remote sensing

MDPI

Article

Training a Disaster Victim Detection Network for UAV Search and Rescue Using Harmonious Composite Images

Ning Zhang *, Francesco Nex, George Vosselman and Norman Kerle

Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, 7514 AE Enschede, The Netherlands; f.nex@utwente.nl (F.N.); george.vosselman@utwente.nl (G.V.); n.kerle@utwente.nl (N.K.)

* Correspondence: n.zhang@utwente.nl

Abstract: Human detection in images using deep learning has been a popular research topic in recent years and has achieved remarkable performance. Training a human detection network is useful for first responders to search for trapped victims in debris after a disaster. In this paper, we focus on the detection of such victims using deep learning, and we find that state-of-the-art detection models pre-trained on the well-known COCO dataset fail to detect victims. This is because all the people in the training set are shown in photos of daily life or sports activities, while people in the debris after a disaster usually only have parts of their bodies exposed. In addition, because of the dust, the colors of their clothes or body parts are similar to those of the surrounding debris. Compared with collecting images of common objects, images of disaster victims are extremely difficult to obtain for training. Therefore, we propose a framework to generate harmonious composite images for training. We first paste body parts onto a debris background to generate composite victim images and then use a deep harmonization network to make the composite images look more harmonious. We select YOLOv5l as the most suitable model, and experiments show that using composite images for training improves the AP (average precision) by 19.4% (15.3% → 34.7%). Furthermore, using the harmonious images is of great benefit to training a better victim detector, and the AP is further improved by 10.2% (34.7% → 44.9%). This research is part of the EU project INGENIOUS. Our composite images and code are publicly available on our website.

Keywords: victim detection; deep learning; unsupervised learning; generative adversarial network; image harmonization; emergency rescue; disaster management; UAV

Citation: Zhang, N.; Nex, F.; Vosselman, G.; Kerle, N. Training a Disaster Victim Detection Network for UAV Search and Rescue Using Harmonious Composite Images. *Remote Sens.* **2022**, *14*, 2977. https://doi.org/10.3390/rs14132977

Academic Editors: Fabio Giulio Tonolo and Daniela Carrion

Received: 17 May 2022
Accepted: 18 June 2022
Published: 22 June 2022

1. Introduction

The object detection task finds predefined categories of objects in an image and generates bounding boxes. It is a popular research topic and has practical value in many fields, such as anomaly detection in surveillance videos [1], wildfire detection [2], and plant diseases detection [3]. Benefiting from advanced deep learning (DL) technology and many available datasets for training, detection model performance has improved greatly compared to traditional detection methods [4].

In this paper, we focus on the detection of victims in debris, which can help save lives after disasters such as earthquakes and building collapses. Human detection based on deep learning has been used in many fields, such as security monitoring and intelligent transportation [5–8]. However, we find that existing human detectors fail to detect real victims because these models only use photographs of people's daily lives for training, which normally contain complete bodies. However, in real disaster scenes, a victim is usually partially buried by debris and only part of the body is visible. In addition, the colors of the victim's skin and clothes are often close to the colors of the surrounding dust and soil. These differences lead to the fact that the human detection models trained on normal datasets such as COCO [9] can not achieve satisfactory results in the detection of disaster

victims. Considering that it is extremely difficult to obtain real victim images we propose a novel framework to generate sufficient composite images to train the victim detector. We start by generating a composite image by pasting human body parts into images of debris background. Since the body parts and the background are from two different image sources, their color, texture, and illumination characteristics are inconsistent. Therefore, we design an unsupervised harmonization network to make the composite image look harmonious. Finally, we use these images to fine-tune a pre-trained detector. The algorithm runs in real time and can be deployed on unmanned aerial vehicles (UAVs) for autonomous searching in post-disaster scenarios. The contributions of this work can be summarized as follows.

- We propose to generate composite images for training a visual victim detector. Specifically, we focus on the detection of human body parts in debris, which is useful for UAV search and rescue in post-disaster scenarios.
- We propose a deep harmonization network to make composite images more realistic and further improve the detection accuracy.

We organize the remainder of the paper as follows. We first review relevant recent research in Section 2. The pipeline of the proposed method is presented in Section 3. The experimental results and discussion are elaborated in Section 4. We conclude this paper in Section 5.

2. Related Work

This section reviews some recent relevant work on object detection and image harmonization.

2.1. Object Detection Based on Deep Learning

Traditional methods usually detect objects in three steps. They first select some regions that may contain objects and then extract features from them. Finally, the features are fed into a classifier to yield the detection. HOG [10] and SIFT [11] features are commonly used, and the classifier is mostly SVM (support vector machine) or Adaboost [12]. These methods are computationally expensive, slow in operation, and low in accuracy. With the rise of deep learning DL-based detection methods have predominated the object detection field, and they can be divided into two categories: anchor-based and anchor-free methods.

Anchor-based methods can be further divided into two-stage and one-stage methods [4]. They use a set of predefined bounding boxes, predict categories through training and regress the positions of the bounding boxes. R-CNN, proposed by the authors of [13], was the first DL-based two-stage detector. The authors first used a selective-search method to extract about 2000 region proposals and then extracted features of these regions using a convolutional neural network (CNN). They finally used an SVM to generate the classification result. R-CNN achieved a great improvement and increased the mean average precision (mAP) to 58.5% on the VOC-2007 dataset [14], but its processing speed is slow. Later work such as Fast R-CNN [15] and Faster R-CNN [16] both focused on improving the way of selecting regions and reducing the reference time. Feature pyramid networks (FPN) proposed by the authors of [17] further improved Faster R-CNN by using a top-down architecture and building high-level semantic information at multiple scales. Two-stage methods can achieve higher accuracy than one-stage methods, but one-stage methods are faster because they skip the step of region proposal. YOLOv1 was the first DL-based one-stage detector [18]. It took the whole image as input and divided it into $S \times S$ grids. Each grid was responsible for predicting two bounding boxes and their corresponding class-specific confidence score. Every grid could only have one label, which made YOLOv1 not good at detecting small objects, although its speed was surprising (155 FPS on VOC-2007). With the integration of the latest deep learning technologies, subsequent YOLOv2 [19], YOLOv3 [20], YOLOv4 [21], and YOLOv5 [22] continuously improved the detection speed and accuracy, and have been applied in industry.

In recent years many anchor-free methods, which do not rely on pre-defined anchor boxes, have been to reduce the computational effort, and have achieved comparable

accuracy to that of anchor-based methods. FCOS is a one-stage fully convolutional detector that regresses bounding boxes at each location on the feature map [23]. It regards a location falling in a bounding box as a positive sample, resulting in a network with more positive samples for training. TTFNet was proposed to pursue a better balance between speed and accuracy [24]. It uses Gaussian kernels in both object localization and size regression, which allows the network to encode more training samples and accelerate the training process. PAFNet [25] extended TTFNet by using a better pre-trained model and combining several existing tricks, such as exponential moving average [26] and CutMix [27].

2.2. Object Detection in Emergency Scenarios

DL-based object detection approaches have been used in emergency scenarios, such as building damage detection. The authors of [28] presented a large dataset and trained a CNN to detect structural building damage after earthquakes. The authors of [29] trained and validated a building damage detector using aerial images of Hurricane Sandy and Hurricane Irma. Smoke or fire detection is another useful research topic in emergency scenarios because early warning of fire enables people to take prompt actions to reduce damages. Many datasets have been built to train fire or smoke detectors [30–33].

Image-based disaster victim detection is useful and can be integrated into advanced low-altitude UAVs for automatic victim search [34,35]. However, due to the lack of real victim datasets, existing victim detection systems [36–38] used common datasets such as INRIA person [39] and PASCAL VOC [14,40] for training. Moreover, these methods were tested on extremely small real datasets. The authors of [34] only used 19 images for testing. The authors of [36] tested their method using 50 images from the INRIA person dataset, which does not contain victim images. In this paper, we verify that detectors trained on the large popular dataset COCO [9] can not effectively detect real disaster victims because it contains regular human photos, which are quite different from the photos of victims in real disasters. Real victims of disasters are usually buried under highly cluttered rubble, with only part of their bodies exposed.

