



Real-Time Implementation of Human Detection in Thermal Imagery Based on CNN

Nazeer Shahid*¹, Gwang-Hyun Yu*², Tan Dat Trinh*³, Do-Seong Sin**⁴, and Jin-Young Kim*⁴

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00314) supervised by the IITP

Abstract

In this paper, an effective human detection method in thermal imaging is proposed using background modeling and convolution neural network(CNN). For real-time implementation, the background modeling is done by modified running Gaussian average and the CNN-based human classification is performed for only detected foreground objects. To enhance human detection accuracy, morphological operators and ellipse testing are adopted to extract Region of Interest. Also, three CNN models with different input sizes and voting method are trained using our own dataset. For real-time system, the whole system is implemented in C++ and it process more than 30 fps with high accuracy.

요약

본 논문은, 열화상 이미지에서 배경 모델링과 합성곱 신경망(CNN)을 이용한 효과적인 인간 탐지 방법을 제안한다. 실시간 구현을 위하여, 배경 모델링은 수정된 연속하는 가우시안 평균에 의해 얻어지고, CNN 기반의 인간 분류는 오직 탐지된 전경의 객체로만 이루어진다. 인간 탐지 정확도 향상을 위해서, 형태학적 연산자와 타원 테스트를 채택하여 관심 영역을 추출한다. 또한 자체 데이터를 이용하여 3개의 다른 입력 크기를 갖는 CNN 모델과 투표 기법을 학습한다. 실시간 시스템을 위해서, 전체 시스템은 C++로 구현되고 초당 30프레임 이상의 처리 속도 및 높은 정확도를 가진다.

Keywords

background modeling, CNN, deep learning, human detection, thermal videos

* Dept. of Electronics and Computer Eng, Chonnam National University
- ORCID¹: <http://orcid.org/0000-0003-4311-7607>
- ORCID²: <http://orcid.org/0000-0003-0434-5260>
- ORCID³: <http://orcid.org/0000-0003-0443-612X>
- ORCID⁴: <http://orcid.org/0000-0002-4896-8980>

** Defence Technology

- ORCID: <http://orcid.org/0000-0003-4924-2062>

· Received: Oct. 05, 2018, Revised: Nov. 06, 2018, Accepted: Nov. 09, 2018

· Corresponding Author: Jin Young Kim

Dept. of Electronics and Computer Eng, Chonnam National University

Tel.: +82-62-530-1757, Email: beyondi@jnu.ac.kr

1. Introduction

Real-time human detection is a challenging and significant field in computer vision research[1]. It includes focusing on building systems for observing humans, which further provides advanced interfaces for creating realistic models of human detection[2].

Researchers have proposed multiple techniques in field of human detection. They have done quality work in this field that covers varying situations like, aerial tracking system for UAVs[3], monocular images [4], Infrared Images[5], varying temperature and brightness[6]. Moreover, researchers have also experimented by combining the result attained from vision and thermal imagery using a dual camera[7]. "Ghost effect" is a problem regarding real time human detection[8]. This demands to gather data keeping in mind multiple factors to properly exploit multispectral information and ensure reliability in result[9]. However, the normal video does not work at night time for human detection task, to overcome the disadvantage of RGB camera, we proposed to use thermal camera for detection task. Several algorithms based on thermal infrared systems provide acceptable results for human detection[10]. The techniques for human detection using thermal imagery include wavelet transform[11], shape distribution[12], histogram feature and modified sparse representation classification [13], region extraction with curvelet space validation [14], Haar-cascade classifier[15]. The conditions for detection also vary and they may range from camera set on a vehicle[16], availability of fuzzy system[17] or tracking of objects from ocean's surface[18]. We have proposed a practical application for human detection which includes background subtraction and human classification using CNN. There are many background subtraction techniques that are applicable to thermal videos such as frame difference, kernel density estimation[19], temporal medium filter(TMF) [20], running Gaussian average[21], mixture of Gaussians (MoG)[22], codebook[23].

In the last few years, deep learning and in particular "Convolutional Neural Networks (CNN)" has created remarkable results in various computer vision tasks including image classification, object detection and segmentation. Generally, deep learning allows to learn representations of data with multiple levels of abstraction[24]. The deep learning based approaches have outperformed the previous gold standards by a large margin[25].

CNN is playing a vital role in real time data manipulation. With this recent successful progress of CNNs in vision community, it is attracting attention of researchers on how to properly explore information in color and thermal images[26]. CNN has led new ways to detect humans in real time. They have experimented CNN with different situations and techniques to get real time people detection[27]. In practical applications, real time processing with high detection accuracy is a critical requirement[28]. Pose calibration, motion prediction and many other factors are used to estimate the tracking path to effectively detect human in real time[29]. One researched technique includes implying Ensemble Inference Network (EIN) to the classification process to accurately detect human[30].

In summary, our process is divided into two main parts, detection and classification of the objects. In the first part, we apply a background subtraction model for getting the foreground. Initial background model is achieved using baseline Running Gaussian Average (RGA). We introduce the robust approach to cater the presence of moving object in the initial frames. Robust principle component analysis(RPCA) is applied on small regions for estimating the background and saving the computational cost. Pixel parameters of background model are updated based on selectivity and random subsampling.

Frames are effected by sudden change in intensity. To handle this problem, the frame values are updated using histogram matching with reference frame based on skewness value. For getting the correct ROI,

post-processing is applied for candidate extraction using morphology operator and candidate removal is applied based on ellipse measurement of ROI. Secondly, in our classification part, pre-trained CNN model is used for classifying the ROI as human or non-human. Owing to variable size of ROI, we use three different models to classify human via voting method for final decision. Fig. 1. is an illustration of our proposed method for realtime human detection.

