6-DoF Occluded Object Semantic Grasp Planning with De-occlusion Instance Segmentation

Zhongming Huang*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
reavenhuang@ieee.org

Shangyun Yang
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
yangshangyun@tiangong.edu.cn

2022 5th International Conference on Intelligent Autonomous Systems (ICoIAS) | 978-1-6654-9838-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOIAS56028.2022.9931318

***Abstract*—Thanks to previous research, computer vision has now endowed 6-DoF robot arms with the intelligence to plan its best trajectory to a given target. One of the problems that the 6-DoF semantic grasp planning still faces is the solution to grasping occluded targets in a cluttered scene. In such a scenario, the robot arm may not be available to grasp the target in one shot. The current potential attempts to grasp an occluded object include re-arranging the cluttered scene in the hope to expose the target object for a more viable trajectory. However, such methods require multiple viewpoints to model the scene for making global re-arranging plans. Also, the process of re-arranging would take considerable steps to achieve. In our work, we use our decision-making algorithm to combine occlusion prediction (BCNet) with grasp pose planning algorithm (GraspNet), enabling the robot arm to understand the relative positions of each object in a scene. Our method is based on a single viewpoint so that we do not require the robot arm to look all around the scene. In addition, our method is target-motivated, which means we only grasp relevant objects instead of re-arranging all objects in the scene, providing a novel efficient solution to grasp occluded targets.**

Keywords-6-DoF Semantic Grasp Planning, Instance Segmentation, Occlusion Prediction, Robotic Grasping, Vision and Perception.

I. INTRODUCTION

Semantic grasping planning for 6-DoF robot arms has been a rising topic of robotic intelligence, this is a process involving RGB-D depth image processing, object localization, grasp pose generation, and trajectory planning. Current works [1] have achieved robust algorithms to solve the problem that how the robot could understand the layout of a certain scenario, as well as how to approach the target object in a correct movement and position. So, that overcomes the barriers of the mentioned processes, everything seemed to be a no problem. But is this pleasure a real relief to researchers? The answer is negative, after the applications of semantic grasping expanded beyond the laboratory environment, reaching real-world occasions. One difference between pure lab environments and practical situations is the layout of the scene where the robot seeks for a target to conduct grasps. In real-world scenes, objects are often arranged tightly to each other, consisting of a cluttered layout which is difficult to conduct traditional semantic grasping [2] such scenes are shown in Fig. 1. Several obstacles are contributing to the difficulty for robots to conduct a successful semantic grasp within such cluttered scenes: a) the difficulty to spot the target object given the target has been occluded by obstacles; b) the unavailability for the robot arm to directly approach the target successfully. To solve the problem, previous researchers have made great efforts such as re-arranging the scene to make the layout more sparse and convenient for a robot arm to move around [3]. However, such a method requires a great amount of robotic observation around the layout, which involves a heavy load of computation since the robot has to reconstruct the whole scene before determining the movements of each object in the scene. In addition, this re-arranging process also requires a considerable number of grasp movements to place each object in the expected location. Admittedly, such re-arranging methods bring advantages to other applications such as robotic tidying, but it is not an optimal choice to simplify semantic grasping in a cluttered scene.

Figure 1. Real-world grasps scenes. Images are by courtesy of GraspNet-1Billion dataset. Occluded target objects in the above four scenes are indicated by yellow arrows: a) the toothpaste package; b) the red cup; c) the electric drill; d) the white bottle.

In our work, we are about to try another method to enable the robot arm to reach its target and make a successful grasp. We focus on removing the objects which directly occludes

our target, instead of trying to divide and re-arrange all objects in the scene. By the word removing, we mean grasping and placing the occluding objects out of the scene. To have our robot understand the spatial relationship of the scene without exhausting observance, we make use of RGB-D data from one camera viewpoint to exploit the depth relationship while segmenting objects and predicting occlusion areas. In the actual implementation of grasping, we adapt a grasp pose detection network [4] to provide potentially viable grasp positions and gestures on each object in the scene. To select the best next grasp, which could be the occluding objects or occluded target, we design a decision-making algorithm. This decision-making algorithm makes use of both the spatial and occlusion relationships from the segmentation-and-occlusion-prediction network (BCNet) [5] and the grasp confidence of the grasp detection network, to generate scores of each object in the scene. The score given by our algorithm is an overall estimation of the viability of a grasp and the profit the grasp can bring after removing the object. Our algorithm always chooses to grasp the object with the highest score until it grasps the target object.

In our decision-making algorithm, we gather parameters from the previous two networks and weigh them to conclude the best next grasp. This decision-making algorithm is always pursuing the optimal grasp on the occluding object and the target, considering factors including the pose confidence, and the area of occlusion. When there is no direct reliable grasp pose on the target, our decision-making algorithm will try to manipulate the robot to grasp the best occluding object until the target is graspable in one shot. We adjusted our decision-making algorithm on the CoppeliaSim virtual grasping environment to finetune the parameters. After that, we tested our model on randomly generated cluttered scenes with the target object occluded and harvested over 4 times success rate improvement. To discuss our criteria to mark the object as the "best next grasp" as well as our standard for an efficient grasp, we will expand our designs in Section III.

II. Related Works

A. Visual Grasp Pose Detection

The most important approach for a robot arm to percept the surrounding environment is by looking around, which has evolved a lot before we can now generate 6-DoF grasp poses [6]. The easiest method for a robot to see the scene with objects to be grasped is pure camera video streams. Since the video streams consist of consecutive RGB image frames, the robot is not able to feel the cubical appearance of objects. Hence, the grasp pose proposals generated from such perception are almost 2D poses. This means the robot can only look from above and vertically at the scene. Redmon and Angelova (2015) [7] trained a single-stage CNN to detect 2D planar grasps and conduct object classification at the same time. Such a method is fast and is suitable for robot arms that are not so flexible. However, observations from such planar images are not sufficient for robots to judge whether the edges of the proposed grasp poses are indeed reliable for the robot to grasp tight. Also, planar grasp poses proposals will limit the possibility for robots to approach target objects from a more viable trajectory since the robot arm being given a planar grasp pose can only grasp vertically from the above. Zhang et.al. (2017) proposed [8] a multi-modal fusion approach to provide 2D grasp poses with the combination of RGB images and depth data. The aid of depth information is critical to understanding the accurate margins and the shape of edges before proposing poses, and this expanded its application from a lab environment where pure backgrounds are guaranteed to many real-world polychrome environments. Moreover, depth data also acts as a new criterion when detecting grasps on strange or unknown objects.

As robots with higher flexibility gradually prevail, mainstream researches focus on how to give poses in a more natural and unlimited manner. Such pose detection methods generate 6-DoF poses from RGB-D data and allow robot arms to approach the targets in various trajectories. Especially in its application in dense clusters, robots can benefit from the variety of approach angles since 2D grasps may not be practical. Gualtieri et.al. (2016) [9] proposed a 6-DoF grasp pose detection algorithm that is capable of scenes with cluttered objects, the robot in their work achieved over 90% success rate when working in active mode to eliminate the clutters. We choose to adapt from GraspNet [4], which also generates 6-DoF grasp poses, but on a more general scale. Since GraspNet was pre-trained and tuned on various objects and over the cluttered scene, it has shown advanced generality to real-world scenarios, providing a wider range of objects that could be grasped.

B. Occlusion Prediction Methods

Other factors that inspired robotic grasping in occluded scenarios are the advancements in occlusion prediction algorithms. Occlusion prediction, also known as de-occlusion methods, is now more of a branch under instance segmentation algorithms. Before the prosperity of instance segmentation algorithms, Chen, X., & Yuille, A. L. (2015) [10] tried to de-occlude human arms in collective photos based on the specific features of a human torso, which is not dependent on the segmentation of different people. Such de-occlusion methods are based on the abstract modeling of specific structures and are not general enough to be transferred to common object de-occlusions. As to the aspect of occlusion prediction with instance segmentation, Chen, Y.-T., Liu, X., & Yang, M.-H. (2015) [11] conducted an early try on a scene of multiple objects. The occlusion area predictions are based on the margins of segmentation, and the network will reason the overlapping area pixel-wise after learning human-labeled annotations which are also masks at the pixel level. By predicting the occlusion areas, the original object being occluded in the scene can be reconstructed. Zhan et.al. (2020) [12] also pointed out the possibility of comprehending the spatial relationship through de-occlusion and reconstructing each segmented object in the scene. We choose BCNet proposed by Ke et.al. in 2021 [5], it is a bi-layer occlusion prediction network, where two parallel networks are predicting the original margins of the two

objects in the front scene and the background. Its leading performance in correctly predicting the margins of small overlapping areas brings benefit to our grasp scenes where small objects are dominant.