2.3. Using Unreal Data for Training

When real training data are hard to collect or annotate, researchers have verified that it is feasible to train the DL model with composite or rendered data. For example, rendered datasets are widely used in training semantic segmentation networks, because advanced computer graphics technologies allow to easily render a large number of RGB images and corresponding segmentation ground-truth [41–45]. Detection tasks also benefit from using composite or rendered data for training. The authors of [46] proposed a method to synthesize drones, planes, and cars in arbitrary poses, using these images to better train detectors. The authors of [47] used 3D CAD models to augment the training data of the few-shot learning detection task. A simple cut-and-paste method was proposed in [48] to generate large training data for indoor object instance detection. The authors of [49] generated synthetic wires for pre-training, and fine-tuned their wire detection network on real data. Similarly, the authors of [50] presented a rendered dataset for household objects detection and post estimation. To solve the shortage of training data, the authors of [51] inserted smoke effects into forest backgrounds to generate synthetic forest fire images, and trained a forest fire detector. The authors of [52] generated synthetic 3D faces with different poses, backgrounds, and occlusions, so as to train more robust face detectors. Inspired by [48], we present a composite victim-in-debris dataset for training a disaster victim detector. Because the foreground and background images used for composition are different in color and illumination characteristics, we further propose to harmonize composite images in a novel self-supervised framework.

2.4. Image Harmonization

Image harmonization improves composite images by reducing appearance differences between foreground and background images, such as color, illumination, and contrast.

Existing work mainly focuses on training a fully supervised deep harmonization network using annotated datasets. The authors of [53] proposed to incorporate semantic information during harmonization. Their network had an additional decoding branch to learn semantic segmentation and concatenated semantic features to the harmonization branch. The authors of [54] addressed painting harmonization, and proposed an algorithm to transfer the local statistics of paintings. The authors of [55] also focused on stylized image harmonization, and designed the Poisson blending loss. The authors of [56] collected iHarmony4 datasets and proposed the domain verification in their generative adversarial network (GAN) framework to harmonize images. The authors of [57] proposed a harmonization framework, which integrated the spatial-separated attention module. These supervised learning methods require input images and corresponding ground-truth for training. Recently, the authors of [58] designed a self-supervised framework for image harmonization. Their method extracted an image's content and appearance features from different image crops and then used these features to reconstruct the image. In the present paper, we propose an unsupervised learning framework that uses the GAN to harmonize our composite victim images. Our purpose is to use these harmonious images to train a better victim detector.

3. VictimDet: Training a Disaster Victim Detector Using Harmonious Composite Images

In this section, we introduce the proposed pipeline of victim detection in detail. As shown in Figure 1 our pipeline consists of three steps. First, we collect some background pictures of earthquakes and building collapses. We randomly cut out body parts from a character image and paste them on a background image to obtain a composite image. Then, the composite image is fed into our proposed network for deep harmonization. These harmonious images are finally used to fine-tune a victim detector. In the following sub-sections, we introduce the details of the three steps respectively.

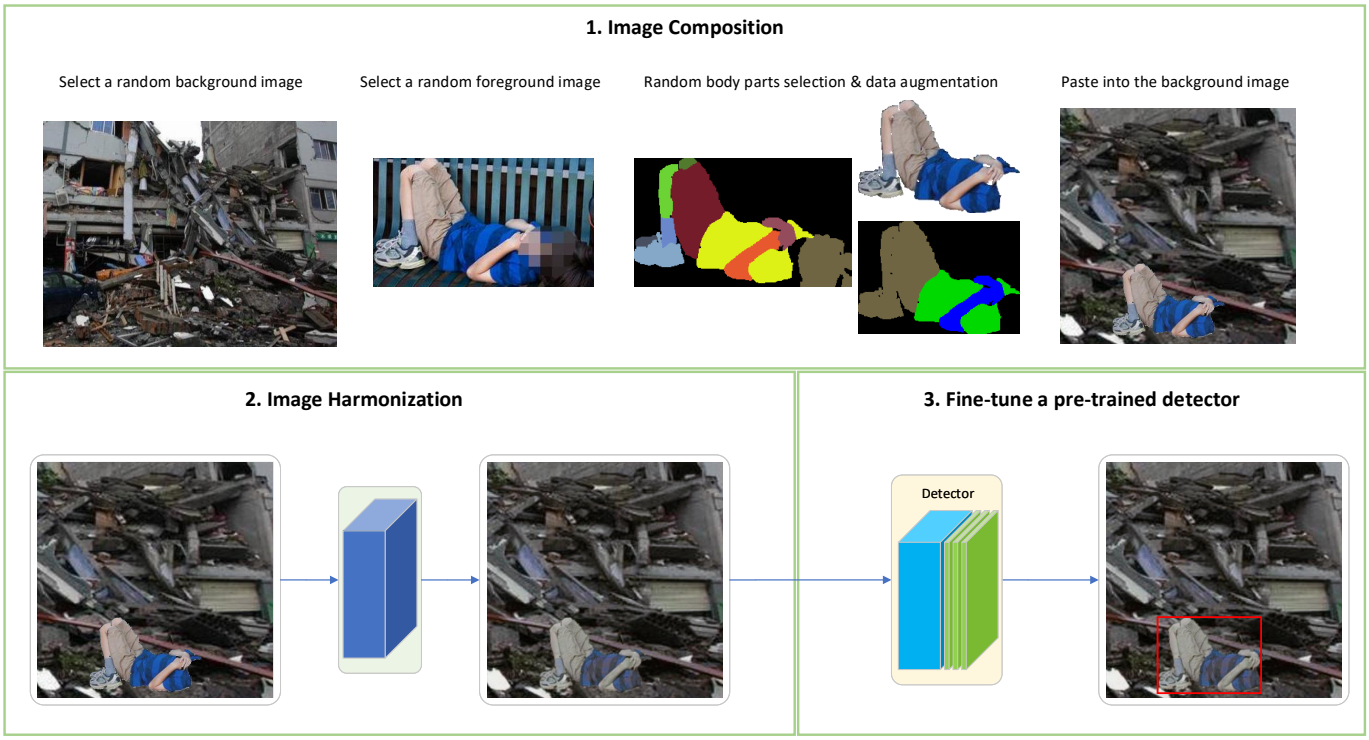


Figure 1. Our proposed framework consists of three steps: (1) image composition, (2) image harmonization, and (3) fine-tune a pre-trained detector.

3.1. Victim Image Composition

In this step, we start by collecting real background and foreground images for image composition. Because we focus on detecting victims that are partially buried in debris, we collect images of real earthquakes and collapsed buildings on the internet as background images. To obtain real foreground images of people we use the Look Into Person (LIP) dataset [59], which was collected for the human parsing task. In the LIP dataset, 50K character images were selected from the COCO [9] dataset, and each image was labeled using 19 pre-defined semantic classes such as left/right arm, left/right leg, and torso. Some sample images in the LIP dataset are shown in Figure 2. We manually check and delete many low-resolution, blurred, monochrome, and non-exposed body part images, as they are not suitable for composite victim images. Figure 2f–i are some good images that we can use to generate victim images.

Figure 2. Some images in the LIP data set are not suitable to be used as the foreground, such as (**a**) black-and-white image; (**b**) blurred image; (**c**) low resolution image; (**d**) severe occlusion image, and (**e**) an image with no body parts exposed. (**f–i**) are good image samples we keep to generate composite images. We blur faces for privacy reasons.

Although the LIP dataset was not designed for image composition, we can leverage it to composite various victim images. Different from previous papers that copy and paste a complete object instance [48,60], we choose to paste human body parts. By selecting specific body parts we are able to simulate victims partially buried by debris. For example, if we only select and paste the lower part of a person into a background image, we can imagine that the victim's upper body was buried in the rubble, and only their lower body was exposed.

We merge all classes into five body parts, which are upper limbs, upper limbs + torso, lower limbs, lower limbs + torso, and full body. Classes such as dress, skirt, pants, etc. are merged into these five body part classes. We do not use the class head because this paper focuses on the detection of body parts, not face or head. What is more, including heads may cause potential privacy issues. Given a background image I^b, a foreground image I^f, and the binary mask M^f of the body parts corresponding to the foreground image, we generate the composite image I^c using the following equation:

$$I^c = I^b \times (1 - M^f) + I^f \times M^f. \quad (1)$$

The positions where we paste the body parts are randomly generated. We also use image augmentation techniques, such as resizing, cropping, flipping, and adjusting contrast.

