

Domain adaptation [26], a form of transfer learning, attempts to use the learned knowledge from the source domain on the new target domain. Early works of domain adaptation used feature transformations (inversion, equalization, and histogram stretching) to convert the thermal images to as close as RGB images. Another approach [21] used a shallow CNN before the main model that transforms the input image to the target. The authors of [26] later tried a top-down domain adaptation approach [25] where pre-trained weights from the RGB spectrum were retrained for the thermal spectrum using top-down loss. Another approach to domain adaptation was explored in [34], where the style of an RGB image was applied to a thermal image to transfer the low-level features of RGB images to thermal images, while still maintaining the high-level features. This was carried out by a multi-style generative network (MS-GNet) which draws inspiration from GANs such as CycleGAN. The resultant image is fed into a cross-domain detection model which is a pre-trained RGB detector, fine-tuned on the outputs of the MS-GNet.

The layer-wise domain adaptation builds on [26] slowing down the training procedure to retain more knowledge from the original RGB domain. The approach also trains using a bottom-up approach rather than top-down loss to train the network. This is done by progressively training the network one layer at a time from the bottom to the top.

The rest of this paper is organized as follows: Sect. 3 describes our approach to create the proposed object detection framework and its finer details. Section 4 describes the datasets used, evaluation metrics, baseline, set-up, and the experimental results. We summarize our contribution and discuss further research plans in Sect. 5.

3 Proposed methodology

Our objective was to improve over the industry-standard architectures used in thermal object detection by implementing a combination of transfer learning and domain adaptation with the help of state-of-the-art research on RGB object detection in the form of EfficientDet [41]. In this section, we discuss the motivation behind choosing the base architecture and the process of multi-stage domain adaptation.

3.1 Choice of base architecture

The first motive for selecting EfficientDet is its ability to support compound scaling as it provides a range of configurations by varying resolution, depth, and width of the architecture. This aids in finding the suitable architecture needed for our motive to transfer knowledge accurately and efficiently from the RGB to thermal spectrum for object detection.

The second criterion that EfficientDet satisfies is the ability of the model to borrow features from the rich visible spectrum [12]. An accurate object detection model trained on a substantial RGB object detection dataset is needed to obtain this information. EfficientDet (D7x configuration) obtained a new state-of-the-art average precision (AP) of 55.1 on the large-scale RGB object detection dataset, MS-COCO [29]. Further, EfficientNet [40], the backbone of EfficientDet, is pre-trained on the ImageNet [10] dataset, providing further insight and information into features in the visible domain.

Third, the modularity of the architecture can efficiently adapt information from the RGB to the thermal domain. EfficientDet comprises four modular components: the backbone,

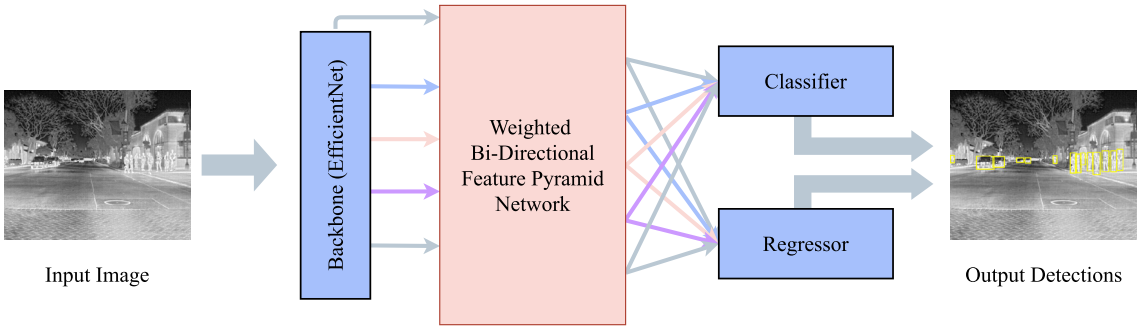

Fig. 1 A simplified diagram of the EfficientDet architecture, highlighting the modules used to construct it, i.e. the EfficientNet backbone, the weighted bidirectional feature pyramid network (BiFPN), the classifier, and the regressor. In our case, a thermal image is given as input to produce appropriate output detections, i.e. bounding boxes around detected objects and the corresponding class label for each object

bidirectional feature pyramid network (BiFPN), regressor, and classifier, as shown in Fig. 1. Further, the EfficientNet backbone comprises a multitude of mobile inverted bottleneck blocks (MBConv) [38] which makes the design within the backbone modular. We utilized these features during block-wise and multi-stage training in our framework, described in Sect. 3.2.

Model efficiency is the last criterion for choosing an architecture. Among the various configurations, EfficientDet-D0 achieves accuracy similar to YOLOv3 [35] with 28x fewer FLOPs. Similarly, even the largest and most accurate configuration EfficientDet-D7x uses 7x lesser number of FLOPs than the prior state-of-the-art methods. With all these considered, we further endeavour to improve training efficiency with the pace parameter used in our framework, described in Sect. 3.2.1. In the next section, we describe the features of the proposed paced multi-staged block-wise approach.

3.2 Multi-stage domain adaptation

The principal component of the proposed framework is the multi-stage domain adaptation approach that transfers knowledge from the RGB to the thermal domain. The base architecture, EfficientDet [41], comes with different components, each serving a specific purpose. The EfficientNet [40] backbone generates features which are fed into the BiFPN to perform fast, multi-scale feature fusion followed by the bounding box and class prediction networks. We hypothesized that training specific components of the architecture in multiple stages could successfully adapt the entire network for the thermal spectrum and give accurate results. The initial state of the EfficientDet model loaded with information-rich pre-trained weights from the RGB domain is used rather than training the network from scratch. Thus, the model, when trained with the multi-staged approach, adds information to the features it has already learned from the COCO dataset [29]. Conversely, replacing the said features would be detrimental to the model's performance since these features contain a large amount of helpful information for object detection. In the proposed design, we carried out the multi-stage procedure in three stages: block-wise backbone training, first round fine-tuning, and second round fine-tuning. We have detailed them in the following sections. To better elucidate the framework, we first introduce and explain some notations.

Notations Let E denote the entire EfficientDet network used in this approach, and let x refer to the input image such that $x \epsilon X$, where X represents the thermal domain, i.e. images in the thermal dataset. As explained in [41], there are different configurations of the EfficientDet model, and we have represented it by a compound coefficient denoted by ϕ. In this paper, we use four configurations of EfficientDet; in other words, the values of ϕ we use are $D0$, $D1$, $D2$, and $D3$. Let the network with configuration ϕ working on an input image x be denoted by $E_\phi(x)$. The network consists of four different modules denoted as follows: the backbone, Γ, the BiFPN, β, the classifier, C, and the regressor, R. Thus, the entire network with default settings of its modules is denoted as $E_\phi(\Gamma, \beta, C, R)$. When a particular module is frozen, the weights for all parameters in all the layers in that module are not learnable, and we denote such modules with a bar. For instance, if the entire BiFPN is frozen, then it is denoted by $\bar{\beta}$. Since the backbone, Γ, is comprised of several MBConv blocks [38], let ψ signify this number. If the first n blocks in the backbone are unfrozen, while the remaining are frozen, then let this setting of the backbone be denoted by, $\bar{\Gamma}_{n:\psi}$. The pace parameter, explained in the next section, is denoted by P.

