Implementation of Victims Detection Framework on Post Disaster Scenario

Indra Adji Sulistijono[¶], Tegar Imansyah[*], Muhammad Muhajir[†], Edi Sutoyo[‡],
Muhamad Khoirul Anwar[§], Edi Satriyanto[‖], Achmad Basuki[**], Anhar Risnumawan[††]

[*§¶††]Mechatronics Engineering Division
[‖]Electronics Engineering Division
[**]Department of Creative Multimedia
Politeknik Elektronika Negeri Surabaya (PENS)
{[*]tegarimansyah,[§]muhkhoi}@me.student.pens.ac.id,
{[¶]indra,[‖]kangedi,[**]basuki,[††]anhar}@pens.ac.id

[†]Department of Statistics
Faculty Mathematics
and Natural Science
Islamic University of Indonesia
[†]mmuhajir@uii.ac.id

[‡]Department of Information Systems
Telkom University
[‡]edisutoyo@telkomuniversity.ac.id

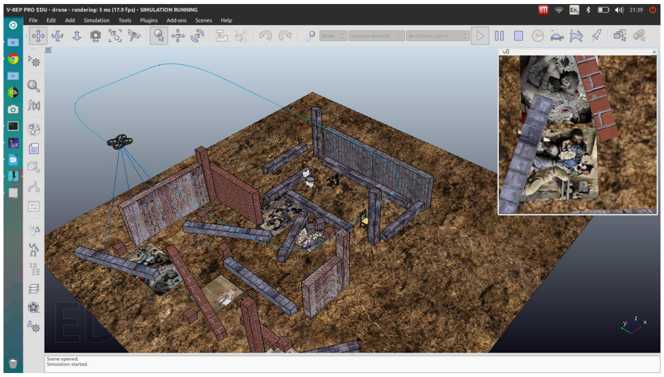
***Abstract*—Disasters are prone to occur in Indonesia due to geographical factors, such as tectonic plate movements, which can cause an earthquake. Earthquakes are one of the most frequent disasters, they have broad impacts in a short time and are unpredictable. Thus, an extensive search process in a short time is highly critical to determine the victims location. In this paper, a victims detection framework is developed starting from acquiring images using an unmanned aerial vehicle and further processing using convolutional neural network (CNN) to locate victims robustly on post-disaster. Input images are then sent to victim detector dedicated ground station server for further high processing robustly locating the possibility of victims. A simulation system mimicking a real environment is developed to test our framework in real time. A transmission protocol is also developed for effectively transmitting data between the robot and the server. The treatment on the detection process of the victim is different from the normal human detection, some pre-processing stages are applied to increase the variation of the given dataset. An embedded system is used for taking images and additional sensors data, such as location and time using Global Navigation Satellite System.**

***Keywords*—Victims detection framework, post disaster scenario, convolutional neural network, embedded system, unmanned aerial vehicle**

I. Introduction

Natural disasters are a phenomenon that can not be resisted but can be anticipated before (mitigation) and after (evacuation). The purpose of disaster evacuation is to minimize the number of victims and losses. One of the most significant hurdles for the rescue team is the lack of information so that the evacuation process becomes slower and fewer lives are saved. Unmanned aerial vehicle (UAV) has been widely available at a reasonably affordable has been able to explore for collecting data in a disaster area.

Existing works using only cameras to detect victims have been performed by [1]–[3]. However, the job uses a background of almost uniform. The background of uniform, e.g., the background is entirely green grass or brown ground, does not describe the original condition after the disaster, and the victim detection is relatively more straightforward because the victim's distinction with the background is obvious. While the original condition after the disaster could be the ruins of buildings, covered victims of soil, fabrics, shrubs, sand, and other materials of various forms scattered that would complicate the detection of victims. Therefore, the detection of post-disaster victims with a complex background is very challenging and interesting to examine.



Scene opened.
Simulation started.

Fig. 1. Real-time victims detection on real simulated images using UAV. The background is quite complex with the real image victims attached on the ground.

An attempt by [4], the concept of deep learning theory has been shown for the detection of post-disaster victims with a complex background, but the work is only a proof-of-concept (the images are not from the bird's eye view, the images tend to be taken from the front, and the full system is not fully elaborated.