3.2. Unsupervised Image Harmonization

A composite image usually does not look harmonious because the background and the foreground have large differences in illumination, color, and texture characteristics. We want to reduce these appearance differences, so that the composite image looks as harmonious as possible. Therefore, we propose a novel deep victim harmonization network utilizing a self-attention mechanism. Figure 3 shows the structure of the proposed network, which is based on the adversarial training. The generator G aims at generating harmonious images, while the global discriminator D_{global} and the local discriminator D_{local} try to distinguish whether an image is composite or real at the global and local levels, respectively.

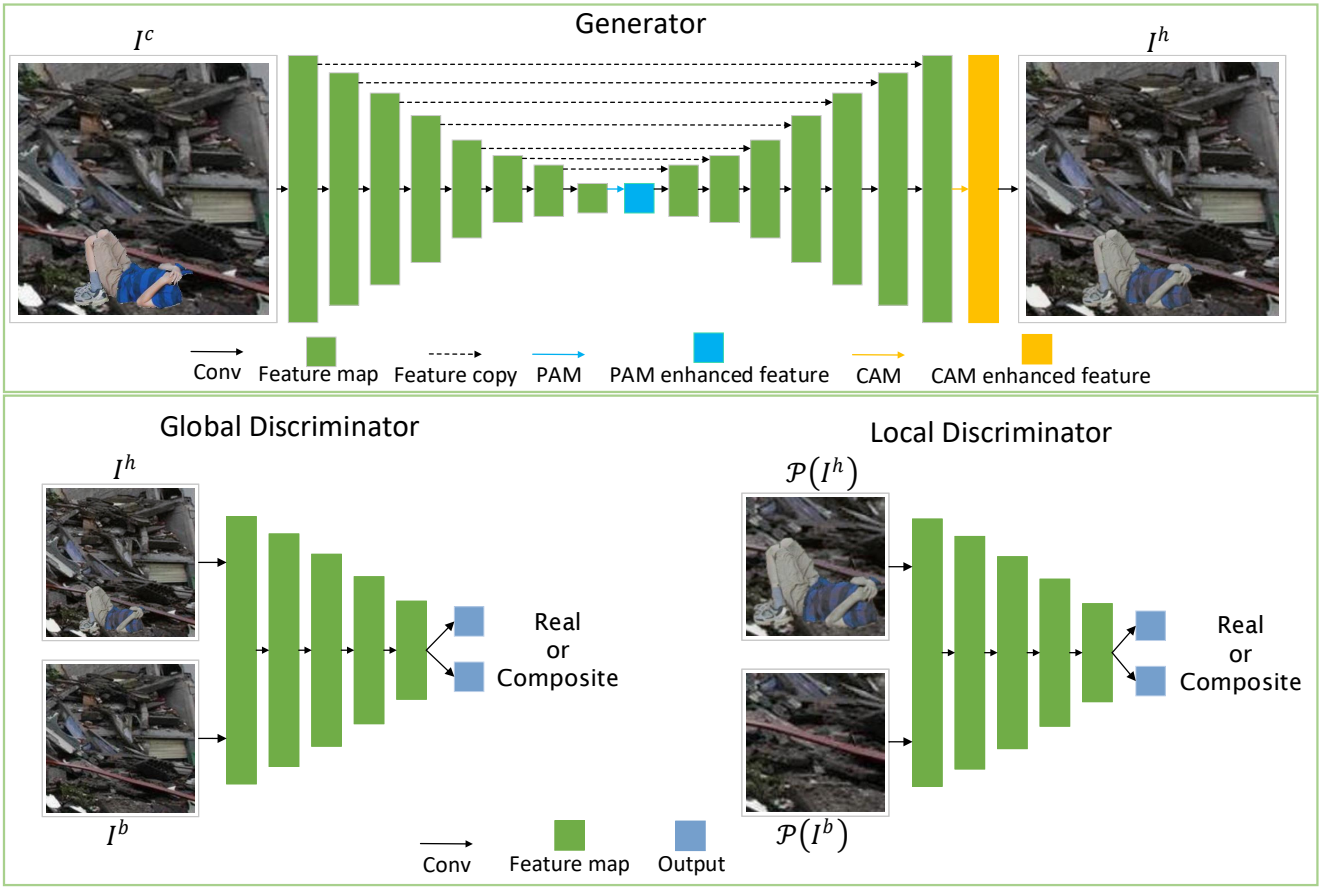


Figure 3. Our framework has a generator G and two five-layers discriminators D_{global} and D_{local}. The generator takes a composite image I^c as input, and generates a harmonious image I^h. Two discriminators discriminate the real images and the generated harmonious images globally and locally, respectively.

3.2.1. Self-Attention Enhanced Generator

The generator G is a U-Net with self-attention layers. It takes a composite image I^c as input and outputs the harmonious image I^h. We use two self-attention modules to enhance the generator. One is the pixel attention module (PAM) which tries to increase the receptive field of the encoder and enhance the deepest features extracted by the encoder. The generator uses a long-skip to concatenate features from the encoder to the decoder, so the channels of features are increased. We use the channel attention module (CAM) in the last layer of the decoder to explore the inter-dependencies between channels. The self-attention mechanism can be illustrated as mapping a query and key-value pairs to an

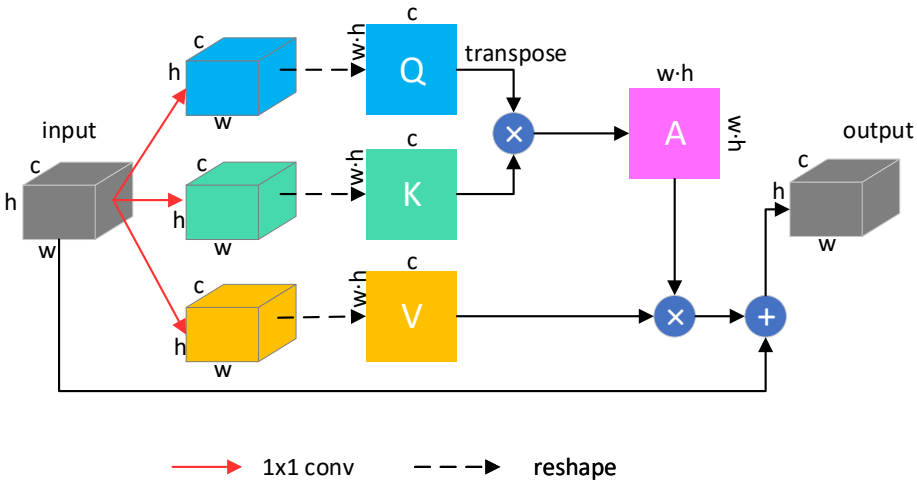
output [61]. The output is weighted by the value, which is computed with the query and the corresponding key. As shown in Figure 4a we take the output of the deepest layer of the encoder as input X and construct the same-shaped query Q, key K, and value V using three convolutional layers, respectively. We transpose the matrix K to get K^{T}, and multiply it by Q. Applying a softmax function to the result QK^{T} we obtain the pixel attention matrix $AP_{j,i}$ that can be expressed as:

$$AP_{j,i} = \frac{exp(p_{ij})}{\sum_{i=1}^{J}\sum_{j=1}^{J} exp(p_{ij})}, \tag{2}$$