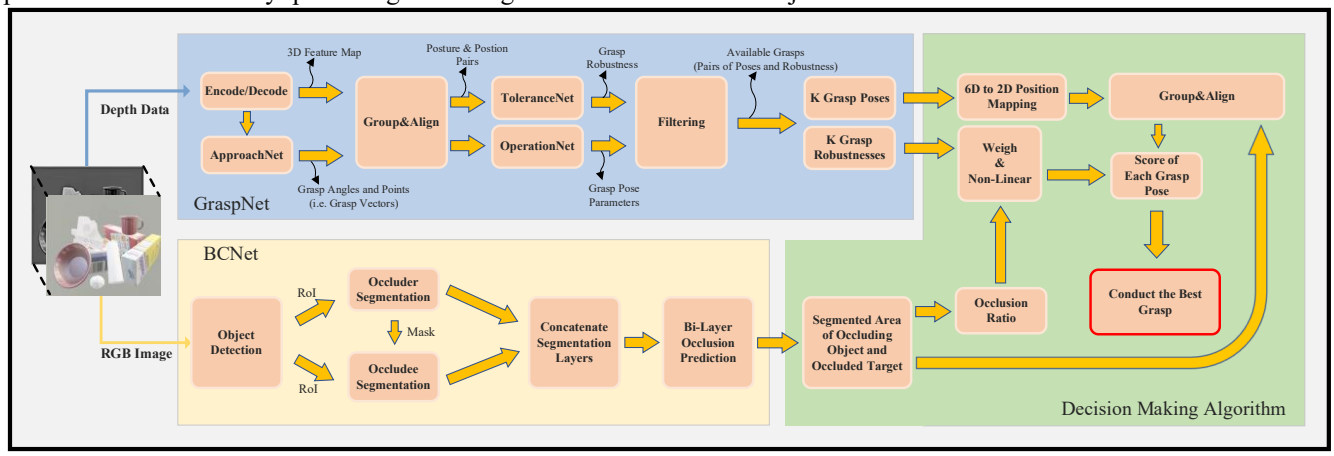

Figure 2. Overview of Our Algorithm. The input is an RGB-D image collected from the fixed camera in the simulated scene. The GraspNet and BCNet will process the depth channel and the RGB color channel respectively. The grasp implementation is determined by our decision-making algorithm. The black twisted arrows indicate the output data from a certain step.

III. Our Method

A. Overview

The general workflow is shown in Fig. 2, we use BCNet and GraspNet as our parallel front ends, and we extract some of the variables from the two networks as our criteria for deciding the next best grasp.

B. Grasp Pose Detection

To endow our model with the ability to detect grasp poses, we adapt GraspNet [4], a novel and precise algorithm. And we utilize its benchmark GraspNet-1Billion as our dataset where we adjust parameters and test our algorithm. The GraspNet uses PointNet++ [13] as its backbone network and is capable of processing RGB-D data captured from the object scene and exploiting information from both the RGB image channels and the depth channel. It was pre-trained and evaluated from massive 3D object models and has shown preferable performance in pose richness and speed when tested on the cluttered object scene. To feature on the actual test result, this network brings up to 29.88 average precision which is leading among other algorithms being compared [4]. As shown in Fig.2, the point cloud depth image is first encoded and decoded into a 3D feature map, and then passed to ApproachNet to detect possible grasp approach angles and points (approach vectors described in [4]). After that, the approach vectors are grouped with the correlating feature points from the previous 3D feature map, at this step, each approach vector points at a certain feature point. In the next step, the approach vectors are evaluated parallelly by OperationNet and ToleranceNet, which respectively predicts the grasp pose parameters and their grasp robustness. Finally, it outputs filtered grasps poses.

In our work, we do not need the robot to directly conduct grasps based on those poses. We here collect some important parameters and the final grasp decision will be based on our choice of the next best grasp (NBG) in section D. We first run it to detect every possible grasp in our scene, assume there are G_{global} viable grasp poses after filtering, upon all the objects detected. And we make use of the output grasp robustness R, *sized* $1\times G_{global}$ which is aligned with the viable G_{global} grasp poses. We temporarily save the correlating grasp pose parameters P_{global}, *sized* $6\times G_{global}$, for a further selection of NBG.

C. Occlusion Prediction

To understand the occlusion relationship, we need another network that is capable of instance segmenting and occlusion prediction. BCNet [5], an efficient and reliable new method to help with understanding the exact margins of each object and estimating the overlapping area of different objects in an image, comes into our sight. This algorithm intakes RGB image data and outputs segmentations of each object as well as the predicted margins of occluded objects. We will combine depth data and its output margins to help the robot understand which objects are occluding the target and how much we can expose the target's occluded area by removing one of the occluding objects. The area that could be exposed by removing occluding objects helps indicate the necessity of conducting a removing grasp in our decision-making algorithm.

When the RGB image is input to BCNet, this network first detects the region of interest (ROI), and further procedures are based on the clipped ROI out of the whole image. The ROI is processed to predict the occluding objects (occluder) first, and the shape and margin of the occluder will help predict the occluded objects (occludee). Finally, the two layers of occluder and occludee will be added together. We need the pixel-wise segmentation of each object, and more than that, we need the overlapping pixel number of different object segmentations. Let's assume the area of ROI to be A_{global} sized $W\times H$, where W and H are equal to 28, as the width and height of the ROI in pixels. And we assume

A_{masks} sized $N_{obj} \times W \times H$, where N_{obj} is the number of segmented objects and the element of A_{masks} is 0 or 1, we can find the segmentation of the i[th] object by looking for element 1 in $A_{mask}(i)$. We first obtain the pixel-wise area A_{object} sized $1 \times N_{obj}$, of each segmented object:

$$A_{object}(i) = \sum_{m=1,n=1}^{W,H} A_{masks}(i,m,n) \tag{1}$$

We then derive an occlusion area matrix A_o sized $N_{obj} \times N_{obj}$ from the equation below:

$$A_o(i,j) = \sum_{m=1,n=1}^{W,H} (A_{masks}(i,m,n) + A_{masks}(j,m,n) - 1) \tag{2}$$

In (2), $A_o(i,j)$ represents the overlap pixels of object i and j, and it is a symmetrical matrix.

D. Decision-making Algorithm

Since we have obtained the occlusion area matrix A_o, the pixel-wise area of each object A_{object}, we can further determine the occlusion ratio O, sized $N_{obj} \times N_{obj}$:

$$O(i,j) = \frac{A_o(i,j)}{A_{object}(j)} \tag{3}$$

In (3), $O(i,j)$ describes the ratio of the occlusion area resulting from the i^{th} and the j^{th} object to the area of the j^{th} object. And we always have $O(i,j) \le 1$. O is an important indicator about the exposed area it can bring if we grasp a certain object, leaving the behind object un-occluded.

We then determine the target object and those objects occluding. If our target object is the m^{th}, we can find all the objects occluding it by calling:

$$O(i,m) \ge \partial \tag{4}$$

In (4), ∂ is a lower limit that we will finetune in Part IV. Each i that satisfies (4) will indicate an occludee indexed i^{th}. We call the object occluding our target the Related Object (RO).

To know which object the grasp poses in section A belong to, we have to map the 6D grasp poses to the planar (2D) space of the RGB image. Assume $t \in R^3$ represents the grasp pose spatial location $t(x,y,z)$, and $t' \in R^2$ to indicate the corresponding planar position $t'(w,h)$ from the RGB image viewpoint, we have the following transformation formula:

$$\begin{aligned} w &= (t - C) \cdot (-sin(\theta), cos(\theta), 0); \\ h &= (t - C) \cdot (cos(\theta) \cdot sin(\phi), \\ &\quad sin(\theta) \cdot cos(\phi), \sin(\phi)) \end{aligned} \tag{5}$$

In (5), we have the following quantities:

$$\begin{aligned} C &= (\cos(\theta) \cdot cos(\phi), sin(\theta) \cdot cos(\phi), \sin(\phi)) \cdot r; \\ r &= |t| = \sqrt{x^2 + y^2 + z^2} \end{aligned} \tag{6}$$

Note that angles θ and ϕ are learned from the camera viewpoint angle. Now, we have obtained the grasp pose location $t'(w,h)$ in the image plane. The grasp of a certain object can now be called searching A_{masks}, and $A_{masks}(i, w, h)=1$ means the grasp pose belongs to the i^{th} object.

Once we have obtained RO's ratio of the occlusion area, we can establish our scoring formula which takes into consideration the grasp robustness R and the ratio of occlusion O, and we use S (sized $1 \times N_{obj}$) to represent the output score:

$$S(i) = \frac{1}{1 + e^{-[O(i,m) \times 5 - 2.5]}} \cdot R(i) \tag{7}$$

In (7), the target object is indexed m, and our next best grasp (NBG) is defined by the object with the highest score. The de-occlusion grasp process is illustrated in Fig.3.

Figure 3. The Process of De-occlusion. As shown in sub-figure (a), A, B and C are the three occluding objects (occludees) indicated by yellow arrows and marks; the target object is the blue can indicated by a green arrow and mark. The occlusion areas of A, B and C are predicted and indicated by OA, OB and OC which are illustrated in purple, cyan and green respectively. Our decision-making algorithm will determine to grasp object A first.

IV. Experiment

A. Dataset Preparation

To train the BCNet to be more adapted to real-world grasp scenes, we synthesized 500 occlusion scene images for its transfer learning. Our synthesized images are based on combinations of basic object models from the GraspNet-1Billion dataset [4]. Each synthesized image has real and complete object contours of occluded and partially occluded objects, allowing explicit modeling of occluded relationships between occluded regions and occluded objects. We use this synthesized dataset to complete the training of BCNet, to better judge the effect of overlapping areas, and make our network prediction more accurate.

For parameter tuning and model testing, we use a grasp simulator to generate random scenes based on object models in [4]. These objects are suitable for the size of the grippers, and in terms of shape, texture, size, material, and diversity. And this object dataset can better simulate the real-world grasps scenes. We use these objects in this dataset to create a more intensively occluded scene by generating dense clutters. In each scene, we randomly put in 6 to 10 everyday objects. Table I shows the basic information of our dataset.

TABLE I. Dataset Information

Synthesized Occlusion Images	500
Occlusions / Image	~4
Objects / Grasp Scene	6~10
Scenes / Test Epoch	100

B. Environment Setup

Our grasp experiment is based on CoppeliaSim V4.2.0 rev5, where a virtual grasp environment is provided, with the BCNet framework based on PyTorch 1.4.0. And we select the UR5 robot arm as our grasp actuator. The computation is based on an RTX2080 GPU and Intel i7 CPU. In the experiment, the camera viewpoint is fixed from an angle where the sight is inclined downwards, the robot will execute grasps based on the perception from the camera.

