Fire Detection for Automated Firefighting Robot byusing EfficientDet

Author's Details

[1]Anh P.T. Nguyen
Faculty of Automation-Hanoi Univ of Science and TechnologyHanoi- Vietnam
anh.nguyenphamthuc@hust.edu.vn Orcid:0000-0001-9838-9971
[2]Son Hoang
Faculty of Electro-Mechanics andCivilization-Vietnam National University of ForestHanoi- Vietnam
hoangsonbk83@yahoo.com.vn
[3]Nguyen X. Nguyen
Electrical Department-Hanoi College of Electro-MechanicsHanoi- Vietnam
nguyencdhn@gmail.com
[4]Huy M. Nguyen
AI & Robotics Laboratory-Hanoi Univ of Science and TechnologyHanoi- Vietnam
Huy.NM211231M@sis.hust.edu.vn
[5]Thanh Nguyen
Foreign Language Specialized SchoolVietnam National University
Hanoi- Vietnam flssk52thanh@gmailcom
[6]Dan D. Hoang
AI & Robotics Laboratory-Hanoi Univ of Science and TechnologyHanoi- Vietnam
dan.hd191727@sis.hust.edu.vn

Abstract
Fire accidents are one of the leading hazards endangering human life, the economy, and the environment. Fires outbreak will lead the firefighters on duty facing with dangers. Smart firefighting systems with Artificial Intelligence may provide safety measures and minimize loose from hazardous fire accidents. Vision-based Mobile robots which can work autonomously or be controlled from a safe distance are good candidate to extinguish fire and rescue victims without risking the life of the firefighter. In this study, we developed a firefighting mobile robot with crawler drive which can move on various terrains, especially can climb up/down in upstairs for firefighting tasks. In order to integrate with vision- based control unit of the firefighting robot, a Deep Convolution Neural Netwwork (DCNN) EfficientDet has been applied to detect fire and smoke regions. Augmentation and transfer learning techniques have been utilized to cope with limitation of fire datasetfor training. After training process, the achieved model has been transplanted to a Jetson Nano for real-time smoke and fire detection. The experimental results verify the effectiveness of the proposed fire detector in real-time detection with high mean average precision. Vision-based control unit has been designed to control the robot moving to the proximity of fire regions.
Keywords*-firefighting robot, convolution neural network,EfficientDet, transfer learning, fire detection.*

I. INTRODUCTION

Fire accidents are one of the hazardous disasters thatdevastate the world, damage seriously to human life and health, and destroy national resources. Rapid urbanization and industrialization recently across the country lead to an increase of fire. According to Police Department of Fire Fighting and Prevention, there were 1,154 fires across the country in the first six months of 2021, that killed 53 people and estimated damage assets VND 288,76 billion. Responsible of firefighters is to rescue people, control and extinguish fires and on the duty they face to difficulties and dangers. Chaotic environment in fire accidents is a major cause of death and injury of trapped people. Firefighters needto receive accurate information of fire regions, fire level, and location of trapped persons in real-time to make exact decision on rescue and firefighting operations. It is stressful for firefighters when vision is limited or obscured, the working environment may change continuously due to thick smoke, high temperature, collapse of ceilings, walls or relocation of furniture.

Research and development of firefighting robots to replace firefighters becomes an urgent requirement. It is expected to design, manufacture and control firefighting robots with high applicability, autonomous, flexile, reliable operation, and reasonable cost. Almost firefighting robots do not meet the expectation that they can observe clearly, implement heavy physical tasks, push or move obstacles on the way. Surrounded smoke, high temperature, noise, obstacles, and debris may reduce the effectiveness of ultrasonic- and laser sensors that commonly installed in traditional navigation and guidance systems [1,2]. The development of artificial intelligence (AI)-especially machine vision created smart for firefighting robots, provided safety measures and minimize losses of fire accidents. Smart of a firefighting robot has been measured by its autonomy and, more importantly, by its ability to cope with disturbance as well as by its adaptivity to a dynamic fire environment. Autonomousity of the firefighting robots depend on speed and precise of fire detection for determination of fire condition.