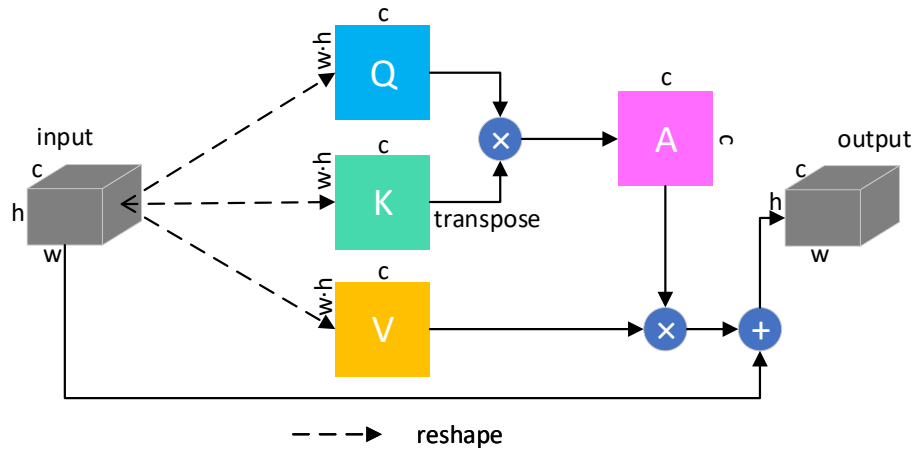
where $p_{ij} \in QK^{\mathrm{T}}$, and $AP_{j,i}$ denotes the influence of the ith feature on the jth feature. We multiply the attention matrix by the value matrix V and add X to get the enhanced output:

$$X_p = \alpha \sum_{i=1}^{J} AP_{j,i}V_i + X_j, \tag{3}$$

where α is a learnable scale parameter to weight features.



(**a**)



(**b**)

Figure 4. The details of two self-attention modules used in the generator. (**a**) Pixel attention module. (**b**) Channel attention module.

Figure 4b shows the channel attention module using the self-attention mechanism. It is similar to the pixel attention module we use, but we do not use convolution operations to construct query, key, and value. Instead, we directly reshape the features of the decoder layer, because we want to keep the channel information from the decoder. We can get a channel attention matrix $AC_{j,i}$:

$$AC_{j,i} = \frac{exp(p_{ij})}{\sum_{i=1}^{J}\sum_{j=1}^{J} exp(p_{ij})}. \tag{4}$$

The enhanced feature maps can be obtained by:

$$X_p = \beta \sum_{i=1}^{J} AC_{j,i} V_i + X_j, \tag{5}$$

where β is a learnable scale parameter.

3.2.2. Global and Local Discriminators

A global discriminator D_{global} is used to discriminate if a complete image is real or composite. A background image I_b used to generate the corresponding composite image is harmonious and can be used as the ground truth of training. Our discriminators are based on the PatchGAN [62], and we define the adversarial loss function of the global discriminator as:

$$\begin{aligned} L_{D_{global}} &= \mathbb{E}[\log D_{global}(I^b)] + \mathbb{E}[\log(1 - D_{global}(I^h))], \\ L_{G_{global}} &= \mathbb{E}[\log(1 - D_{global}(I^h))]. \end{aligned} \tag{6}$$

The foreground areas of some composite images are relatively small, which makes it impossible for the global discriminator to pay attention to local consistency effectively. Therefore, we propose to use another local discriminator D_{local} to mainly focus on the foreground areas. We extend the bounding box of the foreground image by 60 pixels and use this image patch $\mathcal{P}(I^h)$ as the input of the local discriminator. We define the loss function of the local discriminator as:

$$\begin{aligned} L_{D_{local}} &= \mathbb{E}[\log D_{local}(\mathcal{P}(I^b))] + \mathbb{E}[\log(1 - D_{local}(\mathcal{P}(I^h)))], \\ L_{G_{local}} &= \mathbb{E}[\log(1 - D_{local}(\mathcal{P}(I^h)))], \end{aligned} \tag{7}$$

where $\mathcal{P}(I^b)$ is the corresponding patch on the background image.

3.2.3. Loss Functions

The background part of the harmonious image should remain unchanged, while the foreground part should have similar color, illumination, and texture characteristics as the background image without changing the content. To keep the background we compute the masked smooth L_1 loss on each pixel i, whose value is in the range of $[0, 1]$:

$$L_{1,i} = \begin{cases} \frac{1}{2}(I_i^c - I_i^h)^2 \times (1 - M_i^f) & \left|I_i^c - I_i^h\right| < 1, \\ (\left|I_i^c - I_i^h\right| - \frac{1}{2}) \times (1 - M_i^f) & otherwise. \end{cases} \tag{8}$$

The total loss over all pixels is:

$$L_1 = \sum_{i}^{I} L_{1,i}. \tag{9}$$

To harmonize the foreground body parts while keeping the semantic information we take inspiration from [63] and propose the locally constrained perceptual (LCP) loss, which consists of a locally constrained content (LCC) loss L_{LCC} and a locally constrained style

(LCS) loss L_{LCS}. The former keeps the content information and the latter constrains the style (color, illumination, texture, etc.) between two images. Different from the common practice of computing the perceptual loss over a whole image, our proposed loss constrains features extracted from image patches. This is based on our intention to make the body parts harmonious with the background. We define the proposed locally constrained content loss as:

$$L_{LCC} = \frac{1}{C_j M_j N_j} \|\phi_j(\mathcal{P}(I^h)) - \phi_j(\mathcal{P}(I^c))\|_2^2, \tag{10}$$

where $\mathcal{P}(\cdot)$ denotes a cropped image patch as we use in the local discriminator. $C_j \times M_j \times N_j$ is the dimension of the feature map, and ϕ_j represents the feature map of the j-th convolutional layer of a pretrained VGG16 model. The loss uses l^2-norm to measure the distance between two features. Similarly, our locally constrained style loss is defined as:

$$L_{LCS} = \frac{1}{C_j M_j N_j} \|(\mathcal{G}(\phi_j(\mathcal{P}(I^h))) - \mathcal{G}(\phi_j(\mathcal{P}(I^b))))\|_2^2, \tag{11}$$

where $\mathcal{G}$ denotes the Gram matrix [30]. The shallow layers of the VGG model extract low-level features such as texture and color, while the deeper layers capture high-level semantics. We set $j = 8, 11$ in Equation (10), and set $j = 3, 5$ in Equation (11).

In addition, we use the total variation loss to suppress noises on the local patch [41]. It is defined as L_{TV}:

$$\begin{aligned} L_{TV} = &\sum_{m,n} \left| \mathcal{P}(I^h)_{m,n} - \mathcal{P}(I^h)_{m+1,n} \right| \\ &+ \left| \mathcal{P}(I^h)_{m,n} - \mathcal{P}(I^h)_{m,n+1} \right|, \end{aligned} \tag{12}$$

Combining the adversarial losses defined in Equations (6) and (7) the loss function for training the generator G is expresses as:

$$\begin{aligned} L_G = &\lambda_1 L_1 + \lambda_2 L_{LCC} + \lambda_3 L_{LCS} \\ &+ \lambda_4 L_{G_{global}} + \lambda_5 L_{G_{local}} + \lambda_6 L_{TV}, \end{aligned} \tag{13}$$

where $\lambda_1 \sim \lambda_6$ are weight coefficients.

4. Experiments

In this section, we evaluate the results of training a victim detector with composite images and testing on real images. We first present the dataset we use to train the victim detector, then we introduce details of the implementation of training the harmonization network and fine-tuning the detectors. We show both qualitative and quantitative results.

4.1. Dataset

Training set: We use 85 background images and 1500 foreground images to generate a total of 3000 composite images with the size of 512×512, using random background and foreground combinations. We apply separate data augmentation to the foreground and background, such as (1) flip horizontally, (2) change contrast, (3) brighten/darken, (4) rotate, (5) resize, and (6) crop, to increase the diversity of the composite images. Although manually controlling the size of a foreground image can yield a more reasonable foreground-background ratio, it is very time-consuming. Moreover, with different camera heights or focal lengths, the relative sizes of the foreground body parts and the background are different. Therefore, the scales of the body parts should not be fixed. By carefully controlling the scaling factors of different body parts we are able to easily generate composite images with acceptable foreground-background ratios. At the same time, we generate the bounding boxes of the body parts on each image and use them as ground truth when training a victim detector. We use the proposed harmonization network to output corresponding 3000 harmonious images for training the detector.

