Mask Classification and Head Temperature Detection Combined with Deep Learning Networks

Isack Farady
Department of Electrical Engineering
Yuan Ze University,
Taoyuan, Taiwan
School of Electrical Engineering
Mercu Buana University,
Jakarta, Indonesia
isack.farady@mercubuana.ac.id

Chih-Yang Lin
Department of Electrical Engineering
Yuan Ze University
Taoyuan, Taiwan

Amornthep Rojanasarit
Department of Electrical Engineering
Yuan Ze University
Taoyuan, Taiwan

Kanatip Prompol
Department of Electrical Engineering
Yuan Ze University
Taoyuan, Taiwan

Fityanul Akhyar
Department of Electrical Engineering
Yuan Ze University,
Taoyuan, Taiwan
School of Electrical Engineering
Telkom University,
Bandung, Indonesia

***Abstract*— Due to the COVID-19 pandemic, wearing a mask is mandatory in public spaces, as properly wearing a mask offers a maximum preventive effect against viral transmission. Body temperature has also become an important consideration in determining whether an individual is healthy. In this work, we design a real-time deep learning model to meet current demand to detect the mask-wearing position and head temperature of a person before he or she enters a public space. In this experiment, we use a deep learning object detection method to create a mask position and head temperature detector using a popular one-stage object detection, RetinaNet. We build two modules for the RetinaNet model to detect three categories of mask-wearing positions and the temperature of the head. We implement an RGB camera and thermal camera to generate input images and capture a person's temperature respectively. The output of these experiments is a live video that carries accurate information about whether a person is wearing a mask properly and what his or her head temperature is. Our model is light and fast, achieving a confidence score of 81.31% for the prediction object and a prediction speed below 0.1s/image.**

Keywords—neural network, object detection, deep learning, RetinaNet.

I. INTRODUCTION

In recent years, the development of neural networks and deep learning has vastly contributed to object classification and detection performance [13, 16]. The modern deep learning object detection paradigm is based on two-stage detectors and one-stage detectors. The dominant two-stage detectors in the R-CNN family [3, 10] perform well on detection accuracy. Numerous developments on Faster R-CNN [10] have generated a set of candidate proposals called RPN [10] in stage one and integrated them with a strong classifier at the second stage into a single Convolution Neural Network (CNN). Meanwhile, the popular one-stage detectors SSD [8], YOLO [1] have reignited interest in modern object detectors. One-stage detectors have now turned their focus toward speed and accuracy. The RetinaNet detector [7] is a single network designed with a new concept involving anchor boxes by RPN, Feature Pyramid Network [6], like in SSD, and the network's innovation focal loss has been shown to match the accuracy of a two-stage detector at a similar speed.

Fig. 1. The detected object from two different input images from the RGB camera and thermal camera. A) The output of Module 1 detects the position of the mask and classifies it into the corresponding class label with a confidence score. B) Detected "head" label of output image in Module 2 from the thermal camera. C) Output image of Module 3. A combination of the mask-wearing position and head temperature of the detected head are combined into a single output image containing object prediction and object temperature

Our work uses deep learning to detect whether someone is wearing a mask properly or not. To do so, we build a network to learn the features of our input image dataset to recognize three kinds of mask-wearing positions. We define those positions into three classes of the object, then set the classification sub-network and regression sub-network in the

© IEEE 2020. This article is free to access and download, along with rights for full text and data mining, re-use and analysis