II. Detection Process

Object detection deals with detecting instances of semantic objects of a certain class in digital images or videos. Detection algorithms extract features from imagery and use learning algorithms to classify them as specific class. In continuation of our previous work [31], we implemented our background subtraction technique in realtime. In this section, we will introduce our robust method for initialization, update of background model and foreground detection for the basic RGA. In addition, we will explain the technique for catering the intensity variation problem using histogram matching. Finally, for extraction of suitable ROI, we use morphology and introduce our new heuristic algorithms using candidate removal based on shape information.

2.1 Modified Running Gaussian Average

We get the initial background model by applying Running Gaussian Average (RGA). The main idea

behind this approach is to fit Gaussian probability density function(pdf) with mean and standard deviation on each pixel. The starting RGA model is to compute mean and standard deviation from initial 200 training images to make the background model. In practical environment, we cannot avoid presence of moving objects in the initial background model which leads to “ghost effect.” To overcome this problem, we figure out the moving object regions from our initial training frames. Then, Robust principle component analysis (RPCA) is applied on each region for estimating the background. However, to handle the computational cost for RPCA, it is only applied on the specific regions. Steps of background modeling is shown in Fig. 2.

First we compute mean and standard deviation from initial 200 training images and then apply thresholding to get the moving objects. We remove small noise using morphology technique; find out connected components and crop that regions from the frame to apply RPCA on each region.

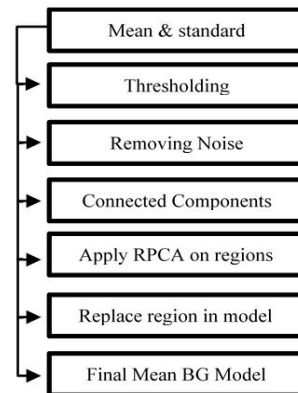


Fig. 2. Background modeling process

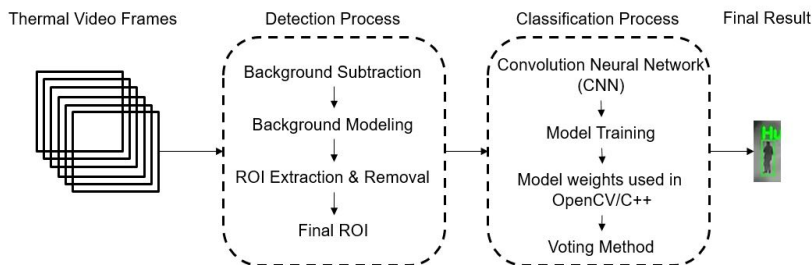


Fig. 1. Overall process of our method

Low rank components is calculated using Eigen::BDCSVD library, these regions are merged into initial mean model to get the final background model. In some cases, moving objects may be overlapped and make a large connected component. Watershed algorithm is used for segmenting these overlapped regions into small regions, which saves the computational cost.

Next important step is background update so that possible changes are incorporated in the initial model. In the RGA, parameters of each pixel in background model can be updated directly. μ_t mean and σ_t variance parameters are updated at each pixel I_t . α is learning rate which controls the slow and fast adaptation. High value of α is applied to background pixels for fast updating, and low value is selected if the pixel is foreground for stability.

Separate updating processes for background and foreground are adopted as follows.

$$\mu_t = \alpha I_t + (1 - \alpha) \mu_{t-1} \quad (1)$$

$$\sigma_t^2 = \alpha |I_t - \mu_t|^2 + (1 - \alpha) \sigma_{t-1}^2, \quad (2)$$

where I_t is background pixel.

$$\mu_t = \frac{\alpha}{3} I_t + (1 - \frac{\alpha}{3}) \mu_{t-1} \quad (3)$$

$$\sigma_t^2 = \frac{\alpha}{3} |I_t - \mu_t|^2 + (1 - \frac{\alpha}{3}) \sigma_{t-1}^2, \quad (4)$$

where I_t is foreground pixel. After several experiments, we choose 12.5% pixels randomly from the background pixels of new frame to update the background. This technique helps to lower the computational cost of update as initial model is not updated at once.

Foreground detection is achieved by satisfying the equation (5). Generally, k is a constant with value 2.5 and I_t is foreground pixel.

$$|I_t - \mu_t| > k \sigma_t \quad (5)$$

Thermal videos are effected by sudden change in intensity. For instance, if a big hotter object like car passes by, the intensity of the whole frame suddenly changes which results in failure of the background subtraction. To overcome this problem, we use the skewness value of the frame and compare with threshold value (90 from experiments). If the value is less than the chosen threshold value then histogram matching is applied and intensity of the effected frame is updated with respect to reference frame.

2.2 ROI extraction & removal

With the help of morphology operator, small noise is removed and the ROI is extracted. Extraction steps are shown in Fig. 3.

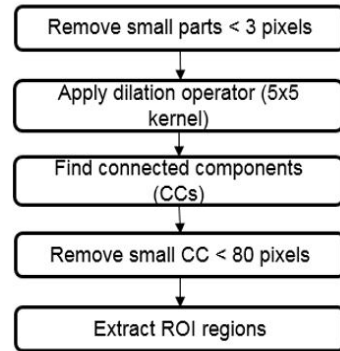


Fig. 3. ROI extraction process

First, regions which are smaller than 3 pixels are removed. Dilation operator with 5x5 kernel is applied to join the unconnected segments of an object. Connected components are found and objects which are smaller than 80 pixels are rejected. Then we find the bounding boxes and ROI regions are extracted.