C. Parameter Finetuning

We first finetune BCNet, which achieved 31.75 AP after the network was sufficiently converged. Compared with other image segmentation networks, our finetuned BCNet showed stronger sensitivity in predicting overlapping regions. We expect that it can be used to accurately judge the area between occlusions and target objects and to lay a foundation for the subsequent adjustment of model parameters.

We adapted the GraspNet framework and its pre-trained parameters, and we finetune the parameter ∂ in our decision-making algorithm based on the pre-trained GraspNet. We test different values of the parameter ∂ in the simulated grasp scene, and of each value, we test it on 100 random scenes as shown in Table I. The following Fig.4 shows the tuning process of the parameter ∂.

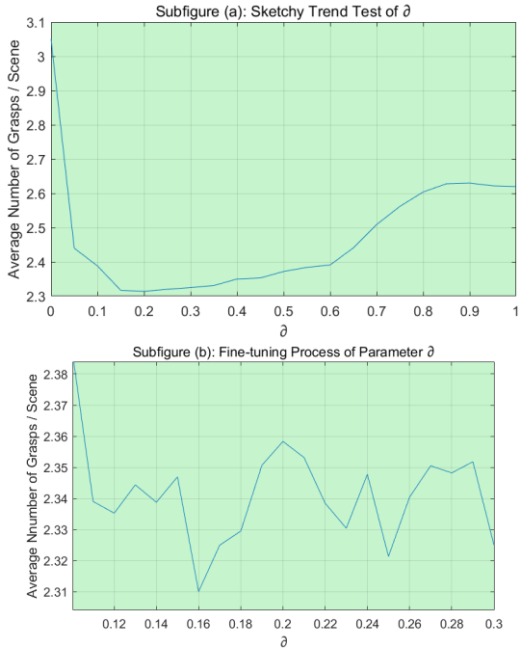

Figure 4. The Finetuning of Parameter ∂. ∂ is the lower limit in (4), and this is critical in deciding whether it is necessary to grasp an object with a small occlusion ratio O. Subfigure (a) indicates the overall trend of the average number of grasps when changing ∂; subfigure (b) shows the finetuning of ∂ within the selected range 0.1~0.3.

We believe that in the process of deciding on a grasp-worthy occluder, there is a threshold value ∂ related to the occlusion ratio that is critical to grasp efficiency. On the one hand, if ∂ is too large, our decision algorithm will not grasp occluders that have already occluded a big area. On the other hand, a too small value of ∂ means grasping small occluders which does not preclude the process of grasping the target. To finetune the value ∂, we divided the test epochs into 3:2. The first part occupying 3/5 of the dataset was used for tuning value ∂, and the second part was used to test the universality of our tuned model.

The average grasps per scene are the average grasp movements the robot completes until finally grasping the target object, including the grasps in the de-occlusion procedure. As Fig. 4 indicates, when ∂ is too small, the robot will be prone to grasp redundant objects, resulting in a greater number of grasps per scene; but when the threshold ∂ is assigned a too big value, it will lead to grasps directly to the target object with poor robustness. After finetuning the threshold ∂ in a selected range, we yield the minimum grasps per scene 2.31 at the ∂ value of 0.16.

D. Grasp Efficiency Test Results

TABLE II. Grasp Efficiency Test Results

Average Success Rate of Grasping the Occluded Object (Before and After De-occlusion)	Before	**After**
	21.7%	**98.4%**
Average Grasp Movements per Scene	2.31	
Average Grasp Movements per Occluder	1.29	

We tested our model based on 10 test epochs indicated in Table.1. We noted the average success rate of grasping the occluded target object without and with the aid of our decision-making algorithm and BCNet occlusion prediction. And we yield a 98.4% success rate after removing necessary occluders, which is around 4.5 times the success rate without de-occlusion movements. This feature is particularly important in facilitating the overall grasping speed in real-world cluttered scenes. Under a cluttered test scene with up to 10 objects (indicated in Table.1), our average grasp movements number in each scene is 2.31, which means we are mostly grasping necessary objects, instead of re-arranging the whole scene which involves a great number of redundant grasping.

V. Discussion

In our model, we combined two novel networks using our decision-making algorithm to decide the next best grasp (NBG) in de-occlusion semantic grasping scenarios. The BCNet has endowed our model with the ability to predict occlusion areas, which is an important criterion for determining the NBG. While making the decision, our model also considers the robustness of grasps, and we extract the grasp robustness from GraspNet. We finetuned our model parameters in the CoppeliaSim virtual environment and harvested considerable grasp efficiency. Compared to other grasp planning methods, our model does not require the robot to re-arrange or remove all the objects in the scene, which shortcuts redundant movements. Also, since our model is worked on a single camera viewpoint, it does not need exhausting observation over the whole grasp scene. However, our model still faces several challenges. Our model is not able to detect more than two occluding objects and is not fully capable of understanding occlusion from the above. We consider this model can better understand the spatial relationship of objects by combining the point cloud data with the instance segmentation masks. We are hoping to conduct further research to better consummate our model.

VI. Conclusion

In our research, we are aimed at enhancing the grasp efficiency in planning to grasp occluded objects. Before grasping the target object, the robot is required to grasp and remove the occluding objects to create convenience for grasp pose detection on the target object. In the process of planning to grasp and remove occluding objects (occluders), we propose a decision-making algorithm to determine the next best grasp (NBG). Our decision-making algorithm considers both the robustness of a proposed grasp pose and the necessity of grasping one occluding object. To integrate BCNet and GraspNet which take care of occlusion prediction and grasp pose detection respectively, we conduct a mapping process that transfers the spatial grasp pose location to the image planar where the objects are segmented. At last, we finetune and test our model on the simulated grasp environment and harvest great grasp efficiency enhancement. Our work is greatly inspired by the real-world grasping obstacles that the robots are faced with. And we look forward that our work providing a little reference for further studies on grasp planning in dense clusters, especially those scenarios where the targets are deeply occluded.

References

[1] Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., & Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. IEEE Transactions on Robotics, 33(6), 1273-1291.

[2] Du, G., Wang, K., Lian, S., & Zhao, K. (2021). Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review, 54(3), 1677-1734.

[3] Murali, P. K., Dutta, A., Gentner, M., Burdet, E., Dahiya, R., & Kaboli, M. (2022). Active visuo-tactile interactive robotic perception for accurate object pose estimation in dense clutter. IEEE Robotics and Automation Letters, 7(2), 4686-4693.

[4] Fang, H. S., Wang, C., Gou, M., & Lu, C. (2020). Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11444-11453).

[5] Ke, L., Tai, Y. W., & Tang, C. K. (2021). Deep occlusion-aware instance segmentation with overlapping bilayers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4019-4028).

[6] Du, G., Wang, K., Lian, S., & Zhao, K. (2021). Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review, 54(3), 1677-1734.

[7] Redmon, J., & Angelova, A. (2015, May). Real-time grasp detection using convolutional neural networks. In 2015 IEEE international conference on robotics and automation (ICRA) (pp. 1316-1322).

[8] Zhang, Q., Qu, D., Xu, F., & Zou, F. (2017). Robust robot grasp detection in multimodal fusion. In MATEC Web of Conferences (Vol. 139, p. 00060).

[9] Gualtieri, M., Ten Pas, A., Saenko, K., & Platt, R. (2016, October). High precision grasp pose detection in dense clutter. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 598-605).

[10] Chen, X., & Yuille, A. L. (2015). Parsing occluded people by flexible compositions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3945-3954).

[11] Chen, Y. T., Liu, X., & Yang, M. H. (2015). Multi-instance object segmentation with occlusion handling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3470-3478).

[12] Zhan, X., Pan, X., Dai, B., Liu, Z., Lin, D., & Loy, C. C. (2020). Self-supervised scene de-occlusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3784-3792).

[13] Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 30.

2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML) | 978-1-6654-9950-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/PRML56267.2022.9882243

Dynamic Feature Extraction Using I-Vector for Video Fire Detection

Zhongming Huang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
reavenhuang@163.com

Yuxiang Wang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
Euson@outlook.ie

Haolan Hu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
haolanhu_2019@163.com

Xun Liu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
vickibaileyraymond@gmail.com

Tongzhen Liu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
liutongzhen0311@126.com

Zhanxu Zhang
School of Computer Science and Software
Tiangong University
Tianjin, China
zhangzhanxu23@outlook.com

***Abstract*—Fire detection technology has been researched and developed for decades. However, in videos and complex scenes, it still lacks fast recognition of fire's existence. The traditional model of fire recognition still needs a large number of samples and time-consuming machine learning progress. Meanwhile, the uncertain shape of fire leads to the reduction of accuracy using CNN. Based on the above problems, we establish a novel method based on I-Vector. We use an adapted I-Vector algorithm to extract the time sequence feature vector on the fire and its surroundings and train a G-PLDA classifier to recognize the dynamic occurrence of fire more quickly and accurately. This model requires fewer samples and a shorter learning time while obtaining an accuracy similar to the traditional fire recognition models, which provides a new effective solution for rapid analysis of whether there is a fire in the video scene. In addition, the algorithm has a sound universality and is easy to deploy in the application fields of video fire supervision, UAV fire inspection, and other related fields.**

Keywords—video fire detection, pixel sampling, sequence feature extraction, signal processing, I-vector

I. INTRODUCTION

In recent years, visual recognition has made great progress in both theoretical and practical applications. Among them, popular applications are progressing in face recognition [1], fingerprint recognition [2], fire recognition [3] and other related fields. Most of the methods for fire recognition in videos are based on convolutional neural networks (CNN) [4], which classify and detect images or videos input by the network, and use frame-by-frame detection methods for videos to detect whether a fire exists. However, this method cannot find out whether there is a fire in a video at the first glance, and it is followed by many confinements including an enormous dataset, time-consuming learning period, and unreliable recognition results due to the uncertainty of shapes of fire, as shown in Fig. 1. In dealing with the problems, it is necessary to develop a video fire detection algorithm that is not dependent on static fire morphology features. After long-term observation, we discover that the flickering of a fire has certain dynamic characteristics when burning. We come up with the idea of developing a feature extraction algorithm to look into the dynamic patterns of fires. When a video is separated into pixels along the time domain, each pixel will change its value as the frame goes with timeline. This means we can cut a video along the time domain, resulting in time sequences of correlating pixels. For pixels on the fire, they carry unique patterns of brightness change. Our algorithm uses the greyscale time series extracted from a video at the fire pixel points and the surrounding environment pixels to learn to differentiate unique patterns of fires. Extracting features from time sequences instead of performing frame-wise convolution enables the model to quickly find fires in videos. With a proper selection of feature extraction algorithm, our method would be lightweight and less dataset-dependent compared with CNNs, making it conducive to fire detection.