The latest achievements in sensor technology open opportunities for detection of the fire. The primary sensing technologies were based on thermal, gas, flame, smoke and some other important fire characteristics and considered as point sensors, such as infrared, ultrasonic, laser and optical sensors. etc. These sensors activate when particles from the fire source reach the sensor body, then certain delay in their responses may cause. These sensors are not well suited to critical environments with large and open spaces. Furthermore, the fire flame and smoke have certain static and dynamic features such as color and motion, but the point sensors do not utilize these important features for fire detection. The task of firefighting becomes easier if the information of flame and smoke location, their severity, height, growth, direction, etc. are determined. Such multidimensional information is not possible with point sensors. For small space locations, these are the low-cost options for detecting smoke and flames and these can be used for multi-sensor systems in a complementing mode. Vision- based fire flame and smoke detection system is an advantage option to overcome most of the disadvantages of point sensor-based smoke and flame detection systems [3,4,5,6].

Almost research on flame detection since the last decade based on methods of traditional feature extraction [3,4]. The disadvantage of the such methods is time consuming and low accuracy, and furthermore can give out false detection when working in shadow, varying lightings and fire-color object. In order to avoid such disadvantages, a potential approach is deep-learning detection system. The deep learning approach has several differences from the conventional computer vision-based fire detection. The first is that the features are not explored by experts, but rather are automatically captured in the network after training with a large amount of diverse training data. Clearly, the effort to find the proper handcrafted features is shifted to designing a proper network and preparing the training data. Another difference is that the detector/classifier can be obtained by training simultaneously with the features in the same neural network. Therefore, the appropriate network structure becomes more important with an efficient training algorithm [5]. In our study, we applied EfficientDet- a multi-scale object detection network for fire detection work [9,10,11].

The remainder of this paper is organized as follows: Section II describes the main components and working principle of firefighting robot. Section III introduces CNN- based fire detection and introduce EfficientDet network. Section IV describes training process for fire detection by EfficientNet and visualization results in detail. Conclusions are presented in Section V.

II. COMPONENTS AND WORKING PRINCIPLE

A. *Components*

The design of firefighting robot is illustrated in Fig. 1. Power supply for system is battery with voltage of 12 Volt and capacity of 400Ah, so the battery could supply energy for the system in at least 10 hours. The robot has two main wheels in front, that are driven by two brushless motors. When the main wheels rotate, a chain track belt-driven mechanism moves to drive small wheels. A lifting mechanism is to create a lifting angle with steps for chain track belt-driven mechanism able to climb up and down on stairs as illustrated in Fig. 2. Besides it is responsible for taking pictures in-site regions. Deep- camera D415i plays a role of vision sensor, it is fixed on a Robot frame and takes in-site pictures and send them to a vision machine unit, smoke and fire areas in the images will be bounded by boxes and labeled by EfficientDet- a novel Convolutional Neural Network (CNN), and then distance from the robot to detected fire-areas will be calculated and sent to a PLC-based control unit. A fire extinguishing systeminclude a water tank and a water pump. The water pump on roller will pump water or soap to extinguish the fire depending on the class of fire that occurs. However, in this paper, we focus only on fire detection based on deep-leaningneural networks- an important task to decide the work of therobot.