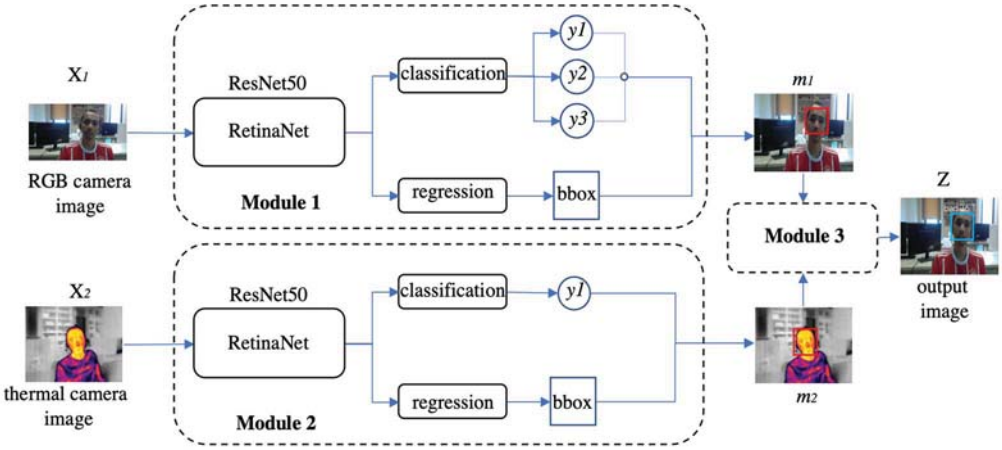

Fig. 2. The network architecture of the 3-module object detection. Two different input images are trained into the object detection backbone and each of the networks produces a class prediction and label. Lastly, Module 3 will use the results from two other modules and combine them to deliver a new output. All training and testing processes are run in one single light pipeline task.

last layer of the RetinaNet. The output of Module 1 and Module 2 determine the class of a mask-wearing position and show the bounding box on the predicted class. To improve the comprehensiveness of preventive action in the real world, in Module 3 we combine two predicted results from Modules 1 and 2 then generate an image with a bounding box and show the highest temperature of a predicted head bounding box from the thermal camera. We design all modules in one single pipeline network as a light deep learning network for other applications.

II. Related Works

Currently, many famous object detection methods are used to detect an object through one or two visible camera combinations to explore more helpful features and collect more additional datasets. Person re-identification recognition is one problem that can be solved by taking advantage of a thermal camera [14]. Visual inputs of thermal cameras have a valid appearance and capture accurate temperature information under any illumination or lighting conditions.

Thermal images are commonly used as additional input images to capture more supplementary feature information via a convolutional neural network. An experiment by [2] shows the object detection improvement in the thermal image domain by borrowing features from the RGB image. A combination system for image recognition in [9] also utilizes the combination of two different input images to reduce captured noise in RGB images. Exploitation data for enhanced object detection was also shown in [5] by using a combination of two networks, Retina and U-Net [11], to fuse a one-stage object detector and U-Net for semantic segmentation. Another famous object detection method that combines two models to achieve the highest accuracy is RefineNet [15]. The idea behind these combinations relies on two inter-connected modules; one for anchor refinement and another for object detection.

We adopt the same idea from those previous works by simply modifying the combination of two similar backbones. Since thermal cameras were designed to capture the heating surface in all captured frames, we needed to apply deep learning image detection to capture heat temperature only on the detected object. Unlike previous experiments, here we formulate two input images from a visible RGB camera and thermal camera to get more additional information about the temperature of the specific object detected. We manage the detection network to get accurate coordinates from both models, especially for the thermal camera, to capture temperature at a specific point. This work aims to produce an image result that contains the detection object and the temperature of the object. The result must include complete information on specific task object detection in a single pipeline network of 2 RetinaNet [7] with ResNet50 [4] as a convolutional neural network backbone for object classification task. An additional module is added to expand the prediction result and combine it into one single output bounding box prediction.

III. Methodology

In this section, we first introduce our mask and thermal detection pipeline before discussing the experiment details of the three main modules: 1) Object detector network, 2) Head thermal detector and 3) Combination of both output images.

A. Framework Description

As mentioned above, in this experiment we implement RetinaNet as an object detector network. RetinaNet is composed of a backbone network and sub-network. We use ResNet50 as a convolutional neural network backbone for computing feature maps of the input image dataset. We adopt all detector network parameters and loss function from [4] as built on FPN. We simply implement focal loss formula from RetinaNet into our multi-class case for both networks to yield a confident prediction on the class of mask and head images. We utilize the advantage of an FPN for a multi-scale feature pyramid from a single image resolution. Since our dataset is limited, we have to optimize ResNet50 and FPN to calculate feature maps and predict the class. Much like RetinaNet, we use the same parameter for the RPN, but modify it for 3 classes of label detection and adjust the threshold. We assign anchor intersection-over-union (IoU) threshold and each anchor is assigned one object class in one bounding box prediction.

In this network, two sub-networks work in parallel between class prediction and regressing the offset from each bounding box. Classification sub-network predicts the probability of object for each anchor with FCN network

Fig 3. Sample of detected images from different angles and different *y* class labels of mask-wearing positions. A, C) Images captured from real-time RGB camera with all classes of mask positions. B, D) Classification images from the testing dataset with class label predictions.

attached to each FPN level, and shares similar parameters for all layers of FPN. Our FPN input image channels for both networks consist of three channels before applying the 3x3 convolution layer size in each subnetwork. Sigmoid activation is applied for class prediction in the last layer. The sub-network of the regression box also attaches FCN to each pyramid level for the regressing offset of an anchor box near the ground truth bounding box.