Before I-Vector is selected as our baseline feature extractor, we have compared it with other two commonly used time series feature vector extraction algorithms. But it turns out that I-Vector has the best adaptability on fire detection and is easier to use. We use the I-Vector algorithm [5] as a baseline to establish the model, where I-Vector is originally a kind of time-series feature processing algorithm commonly used in speaking verification technology and speech analysis. We first establish a dataset containing 1298 time sequences in greyscale, in which fire samples (positive samples) take up 3/4. Then an I-Vector extractor and a G-PLDA classifier are trained one after another. This method reduces dataset size and computational complexity, while keeping a performance similar to that of a CNN. And it is found that our model has a shorter learning period, faster processing speed, and decent accuracy. From the test results, our model can effectively and quickly decide whether a fire is in a video, which contributes to the rapid solution for problems caused by fire based on video records. Due to its good adaptability, our algorithm is also relatively stable.

Our novel time-series feature extraction algorithm based on I-Vector can be used as a supplement or even an alternation for

This is part of the research under the Tianjin Provincial University Student Innovation and Entrepreneurship Program (Project No.202110058107), which all the authors are affiliated to.

current fire recognition algorithms. The fire recognition model established by our algorithm can be widely used in fire spotting in videos, fire time confirmation in monitor records, and other fire detection within videos.

Fig. 1. Fire scenarios: Fire or fire does not have a fixed shape, but the dynamic feature during combustion can be extracted.

II. Related Work

CNN: Convolutional neural network has been widely used in various target recognition scenarios. The mainstream method of CNN to recognize fire is almost to build a neural network and learn the dataset which includes images labelled as fire or non-fire. After repetitive training, a model that can recognize static fire is finally obtained [6] [18]. Arpit Jadon et.al. [7] proposed a specially modified object detection CNN called FireNet for fire recognition, and Shixiao Wu et.al [8] compared performances between many prevailing CNNs tuned for fire detection. Although mainstream convolution neural network models have the potential to achieve high accuracy in recognizing fires in images, they still have limitations. To discuss drawbacks of fire detection CNNs, we classify them into two types. First, for the Object Detection Networks [9], by inputting images or videos, it will output a bounding box confining the fire. However, the accuracy of the algorithm is limited by the scale of dataset. And their work also indicates high dataset dependence and computational complexity. Second, for the Instance Segmentation Networks [10], by inputting images or videos, it will output pixel-wise segmentations for objects. However, it would take great efforts to label the training data. So far, it still lacks of dataset and practical applications in fire detection. Moreover, in practical applications, fires are continuously varying in shape. But CNNs perform frame-by-frame detection, leading to their inability to utilize dynamic features of fires.

Time series processing: To find a feature extractor best suits fire time sequence, we compared three frequently used in sequence feature extracting. First, we looked into GMM-UBM model [11], a combination of Gaussian Mixture Model and Universal Background Model. The GMM-UBM model improves upon the original GMM model by adding background information, thus enhancing the ability for the extractor to focus on the target features. Second, we evaluated the Joint Factor Analysis method as a front end for feature extraction [13]. According to experiments by Kenny P. et.al [14], JFA has better reliability when faced with noisy information and gains purer feature extraction results. Though the above methods have many advantages, they are facing the challenge of I-Vector [15], a better refined and effective method to extract and present high dimensional features in a more compact dimensionality. To absorb the advantages of those methods, it is effective for us to choose GMM-UBM as the front end of I-Vector [16]. With the help of UBM, the feature extractor can more easily spot target features, and I-Vector further carries dimension reduction to output highly compact feature vectors.

Our contribution: Based on the limitations of CNNs and advantages of I-Vector, we propose a novel method that is not dependent on huge datasets or exact shapes of a fire. We use GMM-UBM model as the front end [9], it is followed by an I-Vector feature extractor. At the back end, we choose a G-PLDA classifier [17] to differentiate fires from the background. Details are to be discussed in the next section.

III. Our Method

In this section, we will introduce the structure and principle of our algorithm in detail.

A. Sampling Method

The combustion process of a fire is dynamic. Through this feature, we consider the viability to collect the flickering pattern of each sampled pixel as a time sequence for feature extraction, thus recognizing the fires in a video.

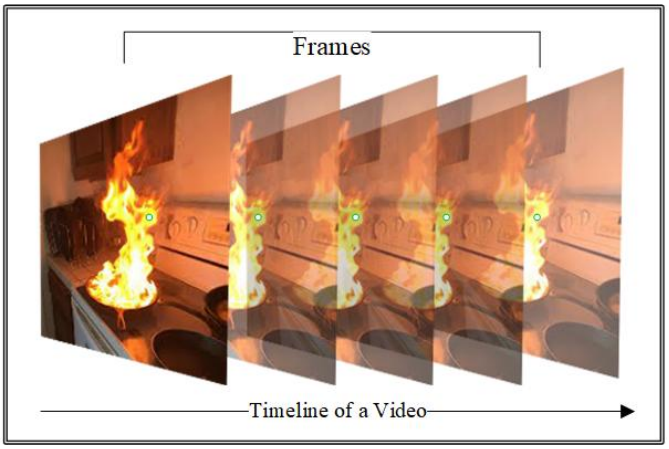

Fig. 2. Sampling method: The green dot is the pixel to be sampled. The sample pixel's brightness value of each frame will be recorded in a sequence.

As Fig. 2 shows, we first locate the pixel to be sampled. Second, considering every sample pixel in the video frames, we record each brightness value in greyscale, to obtain time sequences containing dynamic features. We sample pixels on the fires to accumulate positive instances, while sampling pixels on the surroundings or background to collect negative instances

B. I-Vector Algorithm Baseline

Inspired by [16], we consider that I-Vector can be used to represent the feature of each sample sequence. Firstly, we roughly extract the MFCC feature vectors of each fire and non-fire sample. Then, the MFCC is utilized by GMM-UBM (Gaussian Mixture Model-Universal Background Model) [11][12], and then a large number of data are collected to form further feature extractions. In GMM-UBM, the state occupancy of the Gaussian component of the signal at each time will be calculated. Input MFCC vector of non-fire sequence to GMM-UBM model, the mean value in each Gaussian classification

clutter is m and the Gaussian mean supervector of non-fire UBM can be calculated. After that, Input the MFCC vector of fire sequence to GMM-UBM model, and adaptively obtain M which is the Gaussian mean supervector of fire GMM through MAP algorithm. Meanwhile, the Baum-Welch statistics should be carried out. Assume C to be the number of Gaussian components. The algorithm formula to calculate Baum-Welch statistics shows as Eq1. And Eq2. [16]. At every moment t, $\gamma_t(c)$ is the state occupancy of γ_t in each Gaussian component c, that is, the information γ_t falls into the posterior distribution of the Gaussian component c at time t. And $N_c(s)$ represents the occupancy rate of the given information s falling into the Gaussian model c. The word information mentioned above may refer to time sequences as speaking voices or fire combustion patterns in the time domain, which will be further discussed later.

$$N_c(s) = \sum_t \gamma_t(c) \tag{1}$$

$$\gamma_t(c) = \frac{\pi_c p_c(Y_t)}{\sum_{j=1}^{C} \pi_j p_j(Y_t)} \tag{2}$$

After having Baum-Welch statistics, we collect the mean vector p_i of each Gaussian distribution:

$$m = \left[p_1^T, p_2^T, \ldots, p_C^T\right]^T \tag{3}$$

The target equation of I-Vector extraction is:

$$M(s) = m + T\omega(s) \tag{4}$$

In Eq.4, s is the extracted sequence (in NLP, a sentence; in our work, a pixel sequence of fire or non-fire). The $\omega(s)$ is the I-Vector of a given sequence s, and is also a hidden vector to be solved after having the global spatial difference matrix T. Assume D to be the dimensions of acoustic features extracted from the sequence s, and R to be the dimensions of $\omega(s)$. Hence, the dimensionality of T is $C \times D \times R$.

The global spatial difference matrix T will be obtained using EM adaptive iteration. First, initialize T randomly and bring in Eq.3 to obtain a posterior distribution of the hidden parameter $\omega(s)$. Repeat steps E and M until T meets an appropriate termination threshold. In this process, m will eliminate redundant factors by continuously reducing the dimension in the subspace. When T is obtained, the model parameters reached a global difference space matrix. Thus, the ω with a lower dimensionality is obtained, it carries the difference between classes and the difference between the channels carrying class information. The posterior mean of ω is I-Vector. I-Vector contains the difference between fires and the background environment, as well as the discriminations between sample videos. As Figure 3 shows, the fire flickering pattern is represented by the time series of the correlating pixel-wise greyscale amplitude. We then sample those time series for training and detection.