We propose a new postprocessing method for candidate validation based on object shape information. In detail, to remove the candidate, we use second central moment to present ellipse measurement of

ROI's. Several measurements that are based on approximating regions by ellipses are Orientation (Θ), MajorAxisLength (M), MinorAxisLength (n), Eccentricity (e) and Extent (r).

In ROI, dilation is used to capture the smallest possible objects or overcome unconnected parts (the foreground contains holes and/or several unconnected parts).

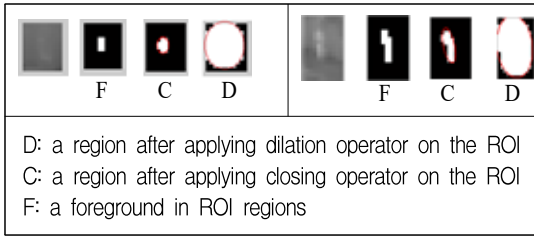


Fig. 4. Illustration of morphology method on given ROI

With the help of these measurements, we propose a Heuristic algorithm to remove unwanted candidate.

Table 1. Heuristic algorithm

Heuristic algorithm: if there is one region in C	
With reference of Fig. 4, for each ROI	
$C(M) < 6$ and $(abs(C(\theta)) = 0 \text{ or } abs(C(\theta)) = 45)$	(6)
$C(Area) < 6$ or $(C(M) < 4 \text{ and } C(n) < 3)$	(7)
Case 1: If equations (6) or (7) are satisfied, remove small noise or smaller objects such as bird, leaf etc.	
$((abs(C(\theta)) \geq 85 \text{ and } abs(C(\theta)) \leq 90) \text{ or } abs(C(\theta)) \leq 4)$	(8)
$(C(M) > 55) \text{ and } (C(e) \geq 0.95) \text{ and } (C(n)/C(M) < 0.2)$	(9)
Case 2: If equations (8) and (9) are satisfied, remove large noise such as ground, tree branch etc.	

Our proposed algorithm successfully reduces the noise and unwanted ROI's in single region and extracts the ROI for classification.

III. Classification Process

Classification of moving objects is a fundamental step for rejecting non-human moving objects. In our method we apply Convolutional Neural Networks (CNN) for classification purposes. CNN achieved significant performance in detection and classification using RGB images. However, there are limited papers applied CNN to classification for thermal image which will be introduced in this section. Using convolution and pooling layers, it is possible to efficiently extract the most relevant features from the images. Results are generated through logistics or softmax output and back-propagation is used to minimize the loss function. Max pooling makes feature detection robust by selecting the maximum value of all selected squares. The main benefit behind using convolutions is parameter sharing (number of weight parameters in one layer are reduced without effecting accuracy) and sparsity of connections.

3.1 Convolution Neural Network

After successful extraction of ROI, these ROIs are treated as the input for classification. Currently CNN is the most famous deep learning classifier thus we propose to use CNN as a classifier in our real-time application. The general structure of CNN contains Convolution layers, max pooling layers and fully connected layers. We use three layer CNN model to train with the data. The architecture of our model is shown in the Fig. 5.

The most important factor for gaining desired results in CNN is the training data. Result efficiency of the model depends on size and quality of data. For real-time application, we use our own dataset to train the model. Details of our dataset is presented in section 4.1.

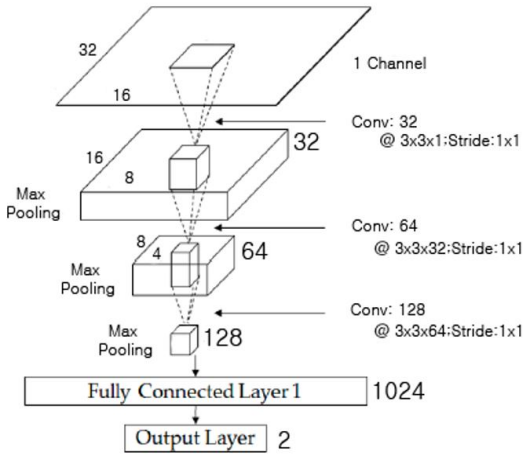


Fig. 5. Our CNN model structure used for training

We train three CNN models with different input sizes for classification. In real-time environment, the object size differs from small to big because distance of objects with respect to camera is different, so the models are trained with three input sizes [32×16], [64×32] and [128×64].

3.2 Training

In this section we explain the structure of our CNN and the training process in detail. For making our CNN model, TensorFlow and Python programming language is used. CNN has an input layer, an output

layer and in between these two layers are convolution, max pooling layer, non-linear activation function (in our case, we used Rectified Linear Unit – ReLU) and the fully connected layer. The 3×3 kernel in convolution layer is applied to get 32, 64 and 128 feature maps. Pooling layer down samples the image data extracted from convolution layer so we use 2×2 (pixel tile) with stride of 1 in pooling layer. In fully connected layer, we get 1024 feature vector which is then linked with output layer to produce two classes for human & non-human.

Different input sizes [32×16], [64×32] and [128×64] are used for training the models. The main idea behind using three models is to improve the classification process as the object sizes differ. Therefore, we also add one and two more convolution layer in second and third model respectively.

After finishing training, we save our models and test their accuracy. By TensorFlow library, our model is saved in the form of computational graphs and weights which are associated with these graph nodes. The two main files are ckpt.meta and .ckpt.data as shown in Fig. 6 in detail. ckpt.meta contains the graph information while .ckpt.data contains the weights related to that graphs. Next section includes more detail regarding conversion of weights readable in OpenCV.