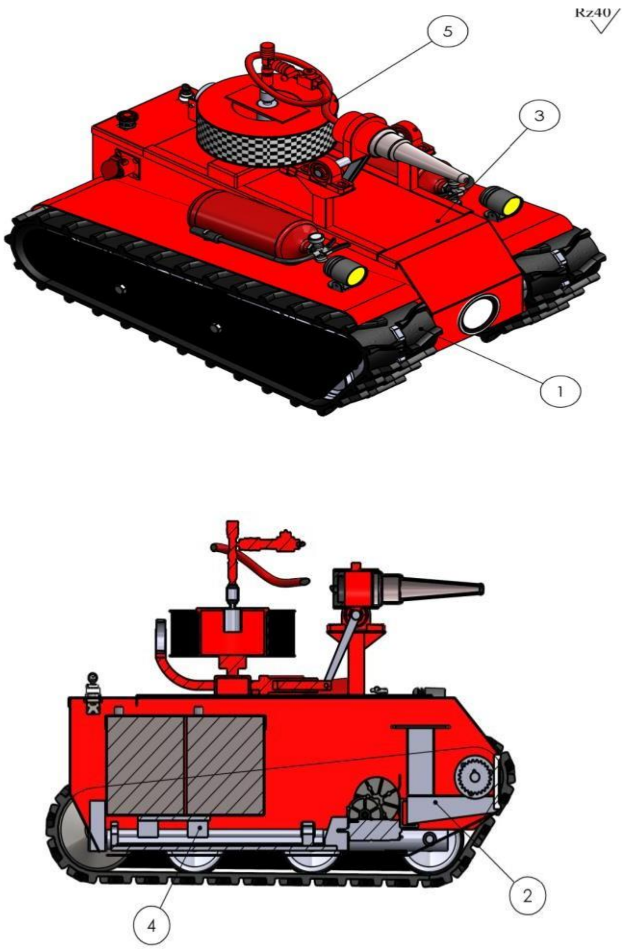

Fig.1. Design of the Automated Firefighting Robot.

(a) (b)

Fig.2. The Firefighting Robot climbs up (a) and down (b)on the upstairs

B. Steering method

The steering method that selected for controlling wheels to drive robot is differential drive, in which a difference of velocities of two wheels will drive robot tracking the required trajectory. The principle of differential driving is given as:

-When two wheels rotates at the same speed and same direction, the robot will move in a straight line.

-When two wheels rotates at the same speed but in opposite directions, the robot will spin in site.

-When one wheel stops, while the other rotates, the robot will spin around the point centered at mid-point of the stopped wheel.

-When one wheel rotates faster than the other, the robot will track a curved trajectory and turn toward the slower wheel.

III. CNN-BASED FIRE DETECTION SYSTEM

In recent years, convolutional neural networks (CNNs) become popular in real-time recognition, object detection and classification [9,10]. CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. In general, a mathematical construct of CNN composed of three types of layers: convolution, pooling, and fully connected layers. The convolution and pooling layers are responsible for feature extraction, whereas fully connected layers map the extracted features into final output, such as classification. A convolution layer plays an important role in CNN, it is composed of a stack of mathematical operations, such as convolution and a specialized type of linear operation. Pixel values in digital images are stored in a two-dimensional grid, such as an array of numbers, and a small grid of parameters called kernel- an optimizable feature extractor, is applied at each image region, then it creates CNNs highly efficient for image processing, since a feature may occur anywhere in the image. As one layer feeds its output into the next layer, extracted features can hierarchically and progressively become more complicated. The process of optimizing parameters such as kernels is called training, which is performed in the direction to minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

Recently there is a lot of approaches has been proposed for fire detection by using CNNs. Early fire detection using fine-tuned AlexNet has been proposed in disaster management with accuracy of 94.39% by Muhammad et al. [6]. Frizzi et al. proposed a CNN-based network for fire detection where the features are simultaneously learned with a Multilayer Perceptron (MLP)- type neural net classifier by training [7]. Muhammad et al. [8] proposed a fire surveillance system based on a fine-tuned CNN fire detector. This architecture is an efficient CNN architecture for fire detection, localization, and semantic understanding of the scene of the fire inspired by the Squeeze Net architecture which achieved an accuracy of 94.43% [8]. Motivated by the recent success of convolutional neural network approaches on fire detection, we integrate pre- trained CNN models into our method.

Advantages of CNNs are: able to (a) directly extract features from data, (b) accomplish state-of-the-art accuracy detection accuracy, (c) retrain the trained network for new detection tasks. However, the main disadvantage of CNNs is that they require huge and diverse data to avoid over-fitting in training the model. For fire detection applications, there is a lack of good datasets consisting of a large collection of images of high dimensionality. Moreover, training from scratch consumes a long time and high memory and the performance is affected by parameter initialization. An effective solution to cope with limitation of annotated data and reduce the training time is using transfer learning to adapt the models to the particular application setting.