In contrast with Network Module 1, as shown in Figure 2, in our experiment of Network Module 2 we highlight the problems on the regression offset of the bounding box to identify the maximum captured temperature of the predicted bounding box. We integrate all modules of this detector framework into one pipeline thread function and add one last module for a combination of two detection results.

The main aim of the detector module is to produce the exact mask-wearing position and head position of an input image. In order to determine the class prediction of our network, we calculate the precision of our class detection result from Module 1 and Module 2 by using the confidence score prediction metric. The confidence score defines the probability of the event or probability of the input to fall into different classes. Confidence value is calculated for all testing images, to determine how confident our algorithm is for *y* class. To comply with real-time application requirements, our network has to reach maximum speed in the prediction process; therefore we show the prediction time of our live video in testing time.

B. Combination Module

Module 3 is a bounding box combination module from two different prediction results. In this case, we assume that the input of both prediction results are different images and resolution scales. The concept of this module is to add two bounding box predictions of two different trained networks with a typical image from each network. Module 1, with an image from the RGB camera, will produce m_1 image $\{x_1, y_1, x_2, y_2\}$ bounding boxes and $y\{1,2,3\}$ class labels. Module 1 will generate *y* class labels based on the loss function of a classification network setting. Module 2, with an image from a thermal camera, produces m_2 images containing bounding box coordinates $\{x_1, y_1, x_2, y_2\}$ and $y\{1\}$ class label and *c* for temperature (*Celsius)*. Module 3 runs through two modules as a thread of two algorithms in parallel. The testing procedure for one image or one frame is shown in Algorithm 1.

Module 1:
  While True:
    Get testing image X_1
    Detect mask coordinate $\{x_1, y_1, x_2, y_2\}$
    If detected:
      Draw m_1 *bbox* $\{x_1, y_1, x_2, y_2\}$
      Classify m_1 into *y* class label
Module 2:
    Set temperature as global variable
    Set initial temperature *c* = 0
    While True:
      Get testing image X_2
      Detect head coordinate $\{x_1, y_1, x_2, y_2\}$
      If detected:
        Draw m_2 *bbox* $\{x_1, y_1, x_2, y_2\}$
        Set $c_{max\ of\ bbox}$
Module 3:
    Set image output resolution
    Draw output image Z with *bbox* $\{x_1, y_1, x_2, y_2\}$, c_{max}, and *y* class label [good, bad, none]

With the assumption that both cameras are positioned at the same angle, we are able to precisely show one identical predicted object. The output of Module 3 is a combination between the bounding box prediction of *y* class labels and temperature *c* of the predicted object as shown in Figure 1.

IV. EXPERIMENTS

In this section, we evaluate our proposed mask and head temperature detector system by experimenting on a live camera as an input of test images. We introduce how we evaluate the training parameters and testing results of each category of detection of our method. Details of all experiment settings for live testing are also described in this section. We additionally show failed and false detection results from our method.

A. Evaluation Method

We train the network for an RGB camera with a public medical mask dataset from the Kaggle dataset[1]. The dataset contains 678 images with annotations and 3 class labels *y* (good, bad, none) that indicate mask-wearing position. From the dataset for Module 2 of the thermal image, we collect 800 captured thermal images. In contrast to the treatment of RGB images, for the thermal image dataset we only define one *y* (head) class label since technically we only consider taking the highest temperature inside the bounding box prediction rather than classifying the class image. We divided the collected dataset into training, validation, and testing data. Practically speaking, we employ all testing images from the live video source to capture in real-time temperature.

We implement Keras RetinaNet [2] object detection framework as our base framework. We follow a similar configuration of RetinaNet basic parameters. To simplify the classification training, we set image size at 800 x 1333 pixels

[1] [1] https://www.kaggle.com/shreyashwaghe/medical-mask-dataset

[2] [2] https://github.com/fizyr/keras-retinanet

TABLE I. PREDICTION RESULTS

Module	Confidence Score			Average Time	Average Score
	class y1 [good]	*class y2 [bad]*	*class y3 [none]*		
1	99.84%	78.69%	65.41%	0.11s	**81.31%**
2	99.7%	-	-	0.10s	**99.7%**

for both modules as a default input size. We build ResNet50 as a backbone model on top of RetinaNet, and load ImageNet pre-trained weight as a starting initial weight. ImageNet with pre-trained weight has shown outstanding performance in the ImageNet Large Scale Visual Recognition challenge for classification tasks [12].