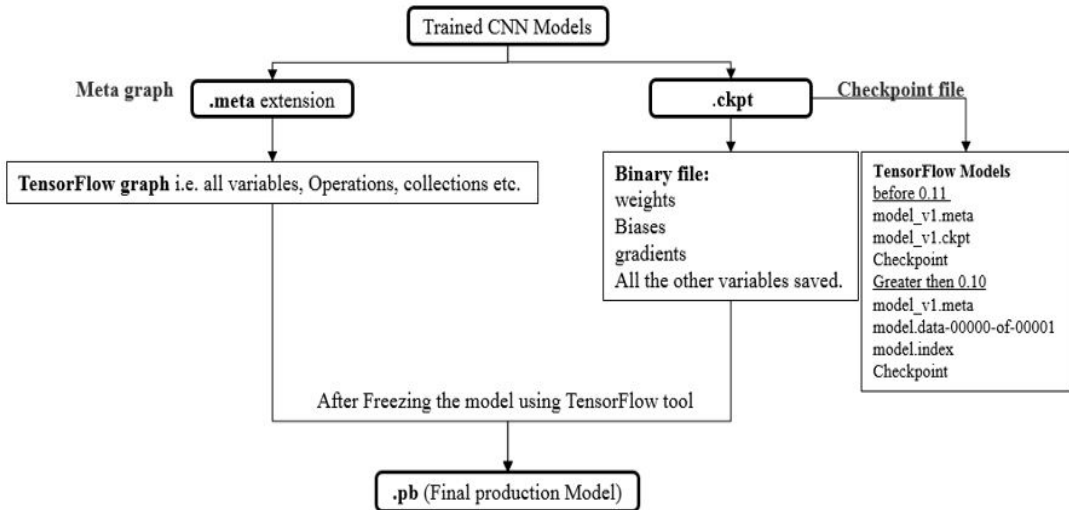


Fig. 6. Conversion of variable weights into constants for a production model

3.3 Weights conversion

After successful training, weight parameters are saved in the form of checkpoint files. These weights are variable so to use them for production to import or load into other programming languages like C++, we require to freeze these weights with respect to graph definition. To do that, we have to make them constant using function “convert_variable_to_constant()” which is available in Tensorflow. Furthermore, only those nodes in the graph are converted into constants which have weights. The final frozen model is made by deleting the dropout layer (with no weights) from the main graph definition. We also define the output_node_names from which we get the output of the model. Finally after saving the frozen model, we get a single file with .pb extension which is the final production model.

3.4 Voting method

The classification performance is effected by object size which keeps on changing in video. To enhance the classification performance, we proposed three trained models with different input sizes as shown in Fig. 7.

The final ROI is resized to the same input of the model. However, the shape of the actual object is able to change little by that step. Due to that, we propose to use different models. Finally, we resize our object into three different sizes and put them as input of the models. Based on voting technique, we finally decide either the object is human or non-human.

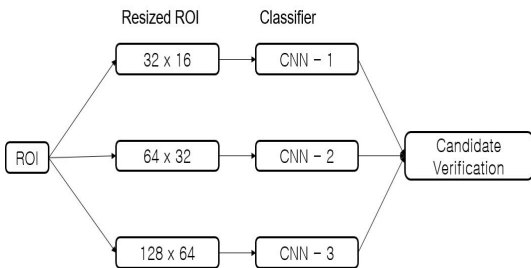


Fig. 7. Candidate verification from three classifiers

IV. Experiment set up and Results

4.1 Dataset

Using Argo S Thermal camera, we recorded several thermal videos in different months and times at Chonnam National University from the 3rd floor of the building as shown in Fig. 8. The average temperature lied between 16°C to 35°C. The camera resolution was 640×480 pixels. The videos were recorded at speed of 31 frames per second. To reduce the computational cost, we resized image into exactly half. The height of the camera was almost 15 meters, so the objects were different in sizes. Illustration of our thermal video frames is depicted in Fig. 9.

When using thermal cameras, if the objects are hotter than the background then they will appear lighter and in opposite condition objects will appear black. In night time, the temperature of the body is more than the background therefore the human appear white. Conversely, in day time, the background is hotter than human body so the objects look completely black. In general, we can conclude that object color is varied due to the temperature difference.



Fig. 8. Camera & site for recording thermal videos

Table 2. Thermal videos dataset

Videos	No of Frames
Video-01	44147 frames
Video-02	32904 frames
Video-03	43993 frames
Video-04	29190 frames
Video-05	43432 frames
Video-06	35301 frames