Validation set and test set: To test our victim detector we collect and annotate 250 real victim images. Most of these images are acquired from the internet using searching keywords such as earthquake victim, and building collapse victim. We also capture some images by ourselves. Many victims in the images are buried by collapsed buildings, and only part of their bodies or limbs are visible. Their clothes and limbs are dirty and covered with dirt or dust. And some victims lay prone or curled up. Due to copyright and privacy reasons, we are not able to show real victim images in the paper.

When fine-tuning a victim detector we need a validation set to evaluate the result of each training epoch and determine the hyper-parameters. In most deep learning tasks we construct the validation set from the training data, that is, we divide all training data into two disjoint subsets, which are the training set and the validation set, respectively [64]. However, this is not applicable in our case, because we cannot guarantee that the training data and the test data have similar feature distributions. If our validation set comes from training data one possible result is that the training achieves good accuracy on the validation set, but the performance on the test set is poor. Therefore, it is meaningless to use composite images as the validation set. In view of the above consideration, we construct the validation set using real victim images. Specifically, we randomly divide the test data into three disjoint subsets. In each training, we use one subset as the validation set and the other two subsets as the test set. Therefore, we train, evaluate, and test our model in a three-fold cross-validation manner. The final result of each model is obtained by calculating the average accuracy of three independent trainings.

4.2. Implementation Details

We run experiments on an Ubuntu 18.04 system with a Nvidia Titan XP graphics card. We implement the deep harmonization network using PyTorch. We do not have a metric to explicitly measure the “quality" of the composite images, so it is difficult to adjust the coefficients according to the generated images. Instead, we carefully observed the loss curves of training, and adjusted the coefficients according to their scale and convergence speed. We set coefficients in Equation (13) as: $\lambda_1 = 100$, $\lambda_2 = 2$, $\lambda_3 = 0.2$, $\lambda_4 = 1$, $\lambda_5 = 1$, $\lambda_6 = 10^{-5}$, to make each loss part have a close scale. We train the network for 10 epochs with $batchsize = 1$.

Since we use the victim detector in real-time scenarios, we hope that the network can have a good trade-off between efficiency and accuracy. Therefore, we select and evaluate four state-of-the-art detectors pre-trained on the COCO dataset, including FCOS, TTFNet, PAFNet, and YOLOv5. They are all one-stage detectors with fast inference speed, and only YOLOv5 is an anchor-based model. For the first three anchor-free networks we use the implementations provided by PaddleDetection [65], which is an open-source development kit for object detection. We use the official YOLOv5 [22] implementation and evaluate three different YOLOv5 models, namely YOLOV5s, YOLOV5m, and YOLOV5l, with increasing depth and width of structures.

4.3. Qualitative Analysis of Harmonized Images

We show some images from the training set in Figure 5. Some of the composite images in the first and third rows do not look real because their foreground and background have large differences in color, illumination, and texture. For instance, the appearance of the “victim” lying in the third image of the first row is bright, which is out of harmony with the earthy background, while the harmonization network generates a more realistic image. In addition, the proposed harmonization method is different from the simple way of smoothing foreground edges used by Dwibedi et al. [48]. Our deep harmonization network can not only produce smooth foreground edges but also transfer the background style to the foreground while maintaining its semantics.

Figure 5. Sample images from our dataset. The images in the first and third rows are composite images obtained using the method illustrated in Section 3.1. The second and fourth rows are the corresponding harmonized images generated by the harmonization network introduced in Section 3.2.

4.4. Quantitative Analysis of Victim Detection

In this experiment we evaluate the average precision (AP) of detecting real victims, using both existing detectors trained on the COCO dataset and fine-tuned using our harmonized composite images. We list results of different models in Table 1. We can see from the fourth and fifth columns of the table that the existing state-of-the-art models trained on the COCO dataset perform poorly in the task of detecting real disaster victims. The FCOS model is slightly improved (17.0%→18.3%) by the use of the deformable convolution (DCN) [66]. TTFNet with a Darknet-53 backbone performs worse than the FCOS model without using DCN. Although the YOLOv5 series models are substantially fast, the results on real images are extremely poor. PAFNet performs the best but the result (AP = 23.1%) is still poor. The poor results are in line with our expectations, because there are great differences in appearance and posture between the photos of people in the COCO dataset and the photos of real victims. These detectors cannot directly apply learned features from the COCO dataset to real disaster victim images.

Table 1. Comparison of state-of-the-art models on real victim images. The best results are highlighted in **bold**.

Model	Params (M)	Speed (FPS)	Trained on COCO		Harmonized Images	
			AP	AP_{50}	AP	AP_{50}
FCOS-R50-FPN	32.2	14	17.0	28.1	(+13.1) 30.1	(+20.5) 48.6
FCOS-DCN-R50-FPN	33.7	11	18.3	30.1	(+18.6) 36.9	(+26.6) 56.7
TTFNet-darknet53	45.8	23	16.5	26.0	(+9.2) 25.7	(+13.1) 39.1
PAFNet	33.8	21	23.1	36.8	(+22.5) **45.6**	(+28.2) 65.0
YOLOv5s	7.2	212	9.2	14.5	(+15.5) 24.7	(+28.4) 42.9
YOLOv5m	21.2	123	11.8	17.4	(+27.9) 39.7	(+42.6) 60.0
YOLOv5l	46.5	89	15.3	21.4	(+29.6) 44.9	(+44.0) **65.4**

From the last two columns of the table, we can find that fine-tuning the pre-trained detectors by fixing their backbones using our harmonized images improves the results by a large margin. The AP of TTFNet increases by 9.2%, and the AP of all other models increase by at least 13.1%. PAFNet increases by 22.5% and achieves the best result. It is particularly noteworthy that the results of YOLOv5 series models have increased substantially, especially YOLOv5m (11.8%→27.9%) and YOLOv5l (15.3%→44.9%). The improvement of the results verifies that our harmonized images are useful in training a victim detector. We also notice that YOLOv5 models outperform the other models in speed, and the largest model, YOLOv5l, has an outstanding balance between efficiency and effectiveness. Therefore, we select YOLOv5 models as our baseline and carry out additional evaluations based on them in the following sections. Figure 6 shows some detection results using YOLOv5l.

4.5. Ablation Study

In this section, we gradually evaluate how different parts of the proposed deep harmonization network affect the detection performance. We copy the results of the baseline models (YOLOv5) to the first row (Exp. A) of Table 2 for clear comparison.

Exp. B First of all, we fine-tune the models using composite images that are not harmonized by our proposed deep harmonization network. These composite images have disharmonious foreground and background, but they still show effectiveness in fine-tuning victim detectors. The improved APs compared with the baseline models are displayed in blue color. The great improvement demonstrates that the victim detection task benefits from using our composite victim images even if the background and the foreground are not harmonious in styles. The effectiveness of composite images is attributed to the fact that we use body parts instead of whole human instances to make the composite images. The semantic information of our composite images is consistent with that in photos of real victims whose body parts are buried and partially visible in real disasters.

Exp. C We start training the proposed deep harmonization network that only has the global discriminator, and generating harmonious images for fine-tuning. The adversarial training makes the style (color, illumination, texture, etc.) of the body parts similar to that of the background image. The introduction of the harmonization network increases the AP of YOLOv5l by 4.5%. We show the improved APs compared with Exp. B in magenta color. As shown in Figure 7, we observe some green artifacts in the foreground when only using the global discriminator, and these artifacts can be removed if we use the local discriminator together (Exp. E).

Figure 6. Visualization of victim detection. Five samples (**a**–**e**) are shown with red rectangles denoting detected victims. The first row is the ground-truth, and the second row is the default COCO pre-trained YOLOv5l model. The third row and the fourth row are the models fine-tuned on our composite images and harmonious images, respectively. Due to copyright and privacy issues we only show our own images.

Table 2. Ablation study of our deep harmonization network modules. The best results are highlighted in **bold**.