3.2.1 Block-wise backbone training

As stated earlier, the chosen backbone architecture, EfficientNet, acts as a feature extractor and incorporates several MBConv blocks. Though this backbone can have varying

Table 1 Compound coefficients $D0$–$D3$ and their corresponding number of blocks in the backbone

Sl. No.	ϕ	ψ
1	$D0$	16
2	$D1$	23
3	$D2$	23
4	$D3$	26

depths, widths, and resolutions based on the compound scaling method, for the first stage of training, we focus on the depth of the backbone, as this controls the number of MBConv blocks used in the architecture. Each block can be assigned a number from 0 to $\psi - 1$ in a sequential manner. The value of ψ is dependent on the configuration of EfficientDet, which is decided by the compound coefficient ϕ. Based on the original implementation of EfficientDet, we have assigned the values of ψ and ϕ as shown in Table 1. Note that we only include the values of ψ for those configurations used in our experiments. After determining the number of blocks used in the backbone, it becomes necessary to adapt their weights to the thermal domain methodically. The authors of [25] demonstrated that a bottom-up adaptation of the network could provide accurate results. We integrate a similar approach to training the backbone of the network, specializing it for the architecture of EfficientNet. Initially from the network $E_\phi(\Gamma, \beta, C, R)$, we set the network to $E_\phi(\bar{\Gamma}, \beta, C, R)$ by freezing all the parameters of every block from 0 to $\psi - 1$ in the backbone. Following this, we unfreeze one block at a time, i.e. unfreezing one block per epoch till all the blocks have unfrozen. Therefore, for any epoch e ($e < \psi$), the network would be $E_\phi(\bar{\Gamma}_{e:\psi}, \beta, C, R)$. This gradual, sequential unfreezing of layers ensures that the backbone has successfully adapted to the thermal domain with satisfactory accuracy. Figure 2 portrays the block-wise backbone training.

Pace The training time of the traditional block-wise backbone training increases in proportion to the values of ϕ as the size of the architecture increases. We have introduced a paced block-wise backbone training to reduce the training time without drastically affecting the detection accuracy by adding a pace parameter (P). This parameter (P) takes an integer value and works as follows: starting with the network $E_\phi(\bar{\Gamma}, \beta, C, R)$, we unfreeze P layers at a time rather than just one. Therefore, for any epoch e ($e < \psi$), the network would be

$$E_\phi(\bar{\Gamma}_{x:\psi}, \beta, C, R), \quad \text{where}$$

$$x = \begin{cases} e * P & \text{if } e * P \leq \psi, \\ \psi & \text{if } e * P > \psi \end{cases} \tag{1}$$

The pace process, visualized in Fig. 3, speeds up the training process, making the framework more efficient. The number of epochs required for this stage of training is $\left\lceil \frac{\psi}{P} \right\rceil$ when compared to ψ using the traditional approach. Here, the number of epochs decreases for a given compound coefficient as the pace parameter increases. However, we expect that, even intuitively, after a certain level, the pace might become too quick for the model to transfer to the thermal domain accurately. Hence, to achieve efficient training and accurate results, a moderate value for the pace parameter is required, which is demonstrated in Sect. 4.6.1.

After this initial stage of adapting the backbone to the target domain, the results can be further improved with the use of the following two stages.

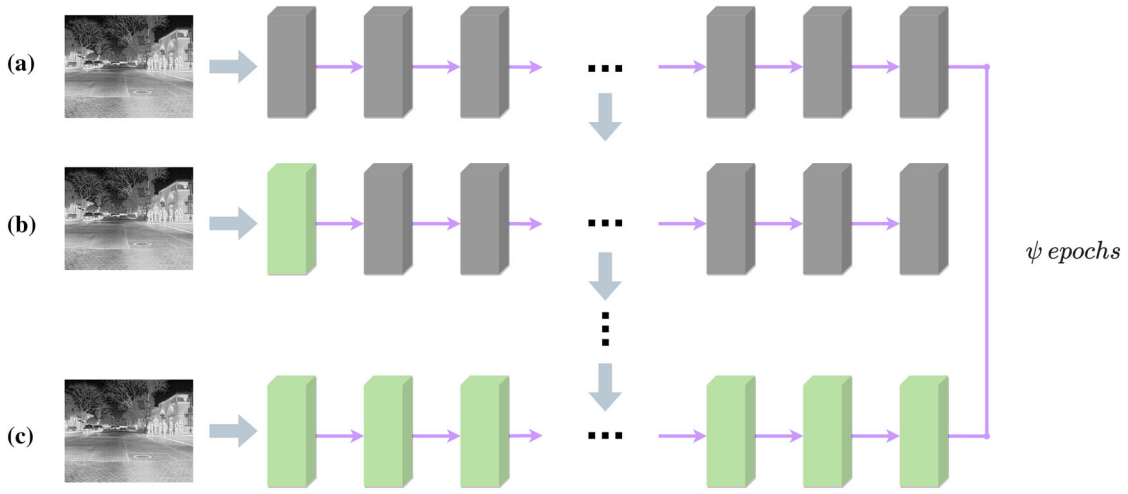

Fig. 2 An overview of block-wise backbone training: **a** exhibits a simplified version of the initial setting of the framework with all blocks frozen (gray), **b** displays the first epoch when the first block is unfrozen (green) and **c** represents the final epoch of the first stage where all the blocks have become unfrozen, after ψ epochs

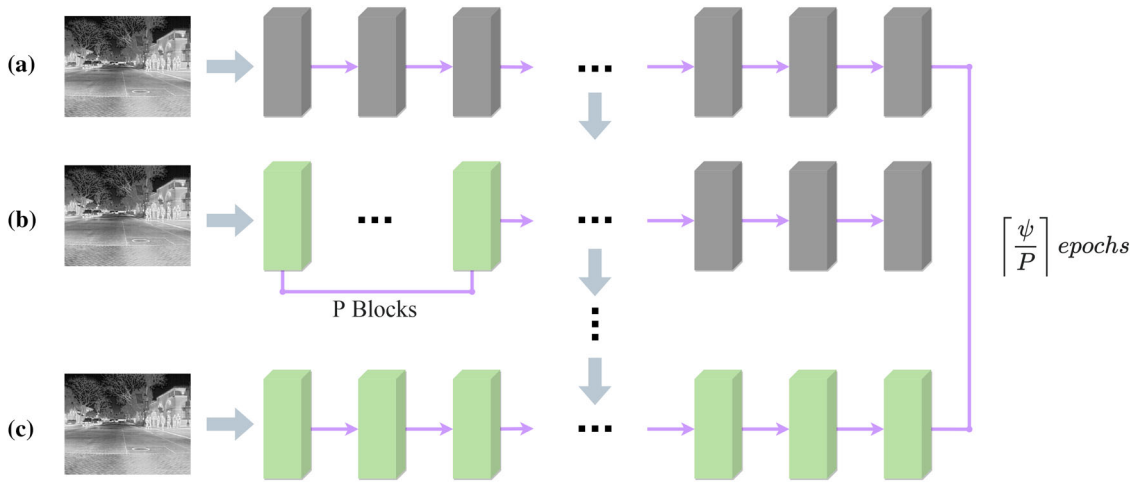

Fig. 3 An overview of block-wise backbone training with the use of the pace parameter to make the training process efficient: **a** a simplified version of the initial setting of the framework with all blocks frozen (gray), **b** displays the first epoch when the first P blocks are unfrozen (green), with each subsequent epoch unfreezing P more blocks and **c** represents the final epoch of the first stage where all the blocks have become unfrozen, after $\left\lceil \frac{\psi}{P} \right\rceil$ epochs

3.2.2 First round fine-tuning

During the first stage, we have trained only the layers of the EfficientNet Backbone, and hence it becomes necessary to train the remaining components of the detector using fine-tuning. The backbone, having been trained for $\left\lceil \frac{\psi}{P} \right\rceil$ epochs, can also be trained further in this stage. Thus, the purpose of this additional stage of training is to focus on fine-tuning the entire network, which allows all the parameters in the network to further adapt to the new domain, thereby improving the framework's accuracy. A fixed network parameter setting, $E_\phi(\Gamma, \beta, C, R)$, is used for the training procedure in this stage; the backbone, BiFPN, classifier, and regressor are unfrozen. Training continues until there is no further improvement in the network.