Transfer learning is a common and effective strategy to train a network on a small dataset, where a network is pre- trained on an extremely large dataset, such as ImageNet or COCO, then applied to the target task. Transfer learning process has two phases: feature extraction and fine-turning. In feature extraction phase, FC layers are removed from a pre- trained network and the remaining network which consists of a series of convolution and pooling layers, referred to as the convolutional base, as a fixed feature extractor is maintained. In this scenario, any machine learning classifier as well as the usual FC layers, can be added on top of the fixed feature extractor, resulting in training limited to the added classifier on a given dataset of interest. The fine tuning phase, is not only to replace FC layers of the pre-trained model with a new set of FC layers to retrain them on a given dataset, but to fine- tune all or part of the kernels in the pre-trained convolutional base by means of backpropagation. All the layers in the convolutional base can be fine-tuned or, alternatively, some earlier layers can be fixed while fine tuning the rest of the deeper layers. This is performed by an idea that the early- layer features appear more generic, including features such as edges applicable to a variety of datasets and tasks, whereas later features progressively become more specific to a particular dataset or task.

CNNs have a large amount of parameters to be learned. High recognition accuracy cannot be achieved unless a large number of training images are provided. Another problem is the imbalanced distribution of training images. It is labor-intensive and exhausting to generate plenty of training images by capturing or collecting fire pictures. Instead, we produce more training data from the training set by using data augmentation technique. Data augmentation isa technique to increase both the size and the diversity of annotated data sets by leveraging input transformations that preserve corresponding output labels. In computer vision, image augmentations work as regulator to improve performance and to reduce overfitting in deep learning models. While most deep learning frameworks implement basic image transformations, the list is typically limited to some variations of flipping, rotating, scaling, and cropping.

There are various powerful CNNs for object detection have been developed such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD), YOLO, EfficientNet**.** Recently the Google Brain team released EfficientDet model for object detection with the goal of crystallizing architecture decisions into a scalable framework that can be easily applied to other use cases in object detection [9]. It is the successor of EfficientNet, and now with a new neural network design choice for an object detection task, it already the RetinaNet, Mask R-CNN, and YOLOv3 architecture. On evaluating EffiecientDet on the COCO dataset, it achieved mAP(mean average precision) of 52.2 with 9.4x less computation and exceed current State of the art (SOTA) models by 1.5 points. EfficientDet has been confirmed as an object detector that balances model accuracy and detection speed simultaneously [10]. The model structure shown in Fig. 3 consists of the backbone network EfficientNet, the bi- directional feature extraction network BiFPN and the box/class prediction net. Bi-FPN is created by using Bi-FPN layers repeatedly which is similar to Nas-fpn repeating structure approach, the first optimization that Bi-FPN has is the removal of singular input nodes to simplify the model as those nodes would not contribute to feature fusion. Furthermore, it also has a direct input to output node to cut down cost for feature fusion.

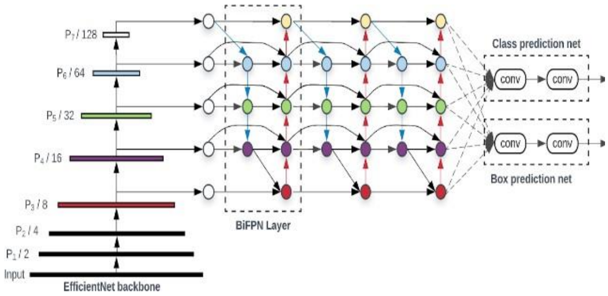

Fig. 3. Architecture of EfficientDet [9]

IV. TRAINING THE EFFICIENTDET MODEL FOR FIREDETECTION

All training processes are performed on Google Colabplatform.

A. *Training Process*

Step 1. Creation of dataset for training.

Diverse and huge dataset is a significant part of deep learning applications. Both Big Data datasets and target dataset are used in the selection of datasets

Big Data: In pre-training stage, dataset is received from COCO 2017 Training and Validating dataset. COCO is a common object in context. The dataset contains 91 objects types of 2.5 million labeled instances across 328,000 images.

***Target Dataset*:** In training stage, we use collected 830 target images containing flames and smoke as depicted in Fig. 4. The target dataset is split into 3 subsets: training subset, a validation subset and test subset. The training set is used in training to train the model as the loss function is calculated by forward propagation and weights are updated by back propagation. The validation dataset is used in training to evaluate for how well the model is generalizing. The test dataset is used after training to test the performance of the final fine-tuning model. All images are resized to 512x512 pixels.