Various visual detection and recognition tasks have been successfully improved by deep learning method [5]. Such application for example image classification [6], [7], image segmentation [8]–[10], and object detection [4], [11]–[14]. Deeper network has a main advantage of the ability to learn effective feature representation automatically, which make appealing for practitioners. All the network parameters are solely learned from the training data.

In this paper, a full victims detection framework is developed by leveraging deep convolutional neural networks to detect victims on post-disaster scenario robustly. A simulation

978-1-5386-8083-4/18/$31.00 ©2018 IEEE

system is developed to test further the overall performance from acquiring data till to detection of victims. At first, frames of video and global position are captured by UAV which is attached an embedded platform for low computation. Those data are then sent to victim detector dedicated ground station server for further high processing locating the possibility of victims robustly. SAR teams carefully observe the detected victims including its location. Then SAR teams arrange a plan to perform rescuing operation or re-investigate the suspected location using UAV. A transmission protocol is also developed for effectively transmitting data between UAV and the server.

This paper is organized as follows. Related work is described in Section II. In Section III explains the proposed method. IV and V describe experiments and conclusion, respectively.

II. Related Work

Research with UAVs and camera sensors to detect victims has been done by [1] and [15]. However, the work uses a colorful background, for example, the whole background is green grass or brown ground, which does not describe the original condition after a disaster. The conceptual proof of the use of Deep Learning for the detection of disaster victims in a complex background has been done by [4], but the use of images that are not from bird's eye view and similarly inclined angles are still in use.

This study uses the Convolutional Neural Network (CNN), one of the Deep Learning branches, to detect disaster victims in both bird's eye view, various viewpoints and complex backgrounds. CNN is used because it has been proven capable of producing a good performance for object detection [16]. The method used studied 10,129 human models that augmented to predict the possibility of human poses when disaster strikes. We believe this research will be of great benefit to post-disaster management and related research.

Combined information from different types of sensors have been recently proposed for autonomous victim detection [17], [18] applications. The work by [17] comes particularly similar with this work from which victim detection is taken from UAVs. The authors proposed to utilize a thermal camera to pre-filter promising image locations and subsequently verify them using a visual object detector. While in [17] people lying on the ground are assumed to be in ideal and nearly uniform background, in this paper we address the significantly more complex problem of detecting people in highly cluttered background. Note that the results of our work can still be used in combination with thermal camera images, which similarly to [17] can be used to restrict the search to image locations likely to contain people or to prune false positives, which contain no thermal evidence.

The combination of multiple sensors for people detection is encouragingly beneficial in many scenarios, however, it comes at the cost especially for unmanned aerial vehicles of an increased payload for the additional sensors. This paper, therefore, aims to evaluate and disaster victim detection in highly cluttered background and to minimize sensor requirements as well. For that detection of victims by just using the camera is very important and will be used in this study.

Research using only cameras to detect victims has been done by [1]–[3]. However, the job uses a background of almost uniform. The background of uniform, e.g., the background is entirely green grass or brown ground, does not describe the original condition after the disaster, and the victim detection is relatively more straightforward because the victim's distinction with the background is obvious. While the original condition after the disaster could be the ruins of buildings, covered victims of soil, fabrics, shrubs, sand, and other materials of various forms scattered that would complicate the detection of victims. Therefore, the detection of post-disaster victims with a complex background is very challenging and interesting to examine.

In [4], the concept of deep learning theory has been shown for the detection of post-disaster victims with a complex background, but the work is only a proof-of-concept (the images are not from bird's eye view, the images tend to be taken from the front. In this study, bird's eye view images will be the main focus because it describes the original image condition when taken from the air.

Detection of the victim in the aerial image (bird's eye view) has the primary challenge of varying pose victims because of different viewpoints. A slight difference in viewpoint may cause the visual features of the victim to be different, which may cause common algorithms such as template matching to fail because of the absence of victim pose in the database. Therefore, we need an algorithm that is robust to variations of victim shape.