3.2.3 Second round fine-tuning

The final stage of our multi-staged approach has a similar purpose as the previous stage, but the target modules are only the classifier and regressor, i.e. only the head of the detector. We observed that training the model for a number of epochs with the network set to the configuration of $E_\phi(\bar{\Gamma}, \bar{\beta}, C, R)$, consistently results in a small boost in performance over the first round of fine-tuning and allows the detector to be as accurate as possible. Performing this round of fine-tuning is also computationally inexpensive as the training is done only on the classifier and regressor, thus serving the purpose of efficiently attaining even higher accuracy, enough to obtain a new state of the art. The number of epochs needed for this round is usually very less before saturation. We cease at this stage of training once the reported accuracies have saturated, which we observed to have more visible outcomes for larger values of ϕ.

The entire training procedure is summarized in Algorithm 1. It was evident that increasing the value of ϕ would also improve accuracies, as shown in [41]. However, the effect of increasing ϕ appears to be more significant, especially when dealing with small objects. This detail is discussed further in Sect. 4.5.

4 Experimental results and discussion

4.1 Datasets

FLIR starter thermal dataset The dataset used for training was the FLIR starter thermal dataset [20]. This dataset contains 10,288 thermal images captured on a FLIR Tau2 camera, with a collection of RGB images that may or may not have a pair with a thermal counterpart. For our approach, we solely utilized the thermal images in the dataset for training and testing. The dataset consists of 8862 images for training and 1366 images for testing. The annotations of the dataset followed the COCO Dataset [29] format, and only the classes of person, bicycle, car, and dog were used during the annotation. However, following the precedence of the baseline and previous research, we train and test on only the three main categories, i.e. person, bicycle, and car. Although the dataset contains RGB images, it does not have any annotations. The annotations corresponding to the thermal counterpart may be inaccurate and misaligned for the RGB images. The final dataset comprises 67,618 annotations (22,372 for person,

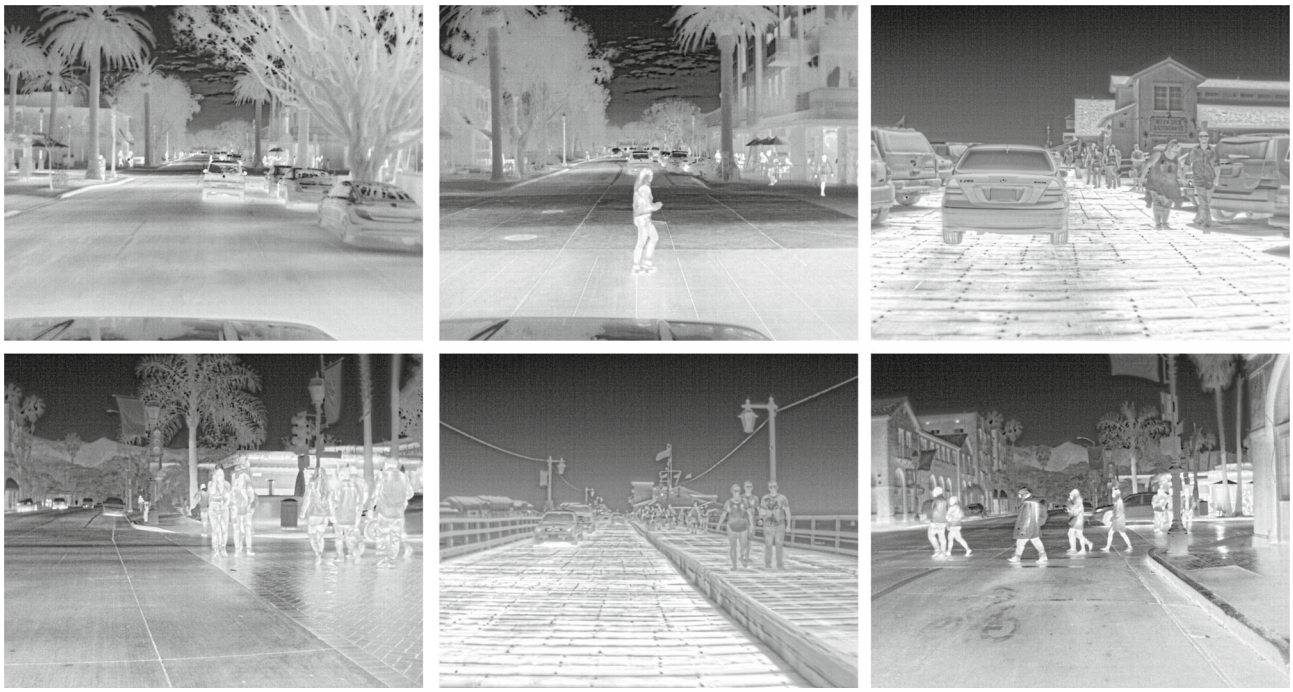

Fig. 4 Examples of the FLIR ADAS dataset

3986 for bicycle, 41,260 for car) in the training set and 11,682 annotations (5779 for person, 471 for bicycle, 5432 for car) in the test set. Each image resolution is 640 × 512. Some sample images can be seen in Fig. 4.

Other datasets

(a) We utilize the thermal images of databases present in the OTCBVS benchmark dataset collection, including the OSU thermal pedestrian database [9], the terravic motion IR database [33], and the BU-TIV (Thermal Infrared Video) benchmark [43]. Thermal cameras such as the Raytheon PalmIR 250D and Raytheon L-3 Thermal-Eye 2000AS were used to capture these datasets. These cameras differ from the FLIR Tau2 images on which we have trained the models. We consider only the person category, as these datasets are specialized for pedestrians/people.

(b) The hiding subset of the LTIR dataset [3] contains 358 images of a single annotated instance of the person category. We sectioned the dataset into two based on occlusion: no occlusion, which consists of 213 images, without any obstruction in front of the person, and full or partial occlusion, which consists of 145 images with some object. We used this dataset to test the model's performance when detecting occluded objects or subjects in an image.

4.2 Evaluation metrics

We use the evaluation metric of mean average precision (mAP) for all experiments using the paced multi-staged block-wise framework for object detection in thermal images. The process of calculating this metric begins with calculating Precision and Recall as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

where TP denotes the number of accurately detected bounding boxes for a given Intersection of Union (IoU). IoU is the ratio of the area of overlap to the union between the ground truth and prediction. FP denotes the number of incorrect detections, while FN denotes the number of ground truth detections that were missed during prediction. To compare with other state-of-the-art methodologies, we follow the COCO 101 point metric of determining AP (average precision), i.e.:

$$\text{AP} = \frac{1}{101} \sum_{x \in \text{MP}} (x) \tag{4}$$

where MP denotes the maximum precision in the recall area (for COCO, it is 0 to 1 in steps of 0.01). As mentioned earlier,

Algorithm 1: Paced Multi-Stage Block-Wise Framework Training Procedure

Input : Training data: $\{x_i\}_{i=1}^{n} \epsilon X$; pretrained initial EfficientDet network: $E_\phi(\Gamma, \beta, C, R)$, where $\phi \epsilon \{D0, D1, D2, D3\}$; pace parameter: P; number of blocks in backbone: ψ; number of epochs for each stage: e_1, e_2, e_3 where $e_1 = \left\lceil \frac{\psi}{P} \right\rceil$

Output: Trained thermal object detection network

Set network to $E_\phi(\bar{\Gamma}, \beta, C, R)$.

Stage 1: Block-Wise Backbone Training

for e_1 **do**
  Let e be the current epoch.
  for x_i, $i = 1, \cdots, n$ **do**
    if $e * P \leq \psi$ **then**
      train the network $E_\phi(\bar{\Gamma}_{e*P:\psi}, \beta, C, R)$ for image x_i.
    else
      train the network $E_\phi(\Gamma, \beta, C, R)$ for image x_i.
    end
  end
end

Stage 2: First Round Fine-Tuning

for e_2 **do**
  for x_i, $i = 1, \cdots, n$ **do**
    train the network $E_\phi(\Gamma, \beta, C, R)$ for image x_i
  end
end

Stage 3: Second Round Fine-Tuning

for e_3 **do**
  for x_i, $i = 1, \cdots, n$ **do**
    train the network $E_\phi(\bar{\Gamma}, \bar{\beta}, C, R)$ for image x_i
  end
end

for evaluation purposes, we use three main classes of the FLIR ADAS dataset: person, bicycle, and car. Let us denote this set of classes as C. We calculate the AP values for these classes, and the following averaging equation gives the final mAP value:

$$\text{mAP} = \frac{\sum_{c \epsilon C} \text{AP}_c}{|C|} \tag{5}$$

where in this case $|C| = 3$.