Exp.	Method	YOLOv5s		YOLOv5m		YOLOv5l	
		AP	**AP$_{50}$**	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**
A	w/o Fine-Tuning	9.2	14.5	11.8	17.4	15.3	21.4
B	Composite Images	(+12.4) 21.6	(+23.8) 38.3	(+19.6) 31.4	(+33.2) 50.6	(+19.4) 34.7	(+32.1) 53.5
C	B + D_{global}	(+0.5) 22.1	(+1.8) 40.1	(+2.8) 34.2	(+2.6) 53.2	(+3.1) 37.8	(+5.4) 58.9
D	C + Attention	(+1.3) 22.9	(+1.6) 39.9	(+3.8) 35.2	(+5.8) 56.4	(+3.5) 38.2	(+7.0) 60.5
E	C + D_{local}	(+0.9) 22.5	(+1.9) 40.2	(+6.2) 37.6	(+6.7) 57.3	(+6.1) 40.8	(+9.6) 63.1
F	D + D_{local}	(+3.1) **24.7**	(+4.6) **42.9**	(+8.3) **39.7**	(+9.4) **60.0**	(+10.2) **44.9**	(+11.9) **65.4**
G	B + Blending [48]	(−0.5) 21.1	(+0.1) 38.4	(+1.4) 32.8	(+2.3) 52.9	(+1.2) 35.9	(+3.1) 56.6

Exp. D We add the attention modules, which are the pixel attention module and the channel attention module. We expect the pixel attention layer to enhance the features of the last encoder layer, and the channel attention module to strengthen the inter-dependencies between feature channels. Compared with Exp. C, the AP of YOLOv5l increased slightly (0.4%). However, we find some checkerboard artifacts as shown in Figure 8. Both the green artifacts mentioned earlier and the checkerboard artifacts here appear in the foreground body parts. The reason is that the global discriminator focuses on discriminating the global

features of the whole image, but the features of local areas are also important for the image composition task of this paper, so it is not enough to use only one global discriminator.

Exp. E We add the local discriminator to the network. It focuses on distinguishing the regions around the foreground body parts, so the green artifacts in Figure 7 can be removed. The AP of YOLOv5l increases by 3.0% compared with Exp. C.

Exp. F This experiment is based upon the full harmonization network. We get the AP of 44.9% on YOLOv5l, which is 10.2% higher than using disharmonious composite images in Exp. B. The AP on YOLOv5s and YOLOv5m also increased by 3.1% and 8.3%, respectively. Moreover, if we compare the results with Exp. E we can find that the attention modules improve the results more when the network has a local discriminator. At the same time, the full model can eliminate the checkerboard artifacts shown in Figure 8.

Exp. G We also test the Gaussian blending method used by [48]. We concur with the authors of [60] that the smooth edges of foregrounds have a negligible impact on AP. This blending method cannot change the color, texture, or illumination of the foreground, which are important in generating useful victim images in this paper.

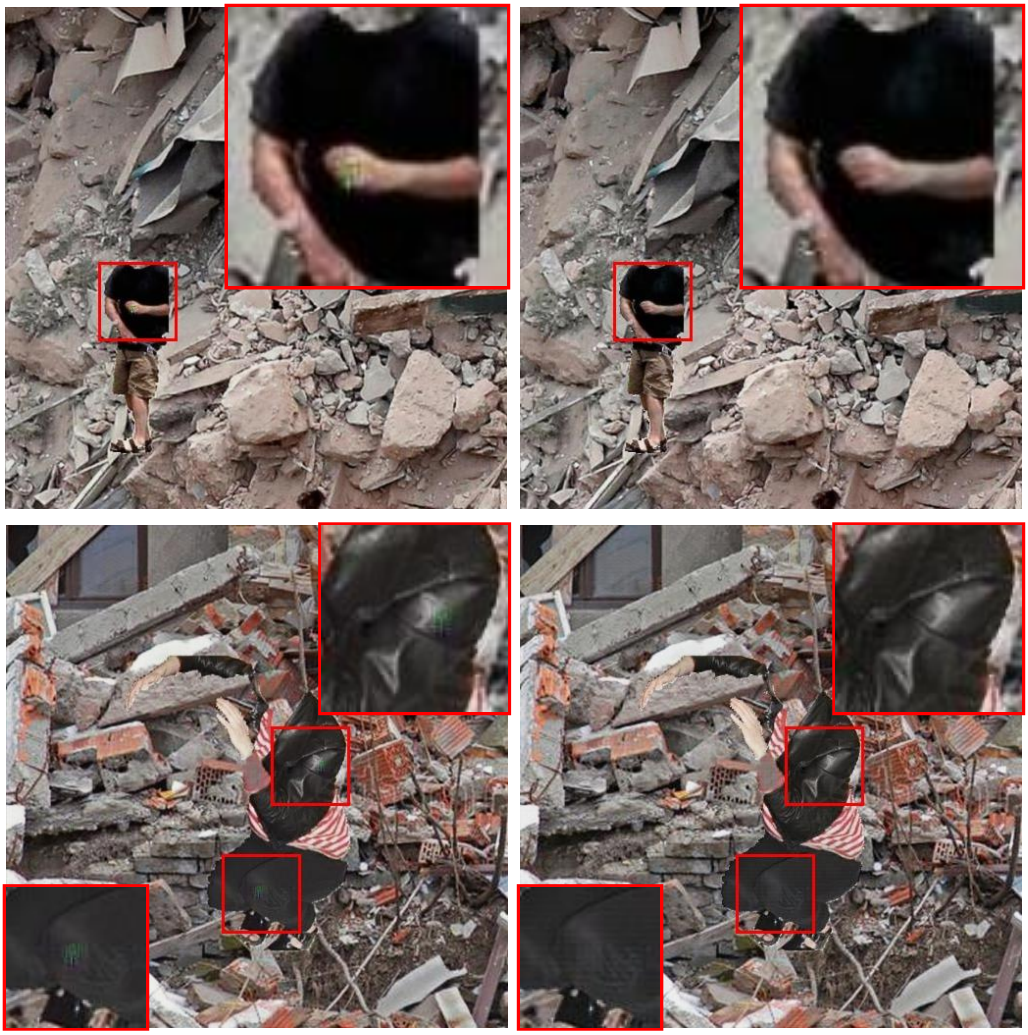


Figure 7. Visual comparison of only using the global discriminator and adding the local discriminator. The first column shows images harmonized by the network that only has the global discriminator, and we can see some green artifacts in the foreground. The second column shows the corresponding images generated by the network with both the global discriminator and the local discriminator. These artifacts are eliminated by introducing the local discriminators. We zoom in on some image areas for better comparison.