Manual design of features is mostly used from the above methods, such as [19]–[21]. Moreover, complex kinematics and dynamic is also used which is non-trivial in practice. Too many manual designs can degrade the accuracy of the tracking due to not optimal parameters obtained. Parameters are not learned from training data but mostly from engineering's knowledge and experience. Global optimization [22] could be used to optimize the parameters.

Various visual detection and recognition tasks have been successfully improved by deep learning method [5]. Such application for example image classification [6], [7], image segmentation [8]–[10], object detection [4], [11]–[14], [23], and text detection [19], [24]. Deeper network has a main advantage of the ability to learn effective feature representation automatically, which make appealing for practitioners. All the network parameters are solely learned from the training data. Therefore, in this paper, we leverage deep learning method to solve main issues in gun turret.

III. Victims Detection Framework

Overall system is shown in Fig. 2. At first, frames of video and global position are captured by UAV. Those data are then sent to victim detector dedicated server for further processing locating the possibility of victims. SAR teams carefully observe the detected victims including its location. Then SAR teams arrange a plan to perform rescuing operation or re-investigate the suspected location using UAV.

A. Victims Detector

Main core of our victims detection is Convolutional Neural Networks (CNN). CNN as a deep learning method has shown

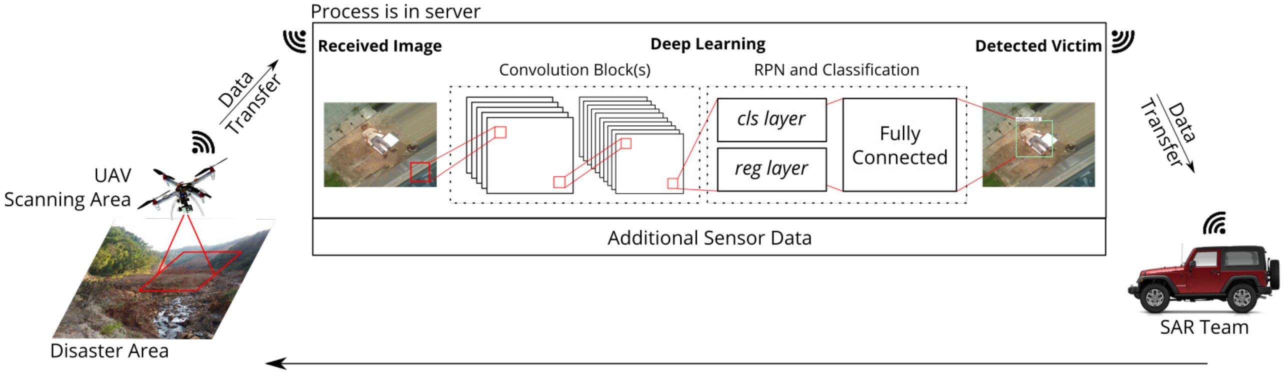


Fig. 2. Overall system of disaster victims detection framework. UAV robot scan a whole disaster area and send images data to high-computational GPU Server, online or offline. Every single image data processed by our method in the server to predict whether there is victim or not. Predicted data could be combined with additional sensor data for SAR Team's early information.

the performance as a notably approach, which is applied in diverse computer vision applications, to learn effective feature automatically from training data and train in an end-to-end [25].

Basically, a CNN comprises of several layers which are staged together. A layer usually consist of convolutional, pooling, and fully connected layers that have different roles. During training forward and backward stages are performed. For an input patch, forward stage is performed on each layer. During training, once the forward stage is performed the output is compared with the ground truth and the loss is used to perform backward stage by updating the weight and bias parameters using a common gradient descent. After several iterations the process can be stopped when the desired accuracy is achieved. All layers parameters are updated simultaneously based on training data.

A convolutional layer consist of N linear filters which is followed by a non-linear activation function h. This work used an activation h on layer m such as the Rectified Linear Unit (ReLU) $h_m(f) = \max\{0, f\}$. In this convolutional layer, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps $f_m(x, y)$, where $(x, y) \in \mathcal{S}_m$ are spatial coordinates on layer m. The feature map $f_m \in \mathbb{R}^{A \times B \times C}$ contains A width, B height, C channels to indicate size of feature map. A new feature map f_{m+1} is produced after each convolutional layer such that,

$$f_{m+1} = h_m(g_m), \quad \text{where } g_m = W_m * f_m + b_m \tag{1}$$

g_m,W_m, and b_m indicate net input, filter kernel, and bias on layer m, respectively. There are three main advantages of the convolution operation 1) the weight sharing mechanism in the same feature map reduces the number of parameters; 2) local connectivity learns correlations among neighboring pixels; 3) invariance to the location of the object.