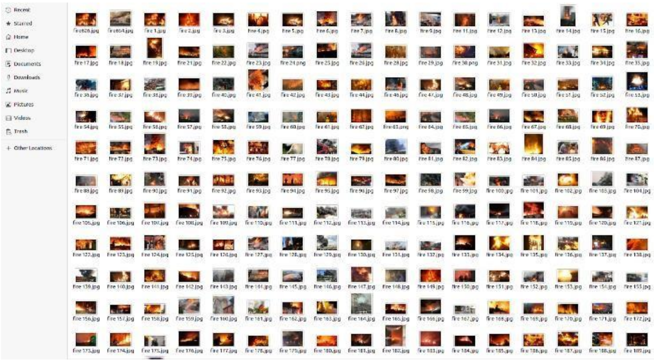

Fig. 4. Collected images containing fires and smokes

Step 2. Label images. The class labeling of the images is very important for the preparation of the dataset. All images are uploaded to LabelImage Tool. Bounding boxes are drawnaround smoke region and fire region in the images and associated them with corresponding class: “smoke” and “fire” as depicted in Fig. 5. A dataset for training is created including images with bounding-box and .xml files, each .xml file provides information about the path to the corresponding labeled image in a directory, size of image frame, label of objects in the image, and coordinates ofbounding boxes. After that the images in the dataset are divided into 3 sub-datasets: 80% for training, the rest is equally distributed for validating and testing. Both datasets (COCO 2017 Training and Validating dataset and the target fire data) are converted into custom TFDS for training modelEfficientDet.

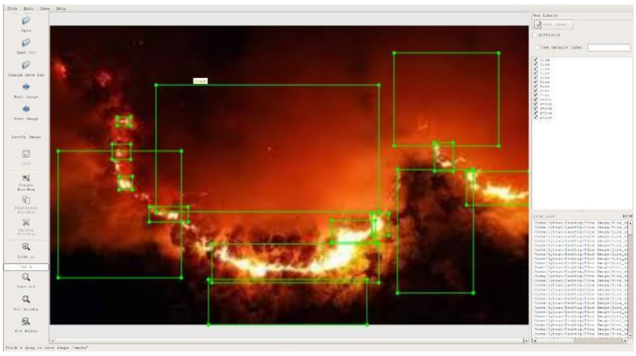

Fig. 5. Labelling smoke and fire regions in an imageStep 3. Data Augmentation

Albumentations- a specialized tool for augmentation is used to integrate with Tensorflow Datasets effectively. The formatPASCAL VOC is used and multiple transformations arecombined together randomly (assigning each transformationa probability p) for augmentation are:

- Image compression with random image quality in the range of 60-100%, performance probability 0.5.
- An order of color channels changes withprobability 0.5.
- Horizontal flip the image horizontally across thecenter axis with probability 0.5.
- Randomly rotate between 0 - 30 degrees left andright with probability 0.4.

- Change brightness and contrast with a maximum factor of 0.35 with probability 0.5.
- Gaussian Blur and Gauss Noise with probability0.4.
- Random crop with size 160x160 with probability0.5.
- Resize the image to 160x160 with probability 1.

Fig. 6. Presents an image after augmentation.

Fig. 6 . An image after augmentation

Step 4. Create base model for pre-training process

The pre-train model EfficientNet is provided with the weights are random initiated. As aforementioned, we usedCOCO2017 Training and Validting Dataset for pre-training.

Then the base model is trained using batch size16 images for 32 epochs with the runtime accelerator of Google Colab set on GPU. The learning rate is set initially at $1e^{-4}$ and maintained throughout the first 8 epochs in order to increase the conversion rate. Then the learning rate follows an exponential weight decay of 0,05 which can be represented as $1e^{-4}0.95t$. During this training process, we also perform checkpoint saving in order to retain the best machine states that have optimized loss.