An output image m_1 from Module 1 contains the prediction of the class label and the bounding box with the threshold setting. In this experiment, we set the NMS threshold to 0.5 for the IoU threshold value to determine when a box should be suppressed. This bounding box configuration will be intuitively selected based on the best score of prediction class results. We show the prediction results for the classification and bounding box in Figure 1. We straightforwardly construct similar parameters on Module 2 to simplify the networks

The main task of Module 2 is to detect the head class after obtaining the maximum temperature value inside the bounding box prediction. During the testing with real-time input video, we did encounter some failed detection problems with Module 2. The output of Module 2 failed to detect head class, after which Module 3 failed to update the real-time temperature for a short period. As depicted in Figure 4.A, the output image showed the temperature of the last detected bounding box.

All training processes of this experiment were run on Microsoft Windows with one NVIDIA GeForce GTX 1080i graphic card. The training accuracy of *y* class predictions are depicted in Figure 3. We built a visual testing environment from a high-resolution RGB camera with a maximum resolution of 720 pixels/30 fps and one radiometric longwave infrared camera with 160x120 active pixels.

B. Results

We evaluated our network prediction by calculating the confidence score of the prediction *y* class label in Modules 1 and 2. We used an unseen test image to calculate the prediction results. The confidence score indicates the percentage of correct predictions out of all predictions. As we defined in our method, we only consider a prediction to be correctly labeled if the IoU ≥ 0.5. The average scores are shown in Table I.

As shown in Figure 4, some prediction results of head detection were not as accurate as expected. Module 3 sometimes had difficulty detecting an input from another thermal camera with different technical specifications. In contrast, when we experiment with the same input image from the same thermal camera, Module 3 works well. This result impacts the output of Module 3, but this problem is instantly

Fig 4. Sample of the failed detected images from Module 1 and Module 2. In A, C) our model gives two high confidence scores for different classes. In B, D) Module 2 fails to detect the different quality images from another output camera.

resolved after Module 2 detects the head position and updates it in real-time in Module 3. We will consider enlarging the study in terms of variation and producing more datasets in the next experiment for both modules.

V. CONCLUSION

In this work, we successfully construct a deep learning object detection network to detect and capture the temperature of a specific point inside a predicted bounding box. We propose a novel approach consisting of two networks trained simultaneously from two different inputs, and combined on the last module to fuse certain pieces of information. By utilizing ResNet50 on top of RetinaNet we successfully detect and classify 3-class labels on Network Module 1 and Module 2. Future work could involve more training and diverse thermal head or facial images to enlarge the dataset.

REFERENCES

[1] C. Devaguptapu, N. Akolekar, M. M Sharma, and V. N Balasubramanian, "Borrow from anywhere: Pseudo multi-modal object detection in thermal imagery," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.1029-1038 , 2019.

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

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

[4] P. F. Jaeger, S. A. Kohl, S. Bickelhaupt, F. Isensee, T. A. Kuder, H.-P. Schlemmer, and K. H. Maier-Hein, "Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection," Machine Learning for Health Workshop, pp. 171-183, 2020.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125, 2017.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Proceedings of the IEEE international conference on computer vision, pp. 2980-2988, 2017.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," European conference on computer vision, pp. 21-37, 2016.

[8] D. T. Nguyen, H. G. Hong, K. W. Kim, and K. R. Park, “Person recognition system based on a combination of body images from visible light and thermal cameras,” Sensors, vol. 17, no. 3, pp. 605, 2017.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[10] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2017.

[11] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

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

[13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” International Conference on Medical image computing and computer-assisted intervention, pp. 234-241, 2015.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, “Imagenet large scale visual recognition challenge,” International journal of computer vision, vol. 115, no. 3, pp. 211-252, 2015.

[15] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” Advances in neural information processing systems, pp. 2553-2561, 2013.

[16] M. Ye, X. Lan, J. Li, and P. C. Yuen, “Hierarchical discriminative learning for visible thermal person re-identification,” Thirty-Second AAAI conference on artificial intelligence, pp. 7501-7508, 2018.

[17] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4203-4212, 2018.

[18] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212-3232, 2019.