To reduce the dimensions of feature maps, pooling layer is usually used which is then followed by convolutional layer. Pooling layers are invariant to translation since it takes the neighboring pixels of feature maps. Max pooling is the most commonly used in many applications. Max pooling is simply taking the maximum value from a predetermined window.

Fully-connected layers perform similar as feed forward neural network. It provides us to convert previous multidimensional feature maps into a pre-defined length. It acts as a classification and it could be used as a feature vector for the next processing.

CNN is usually employed to learn a richer features representation for many applications. All layers are learned simultaneously without much tedious jobs of trial and error tuning features and classifier. This is differ from the previous manual features design.

An image patch u as an input to the CNN, then begin forward stage layer-by-layer, and ends by fully-connected layers producing certain labels with its probability. All the parameters are learned from the training data using the common stochastic gradient descent (SGD) by minimizing the loss over ground truth training labels.

B. Hardware Specification

We use a DJI F450 UAV with frame design APM 2.6 for flight controller. It is attached Raspberry Pi 3 for collecting image, the main data, global positioning, and additional sensors. The reasons of using Raspberry Pi 3 as embedded system are relatively capable than common microcontroller with rich connectivity option while still lightweight. Camera module for the embedded system is designed for high data transfer to specific BCM283x processor using Camera Serial Interface (CSI) and optimized in its GPU than common USB camera [26].

Choosing camera specification has its own consideration, one of them is how to make wide area image but the victim still clearly visible. The camera has a fixed focal length 3.6mm, f, with 2592×1944 pixel of maximum sensor resolution and $1,4 \times 1,4\mu$m of pixel size. We can calculate area coverage (Field of View / FOV) on single picture in specific altitude (working distance) with Eq. 2.

$$f \times FOX = SensorSize \times WorkingDistance \tag{2}$$

Usually, the sensor size is not available in every camera datasheet so calculate it with Eq. 3.

$$SensorSize = SensorResolution \times PixelSize \quad (3)$$

Using both equation, we get number of pixel describe full body victim on specific altitude and image resolution shown in Table I. This data can be used for optimal flight planning.

TABLE I. PERSON PIXEL AT SEVERAL ALTITUDE

Person Height (px)		Altitude (m)				
		5	10	15	20	25
Res (px)	640x480	214	107	72	54	43
	800x600	268	134	89	67	54
	1280x720	321	161	107	80	64
	2560x1920	868	434	289	217	173

Even we can attach several sensors, but our method suggest using only camera to reduce weight of payload carried by the UAV. By using image data, at least we can generate victim existence, pose, location and condition. We believe that visible victims has more chance to saved. However to get more specific data available, system was built so that attaching additional sensor will be ease even with extra effort.

For processing complex image, deep learning, or more specific Convolutional Neural Network (CNN), have recently achieved great performance results in many visual perception task, either image classification [6] [7] or object detection [16]. On modern CNN architecture, more than million parameters calculate together to predict learned classes. For faster learning and predicting process, high-computational GPU-based server is necessary. Deep learning involves huge amount of matrix multiplications and other operations which can be massively parallelized. A single GPU might have thousands of cores while a CPU usually has no more than 12 cores. Although GPU cores are slower than CPU cores, it will be faster with their large number and faster memory if the operations can be parallelized. Sequential code is still faster on CPUs.