Step 5. Training. As mentioned, the Fire dataset will also be encoded to TFRecord and apply random augmentation whentraining. But to be able to train with Fire Dataset we will needto change the last class of the classification subnet in the model trained with COCO because the number of classes in the two datasets is different (COCO is 80% and Fire Dataset is 20%). This change will affect training results because this final layer is out of sync with the rest of the model as the baseis unchanged so the weights of this part of the model are stillbeneficial while the weights of the old classifier is rendered and require significant optimization to fit the new classes, and when training the model components altogether, all weights would be updated; this can damage the already useful and active base weights. To solve this problem, we will divide into 2 training phases for the main model:

Feature Extraction: In this procedure, the base model is frozen while the top classifiers are trained for value update in order to be repurposed to optimize weights to fit the new classifiers. First, the classifier was trained using the training set of the fire data set, after has undergone data augmentation, which be divided into batches of 32 images and trained for 35 epochs with a similar learning rate as pre- train process set initially at $1e^{-4}$ and maintained throughout the first 8 epochs in order to increase the conversion rate. Then the learning rate follows an exponential weight decay of 0,05 models value during training. Fig. 7 demonstrates both loss function in the validation dataset (blue line) and theloss function in the train loss (red line) reduces significantly from 15th epoch, that means the model starts to adopt the newclassifier. From 15th to 50th epoch, the loss functions still reduce but slower, that indicates the model has adapted well to the classifier and then we can transfer to the fine-tuning phase.

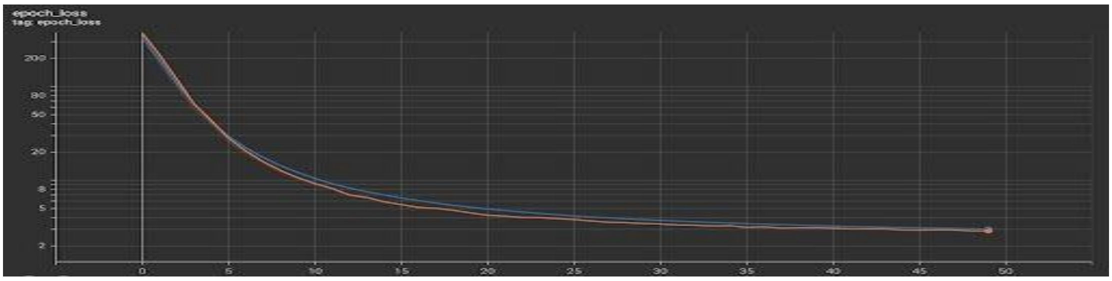

Fig. 7. Loss function in feature extraction process

Fine-Tuning: In this process, we unfreeze the base model and connect it to the classifier that has trained in feature extraction. Then we train the whole model to fine-tune the higher-order feature representations in the base model. The weights will be update from generic feature maps to featuresassociated specifically with the dataset. The batch size is of 16 images and learning rate of initially set at $4.16e^{-3}$.

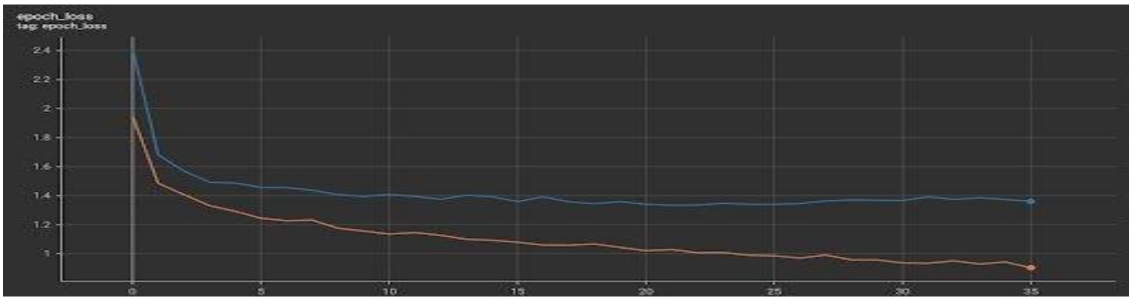

Fig. 8. Loss function in fine-tuning process

It is possible to see the stability of the loss functionsin fine-tuning phase in Fig. 8. After training process, we receive a detection model with trained weights for detecting and identifying smoke and fire regions in images. Next, wetransplant the trained model to Jetson Nano for test process.