We have used the official pycocotools package (https://pypi.org/project/pycocotools/) for the source code. The IoU is fixed at 0.5, similar to state-of-the-art models. We also calculate the mAP for different object sizes: small, medium, and large, denoted by AP_S, AP_M, and AP_L, respectively. Small objects have an area less than 32^2, medium objects between 32^2 and 96^2 and large objects greater than 96^2.

4.3 Baseline

A baseline accuracy was established from the FLIR ADAS dataset [20], with an mAP of 54.0% at IoU of 0.5. Further, we compare the performance of our framework with the state-of-the-art models, primarily those who have dealt with the FLIR ADAS dataset, to show that our approach is not only competitive with prior work but also provides the highest overall mAP values. As a point of reference, the prior state-of-the-art method [6] achieved an mAP of 74.6%.

4.4 Experimental setup

We have conducted all the experiments using PyTorch implementation of EfficientDet, and we have made the training code available at https://github.com/shreyas-bk/PMBW_Object_Detection_In_Thermal_Images. The EfficientDet series comes with pre-trained weights on the COCO dataset, which serves as the starting point for all training instances. The maximum coefficient used in our experiments is $D3$, for two reasons: we found no significant increase in performance between coefficients $D2$ and $D3$, and the computational power increases for higher network configuration. We apply only the first two stages of our framework when working with coefficients $D0$ and $D1$ and the complete framework for coefficients $D2$ and $D3$. The pace parameter is varied primarily when using coefficient $D1$ to determine which pace is the most optimal, as shown in Table 4. Before feeding the thermal images into the network, we normalize them with the calculated mean and standard deviation of 0.53 and 0.19, respectively (assuming image pixel intensities are in the range [0, 1]). The optimizers we have used vary for the different stages of training: we use the AdamW optimizer [32] (a variant of the Adam optimizer that uses decoupled weight decay regularization) for the block-wise backbone training for both the first and second round of fine-tuning we select the stochastic gradient descent (SGD) optimizer with a Nesterov momentum of 0.9. A vital factor in our setup was the learning rate. When training is in the first stage, we used an exponential decay of learning rate, as we found it provided better performance without affecting the efficiency. The γ parameter (multiplicative factor of the learning rate for every epoch) was set to 0.75 for the given decay, with a base learning rate of 0.001. Also, this exponentially decaying learning rate was applied to coefficients $D2$ and $D3$ during the first round of fine-tuning. For all other training stages, we have used a fixed learning rate of 0.001.

4.5 Results

4.5.1 Systematic results for the coefficients

This section systematically describes how each coefficient from $D0$–$D3$ was trained and tested. Here, in order to represent a completely trained system of our proposed paced multi-stage block-wise framework, we denote it with PMBW(ϕ, P), where ϕ is the compound coefficient and P is the pace parameter used in the block-wise backbone stage of training. More precisely, the term PMBW entails all train-

Table 2 Results with $\phi = D0$

Framework	AP_S	AP_M	AP_L	mAP
PMBW($D0$, 1)	29.0	81.2	81.6	55.6

Table 3 EfficientDet results for different sizes on the COCO dataset

ϕ	AP_S	AP_M	AP_L	mAP
$D0$	12.0	38.3	51.2	33.8
$D1$	17.9	44.3	56.0	39.6
$D2$	22.5	47.0	58.4	43.0
$D3$	26.6	49.4	59.8	45.8

Table 4 Results with $\phi = D1$

Framework	AP_S	AP_M	AP_L	mAP
PMBW($D1$, 1)	43.6	86.3	85.2	67.7
PMBW($D1$, 2)	43.8	85.0	85.2	67.4
PMBW($D1$, 3)	42.7	85.3	86.0	67.2
PMBW($D1$, 4)	35.3	78.1	74.8	58.6

ing stages of the multi-stage approach detailed earlier with its various settings of the base network $E_\phi(\Gamma, \beta, C, R)$. All three stages are assumed to have been performed unless specified otherwise.

Compound coefficient $\phi = D0$ The framework used for this coefficient was PMBW($D0$, 1), with only the initial two stages of training. The results for this setting are tabulated in Table 2. Notably, the mAP value, 55.6%, exceeded the baseline using only the first coefficient. We can easily observe that the performance of this setting on small objects is nearly 50% off the mAP values for medium and large objects. With this observation, we can discuss the base object detector's capabilities when detecting objects of varied sizes. From Table 3 taken from [4] we can see that the AP values are proportional to the coefficient value. However, it is also important to note that the values of AP_S increase quicker than the values of AP_M and AP_L; for example, the increase in AP_S from $D0$ to $D3$ is 14.6%, which is a substantially larger increase when compared to the 8.6% increase of AP_L. Since the AP_S values of our framework are poor for this coefficient, we shifted our attention majorly to $\phi = 1$ and beyond. The values for AP_M and AP_L are already comparatively high and are not the primary area that needs to be enhanced, though there is room for improvement. Further, the mAP is still nearly 20% off the state of the art, giving us another reason to focus our efforts on higher compound coefficients.

Compound coefficient $\phi = D1$ Starting with $\phi = 1$ we implemented different values for the pace parameter of the block-wise backbone training with the networks PMBW($D1$, 1), PMBW($D1$, 2), PMBW($D1$, 3), and PMBW($D1$, 4). All implementations used just the first two stages. The results for this setting can be seen in Table 4. Among the different pace variations, the trained network PMBW($D1$, 1) obtained the greatest performance with an mAP of 67.7%. However, as we had hypothesized, there was a marginal difference compared to the other pace settings. For example, using a larger pace parameter resulted in similar mAP values for PMBW($D1$, 2) and PMBW($D1$, 3). Both had a marginal drop in performance of 0.3% and 0.5%, respectively, from PMBW($D1$, 1). As delineated in the methodology, the training time for these two settings was markedly lower than PMBW($D1$, 1). Hence, the training efficiency (explained in Eq. 6) improved without drastically hampering the performance. As predicted earlier, improvement in efficiency without a significant loss in accuracy is observed when we increased the pace parameter further, which is apparent from the results for PMBW($D1$, 4), where the mAP values were nearly 9% off PMBW($D1$, 1). Thus, we can find an optimal value of the pace parameter, providing increased efficiency with competitive accuracy. From the results, it is evident that there was a considerable boost of 13.3% in overall mAP compared to $\phi = D0$. However, the highest mAP value obtained was still 5% off of the state of the art. Also, the performance on small objects was poor, i.e. 40% away from medium and large objects. We also experimented with the effect of block-wise backbone training compared to training without block-wise domain adaptation. The network setting of $E_{D1}(\Gamma, \beta, C, R)$, when trained for the same number of epochs as the block-wise backbone stage of PMBW($D1$, 1), resulted in a maximum mAP of only 60%.

Compound coefficient $\phi = D2$ Following the idea from $\phi = D1$, that implementing pace could improve training efficiency without reducing performance, we directly applied a pace of 2 and 3 for $\phi = D2$, i.e. the networks used were PMBW($D2$, 2) and PMBW($D2$, 3). The results can be found in Table 5. The first stage was carried out normally, using the specified pace, while for the second stage, we observed an average increase of 1% in mAP using the exponential decay of learning rate. The network PMBW($D2$, 3) gave the highest mAP of 75.6% after the second stage. Following this, we applied the final stage of the second round of fine-tuning for both configurations. From the results, we can verify that an increase of 0.5% and 1.6% were achieved for PMBW($D2$, 2) and PMBW($D2$, 3), respectively. Notably, PMBW($D2$, 3) reached an mAP value of 77.2%, thereby improving upon the state of the art. In this case, the gap between mAP values for small objects and medium and large objects reduced to 25%.