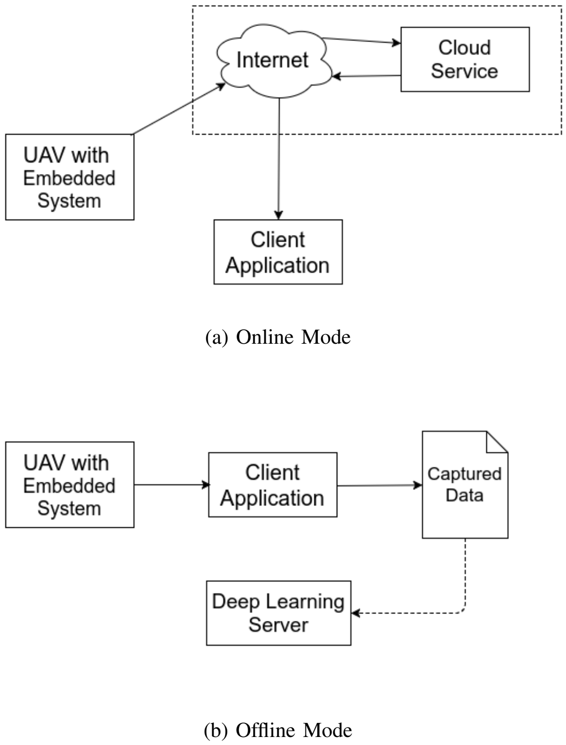
C. Scenario Mode

The design system has been explained in previous section. In practice, sometimes after disaster happen, there is limitation on communication infrastructure that makes us propose two scenario mode shown if Fig 3. The propose scenario mode consider possibility of fastest receiving predicting data of the area.

UAV attached with embedded system is taking image data and possible to attach additional sensors such as location from Global Navigation Satellite System (GNSS) receiver, thermal, etc. Cloud service could be own deep learning server or cloud server such as Google Cloud Service or Amazon EC2.

D. Augmented Dataset

The dataset used in this paper is PASCAL Visual Object Classes (PASCAL VOC) 2012 [27], a well-known dataset of classification and object detection that consist 21 object classes in 17.125 sample image. Each image has anotation of object class label and bounding box for each object that follow PASCAL VOC xml format. The bounding box is four pixel coordinate of image, where $bbox = [top_left, top_right, bottom_left, bottom_right]$ of encapsulate object. An image may be has multiple objects from multiple classes make the dataset more rich.



(a) Online Mode

(b) Offline Mode

Fig. 3. Scenario mode for proposed system (a) if internet network is available and reliable, embedded system connected to cloud server to predicting real-time data. Ground station receive predicted data and display it (b) if internet network not available neither reliable, embedded system connected to ground station and stream data via IEEE 802.11 WiFi Network or radio. Stream data will have predicted with onsite server or cloud server.

For victim detection, we have modified the dataset to annonate only person object in 10.129 images and applied augmentation of each data. The modified dataset has labels $L = background, victim$ that definied in both train or test. Augmentation process is some set of pre-processing image generating numerous image that randomly rotate, horizaontal flip, vertical flip, filling and scaling to make more unique pose that represent victim.

E. Network Architecture

Simple CNN architechture introduced in [25] for digit recognition that consist of two convolution layers, two subsampling layers and closed by 10 classes fully-conected layer composed in series. Modern CNN architectures are similiar with some improvements layer, such as commonly used maxpooling layer than subsampling layer, applying RELU [6], rectification layer, after convolution layer, etc.

The most visible difference, modern CNN architectures are commonly very deep due to capability of hardware resource. The deep itself explain how many layer stack together in series or parallel, even more than 150 layers [28]. For convinience, some layers devided into several blocks contain combination of convolution, RELU and pooling layers. VGGNet [7] for example, the runner up of ILSVRC 2014 that show how depth of the network is a critical component for good performance, is a CNN architecture with five blocks consist of two or three convolution layer that each following by RELU layer and closed with MaxPooling layer. The output of last block going to three stack fully-connected layer. The last layer output is determine how many classes the architecture will determine.

We investigate VGG16 [7] and ResNet50 [28] as base of convolutional block shown in Fig. 2. VGG16 has a great result for 16 convolution and fully-connected layers with only performs 3x3 convolutions and 2x2 pooling from the beginning to the end. The downside of this network is expensive to evaluate with a lot of memory and parameters usage. On the other hand, newer architecture and the winner of ILSVRC 2015, ResNet, works faster and require less parameters even has 50 layers (for ResNet50). ResNet heavy use of batch normalization and introduce *skip connection* that could improve performance from previous layer.