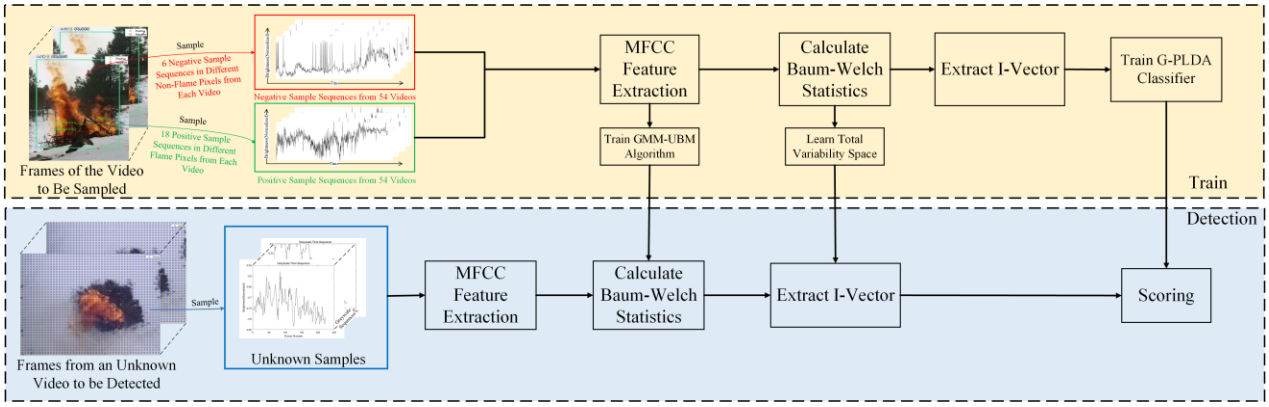

Fig. 3. Our algorithm: a) Training. Fire and non-fire pixels are respectively sampled and labelled into positive and negative samples. All the samples are in the format of time sequences shown in the above coloured squares. Features are extracted from the times sequences to train the I-Vector Extractor and G-PLDA classifier; b) Detection. Pixels of an unknown video are uniformly sampled as the blue circles indicate. Each sampled sequence will have its I-Vector. Then, the I-Vectors of the unknown pixels are scored by the G-PLDA classifier. A threshold in the classifier will determine the output class.

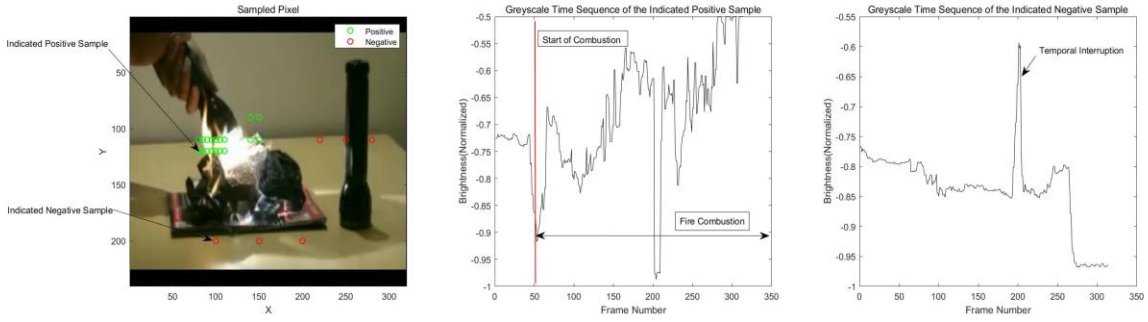

Fig. 4. The sampling of time sequences: it shows one of the 54 videos and its manual sampling. Left) select 24 sample pixels in each of the 54 videos, where 18 pixels are positive instances (coloured in green) and other 6 pixels are negative instances (coloured in red); middle) the time sequence of the arrow indicated positive sample pixel (in greyscale); right) the time sequence of the arrow indicated negative sample pixel (in greyscale).

C. Fire Feature Extraction Using Adapted I-Vector

We transform the original I-Vector algorithm to be applied to fire dynamic feature extraction. In the video to be sampled, the algorithm will collect the specified pixels of a series of time sequences as shown in Fig. 3 and result in sequences $S = [s_1, s_2, \dots, s_{24}]$, which to be fed into Eq.1 and Eq.4. And each sequence s_i has the dimensionality of $1 \times F$, where F stands for the length in frames of a video. When the fire occurs on a certain pixel, its greyscale value will show a specific pattern. Though other background parts may flicker as well, they can be discriminated by the GMM-UBM model at the front end of I-Vector extractor. Thus, we use both fire and non-fire sequences for training this I-Vector extractor by using fire greyscale series as the positive samples while the others as negatives (i.e. background).

Through sampling and training this way, if the greyscale of all or almost all acquisition pixels change greatly, our algorithm will believe that the overall brightness change is caused by other factors that are not relevant to the occurrence of fire. In this way, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) of our algorithm can be greatly reduced.

D. G-PLDA Classifier

Different from the frame-by-frame processing based on a convolutional neural network, the time sequence based on the dynamic characteristics of fire takes the whole video as input. Before G-PLDA scoring, the I-Vector has already been acquired. Then, the G -PLDA classifier intakes the I-Vectors of each sequence and outputs correlating scores. After training, the G-PLDA classifier can score unknown sequences and judge whether some of them contains fires, where the judgement is based on a well-tuned threshold. According to the classification results, we know whether the unknown subject of certain corresponding pixels is burning, thus locating the fire in the video scene. This classifier is rigorous and accurate. Compared with traditional algorithms and neural networks, it not only reduces a lot of computation and speeds up the operation speed, but also greatly improves the robustness of recognition.

IV. Experiment

In this section, we will show how the proposed method is achieved and the experiment results comparing our proposed method with several mainstream approaches in fire detection.

A. Dataset

Unlike the routine of labelling bounding boxes used in CNNs, the proposed method needs time sequences for training and detection. Therefore, we do not consider the sampling and labelling in each frame of the videos, but labelling the collected time sequences. We first select 54 different videos containing at least one clutter of fire combusting at a relatively fixed location of the scene. To cover as many detection environments, the videos varies in viewpoint, lighting condition and resolution. Then, we choose pixels on obvious fires by saving their coordinate values. After that, each selected pixel will be sampled along time domain, resulting in a sequence with a length equal to the total frame number of the video. Each element value of a sampled sequence is identical to the greyscale amplitude of the correlating pixel at the matching frame. The process of manually sampling one of the 54 videos is shown in Figure 4. Through careful sampling, it results a dataset with an amount of 1296 sequences. The samples are labelled into two categories: fire or non-fire. This is different from the I-Vector applied in speaking verification where each person needs to be annotated. Because in the case of fire detection, what the algorithm faces is a binary classification problem. The total positive samples (representing fire) takes up 3/4 of the dataset, and the rest are regarded as negatives representing the environment or confusable noise.

B. Training and Detection

We split the dataset into training set and test set to train our proposed algorithm, and the two sets hold a size ratio of 3:1. During different iterations of training, we reallocate the samples that go to training or test sets randomly in order to exploit more features from different scenes. The algorithm trains I-Vector feature extractor first. At this step, the key parameters in Eq. 3 will be iterated. In this process, the algorithm learns the differences between fire and the background environment, as well as the discriminations between fires sampled in various environments. And it is a well-tuned Eq. 3 that ensures our model's feature extraction precision. By using Eq. 3, the extractor knows how to best extract features regarding fires as the target. After the I-Vector feature extractor is finely adjusted, we train the G-PLDA classifier to endow the algorithm the ability to make precise decisions. By using a DNN, the threshold at the back end of our classifier could be fine-tuned based on I-Vector features and given labels, contributing a 91% training accuracy as Fig. 5 shows.

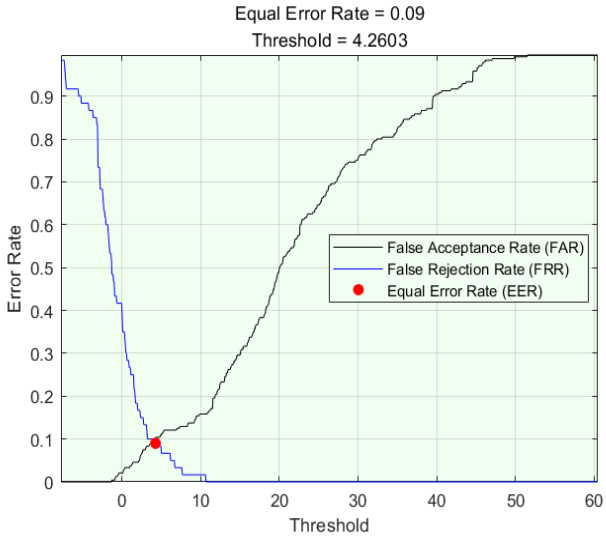

Fig. 5. G-PLDA classifier threshold tuning: After iterating the threshold using DNN, EER reaches 0.09. At this point, recall rates for acceptance or rejection are identical, the G-PLDA classifier reaches best overall performance.

In detection, the unknown video is sampled into sequences in the same approach as the training process does. But the pixels are sampled at a regular adjustable interval as Fig. 6 indicates, resulting in a multi-dimensional matrix. The algorithm will then process all the input sequences in parallel using matrix

manipulation. Each input sample will have an I-Vector and an output classification. By applying the output classes back onto the unknown video, the position and approximate shape of fires would be indicated. It is understandable that the resolution and accuracy would increase as the sampling interval narrows.