Compound coefficient $\phi = D3$ For this final coefficient, using the knowledge we had gained from the previous experiments, we took the single configuration of PMBW($D3$, 3) with all three stages. The results given in Table 6 show that

Table 5 Results with $\phi = D2$

Framework	After second stage	After third stage	AP_S	AP_M	AP_L
PMBW($D2$, 2)	73.4	73.9	58.8	85.3	84.8
PMBW($D2$, 3)	75.6	77.2	61.4	88.7	87.6

Table 6 Results with $\phi = D3$

Framework	AP_S	AP_M	AP_L	mAP
PMBW($D3$, 3)	64.6	86.7	82.1	77.3

the final value procured only slightly exceeds that of $\phi = D2$, but it is the highest mAP value obtained in all the configurations of our framework and a new state-of-the-art on the FLIR ADAS dataset. Examples of detections using PMBW($D3$, 3) can be found in Fig. 5.

4.5.2 Final results and discussion

Table 7 shows the highest mAP value and paced multi-stage block-wise setting for each value of ϕ. To the best of our knowledge, the proposed PMBW($D3$, 3) framework achieves the highest overall mAP, among existing thermal object detectors, for the FLIR ADAS dataset. PMBW($D3$, 3) yields the highest mAP for person and car, whereas PMBW($D2$, 3) for the bicycle class. Table 8 demonstrates this by comparing with other methodologies [6,12,15,25,34]. We consider the mAP for only the person, bicycle, and car categories from [15], which is simply the mean of the reported AP's.

Our new state-of-the-art mAP value is 77.26% (person-81.19%, bicycle-64.04%, car-86.55%), and in Table 8 these values are marked in bold. We demonstrate the utility of the procedural components we have introduced in this paper in the following section.

4.6 Further discussion

4.6.1 Effect of pace

To demonstrate the effectiveness of Pace, we are required to quantify the training efficiency. The training efficiency depends on the performance of the model and the number of epochs required to achieve the desired performance. The time taken per epoch is nearly the same across different values of Pace, as we consider only the coefficient $D1$ for our experiments. To measure the performance, we have devised Eq. (6):

$$\eta_{P,\phi} = \frac{\text{mAP(PMBW}(\phi, P)) - \text{mAP(Baseline)}}{e} \tag{6}$$

where $\eta_{P,\phi}$ is the training efficiency and mAP(PMBW(ϕ, P) is the achieved mAP value for a particular coefficient and pace. mAP(Baseline) is the mAP value of the baseline (54.0%), and e is the number of epochs taken for training. The results can be seen in Fig. 6 for different pace values when $\phi = 1$. Thus, as we had intended, adding a certain amount of pace can speed up the training process without hampering the overall performance. Further, we had also expected that after a certain point, the pace would be too quick for the model to successfully adapt to the thermal domain, which is visible when using the PMBW($D1$, 4) framework.

4.6.2 Effect of multiple stages

We can demonstrate the effect of multiple stages by plotting the mAP values for each value of ϕ across stages. In Fig. 7, we represent the stages on the x-axis, with stage 0 implying the mAP value calculated using the EfficientDet detector loaded with the pre-trained weight for the respective value of ϕ, without any training. As shown in Fig. 7 for every value of ϕ taken, there is an increase in mAP values for each consecutive stage. The average increase in mAP was 7.7% for block-wise backbone training, 5.45% for first-round fine-tuning, and 2.55% for second-round fine-tuning. Thus, the necessity of multiple stages as well the effectiveness of the block-wise backbone training is evident.

We have examined the training efficiency of the fine-tuning round by calculating the average increase in mAP per epoch from the first stage to the second stage. The results from Fig. 8 show that for pace value of 2 for PMBW($D1$, P) has a large improvement in training efficiency. The low training efficiency for PMBW($D1$, 1) may be since it took a larger number of epochs to train in the first stage and may already be nearly saturated before the second stage. However, the result that this ablation study indicates in Fig. 8 is similar to that of Sect. 4.6.1; there is an initial increase in training efficiency followed by a decrease.

4.6.3 Performance of the trained model on other datasets

OTCVBS We tested our trained model on the images of the OTCVBS [9,33,43] dataset to show that the model can adapt other thermal datasets that the detector has not previously seen. Visual evaluation of the model is shown in Fig. 9. Because they were captured on different sensors as mentioned in Sect. 4.1 the images of these datasets are diverse

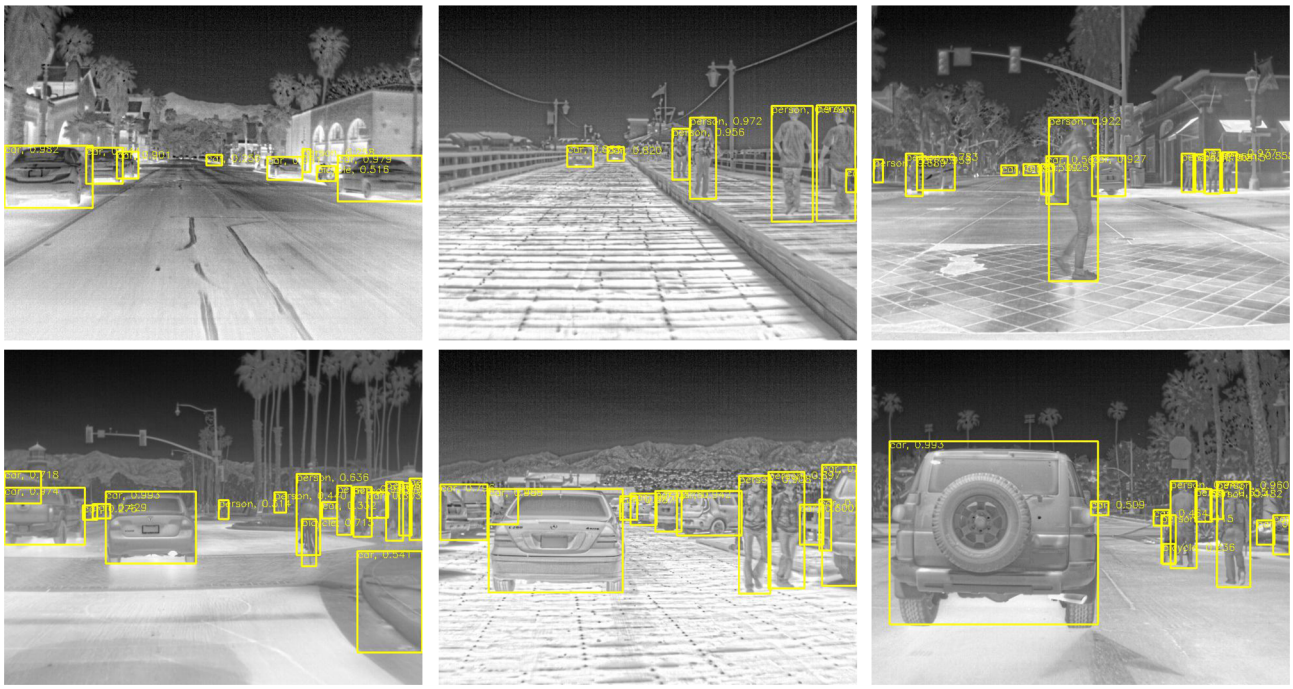

Fig. 5 Examples of detections on thermal images from the FLIR ADAS dataset using our PMBW($D3$, 3) framework

Table 7 Best result for each value of ϕ

Framework	Person	Bicycle	Car	mAP
PMBW($D0$, 1)	58.2	41.9	66.7	54.4
PMBW($D1$, 1)	70.2	55.7	77.1	67.7
PMBW($D2$, 3)	80.6	66.5	84.5	77.2
PMBW($D3$, 3)	81.2	64.0	86.5	77.3

and present a contrasting collection of thermal images than the FLIR ADAS dataset. We can infer from the detections in Fig. 9 that the trained model is capable of generalizing over a broad set of thermal images with reasonable accuracy and no additional training cost.