Table 3. Training images for CNN model

Human	Non-Human	Total
35649	152651	188300

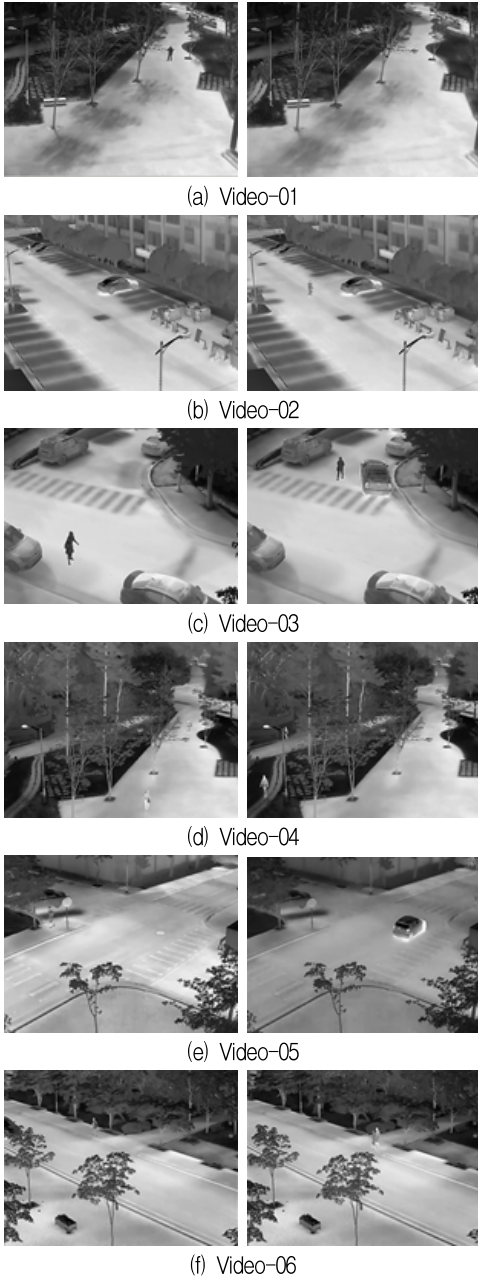


Fig. 9. Illustration of our thermal video frames

To make our dataset for CNN training, we cropped both the human objects and background images to train our model with binary labels. In detail, after cropping the images, we made the dataset into CSV and assigned class label for each class, 0 for human and 1 for non-human. For three models, we changed

the input sizes $[32 \times 16]$, $[64 \times 32]$ and $[128 \times 64]$ to handle the objects varying in sizes.

4.2 Real-time application

Making a real-time application for detecting some moving objects is a challenging task. Our real-time application is explained in detail as follows. We primarily used OpenCV & C++ for making the application. First of all, we explain the structure of our application. The core part of our application is built in C++, VS community version 2017. OpenCV is used for reading & writing the frames, finding connected components, morphology operations, skewness of the frame and histogram matching. Calculating low rank matrix for RPCA, we used Eigen::BDCSVD library for large matrices and other custom functions which are not supported in OpenCV.

We have already explained in detail the training of CNN model and freezing the weights in section 3.3. For classification purposes, we integrated the frozen weights of our model into real-time application through OpenCV Deep Neural Network module which permitted us to load the CNN production model to C++ environment.

Finally, we set the extracted ROI as the input of three CNN networks. First, we checked the ROI with the small network size and then with greater sizes. Based on voting technique, we finally decided if the resulted ROI is human or not.

4.3 Results

The Graphical User Interface (GUI) of our real-time application is shown in Fig. 10. The supported functions in our application are selecting any video file with format (.mp4 or .avi), stop/pause and change/select new video. The original video, background subtraction, post processing and final result are illustrated in the 1st, 2nd, 3rd, 4th column respectively.

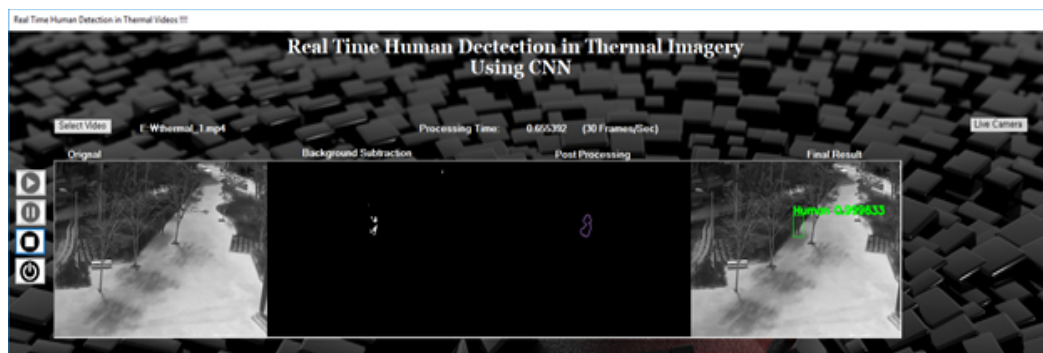


Fig. 10. Our real-time application

The proposed background subtraction technique is successfully applied. The result of background subtraction is shown in Fig. 11. In these thermal videos, we have various moving objects including human, car and bicycle. Moreover, the polar inversion is examined in videos by white and dark objects.

After successful background subtraction, we got ROI as the objects are distinguished from the background. The regions of interest occurred on different locations of the frame. We then passed these ROI through the classification process for classifying them as human or non-human. The result of our ROI extraction in our thermal videos is shown in Fig. 12.

Finally, the classification results of human objects from our classifiers is shown in Fig. 13. Each classifier returns a probability value against ROI. In case ROI is classified as human via voting technique, the maximum probability value generated from classifiers is shown in the results. The ROI was passed to three CNNs separately. If at least any two CNN models classify the ROI as human then it is shown in the results as human with maximum probability value.

Furthermore, to evaluate the efficiency of our technique, we selected 100 frames randomly in each video. Then on these frames, we calculated recall, precision and F-1 score metrics. The performance of our proposed method is shown in Table 4.

Our proposed approach achieved up to 92.10, 96.86

and 94.37 for recall, precision, and F-1 score, respectively.