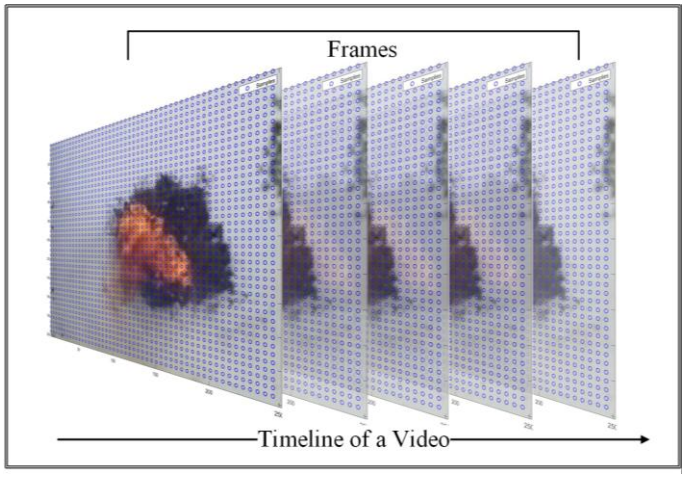

Fig. 6. Sampling of unknown videos before detection: Pixels are sampled at a regular adjustable interval (sampled pixels are spotted by blue circles). Sampled sequences are stored in a multi-dimensional matrix.

V. Results

A. Speed and Accuracy

Usually, smaller parameters are prone to reduce the accuracy of recognition. However, the proposed algorithm not only speeds up the FPS, but also increases the accuracy of recognition.

We use a 480 × 460 size video consisting of 3491 frames to test speed performance on four different algorithms. We operated the following test data under the hardware test conditions of an Intel 10875H CPU and an NVIDIA RTX 2070 MaxQ GPU. The operation results are shown in Table II. As can be seen from the table, the shortest time consumption is 17s which is performed by YOLOv5-S on the GPU. However, our algorithm can complete the same task using only 23s under GPU mode, while obtaining over 1.4 times higher the accuracy of YOLOv5-S, despite a slight cost of speed. Compared with other methods, the proposed algorithm greatly reduces the time consumption of calculation, showing great process capacity.

For the comparison of accuracy, the first three networks are trained based on pretraining models with the same dataset. This dataset includes the Kaggle data set of 1000 images and 500 images about fire we collected from the Internet (to be described later). Our proposed algorithm uses smaller datasets containing 1296 sequences. After training, those four methods are tested on the same unknown labelled video. Although the accuracy of Inception-V4-OnFire is the highest among the three mainstream networks, reaching 83% our algorithm achieves 91% accuracy in the case of smaller data set training. This means that our algorithm achieves higher dataset utilization and computational efficiency, while remaining relatively high accuracy.

TABLE I. Speed and Accuracy of the Four Models

Detection Methods	Speed	Accuracy (%)
FireNet	61s	51
InceptionV4-OnFire	108s	83
YOLOv5-s	17s	67
Our Method	23s	91

B. Parameters

The number of parameters of the four models is shown in Table I. It can be seen that the parameter size of our method is between FireNet and YOLOv5-S, which is less than that of Inception-V4-OnFire. For our method, the parameter size is 14.3 million, which is about 75% of the number of parameters in the Inception-V4-OnFire network. Although the parameters size of our method is not minimal, it is still applicable for deployments on small mobile platforms.

Although the parameter size of our method is larger than that of FireNet and YOLOv5-S, its accuracy is much higher than those of the two models

TABLE II. Parameters of the Four Models

Detection Methods	Parameters
FireNet	3.1M
InceptionV4-OnFire	Over 20M
YOLOv5-s	7.01M
Our Method	14.3M

C. Datasets

We compared the datasets only detecting fire or no fire (i.e. binary classification) with our dataset. In the first two rows of Table III, the dataset sizes for training the corresponding network for fire detection are shown. Arpit Jadon et.al. [7] used a dataset of 2.4K images to train the FireNet, and Shixiao Wu et.al [8] used 1.1K images for YOLOv3 transfer learning towards fire detection. Our dataset is much smaller than the common ones, but our method can still perform equally well when feeding unknown data, compared with the former two studies.

TABLE III. Datasets Comparison

Datasets	Contents
FireNet Dataset [7]	2.4K(image)
YOLOv3 Dataset [8]	1.1K(image)
Our Dataset	1K(greyscale sequence)

VI. Discussion

In this paper, we sample the video into greyscale sequences, and to extract the dynamic features along the time domain. The algorithm makes full use of the brightness patterns of fire in greyscale and achieves decent accuracy. However, in some cases, the occurrence of fire cannot be well determined only from the fluctuation of the object. Therefore, we will add color channels to assist analysis in our future research. Through the time series acquisition of RGB channels, the changes of the target on the three color elements can be analyzed respectively.

Based on specific colors, amplitude difference between color channels could be further researched to discover the relative patterns. Our method proposed in this research can also be used an auxiliary algorithm of the existing ones, providing additional judgements about the existence of fire.

VII. Conclusion

In this paper, we propose an algorithm based on I-Vector to detect the occurrence of fire in videos by using time series. Having set off from fire dynamic features, our algorithm achieves a unique approach using a lighter model while only requiring smaller datasets. The main advantage of our algorithm is that it recognizes fires within video frames at the first glance, which counters the low recognition efficiency of the current CNN networks which process frame by frame. The application of this model has great potentials varying from video fire monitoring and UAV fire inspection. Compared with mainstream convolutional neural networks, our method is easier for both training and deployment. Containing fewer parameters also means lower performance requirements for computing platforms.

Acknowledgement

We here sincerely thank Tiangong University and our group members of Tianjin Provincial University Student Innovation and Entrepreneurship Program (No.202110058107). We would also like to express our appreciation to Mr. Yukuan Sun and Mr. Zijing Zhang for their inspirations and professional suggestions.

References

[1] Valizadeh, M., & Wolff, S. J. (2022). Convolutional Neural Network applications in additive manufacturing: A review. *Advances in Industrial and Manufacturing Engineering*, 100072.

[2] Maheswari, B. U., Rajakumar, M. P., & Ramya, J. (2022). Dynamic differential annealing-based anti-spoofing model for fingerprint detection using CNN. *Neural Computing and Applications*, 1-17.

[3] Zhu Yan, Zhang Bin, Zhang Yaping, Li Weimin, & Cai Likang. (2018). Research on multi-point distributed fire monitoring method based on visual recognition. *Measurement and Control Technology, 37(10), 5*

[4] Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. W. (2018). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, *6*, 18174-18183.

[5] Glembek, O. , Burget, L. , P Matějka, M Karafiát, & Kenny, P. . (2011). Simplification and optimization of i-vector extraction. *2011 IEEE International* Conference *on Acoustics, Speech and Signal Processing (ICASSP). IEEE*

[6] Kim, Y. J. , & Kim, E. G. . (2017). Fire Detection System using Faster R-CNN. *International Conference on Future Information & Communication Engineering.*

[7] Arpit Jadon, A. , Omama, M. , Varshney, Ansari, M. S. , & Sharma, R. . (2019). Firenet: a specialized lightweight fire & smoke detection model for real-time iot applications. *arXiv:1905.11922v2 [cs.CV] 4 Sep 2019*

[8] Wu, S. , & Zhang, L. . (2018). Using Popular Object Detection Methods for Real Time Forest Fire Detection. *2018 11th International Symposium on Computational Intelligence and Design (ISCID). IEEE.*

[9] ZHAO Baojun, ZHAO Boya, TANG Linbo, WANG Wenzheng, WU Chen.(2019) Multi-scale object detection by top-down and bottom-up feature pyramid network. *Journal of Systems Engineering and Electronics,2019,30(01):1-12.*

[10] Pi, L., & Wu, J. (2021, May). FPNet: Fusion Attention Instance Segmentation Network Based On Pose Estimation. *In 2021 33rd Chinese Control and Decision Conference (CCDC) (pp. 2426-2431). IEEE.*

[11] Mclaughlin, J. , Reynolds, D. A. , & Gleason, T. P. . (1999). A study of computation speed-UPS of the GMM-UBM speaker recognition system. *European Conference on Speech Communication & Technology. DBLP.*

[12] Reynolds, D. A. (2009). Gaussian mixture models. *Encyclopedia of biometrics, 741(659-663).*

[13] Bond, S. R. , Hoeffler, A. , & Temple, J. . (2001). Gmm estimation of empirical growth models. *Cepr Discussion Papers, 159(1), 99-115.*

[14] Kenny, P. , Boulianne, G. , Ouellet, P. , & Dumouchel, P. . (2007). Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing.*

[15] Kenny, P. , Stafylakis, T. , Ouellet, P. , & Alam, M. J. . (2014). JFA-based front ends for speaker recognition. *IEEE International Conference on Acoustics. IEEE.*

[16] Garcia-Romero, D. , & Espy-Wilson, C. Y. . (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems. *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011.*

[17] Matejka, P. , Glembek, O. , Castaldo, F. , Alam, M. J. , Kenny, P. , & Burget, L. , et al. (2011). Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. DBLP.

[18] Ying Liu, Luyao Geng, Weidong Zhang, Yanchao Gong, and Zhijie Xu (2021), Survey of video based small target detection," Journal of Image and Graphics, Vol. 9, No. 4, pp. 122-134. doi: 10.18178/joig.9.4.122-134.