B. Visualization Results

In order to confirm the effectiveness of the trained model in fire detection, we input some images in test datasetinto the model. The results have been shown in Fig. 9. It canbe seen that fire areas are box-bounded by green rectangles with number “1” and confidence score, the smoke areas are box-bounded by rea-rectangles with number “2” and confident score. It is possible to realize that the confidence scores of fire detection are higher than those of smoke detection, it may from the color of smoke is similar to color of surrounding environment. Almost smokes and fire regionsare detected already in images of test dataset. The rate of detection is 12 frames per second. The good visualization results are motivation to integrate the CNN-based detection model with the vision-based control unit of the firefighting Robot in the next study.

Fig. 9. Visualization results on some images of test dataset. Fire areas are box-bounded by green rectangleswith number "1" and confidence score, the smoke areas are box-bounded by rea-rectangles with number "2" andconfident score.

V. CONCLUSIONS

The paper presented a creation of CNN-based model towards the detection of fire and smoke to provide visualization information to firefighting robots. This work utilized EficientDet for carrying out the detection tasks. Transfer learning has been utilized to learn feature from data and fine-tuning on the performance of fire flame and smoke detection due to collected data on fire is limited. The pre-trained model EfficientNet is trained on a very large-scale dataset COCO2017 to provide a feature extractor with whichwe can feed our image dataset through the network for extracting features at a given layer. Fine-tuning involves a network modification, the new set of FC layers is trained using a very small learning rate until they can learn high-level features and empower identification ability. Experiment results have verified the effectiveness of the achieved model in fire and smoke detection and assist to integrate to the vison-based control unit of firefighting robot. In the next study, we intend to explore the state of the art detectors and novel algorithms to enhance the accuracy of fire and smoke detection

ACKNOWLEDGMENT

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2022-PC

REFERENCES

i. *B.U. Töreyin, Y. Dedeoglu, U. Güdükbay, A.E. Cetin, "Computer vision based method for real-time fire and flame detection", Pattern Recogn. Lett. 27 (2006) pp. 49–58.*

ii. *Rahman, M. Murshed, "Detection of multiple dynamic textures using feature space mapping", IEEE Trans. Circuits Syst. Video Technol. 19 ,2009, pp. 766–771.*

iii. *T. Qiu, Y. Yan, G. Lu, "An autoadaptive edge-detection algorithm for flame and fire image processing", IEEE Trans. Instrum. Meas. 61 , 2012, pp. 1486–1493.*

iv. *C.-B. Liu, N. Ahuja, "Vision based fire detection", Proceedings of the 17th International Conference on Pattern Recognition, 2004,pp. 134–137.*

v. *Khan Muhammad, Jamil Ahmad, Zhihan Lv, Paolo Bellavista, Po Yang, Sung Wook Baik, Member, "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications", IEEE Transactions on Systems, Man, and Cybernetics: Systems PP(99), March 2018, pp. 1-14.*

vi. *K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks* *during surveillance for effective disaster management," Neurocomputing, vol. 288, pp. 30- 42, 2018/05/02/ 2018.*

vii. *Frizzi, S.; Kaabi, R.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Fnaiech, F., "Convolutional neural network for video fire and smoke detection", Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; IEEE: Piscataway, NJ, USA, 2016;pp. 877–882.*

viii. *Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik S.W., "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications", IEEE Trans. Syst. Man Cybern. Syst.* ***2018****, 99, 1–16.*

ix. *Tan M, Pang R, Le Q V, "EfficientDet: Scalable and Efficient Object Detection," Proceeding of the IEEE Conference on Computer Visionand Pattern Recognition (CVPR), 2020, pp.10778-10787.*

x. *M. Y. Cao, J. Zhao, "Fast EfficientDet: An Efficient Pedestrian Detection Network", Engineering Letters, 30:2, EL_30_2_18, Volume 30, Issue 2: June 2022.*

xi. *Chao Liu and Shouying Lin, " Research on Mini-EfficientDet Identification Algorithm Based on Transfer Learning", Journal of Physics: Conference Series, 2022, pp.1-9.*