Table 4. Performance Results

Videos	Recall(%)	Precision(%)	F Value(%)
Video-01	88.72	100	94.02
Video-02	89.28	96.15	92.59
Video-03	96.66	93.54	95.08
Video-04	94.52	98.57	96.50
Video-05	88.54	94.44	91.40
Video-06	94.89	98.48	96.65
Average	92.10	96.86	94.37

V. Conclusion

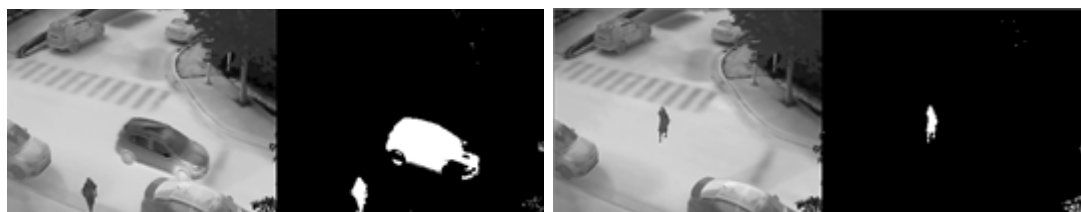
In this paper, we used two different platforms of Python and OpenCV/C++ to implement real-time human detection in thermal images. We introduced our robust method for initialization, updating of background model and foreground detection for the basic RGA. Our method can effectively handle different problems like Ghost effect and intensity sudden change. The success of our technique lies in manipulation of results in real time effectively. The Morphology was applied to extract ROI. Shape based candidate removal was applied using ellipse measurement of the ROI. With the combination of three different CNN models by voting technique, we achieved a reliable classification result in thermal imagery.



(a) Video-01



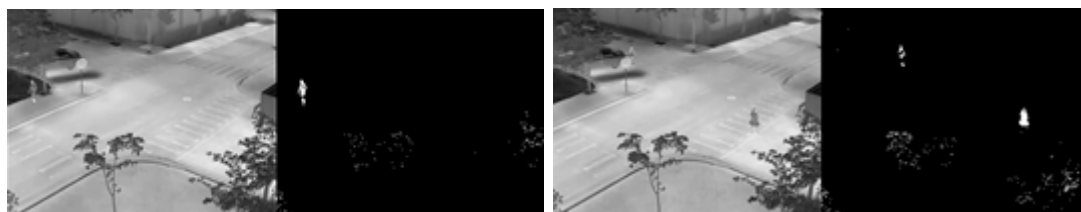
(b) Video-02



(c) Video-03



(d) Video-04



(e) Video-05



(f) Video-06

Fig. 11. Background subtraction results on different frames in our thermal videos