LTIR Previous works [12,25] yielded hindered performance when these thermal object detectors were presented with occluded objects, both fully or partially. To get a concrete idea of the impact of occlusion, we consider the hiding subset of the LTIR dataset. For testing on this dataset, we have considered the PMBW($D2$, 3) framework. As shown in Table 9, the mAP when there is no occlusion present is much higher than when there is some form of occlusion present. Further, the confidence scores for these images are very high, as shown in Fig. 10a. As expected, our proposed detector suffers when there is close to full-occlusion as shown in Fig. 10c. However, when there is partial occlusion, the model can detect a person, albeit with low confidence, as shown in Fig. 10b.

4.6.4 Importance of thermal object detection at night-time

Although images taken in the visible domain can be rich in semantic information during the daytime, the same is not true about night conditions. From the road safety viewpoint, especially in the context of self-driving cars, errors in detections made on images captured in the RGB spectrum can have untoward consequences. However, we can overcome these unfavourable outcomes using object detection in thermal images. It can be made evident by comparing the detections made on RGB images by a fully trained RGB detector (row 1 of Fig. 11) and comparing it with the respective thermal counterpart, using our framework (row 2 of Fig. 11). It is clear that when compared to the thermal, RGB detector may fail in certain instances such as unfavourable lighting, missing objects that are smaller/more concealed, or even misconstruing a crowd and output the wrong number of detections. More accurate results can be obtained, simply by switching to thermal inference as shown in Row 2 of Fig. 11.

4.6.5 Retention of visible information

As mentioned earlier, we intended to perform transfer learning without replacing the information possessed by the RGB pre-trained weights to prevent the loss of a vast amount of readily available knowledge. Thus, the effect was an amelioration of the model's performance in this domain while still retaining an admissible detection capability in the visible sphere. To demonstrate this, we visually compare the results

Table 8 Comparison with baseline and prior state-of-the-art methods (all values rounded to 1 decimal place for consistency)

Method	Person	Bicycle	Car	mAP
Baseline	54.7	39.7	67.6	54.0
MMTOD-UNIT(MSCOCO) [12]	64.5	49.4	70.7	61.5
Transfer learning on SSD + VGG16 [15]	61.9	46.1	85.1	64.4
ODSC (SSD512 + VGG16) [34]	71.0	55.5	82.3	69.6
BU(LT, T) [25]	75.6	57.4	86.5	73.2
ThermalDet [6]	78.2	60.0	85.5	74.6
PMBW($D2$, 3) (ours)	80.6	**66.5**	84.5	77.2
PMBW($D3$, 3) (ours)	**81.2**	64.0	**86.5**	**77.3**

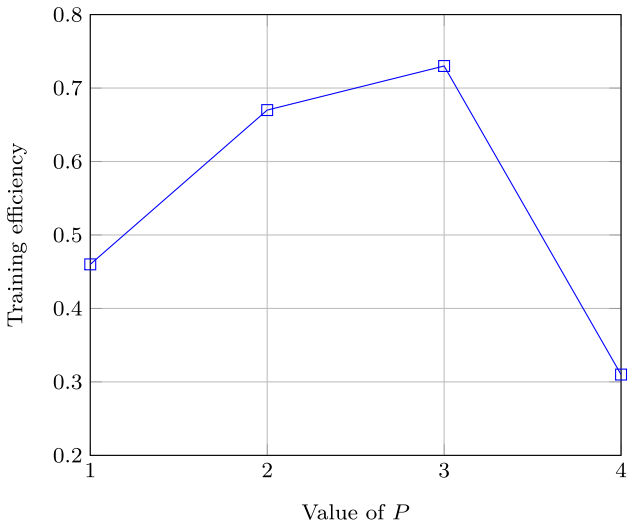

Fig. 6 Training efficiencies for different values of pace for PMBW($D1$, P)

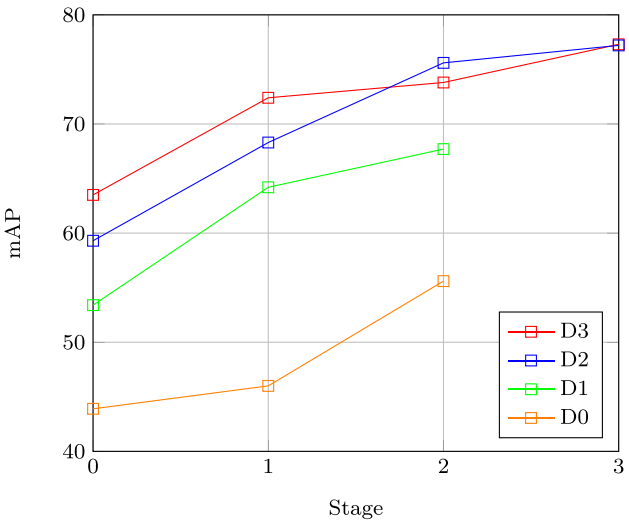

Fig. 7 mAP values for different coefficients across the stages

of the RGB trained detector, i.e. EfficientDet trained on the COCO dataset and our framework, using the value of ϕ as $D3$. As shown in Fig. 12, both RGB and thermal predictions do not differ by a vast amount when considering detections among person, bicycle, and car classes. Hence, there was minimal visible information loss during the training process.

4.6.6 Failure cases and explanations

The proposed framework, when applied to EfficientDet, performs with significantly high accuracy. However, there are still cases where the trained model fails. The most accurate setting (PMBW($D3$, 3)) still fails to detect small, i.e. distant objects. Although the performance concerning small objects increased with each value of ϕ, the mAP value of PMBW($D3$, 3) for small objects (64.6%) is still much lower than that of medium and large objects (86.7% and 82.1%, respectively). The qualitative results also show this result, as is evident in the first two columns of Fig. 13, where we can observe that the model is capable of detecting nearer cars, which appear larger in the image, while the cars farther away and smaller in the image are undetected.

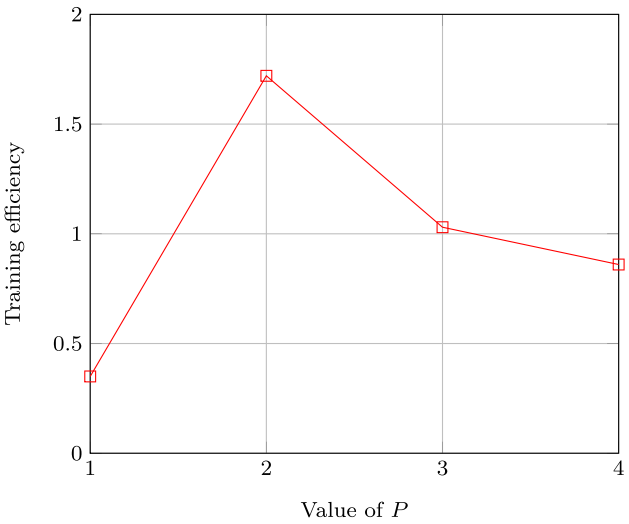

Fig. 8 Training efficiencies of fine-tuning for different values of pace for PMBW($D1$, P)

Fig. 9 Predictions on OTCBVS dataset

Table 9 Results of occlusion testing with PMBW($D2, 3$) on LTIR-Hiding

Subset of LTIR-Hiding	Number of images	mAP
No occlusion	213	89.2
Full or partial occlusion	145	55.4
Overall	358	72.0

Further, occlusion was also a hindrance to performance, as already observed in the LTIR dataset in Sect. 4.6.3. We have demonstrated this with the FLIR dataset in the last two columns of Fig. 13, where the presence of multiple persons or cars close to each other are labelled erroneously, primarily because one object occludes the others.

5 Conclusion

In this paper, we have explored a domain adaptation approach to re-purpose the state-of-the-art, EfficientDet object detector, to work in the thermal domain. We have created a paced multi-staged block-wise framework for efficient and accurate training of the EfficientDet model to detect objects in thermal imagery. By introducing block-wise adaptation and pace parameter, we have also shown that we can improve the training efficiency for larger and more complex detectors. The highlight of this paper was the creation of a framework that provides state-of-the-art performance for object detection in the thermal dataset, namely the FLIR ADAS dataset, with an mAP of 77.3%. In doing so, we obviated the necessity of RGB counterpart images during training to make the model more suitable for real-life applications.