IV. Experimental Results

In this experiments, learning and predicting have been performed using Intel i7-6700K processor with 24 GB of RAM and ZOTAC GTX 1080 AMP Extreme with 2560 CUDA Cores. The computer runs Ubuntu 16.04 with TensorFlow [29] and Keras [30]. Table II shows the time needed for transmitting dan receiving data. We employ a common MQTT protocol for transmitting and receiving image data.

TABLE II. Time needed for sending and receiving image.

Resolution (W×H)	Data Size (Bytes)	Sending Time (ms)	Encoding Time (ms)
640 × 480	0.478.377	50	66
800 × 600	0.749.899	54	86
1296 × 972	2.589.624	150	230
1920 × 1080	4.465.457	235	391

A. Learning Process

Learning process is done by using the modification of the VOC PASCAL dataset described earlier using ResNet50 architecture combined with RPN [16]. The process is done with 70 epochs and 1000 iterations, which means running 70,000 times forward and backward pass, as shown in Fig. 4.

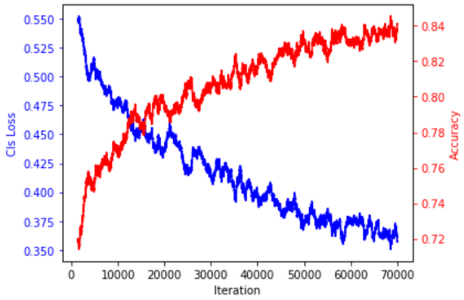


Fig. 4. More iteration make lower loss and higher accuracy, but possibly to overfitting.

Learning and predicting are performed on four architectures, ZF, VGG CNN M 1024, VGG16 and ResNet50 with 5, 13, 16, and 50 layers, respectively. We use Keras with TensorFlow library for the code implementation. RPN layer in faster R-CNN is intertwined at the end of the layer for bounding box formation. We use PASCAL VOC 2007 datasheet with learning iteration is 35,000 and the result is shown in Table III.

TABLE III. Learning and Predicting Time

Architecture	Time (s)	Accuracy (%)	Detection Time (s/image)
ZF	23760	73.2	0.35
VGG CNN M 1024	22032	74.3	0.42
VGG16	44434	76.6	0.85
ResNet50	31262	84.1	0.45

B. Qualitative Results

Qualitative results on several datasheets and different viewpoints have been performed as follows:

1) Simulated Environment: As shown in Fig. 5 when the full framework is simulated in real-time. We build the environment using V-REP educational version.

2) Victims Datasets: In this study, models were tested based on various disaster datasets. In some images still occur false positive or false negative. False positive is a condition in which the detection process finds the victim but is not actually a victim. In contrast to false negative where the detection process did not find the victim but actually there were victims. This study provides a threshold to detect victims with a confidence level of 80%.

As shown in Fig. 6, the first dataset tested was IDV50 containing 50 images of disaster victims with the primary purpose of detecting on a chaotic background. In Figure 6, there are three examples of detected data. The first image (left) shows three objects that have different sizes and locations against the background of the house ruins. The second image (center) shows the model can detect even when the face only looks and tends to have the same color as the background. However, it also detects two false positives. The third image (right) is also able to detect two objects where the first one looks partially face down and the second object is only visible hands and somebody.

The second dataset uses Freiburg Disaster which focuses on indoor casualty testing. In the first and second pictures in figure 6, the detection can be done on the human body that is visible or covered by a third of the lower body. What is interesting is that in the third picture where the head is not visible, and one leg is partially closed, the detection can tell the whole body part or only the leg part.

The third dataset uses a dataset constructed in this study which focuses on the supine, middle or center covered poses called PENS Victim Detection Research 2017 (PVDR2017). In the first and second images have the same result that has a high level of confidence in the detected object, but there are two false positive that is on the victim reflection on glass and plants. In the third picture, in addition to the same false positive, the victim can still be detected on the head, leg, or whole.