2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS) | 978-1-6654-9463-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIS54925.2022.9882470

High precision detection of small hepatocellular carcinoma using improved EfficientNet with Self-Attention

Yuxiang Wang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
Euson@outlook.ie

Zhongming Huang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
reavenhuang@163.com

***Abstract*—Small hepatocellular carcinoma (SHCC) is among the most fatal cancers, and spotting SHCC symptoms in the early stage is vital for conducting timely treatments. Thus, auxiliary detection algorithms have been developed, especially after convolutional neural networks (CNN) made great progress in processing medical images. However, their performance is confined by dataset, resulting in limitations to accurately detect SHCC appearing small and diffusive in CT images. In our work, Self-Attention mechanism has been introduced as the front end and EfficientNet as our backbone network, contributing to a novel SHCC detection algorithm able to automatically spot subtle lesions through our image-wise annotated dataset in a weakly supervised manner. In our model, the Self-Attention module extracts ROI and background features from original CT images and generates weighed feature map to the EfficientNet. In our backbone network, the EfficientNet learns from input feature maps and is weakly supervised under image-wise annotations. With the pre-process of Self-Attention, our data size for EfficientNet has been reduced, thus enhancing learning efficiency and reducing time consumption. After training on over 1.5k CT images, our model has achieved decent detection performance comparing to other state-of-the-art methods while remaining acceptable complexity.**

Keywords—Small Hepatocellular Carcinoma (SHCC) Detection, Self-Attention, Convolutional Neural Network, Medical Image Processing

I. INTRODUCTION

Small hepatocellular carcinoma (SHCC) is accountable for a large number of disease mortalities each year. According to WHO observations [1], SHCC has been the third of major causes of deaths from cancer. Given the perilous risk of exacerbating, it is of high-risk misdiagnosing or ignoring early-stage tumours. To manage to preclude the development of SHCC and conduct therapies earlier, several examine methods were invented, and computed tomography (CT) is widely applied among other techniques. With CT scans, medical workers are able to detect SHCC from observing the darkness of scan images [2]. However, therapists are not always accurate, and early-stage SHCC could be left out [3,4]. Since the early tumours are mostly shaped small and diffusive, they have the potential to escape prudent inspection or cause disagreements between therapists. Fig. 1. shows the SHCC CT images and marks the location of the lesion.

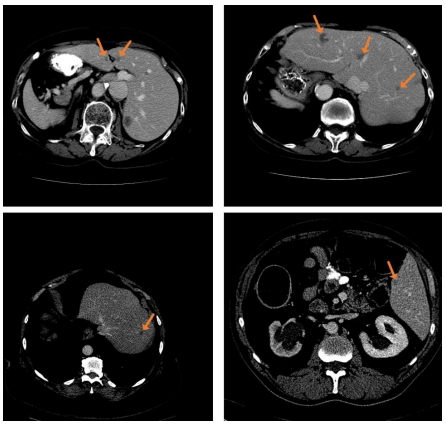

Fig. 1. SHCC CT images and the location of lesions. The lesions of SHCC are usually small which lead to the difficulty of accurate detection.

Predecessors have always been endeavouring to prevent misjudgements of SHCC based on CT scan examinations. At the early stage of computer aided SHCC detection algorithms, simple and mathematical AI approaches were used [5,8], but the overall accuracy was not enough reliable for clinical advising. The advent of convolutional neural network (CNN) further provides researchers a practical and promising solution to process CT scan images [2,3]. By annotating pixel-wise on the CT scan images which have been manually diagnosed, the CNN model is able to learn to distinguish and segment cancerous areas from CT sections. In recent studies of SHCC detection from CT images, Cui et.al. [6] proposed a 3D convolutional neural network; Duc et.al. [7] introduced an adapted convolutional neural network based on U-Net, an originally cellular segmentation-oriented algorithm.

Though prosperous discoveries continue to spring up in SHCC segmentation methods using CNN, knotty problems always occur when those methods are put into practice. To establish a suitable dataset for training traditional CNNs for SHCC detection, huge images should be collected and they are required to be annotated meticulously. This procedure to create a qualified dataset is always exhausting. And because the annotation masks are based on diagnosis from therapists, the judgement of cancerous regions, especially those in early stage, may not be indisputable. Therefore, traditional highly supervised learning is limited by artificial diagnosis, and their

978-1-6654-9463-2/22/$31.00 ©2022 IEEE
ICIS2022, June 26-28, 2022, Zhuhai, China

huge dataset is related to higher computational capacity and longer time consumption.

To further excavate the sensitivity of computer aided algorithms for SHCC CT detection, we would like to explore a novel approach which does not require artificial diagnosis on CT raw data or exhausting pixel-wise annotating. We introduce a novel idea to utilize the attention mechanism and CNN. We adapt the Self-Attention (SA) mechanism [9] into a feature encoder at the front end of CNN backbone, in order to enable our model to automatically learn the region of interest and compare differences between healthy and cancerous liver CT images. For the CNN backbone, we select EfficientNet as the prototype to endow our model high learning efficiency and detection accuracy [10]. Our model is trained on image-wise annotated datasets adapted from the open LiTS benchmark [11], and it gives a classification as the output. The exact detection of SHCC can be decoded from the feature map given by our model. After training, the average detection precision of our model is over 98%, leading congeneric CNNs in SHCC CT segmentation and remaining comparable high accuracy with state-of-the-art methods.

II. Related Work

Computed tomography (CT) examination is an important procedure before the SHCC malady is finally diagnosed [12]. When ultrasonic examination is not enough clear to support analysis of SHCC existence, it becomes necessary to acquire a slice image in the section of abdomen where therapists scrutinize the characteristics inside a liver. Most tumours in a large size are able to be diagnosed by therapists based on CT images. However, in some cases when tumorous shadows in CT scans exist in an ambiguous state, therapists could easily be confused and biopsy may take place. In this case, other risks also lurk [2]. Therapists may ignore early-stage SHCC because of its diffused border or minuscule size, and biopsy is not always practical on patient with fundamental diseases. It is of high risk not able to detect SHCC in time according to the HCC guideline [14].

Therefore, computational approaches aimed to assist SHCC detection based on CT scans were developed. In the earlier stages, machine learning (ML) methods were put into practice [15]. Most ML approaches were based on primary mathematic modelling theories and they do not have fulfilling potentials to handle massive patient data. After the advent of convolutional neural network, the ability for computers to extract patterns in images has been greatly improved. Researchers also adapted achievements in CNN to better suit the specific conditions of SHCC detection. Li W et.al. [16] proposed an CNN automatic SHCC detection based on CT images, their model was trained to segment tumour positions pixel-wise and it is a relatively early implementation in SHCC segmentation algorithms based on a CNN. With delicate labelling of CT image data and sufficient training, such segmentation CNNs have the potential to achieve an accuracy over 70%, and this is not enough for practical diagnosis. As the CNN structure continues to make progress, new models were introduced in SHCC CT detection, especially those of SHCC. Recently, Cui et.al. [6] published a multi-channel 3D-CNN to segment SHCC in CT images and this model was compared between 3D-CNNs of segmentation or non-segmentation sampling, showing robust detection reliability.

However, CNN methods based on traditional supervised learning often demands huge dataset with meticulous labelling. Because the annotations are based on manual diagnosis by therapists, the accuracy and quality of a dataset is confined by the precision of artificial diagnostic advices where mistakes are prone to emerge. Since the attention mechanism came to public, CNN models are endowed with the ability to discriminate subtle differences between sample images in a homogeneous dataset [17]. With the aid of attention mechanism, researchers no longer need to label segmentation datasets in a pixel-wise manner. If the attention mechanism is placed at the front end of a CNN, this weakly supervised learning strategy has the ability to direct the CNN how to segment SHCC regions through comparing mass samples containing SHCC liver CT scans and those which are healthy.

In our model, we utilize the Self-Attention mechanism (SA) as the front end of the CNN backbone. Diao et.al. recently proposed a multi-scale attention mechanism (MSAN-CNN) method in processing whole-slide pathologic images (WPI) [4], contributing to a new method for high resolution medical image processing. But in CT scan examinations, where each image is formatted in grey scale, lower in resolution, and has similar interest regions, we select SA mechanism to achieve lower computational capacity while obtaining decent accuracy. As to the backbone network, we choose to adapt EfficientNet [10] to intake the encoded features from SA front end. After training the backbone on features extracted by SA from SHCC and non-SHCC CT scans, our model turns out to achieve over 98% detection accuracy for SHCC regions, showing improved reliability among traditional SHCC detection methods based on CNN.

III. Our Method

A. Solution

According to the previously mentioned features and problems, we propose to use adapted EfficientNet with Self-Attention [18] to detect SHCC in CT images. Self-Attention mechanism [17] at the front end can effectively screen out the subtle differences in the image and give more "attention" which means our algorithm will focus on analysing these subtle differences and effectively classify them. This Self-Attention module works as a feature extractor and encoder for future learning of the CNN backbone. Meanwhile, the Self-Attention mechanism will automatically select the region of interest, meaning the same parts in a transverse CT section image will be considered as the background while those discriminative parts will be used as the key of our algorithm to learn and analysis. Self-attention mechanism focuses on the blurs and uneven shadows of the liver that may

become a tumour, and this mechanism judges irrelevant organs inside and around the liver as the background. Fig. 2. shows where the Self-Attention focused. Through the repeated learning of a large number of SHCC CT images, our algorithm will make more detailed perceptions of the features of SHCC, achieving higher detection accuracy over average therapists according to previous data [19].

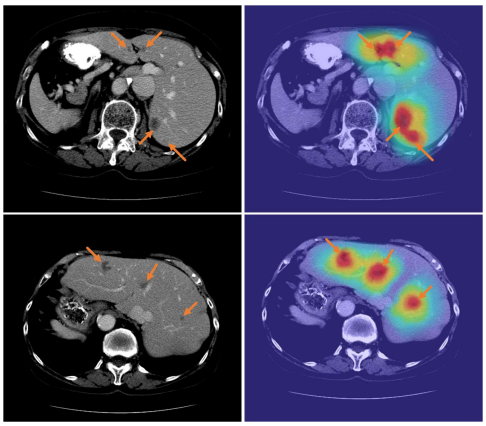

Fig. 2. The locations of SHCC is accurately focused and other locations are marked as the background after the CT image passes through the convolution kernel.