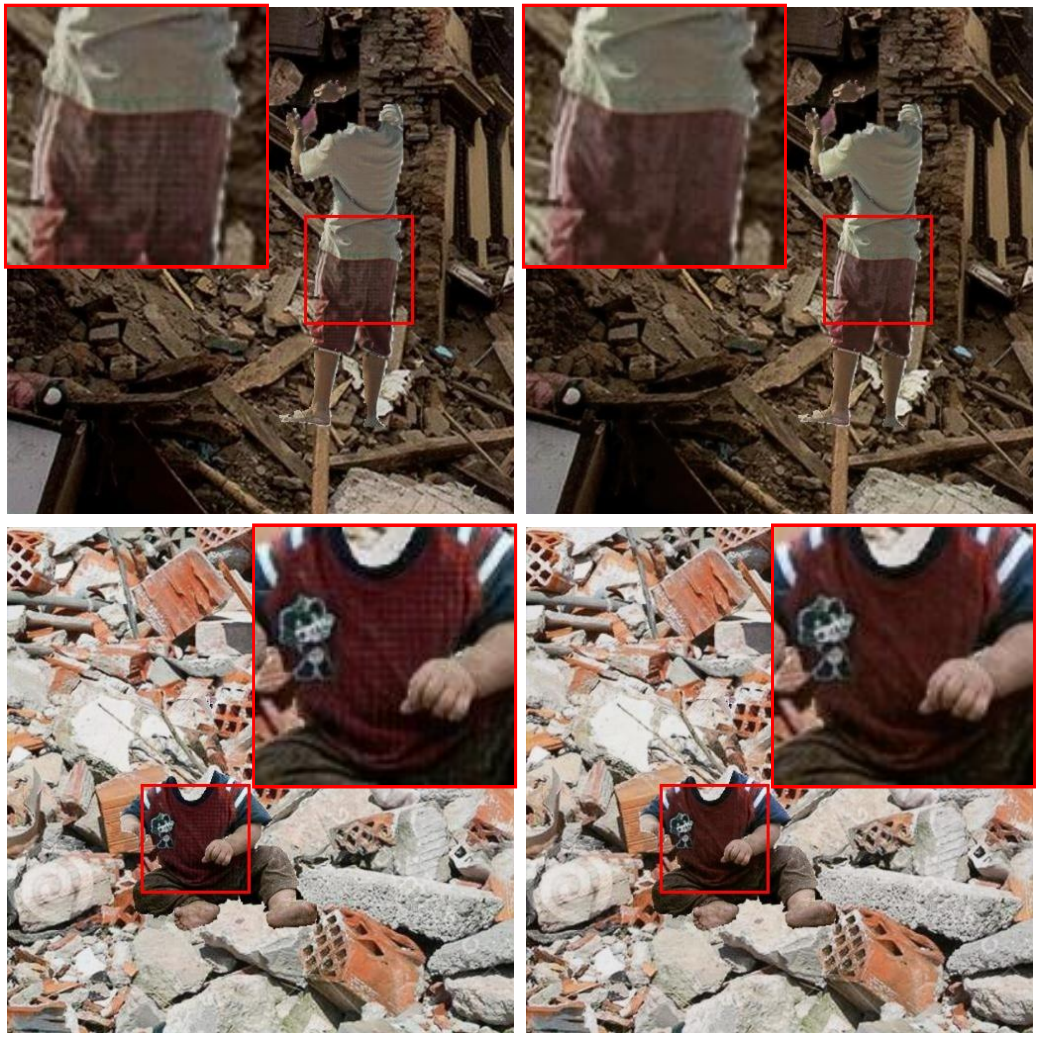


Figure 8. Visual comparison of using the global discriminator + attention modules and the full model. The first column shows images harmonized by the network without the local discriminator, on which some checkerboard artifacts are observed. The second column shows the corresponding images generated by the full harmonization network. We zoom in on some image areas for better comparison.

4.6. Study on Freezing Layers in Fine-Tuning

Our fine-tuning of the pre-trained models utilizes the ability of transfer learning so that the models can quickly apply what they have learned from the large-scale COCO dataset to relevant new datasets or new tasks. In this experiment, we evaluate the influence of fine-tuning YOLOv5 models by freezing different layers on the results. The structure of a YOLOv5 model consists of three parts. It has the CSP bottleneck [67] to extract features and uses PANet [68] as the neck to aggregate features from the backbone. The final prediction results are outputted by a head. Table 3 shows three different fine-tuning settings and their results.

Freeze the backbone We first only freeze the backbone, so the weights of the neck and the head can update during training. We also use this setting in other experiments in this paper, and it achieves the highest APs. The backbones pre-trained on the COCO dataset have a good ability for feature extraction, so freezing the backbones allows the networks to extract effective features for learning the victim detection task. We compare other fine-tuning strategies with this one, and the differences of APs are shown in teal color.

Freeze the backbone and the neck We fine-tune the prediction head by freezing both the backbone and the neck, which means only the last layers responsible for generating the final detection are able to update weights. This fine-tuning strategy is usually effective

when the pre-training data and the new data have great similarities, which is not applicable to our case. We get the AP of 22.5% on YOLOv5l, which is 22.4% lower compared with only freezing the backbone. The results on YOLOv5s and YOLOv5m also decrease substantially.

No frozen layer We also evaluate when there is no frozen layer in the pre-trained models. That is, all the layers of the models can update their pre-trained weights during training. The AP of YOLOv5l is 29.2%, which is better than that of freezing the backbone and the neck together, but worse than that of only freezing the backbone.

Table 3. Study on fine-tuning pre-trained models.

Frozen Layers	YOLOv5s		YOLOv5m		YOLOv5l	
	AP	**AP_{50}**	**AP**	**AP_{50}**	**AP**	**AP_{50}**
Backbone	24.7	42.9	39.7	60.0	44.9	65.4
Backbone + Neck	(−12.3) 11.8	(−20.2) 22.7	(−20.8) 18.9	(−25.9) 34.1	(−22.4) 22.5	(−24.6) 40.8
No Frozen Layer	(−6.4) 18.3	(−3.5) 39.4	(−15.5) 24.2	(−14.3) 45.7	(−15.7) 29.2	(−9.9) 55.5

4.7. Discussion on Failure Cases

We observe some failed detection on the victim images because not enough body parts were exposed, or the photos were low-light. For example, in one image, the victim was fully stuck in the debris, without any body parts exposed, and only the side of his trousers could be seen. In this case, the trained detector fails to detect the victim. In another image, the victim lies in the triangle structure formed by the collapse, with low brightness, only showing his back and half of his right upper arm. The victim's skin is covered with dust, showing a white-gray color. Another point worth discussing is that false detection has been found in very few pictures. We can reduce false detection by increasing the confidence when inferring the model, but at the same time, it may also make the detector miss some victims with low confidence. Considering this real victim detection task we believe that false detection is more acceptable than missed detection.

In a real emergency scenario, a successful detection might not achieve high IoU, but we regard it as a successful detection as long as it can detect (part of) a victim. This also brings an open question: is there a more reasonable evaluation metric than the average precision in this specific victim detection task?

5. Conclusions

In this paper, we have explored the use of composite images to fine-tune an effective victim detector. Our motivation comes from the fact that the existing state-of-the-art detectors trained on the COCO dataset cannot successfully detect disaster victims, and the real victim images for training are hard to obtain. Therefore, we propose to generate composite victim images by copying and pasting human body parts onto a debris background. Our method especially considers that the real victims are often buried in the debris, and only part of their bodies are visible. Therefore, unlike previous methods that copied and pasted a whole object instance, we choose to randomly paste the body parts. We have tested some state-of-the-art detectors and the experimental results demonstrate that fine-tuning the detectors using our composite images can largely improve the AP. Additionally, we verify the effectiveness of our unsupervised deep harmonization network, which can produce harmonious composite images for training, and helps to enhance the detectors further. Our image composition and harmonization methods can also be used for other tasks that lack training images, such as aircraft detection using remote sensing images. As part of the INGENIOUS project (https://ingenious-first-responders.eu, accessed on 17 May 2022) we have integrated the algorithm into a platform, which uses a customized autonomous unmanned aerial vehicle (UAV) for victim detection in post-disaster scenarios. The UAV captures images and sends them to a ground control station (GCS) for victim detection. Although we have not tested them, the YOLOv5 series algorithms themselves can run on embedded computing boards, such as Nvidia Jetson, at a satisfactory speed. In follow-

up research, we will focus on victim detection in low-light environments because many post-disaster rescues occur at night or in low-light environments.

Author Contributions: Conceptualization, N.Z. and F.N.; methodology, N.Z.; validation, N.Z.; formal analysis, N.Z.; investigation, N.Z.; writing—original draft preparation, N.Z.; writing—review and editing, N.Z., F.N., G.V. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No. 833435. The content reflects only the authors' view and the Research Executive Agency (REA) and the European Commission are not responsible for any use that may be made of the information it contains.

Data Availability Statement: Publicly available datasets were analyzed in this study. This dataset can be found here: https://github.com/noahzn/VictimDet, accessed on 17 May 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sreenu, G.; Durai, M.S. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *J. Big Data* **2019**, *6*, 1–27. [CrossRef]
2. Govil, K.; Welch, M.L.; Ball, J.T.; Pennypacker, C.R. Preliminary results from a wildfire detection system using deep learning on remote camera images. *Remote Sens.* **2020**, *12*, 166. [CrossRef]
3. Loey, M.; ElSawy, A.; Afify, M. Deep learning in plant diseases detection for agricultural crops: A survey. *Int. J. Serv. Sci.* **2020**, *11*, 41–58. [CrossRef]
4. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
5. Wei, H.; Laszewski, M.; Kehtarnavaz, N. Deep learning-based person detection and classification for far field video surveillance. In Proceedings of the 2018 IEEE 13th Dallas Circuits and Systems Conference (DCAS), Dallas, TX, USA, 12 November 2018; pp. 1–4.
6. Wei, H.; Kehtarnavaz, N. Semi-supervised faster RCNN-based person detection and load classification for far field video surveillance. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 44. [CrossRef]
7. Braun, M.; Krebs, S.; Flohr, F.; Gavrila, D.M. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1844–1861. [CrossRef]
8. Zhang, S.; Xie, Y.; Wan, J.; Xia, H.; Li, S.Z.; Guo, G. Widerperson: A diverse dataset for dense pedestrian detection in the wild. *IEEE Trans. Multimed.* **2019**, *22*, 380–393. [CrossRef]
9. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
10. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
11. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
12. Zhang, X.; Yang, Y.H.; Han, Z.; Wang, H.; Gao, C. Object class detection: A survey. *ACM Comput. Surv.* **2013**, *46*, 1–53. [CrossRef]
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 2007. Available online: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (accessed on 17 May 2022)
15. Girshick, R. Fast r-cnn. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
17. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
19. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
20. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.

21. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
22. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Liu, C.; Fang, J.; Abhiram, V.; Skalski, S.P. Ultralytics/yolov5: v6.0—YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. *Zenodo Tech. Rep.* **2021**. [CrossRef]
23. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9627–9636.
24. Liu, Z.; Zheng, T.; Xu, G.; Yang, Z.; Liu, H.; Cai, D. Training-time-friendly network for real-time object detection. *AAAI Conf. Artif. Intell.* **2020**, *34*, 11685–11692. [CrossRef]
25. Xin, Y.; Wang, G.; Mao, M.; Feng, Y.; Dang, Q.; Ma, Y.; Ding, E.; Han, S. Pafnet: An efficient anchor-free object detector guidance. *arXiv* **2021**, arXiv:2104.13534.
26. Lawrance, A.; Lewis, P. An exponential moving-average sequence and point process (EMA1). *J. Appl. Probab.* **1977**, *14*, 98–113. [CrossRef]
27. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6023–6032.
28. Nex, F.; Duarte, D.; Tonolo, F.G.; Kerle, N. Structural building damage detection with deep learning: Assessment of a state-of-the-art cnn in operational conditions. *Remote Sens.* **2019**, *11*, 2765. [CrossRef]
29. Li, Y.; Hu, W.; Dong, H.; Zhang, X. Building damage detection from post-event aerial imagery using single shot multibox detector. *Appl. Sci.* **2019**, *9*, 1128. [CrossRef]
30. Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep convolutional neural networks for forest fire detection. In *2016 International Forum on Management, Education and Information Technology Application*; Atlantis Press: Amsterdam, The Netherlands, 2016.
31. Sharma, J.; Granmo, O.C.; Goodwin, M.; Fidje, J.T. Deep convolutional neural networks for fire detection in images. In *International Conference on Engineering Applications of Neural Networks;* Springer: Cham, Switzerland, 2017; pp. 183–193.
32. Jadon, A.; Omama, M.; Varshney, A.; Ansari, M.S.; Sharma, R. FireNet: A specialized lightweight fire & smoke detection model for real-time IoT applications. *arXiv* **2019**, arXiv:1905.11922.
33. Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M.A. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Saf. J.* **2017**, *92*, 188–194. [CrossRef]
34. Sulistijono, I.A.; Risnumawan, A. From concrete to abstract: Multilayer neural networks for disaster victims detection. In Proceedings of the 2016 International Electronics Symposium, Denpasar, Indonesia, 29–30 September 2016; pp. 93–98.
35. Andriluka, M.; Schnitzspan, P.; Meyer, J.; Kohlbrecher, S.; Petersen, K.; Von Stryk, O.; Roth, S.; Schiele, B. Vision based victim detection from unmanned aerial vehicles. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1740–1747.
36. Hartawan, D.R.; Purboyo, T.W.; Setianingsih, C. Disaster victims detection system using convolutional neural network (CNN) method. In Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, Bali, Indonesia, 1–3 July 2019; pp. 105–111.
37. Hoshino, W.; Seo, J.; Yamazaki, Y. A study for detecting disaster victims using multi-copter drone with a thermographic camera and image object recognition by SSD. In Proceedings of the 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Delft, The Netherlands, 12–16 July 2021; pp. 162–167.
38. Sulistijono, I.A.; Imansyah, T.; Muhajir, M.; Sutoyo, E.; Anwar, M.K.; Satriyanto, E.; Basuki, A.; Risnumawan, A. Implementation of Victims Detection Framework on Post Disaster Scenario. In Proceedings of the 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), Bali, Indonesia, 29–30 October 2018; pp. 253–259.
39. Dalal, N.; Triggs, B. INRIA Person Dataset. 2005. Available online: http://pascal.inrialpes.fr/data/human (accessed on 17 May 2022).
40. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 2012. Available online: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (accessed on 17 May 2022).
41. Handa, A.; Patraucean, V.; Badrinarayanan, V.; Stent, S.; Cipolla, R. Understanding real world indoor scenes with synthetic data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4077–4085.
42. McCormac, J.; Handa, A.; Leutenegger, S.; Davison, A.J. Scenenet rgb-d: Can 5 m synthetic images beat generic imagenet pre-training on indoor segmentation? In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2678–2687.
43. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
44. Zhang, N.; Nex, F.; Kerle, N.; Vosselman, G. Towards Learning Low-Light Indoor Semantic Segmentation with Illumination-Invariant Features. *Int. Arch. Photogramm. Remote Sens.* **2021**, *43*, 427–432. [CrossRef]
45. Zhang, N.; Nex, F.; Kerle, N.; Vosselman, G. LISU: Low-light indoor scene understanding with joint learning of reflectance restoration. *ISPRS J. Photogramm. Remote Sens.* **2022**, *183*, 470–481. [CrossRef]

46. Rozantsev, A.; Lepetit, V.; Fua, P. On rendering synthetic images for training an object detector. *Comput. Vis. Image Underst.* **2015**, *137*, 24–37. [CrossRef]
47. Peng, X.; Sun, B.; Ali, K.; Saenko, K. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1278–1286.
48. Dwibedi, D.; Misra, I.; Hebert, M. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1301–1310.
49. Madaan, R.; Maturana, D.; Scherer, S. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3487–3494.
50. Tremblay, J.; To, T.; Birchfield, S. Falling things: A synthetic dataset for 3d object detection and pose estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 2038–2041.
51. Zhang, Q.X.; Lin, G.H.; Zhang, Y.M.; Xu, G.; Wang, J.J. Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Eng.* **2018**, *211*, 441–446. [CrossRef]
52. Han, J.; Karaoglu, S.; Le, H.A.; Gevers, T. Object features and face detection performance: Analyses with 3D-rendered synthetic data. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 9959–9966.
53. Tsai, Y.H.; Shen, X.; Lin, Z.; Sunkavalli, K.; Lu, X.; Yang, M.H. Deep image harmonization. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3789–3797.
54. Luan, F.; Paris, S.; Shechtman, E.; Bala, K. Deep painterly harmonization. In *Computer Graphics Forum*; Wiley: Hoboken, NJ, USA, 2018; Volume 37, pp. 95–106.
55. Zhang, L.; Wen, T.; Shi, J. Deep image blending. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 231–240.
56. Cong, W.; Zhang, J.; Niu, L.; Liu, L.; Ling, Z.; Li, W.; Zhang, L. Dovenet: Deep image harmonization via domain verification. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8394–8403.
57. Cun, X.; Pun, C.M. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Trans. Image Process.* **2020**, *29*, 4759–4771. [CrossRef] [PubMed]
58. Jiang, Y.; Zhang, H.; Zhang, J.; Wang, Y.; Lin, Z.; Sunkavalli, K.; Chen, S.; Amirghodsi, S.; Kong, S.; Wang, Z. SSH: A Self-Supervised Framework for Image Harmonization. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4832–4841.
59. Gong, K.; Liang, X.; Zhang, D.; Shen, X.; Lin, L. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 932–940.
60. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 2918–2928.
61. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
62. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
63. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 694–711.
64. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
65. PaddlePaddle. PaddleDetection: Object Detection and Instance Segmentation Toolkit Based on PaddlePaddle. 2019. Available online: https://github.com/PaddlePaddle/PaddleDetection (accessed on 17 May 2022).
66. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773.
67. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
68. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.