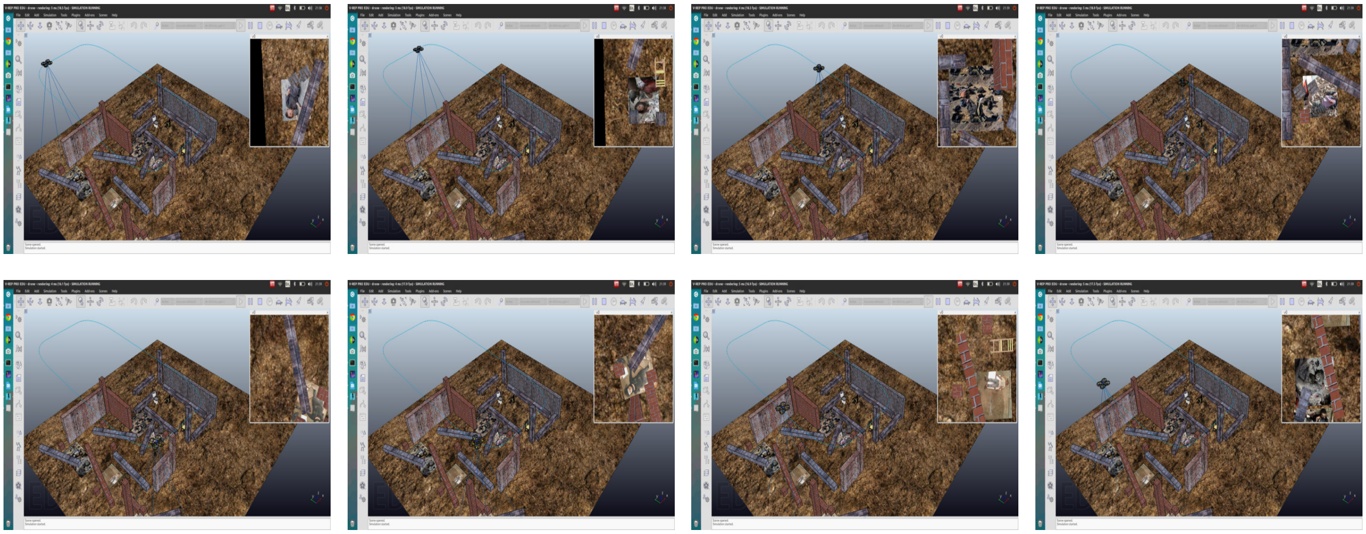


Fig. 5. Simulation results of our framework for detecting people in real time while the UAV moves in a predefined path. Image of victims are real which is attached on the terrain surface.

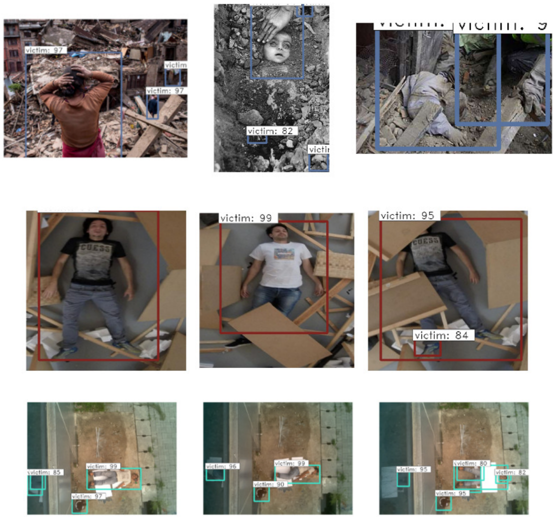


Fig. 6. Qualitative Results from IDV50 [4] (first-row), Freiburg [31] (second-row), and PVDR2017 (last-row) images datasheets.

3) Different Viewpoints: This test, as shown in Fig. 7 still uses the dataset created in this study but augmentation process is done to see the victim at various point of view. One image on the dataset is converted into 25 different images.

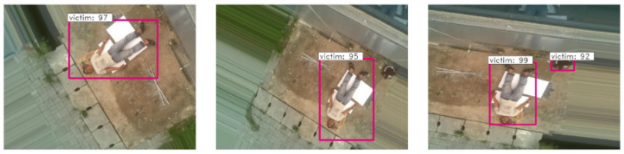


Fig. 7. Qualitative results for different viewpoints on PVDR2017 datasheet.

The predicted results performed on the dataset show good results at different points of view. It only happens a few times false positive in plant pots but does not occur false negative. This is good because it is better to misidentify the victim than not detect the victim.