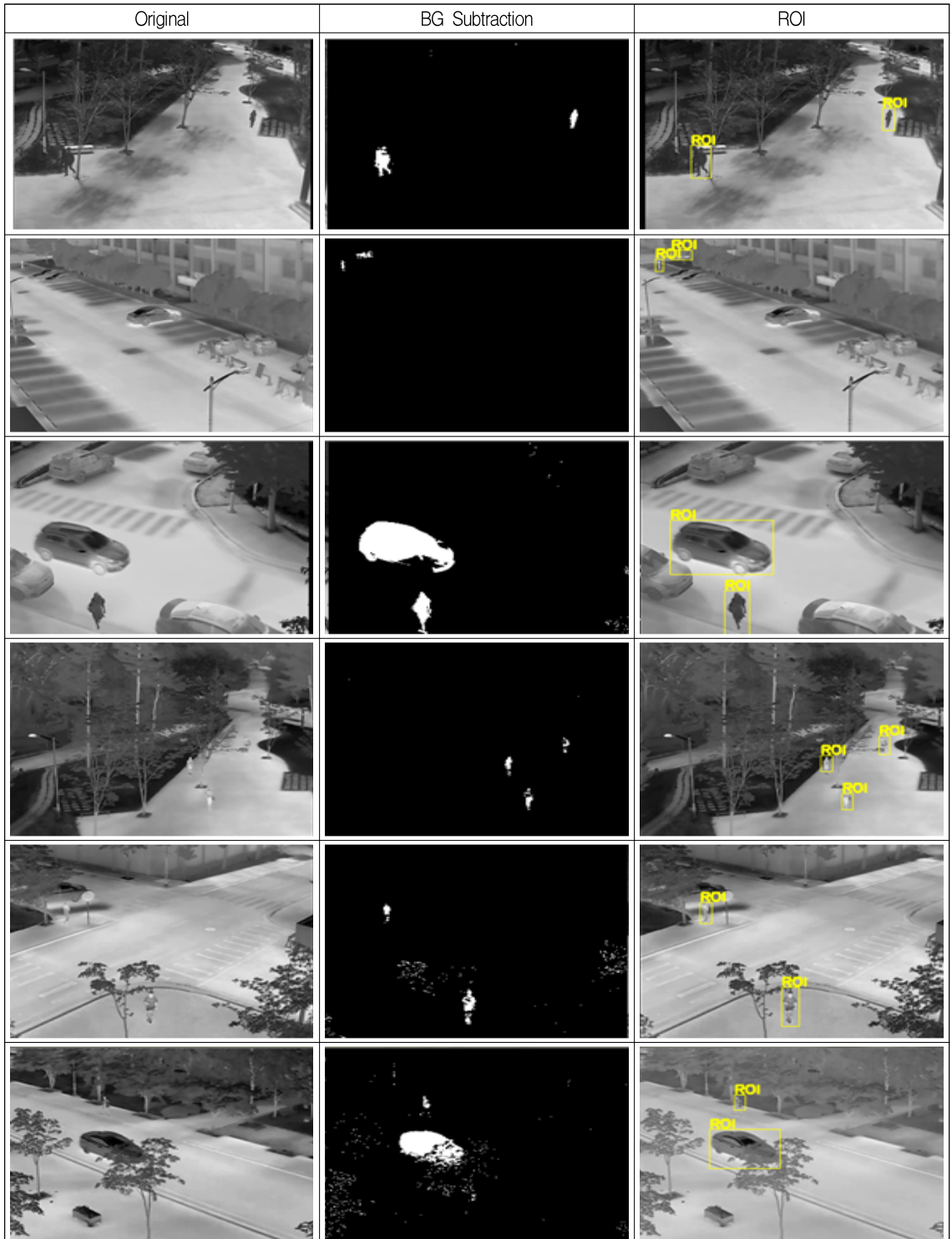


Fig. 12. ROI extraction in thermal videos


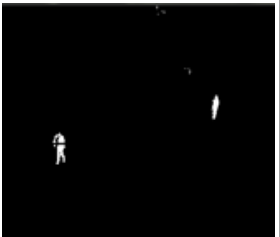
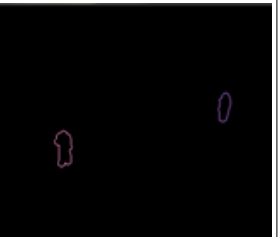


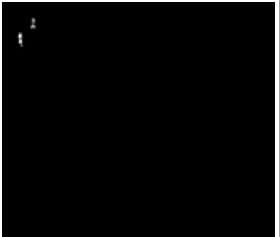
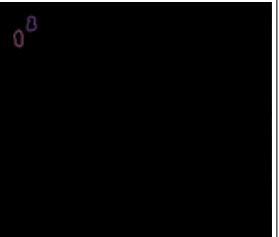






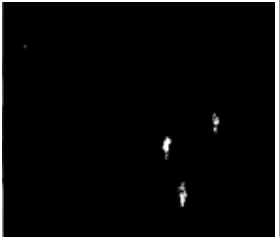




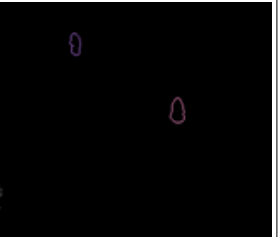


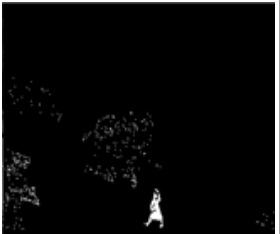
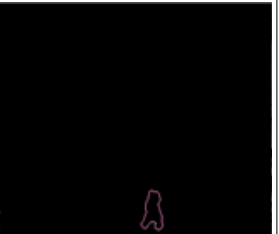

Original	BG Subtraction	Post Process	Final Results
			
			
			
			
			
			

Fig. 13. Final Classification results in thermal videos

References

- [1] S. V. Tathe and S. P. Narote, "Real-time human detection and tracking", IEEE Annual India Conference (INDICON), pp. 1-5, Dec. 2013.
- [2] N. Ogale, "A Survey of Techniques for Human Detection from Video", Master's thesis, University of Maryland, Jul. 2006.
- [3] M. Mueller, G. Sharma, N. Smith, and B. Ghanem, "Persistent Aerial Tracking system for UAVs", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1562-1569, Oct. 2016.
- [4] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 4, pp. 743-761, Apr. 2012.
- [5] R. Soundrapandiyam and C. Mouli P.V.S.S.R, "Adaptive Pedestrian Detection in Infrared Images Using Background Subtraction and Local Thresholding", Proceedings of Second International Symposium on Computer Vision and the Internet (VisionNet'15), Vol. 58, pp. 706-713, Aug. 2015.
- [6] T. Kim and S. Kim, "Pedestrian detection at night time in FIR domain: Comprehensive study about temperature and brightness and new benchmark", Pattern Recognition, Vol. 79, pp. 44-54, Jul. 2018.
- [7] J. H. Lee, J. S. Choi, E. S. Jeon, Y. G. Kim, T. T. Le, K.Y. Shin, H. C. Lee, and K. R. Park, "Robust Pedestrian Detection by Combining Visible and Thermal Infrared Cameras", Sensors, Vol. 15, pp. 10580-10615, May 2015.
- [8] F. Yin, D. Makris, and S. A. Velastin, "Real-time ghost removal for foreground segmentation methods", In proceedings of International Workshop on Visual Surveillance - VS2008, Oct. 2008.
- [9] Y. L. Hou, Y. Song, X. Hao, Y. Shen, and M. Qian, "Multispectral pedestrian detection based on deep convolutional neural networks", IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1-4, Oct. 2017.
- [10] N. K. Negied, E. E. Hemayed, and M. B. Fayek, "Pedestrians' detection in thermal bands - Critical survey", Journal of Electrical Systems and Information Technology, Vol. 2, No. 2, pp. 141-148, Sep. 2015.
- [11] J. Li, W. Gong, W. Li, and X. Liu, "Robust pedestrian detection in thermal infrared imagery using the wavelet transform", Infrared Physics & Technology, Vol. 53, No. 4, pp. 267-273, Jul. 2010.
- [12] B. Qi, V. John, Z. Liu, and S. Mita, "Pedestrian detection from thermal images: A sparse representation based approach", Infrared Physics & Technology, Vol. 76, pp. 157-167, May 2016.
- [13] X. Zhao, Z. He, S. Zhang, and D. Liang, "Robust pedestrian detection in thermal infrared imagery using a shape distribution histogram feature and modified sparse representation classification", Pattern Recognition, Vol. 48, No. 6, pp. 1947-1960, Jun. 2015.
- [14] A. Lakshmi, A. G. J. Faheema, and D. Deodhare, "Pedestrian detection in thermal images: An automated scale based region extraction with curvelet space validation", Infrared Physics & Technology, Vol. 76, pp. 421-438, May 2016.
- [15] C. H. Setjo, B. Achmad, and Faridah, "Thermal image human detection using Haar-cascade classifier", In proceedings of 7th International Annual Engineering Seminar (InAES), pp. 1-6, Aug. 2017.
- [16] K. Lenac, I. Maurović, and I. Petrović, "Moving objects detection using a thermal Camera and IMU on a vehicle", In proceedings of International Conference on Electrical Drives and