The experimental results have shown us a highly flexible, paced multi-staged block-wise framework that achieves increased accuracy while striking a balance with available computational power. Further, the results demonstrate its versatility and capability to variations in the thermal domains, especially when it comes to occlusion. We have also shown that thermal domain features add to the pre-existing knowledge from the RGB spectrum, giving favourable results on visible images even after training on a thermal dataset.

The thermal object detector we have presented here is a step forward, but still, there is much to improve. We have observed that small or distant objects have the chance of not being detected, for example, PMBW($D3, 3$), which provided the best overall results, still had an mAP disparity of roughly 20% for small objects.

(a) No Occlusion **(b)** Partial Occlusion **(c)** Nearly Full Occlusion

Fig. 10 Results on LTIR-Hiding dataset with varying levels of occlusion

Fig. 11 Contrast in detections between RGB and thermal detectors on night-time images. Row 1 contains detections made by EfficientDet ($\phi = 3$) trained on the COCO dataset, while Row 2 contains the detections made on thermal images by our framework PMBW($D3$, 3)

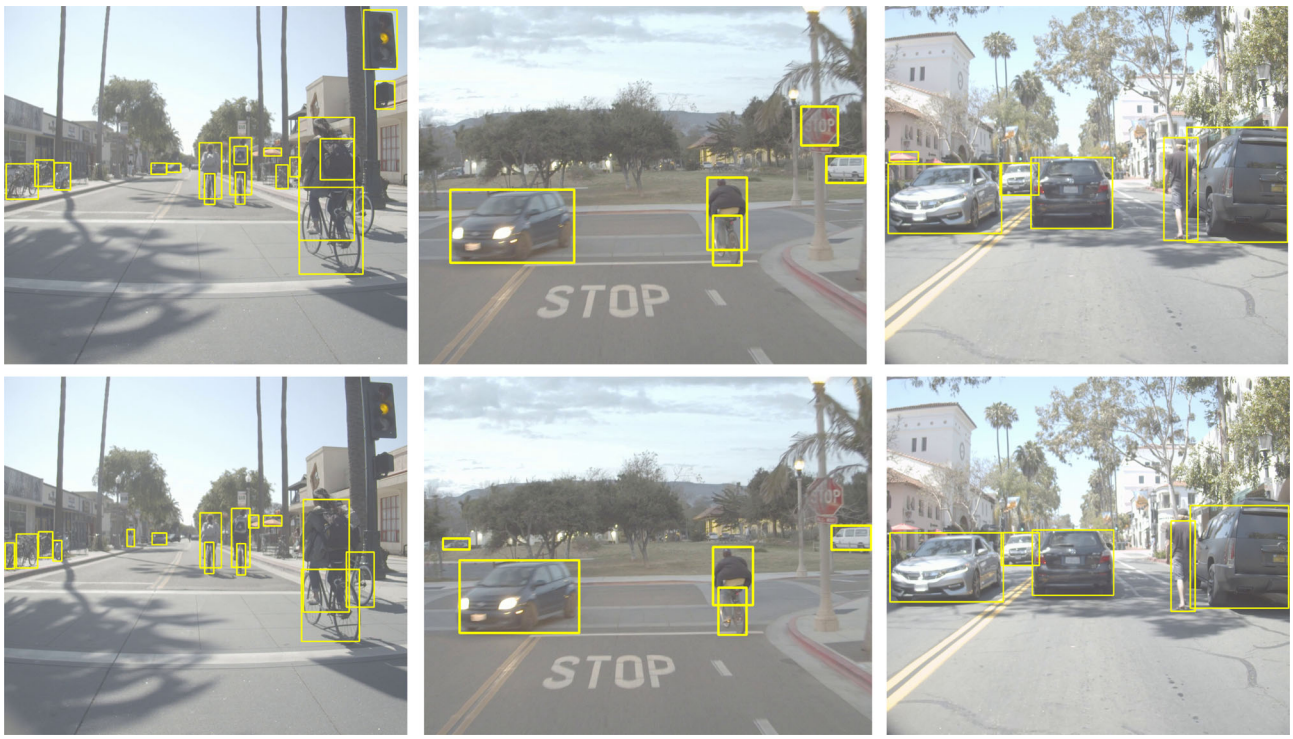

Fig. 12 Information retention tested on RGB images. Row 1 contains detections made by EfficientDet ($\phi = 3$) trained on the COCO dataset, while Row 2 contains the detections made on thermal images by our framework PMBW($D3$, 3)

Additionally the proposed approach is practically applicable and can potentially be implemented in self-driving cars and surveillance, as it can be inexpensively trained only on thermal images, thereby preserving privacy and still

Fig. 13 Failure cases of PMBW($D3$, 3). Row 1 contains the predictions made by PMBW($D3$, 3), while Row 2 contains the ground truths. The first two columns demonstrate failure to detect smaller objects, while the last two columns demonstrate failure to distinguish occluded objects

acquiring accurate results. Further research can tackle these problems through increased resolution or augmented data and can push the state-of-the-art further. Approaches involving thermal image-based pre-processing could yield better results for small objects. Additionally, implementing this approach on other state-of-the-art detectors could produce improvements. Our framework provides new insights into domain adaptation, especially for object detection in thermal images. However, we can utilize the maximum potential of this framework by making it a general guideline for improving efficiency while maintaining accuracy to enhance performance in various other computer vision and domain adaptation tasks.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Baek, J., Hong, S., Kim, J., Kim, E.: Efficient pedestrian detection at nighttime using a thermal camera. Sensors **17**(8), 1850 (2017)
2. Benenson R., Omran, M., Hosang J., Schiele, B.: Ten years of pedestrian detection, what have we learned? In: Agapito, L., Bronstein, M., Rother, C. (eds.) Computer Vision–ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol. 8926. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_47
3. Berg, A., Ahlberg, J., Felsberg, M.: A thermal object tracking benchmark. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2015). https://doi.org/10.1109/AVSS.2015.7301772
4. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection (2020). arXiv:2004.10934
5. Braun, M., Krebs, S., Flohr, F.B., Gavrila, D.M.: Eurocity persons: a novel benchmark for person detection in traffic scenes. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1844–1861 (2019). https://doi.org/10.1109/TPAMI.2019.2897684
6. Cao, Y., Zhou, T., Zhu, X., Su, Y.: Every feature counts: an improved one-stage detector in thermal imagery. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), pp. 1965–1969 (2019). https://doi.org/10.1109/ICCC47050.2019.9064036
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223. https://doi.org/10.1109/CVPR.2016.350
8. Dai, X., Yuan, X., Wei, X.: Tirnet: object detection in thermal infrared images for autonomous driving. Appl. Intell. **51**(3), 1244–1261 (2021). https://doi.org/10.1007/s10489-020-01882-2
9. Davis, J.W., Keck, M.A.: A two-stage template approach to person detection in thermal imagery. In: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05), vol. 1, pp. 364–369 (2005). https://doi.org/10.1109/ACVMOT.2005.14
10. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
11. Deng, Z., Hu, X., Zhu, L., Xu, X., Qin, J., Han, G., Heng, P.A.: R^3net: recurrent residual refinement network for saliency detection. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pp. 684–690. International Joint Conferences on Artificial Intelligence Organization (2018). https://doi.org/10.24963/ijcai.2018/95
12. Devaguptapu, C., Akolekar, N., Sharma, M.M., Balasubramanian, V.N.: Borrow from anywhere: pseudo multi-modal object detection in thermal imagery. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019). https://doi.org/10.1109/cvprw.2019.00135
13. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 304–311 (2009). https://doi.org/10.1109/CVPR.2009.5206631

14. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
15. Gaus, Y.F.A., Bhowmik, N., Isaac-Medina, B.K., Breckon, T.P.: Visible to infrared transfer learning as a paradigm for accessible real-time object detection and classification in infrared imagery. In: Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies IV, vol. 11542, p. 1154205. International Society for Optics and Photonics (2020)
16. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Rob. Res. **32**(11), 1231–1237 (2013). https://doi.org/10.1177/0278364913491297
17. Ghose, D., Desai, S.M., Bhattacharya, S., Chakraborty, D., Fiterau, M., Rahman, T.: Pedestrian detection in thermal images using saliency maps. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 988–997 (2019). https://doi.org/10.1109/CVPRW.2019.00130
18. globenewswire: The global thermal scanners market size is expected to reach $6.7 billion by 2025. https://www.globenewswire.com/news-release/2020/04/17/2017896/0/en/The-Global-Thermal-Scanners-Market-size-is-expected-to-reach-6-7-billion-by-2025-rising-at-a-market-growth-of-10-3-CAGR-during-the-forecast-period.html (2020)
19. govtech: Interest in thermal imaging is growing as covid-19 rages on. https://www.govtech.com/products/Interest-in-Thermal-Imaging-Is-Growing-as-COVID-19-Rages-On.html (2020)
20. Group, F.A.: Flir starter thermal dataset. https://www.flir.com/oem/adas/adas-dataset-form (2018)
21. Hazan, A., Shoshan, Y., Khapun, D., Aladjem, R., Ratner, V.: Adapternet-learning input transformation for domain adaptation. arXiv preprint arXiv:1805.11601 (2018)
22. Herrmann, C., Ruf, M., Beyerer, J.: CNN-based thermal infrared person detection by domain adaptation. In: Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything, vol. 10643, p. 1064308. International Society for Optics and Photonics (2018)
23. Hwang, S., Park, J., Kim, N., Choi, Y., Kweon, I.S.: Multispectral pedestrian detection: Benchmark dataset and baseline. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1037–1045 (2015). https://doi.org/10.1109/CVPR.2015.7298706
24. John, V., Mita, S., Liu, Z., Qi, B.: Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In: 2015 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 246–249 (2015). https://doi.org/10.1109/MVA.2015.7153177
25. Kieu, M., Bagdanov, A.D., Bertini, M.: Bottom-up and layer-wise domain adaptation for pedestrian detection in thermal images. ACM Trans. Multimed. Comput. Commun. Appl. (ACM TOMM) **17**, 1–19 (2020)
26. Kieu, M., Bagdanov, A.D., Bertini, M., Del Bimbo, A.: Domain adaptation for privacy-preserving pedestrian detection in thermal imagery. In: International Conference on Image Analysis and Processing, pp. 203–213. Springer (2019)
27. Li, C., Song, D., Tong, R., Tang, M.: Illumination-aware faster R-CNN for robust multispectral pedestrian detection. In: Pattern Recognition, vol. 85, pp. 161–171 (2019). https://doi.org/10.1016/j.patcog.2018.08.005
28. Li, C., Song, D., Tong, R., Tang, M.: Multispectral pedestrian detection via simultaneous detection and segmentation. In: British Machine Vision Conference (BMVC) (2018)
29. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014. Lecture Notes in Computer Science, vol. 8693. Springer, Cham. (2014). https://doi.org/10.1007/978-3-319-10602-1_48
30. Liu, N., Han, J., Yang, M.H.: Picanet: learning pixel-wise contextual attention for saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3089–3098 (2018)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: single shot multibox detector. In: Lecture Notes in Computer Science, pp. 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of 7th International Conference on Learning Representations (ICLR) (2019)
33. Miezianko, R.: Terravic research infrared database. http://vcipl-okstate.org/pbvs/bench/
34. Munir, F., Azam, S., Rafique, M.A., Sheri, A.M., Jeon, M.: Thermal object detection using domain adaptation through style consistency (2020). arXiv:2006.00821v1
35. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement (2018). arXiv:1804.02767
36. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems Conference, vol. 28 (2015)
37. Saeidi, M., Arabsorkhi, A.: A novel backbone architecture for pedestrian detection based on the human visual system. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02280-6
38. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520 (2019)
39. Schwarz, M., Schulz, H., Behnke, S.: RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1329–1335 (2015). https://doi.org/10.1109/ICRA.2015.7139363
40. Tan, M., Le, Q.V.: Efficientnet: rethinking model scaling for convolutional neural networks (2020). arXiv:1905.11946
41. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection (2020). arXiv:1911.09070
42. Wagner, J., Fischer, V., Herman, M., Behnke, S.: Multispectral pedestrian detection using deep fusion convolutional neural networks. In: ESANN, vol. 587, pp. 509–514 (2016)
43. Wu, Z., Fuller, N., Theriault, D., Betke, M.: A thermal infrared video benchmark for visual analysis . In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 201–208 (2014). https://doi.org/10.1109/CVPRW.2014.39
44. Xu, D., Ouyang, W., Ricci, E., Wang, X., Sebe, N.: Learning cross-modal deep representations for robust pedestrian detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4236–4244 (2017). https://doi.org/10.1109/CVPR.2017.451
45. Zhang, G., Liu, J., Li, H., Chen, Y.Q., Davis, L.S.: Joint human detection and head pose estimation via multistream networks for RGB-D videos. IEEE Signal Process. Lett. **24**(11), 1666–1670 (2017). https://doi.org/10.1109/LSP.2017.2731952
46. Zhang, G., Liu, J., Liu, Y., Zhao, J., Tian, L., Chen, Y.Q.: Physical blob detector and multi-channel color shape descriptor for human detection. J. Vis. Commun. Image Represent. **52**, 13–23 (2018). https://doi.org/10.1016/j.jvcir.2018.01.013
47. Zhang, H., Hong, X.G., Zhu, L.: Detecting small objects in thermal images using single-shot detector. Autom. Control Comput. Sci. **55**(2), 202–211 (2021). https://doi.org/10.3103/S0146411621020097
48. Zhang, L., Liu, Z., Chen, X., Yang, X.: The cross-modality disparity problem in multispectral pedestrian detection. arXiv preprint arXiv:1901.02645 (2019)

49. Zhang, X., Wang, X., Gu, C.: Online multi-object tracking with pedestrian re-identification and occlusion processing. Vis. Comput. **37**, 1089–1099 (2021). https://doi.org/10.1007/s00371-020-01854-0
50. Zheng, Y., Izzat, I.H., Ziaee, S.: GFD-SSD: gated fusion double SSD for multispectral pedestrian detection (2019). arXiv:1903.06999
51. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244

Shreyas Bhat Kera is currently pursuing his bachelor's degree in Computer Science with the Birla Institute of Science and Technology, Pilani, Rajasthan, India. His research interests include machine learning, deep learning, computer vision, processing of images in various domains, object detection in images, and activity recognition of humans in videos.

Anand Tadepalli is currently pursuing his bachelor's degree in Computer Science with the Birla Institute of Technology and Science, Pilani, Rajasthan, India. His research interests include machine learning, neural networks, deep learning and computer vision in the area of object detection.

J. Jennifer Ranjani received her B.E. degree in Electronics and Communication Engineering, from Noorul Islam College of Engineering, Nagercoil, India, M.Tech. Degree in Computer and Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India, and Ph.D. in Information and Communication Engineering from Anna University, Chennai, India, in 2002, 2005, and 2011, respectively. From July 2005 to August 2012, she was with the Department of Information Technology, Thiagarajar College of Engineering, Madurai. Later she joined Vivekanandha College of Engineering for Women, Tiruchengode, as an Associate professor. From June 2014 to November 2017, she was with the School of Computing, SASTRA University, Thanjavur, India. From December 2017 to August 2021, she was working in the Department of Computer Science and Information Systems at Birla Institute of Technology and Science, Pilani, Rajasthan, India. Her research interests include computer vision, statistical image processing, multiresolution signal analysis, data hiding, and embedded systems. She is a reviewer for journals like IEEE Transaction on Geoscience and Remote Sensing, IEEE Geoscience and Remote Sensing Letters, IET Image Processing, IET Radar, Sonar and Navigation, Multimedia Tools and Application, etc.