4) Different Altitudes: Another image variation of the PVDR2017 dataset is the taking of victims at various altitudes, as shown in Fig. 8. There are some images with a height of 5.4, 9.7, 14, 18.3, 22.6, and 26.9 meters.

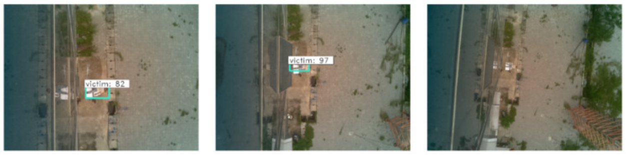


Fig. 8. Qualitative results for different altitudes on PVDR2017 datasheet.

In this test is done with the same pose with different heights. The first and second images (left - center) respectively are at altitude 9,7 and 18,3 meter still able to be detected without false positive. However, in the third picture at 26.9 meters height, there is a false negative where the victim is not detected. This is because of the calculation of human form that is not in accordance with the number of pixels available to describe the shape of the victim.

V. Conclusion

In this paper, a full victims detection framework has been developed by leveraging deep convolutional neural network for robust detection in complex background and large appearance of victims. A simulation system also has been developed for testing a full framework to the real simulated scenario. A transmission protocol is also developed for effectively transmitting data between UAV and the server. The experiments show encouraging results that it would be beneficial for the future works on the related field.

VI. Acknowledgements

We would like to thank Ministry of Research, Technology, and Higher Education of Indonesia for supporting the research with the grant INSINAS 13/E/KPT/2018.

References

[1] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *IROS*. IEEE, 2010, pp. 1740–1747.

[2] P. Blondel, A. Potelle, C. Pégard, and R. Lozano, "Fast and viewpoint robust human detection for sar operations," in *SSRR*. IEEE, 2014, pp. 1–6.

[3] ——, "Human detection in uncluttered environments: From ground to uav view," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2014, pp. 76–81.

[4] I. A. Sulistijono and A. Risnumawan, "From concrete to abstract: Multilayer neural networks for disaster victims detection," in *Electronics Symposium (IES), 2016 International*. IEEE, 2016, pp. 93–98.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[9] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[10] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[13] ——, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[15] P. Blondel, A. Potelle, C. Pégard, and R. Lozano, "Human detection in uncluttered environments: From ground to uav view," in *ICARCV*, Dec 2014, pp. 76–81.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007, pp. 1–13.

[18] A. Kleiner and R. Kummerle, "Genetic mrf model optimization for real-time victim detection in search and rescue," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3025–3030.

[19] A. Risnumawan, I. A. Sulistijono, and J. Abawajy, "Text detection in low resolution scene images using convolutional neural network," in *International Conference on Soft Computing and Data Mining*. Springer, 2016, pp. 366–375.

[20] A. Risnumawan and C. S. Chan, "Text detection via edgeless stroke width transform," in *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on*. IEEE, 2014, pp. 336–340.

[21] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8027–8048, 2014.

[22] Y. Saadi, I. T. R. Yanto, T. Herawan, V. Balakrishnan, H. Chiroma, and A. Risnumawan, "Ringed seal search for global optimization via a sensitive search model," *PloS one*, vol. 11, no. 1, p. e0144371, 2016.

[23] M. K. Anwar, A. Risnumawan, A. Darmawan, M. N. Tamara, and D. S. Purnomo, "Deep multilayer network for automatic targeting system of gun turret," in *Engineering Technology and Applications (IES-ETA), 2017 International Electronics Symposium on*. IEEE, 2017, pp. 134–139.

[24] M. L. Afakh, A. Risnumawan, M. E. Anggraeni, M. N. Tamara, and E. S. Ningrum, "Aksara jawa text detection in scene images using convolutional neural network," in *Knowledge Creation and Intelligent Computing (IES-KCIC), 2017 International Electronics Symposium on*. IEEE, 2017, pp. 77–82.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] J. Creasey, *Raspberry Pi Essentials*. Packt Publishing Ltd, 2015.

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[30] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[31] G. L. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox, "Deep learning for human part discovery in images," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1634–1641.