- Power Electronics (EDPE), pp. 212-219, Sep. 2015.
- [17] V. John, S. Mita, Z. Liu, and B. Qi, "Pedestrian detection in thermal images using adaptive fuzzy C-means clustering and convolutional neural networks", In proceedings of 14th IAPR International Conference on Machine Vision Applications (MVA), pp. 246-249, May 2015.
- [18] F. S. Leira, T. A. Johansen, and T. I. Fossen, "Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera", IEEE proceedings of Aerospace Conference, pp. 1-10, Mar. 2015.
- [19] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos", Computer Vision and Image Understanding, Vol. 122, pp. 4-21, May 2014.
- [20] M. H. Hung, J. S. Pan, and C. H. Hsieh, "A fast algorithm of temporal median filter for background subtraction", Journal of Information Hiding and Multimedia Signal Processing, Vol. 5, No. 1, pp. 33-40, Jan. 2014.
- [21] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body", IEEE Trans. on Patt. Anal. and Machine Intell, Vol. 19, No. 7, pp. 780-785, Jul. 1997.
- [22] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection - a survey", Recent Patents on Computer Science, Vol. 1, No. 3, pp. 219-237, Nov. 2008.
- [23] J. Y. Jung, "Codebook-based foreground background segmentation with background model updating", Journal of Digital Contents Society, Vol. 17, No. 5, pp. 375-381, Oct. 2016.
- [24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning", Nature. Vol. 521, pp. 436-44, May 2015.
- [25] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural Networks, Vol. 61, pp. 85-117, Jan. 2015.
- [26] D. C. D. Oliveira and M. A. Wehrmeister, "Towards Real-Time People Recognition on Aerial Imagery Using Convolutional Neural Networks", IEEE proceedings of 19th International Symposium on Real-Time Distributed Computing (ISORC), pp. 27-34, May 2016.
- [27] J. H. Kim, H. G. Hong, and K. R. Park, "Convolutional Neural Network-Based Human Detection in Nighttime Images Using Visible Light Camera Sensors", Sensors, Vol. 17, No. 5, pp. 1-26, May 2017.
- [28] S. V. Tathe and S. P. Narote, "Real-time human detection and tracking", IEEE Annual India Conference (INDICON), Dec. 2013.
- [29] Y. Sun, L. Sun, and J. Liu, "Real-time and fast RGB-D based people detection and tracking for service robots", In proceedings of 12th World Congress on Intelligent Control and Automation (WCICA), pp. 1514-1519, Jun. 2016.
- [30] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, "Pedestrian detection based on deep convolutional neural network with ensemble inference network", IEEE Intelligent Vehicles Symposium (IV), pp. 223-228, Jul. 2015.
- [31] T. D. Trinh, X. Ma, and J. Y. Kim, "Improved Running Gaussian Average for Background Subtraction in Thermal Imagery", Journal of Korean Institute of Information Technology, Vol. 15, pp. 101-117, Jul. 2017.

Authors

Nazeer Shahid



2007 : BEIT degree in Dept. of Engineering & Information Technology.
2008 ~ 2017 : Worked in Research & Training Institute (Software Development)
2017 ~ 2019 : Student of MS degree in Dept. of Electrics Eng, Chonnam National University
Research interests : Digital Signal Processing, ML, DL, Information Security, Cyber Security

Gwang-Hyun Yu



2017 : Founder of FTEN company, Korea
2018 : MS degree, in Dept. of Electronics Eng, Chonnam National University
2018 : Student of Ph.D degree, in Dept. of Electronics Eng, Chonnam National University
Research interests : Digital Signal Processing, Image Processing, Speech Signal Processing, ML, DL

Tan Dat Trinh



2010 : BS degree in Mathematics and Computer Sciences from HCMC University of Natural Sciences.
2013 : MS degree, in Dept. of Electronics Eng, Chonnam National University
2017 : Ph.D degree, in Dept. of Electronics Eng, Chonnam National University
Research interests : speaker recognition, computer vision, machine learning and pattern recognition.

Do-Seong Sin



1998 : MS degree, in Dept. of Electronics Eng, Chonnam National University
2010 ~ 2016 : Research professor, in Information Technology Research Center
2017 : Ph.D degree, in Dept. of Electronics, Information and Communication Eng, Chonnam National University
2017 : Reaearch Manager, in DefenseTech Inc.
Research interests : Digital Signal Processing, Image Processing, Machine Learning, Biometrics

Jin-Young Kim



1986 : BS degree, in Dept. of Electronics Eng, Seoul National University
1988 : MS degree, in Dept. of Electronics Eng, Seoul National University
1994 : Ph.D degree, in Dept. of Electronics Eng, Seoul National University
1995 ~ now : Professor in Department of Electronics and Computer Engineering, Chonnam National University
Research interests : Digital Signal Processing, Image Processing, Speech Signal Processing, ML, DL