B. Self-Attention Convolution Kernel

Attention mechanism was inspired by human neural science, and self-attention is a derivative of the rudimentary concept of attention [9]. In our implementation of Self-Attention mechanism [20], we assign weights to the pixels of the input images as the pre-processing for latter learning of EfficientNet. We first generate embeddings a of each input image by matrix multiplication, a^i stands for the embedding of the i^{th} input image. Then we extract three attribute vectors Query, Key and Value from the previous embeddings. The three attributes are represented by q^i, k^i, v^i where i is the index of input images in the following formula. Next, we implement self-attention by multiplying the embeddings with Query, Key and Value respectively. Then, by calculating the correlation between Q and K, the matrices Q, K and V are obtained [18]. The weight coefficient of V corresponding to each k is also calculated. The formulas (1)(2)(3) show the above process.

$$q^i = W^q a^i \tag{1}$$

$$k^i = W^k a^i \tag{2}$$

$$v^i = W^v a^i \tag{3}$$

Then, we normalize attention values by referring to SoftMax with the probability distribution with the sum of the weights of all elements as one [18]. According to the probability distribution, the detection characteristics of the attention output matrix can be carried out. The attributes Query, Key and Value are the appropriate parameters learned through separate training, namely weight. By adding these weights to the corresponding pixels which are reduced dimension, the matrix becomes the feature map. The Self-Attention mechanism on our dataset is shown in Fig. 3.

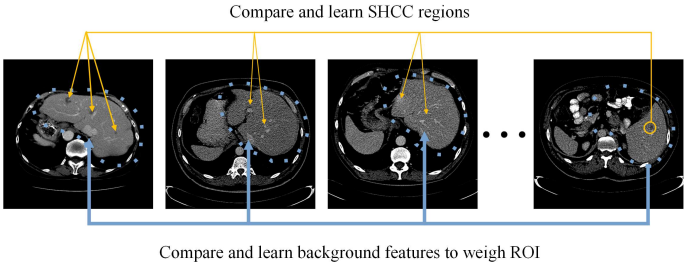

Fig.3. The process of Self-Attention convolution kernel extracting feature. Convolution kernel learns the subject features and background features by comparing a series of pictures.

C. EfficientNet with Self-Attention

The main structure of our algorithm is quite clear. It could be mainly divided into three modules: construct our Self-Attention mechanism, pre-training EfficientNet model [10] and the fusion network. Self-Attention mechanism and EfficientNet model are constructed separately before fusion. After the implementation of these two modules, they will be placed in the fusion connector for encapsulation.

In our whole model where the front mechanism and the backbone CNN are jointed, Self-Attention mechanism functions as an attention convolution kernel. As an instanced kernel stored in the fusion network, its output features can be directly passed to EfficientNet. The existence of this convolution kernel replaces the original feature extraction process originally embedded in EfficientNet. The subtle features of the original data will be extracted one by one and encoded into the "main body", while the same part of each image will be encoded into the "background" through the attention convolution kernel. These two parts of features will be assigned different weights. When the overall weights are summed to one, more weight is assigned to the main body of extracted features while the background part weighs less which could be called heat features map. Through the encoding process by Self-Attention module, we will get a series of image feature vectors that only retain effective detail information and the irrelevant background will not be fed into the EfficientNet. After encoding features, our attention convolution kernel outputs the resulted feature tensor to the pre-training EfficientNet. EfficientNet expands the network depth (Li), network width (Ci) and input resolution (Hi, Wi) on the basis of convolution network [10]. The particle elements of the input heat feature vectors will be magnified by high resolution. And then, they will perform more complex convolution in the extended three-dimensional parameters. EfficientNet will largely improve the training speed and accuracy of our model by sensing and learning based on encoded features. Fig. 4. shows the overall process of data flowing through the improved network.

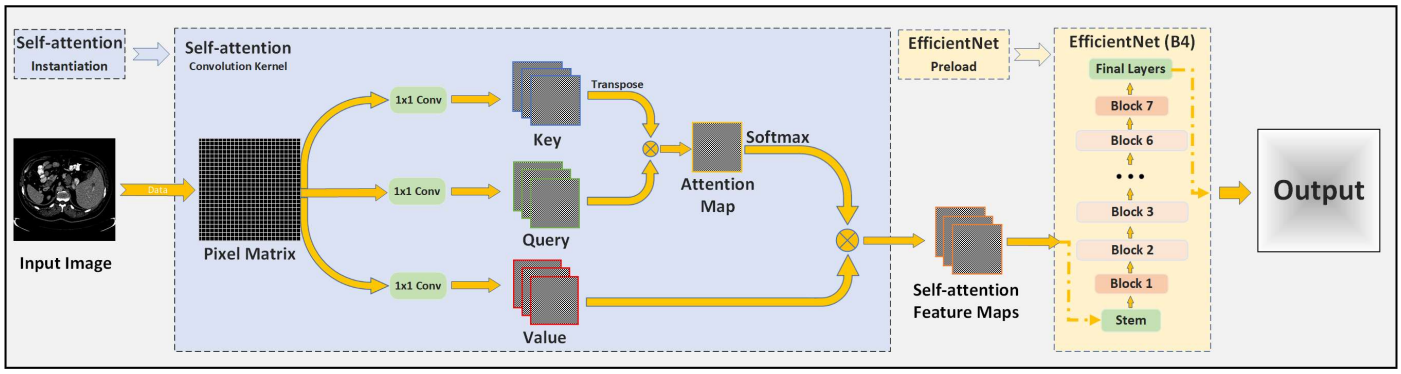

Fig. 4. The whole process of improved network. The original data is extracted from the weighted "main body" and "background" through the Instantiated Self-Attention convolution kernel which will input to EfficientNet for learning in the form of matrix data. Finally, it could get more accurate training and detection results.

IV. Experiment

A. Dataset

Our dataset is selected and re-labelled from the open SHCC CT scan benchmark LiTS [11]. This comprehensive dataset contains three dimensional CT scans of the abdominal region of HCC patients. Totally 130 individuals are covered in LiTS to meet adequate variety in HCC pathology. We selected those CT images taken from the transverse direction, and re-annotated them image-wise into SHCC and non-SHCC groups. Our adapted dataset is consisted of over 10000 image-wise annotated CT scans averagely sampled from each patient's raw data, and we added over 5000 healthy liver section images to provide background contrast for our Self-Attention module.

B. Training and Validation

We divide the dataset into three sets for training, verification and test where their ratio is 3:1:1. The Self-Attention convolution checking accurate feature recognition is a process with continuous comparing and correcting. Therefore, the loss function will fluctuate greatly in the earlier epochs of training. However, as the correct features are focused, loss will be significantly reduced in the following training process. The changes of loss during training are shown as Fig. 5.

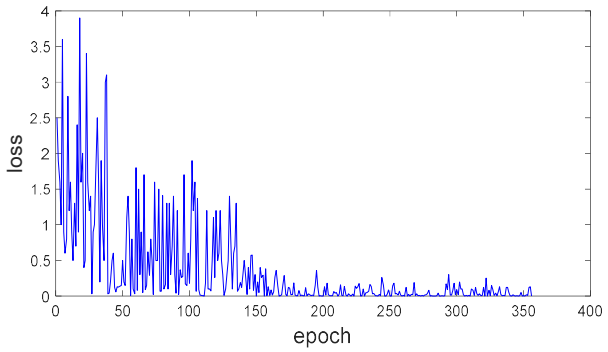

Fig. 5. At the beginning of training, the number of loss is large and fluctuates obviously. After a short period of time, the loss fluctuation becomes smaller. In the middle of the training process, the loss amplitude becomes small and basically remains steady.

Since the accuracy of our detection model is particularly important in providing advice for further detect SHCC. In order to further improve the accuracy of our model, we utilize a small number of the validation set which is about 1500 CT images to evaluate our model and further increase the generalization ability and accuracy of our model. The data of the validation set will also go through all the above processes. After the model is verified, it will be evaluated and corresponding adjustments will be made to further improve the accuracy. The training accuracy changes of mainstream is shown as Fig. 6.

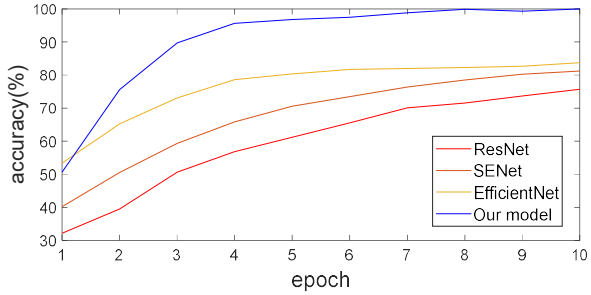

Fig.6. The rising speed of the models using EfficientNet network are significantly higher than the other two networks. And the top accuracy of our model is much higher than other CNNs.

V. Result

A. Top Accuracy

In order to make the accuracy of the algorithm approach near 100%. We have optimized the accuracy of our model in terms of algorithm structure and dataset. Through our experiments, we found that the accuracy of the model can be as high as 98.81%. TABLE I show the accuracy comparison between our model and other mainstream networks on our data set.

TABLE I. Comparison of Top Accuracy of Mainstream CNNs

Detection models	Top Accuracy (%)
ResNet [21]	75.65
SENet [22]	81.24
Original EfficientNet-B4[10]	82.73
Our model	**98.81**

Accuracy is the most important indicator of all auxiliary detection algorithms. In the diagnosis of early SHCC, if we can accurately detect the disease in the early stage of onset, timely therapies would have the opportunity to preclude exacerbation of cancer. Our model has the top accuracy is 98.81% which is close to 100% meaning that our model can detect the occurrence of early cancer with high accuracy, and the fault rate is relatively low. If more data are to be fed in, our accuracy could be further improved. Using this model as the auxiliary method to help detect early liver cancer has high reliability. To a great extent, our method solves the problem that some therapists cannot accurately judge the occurrence of early liver cancer through liver CT examinations. In this respect, it shows that our model has high practical potential in the detection of early SHCC through liver CT images.

B. Parameters and Speed

Usually, parameter complexity and speed are difficult to be equally balanced. However, while adding many parameters to our model, there is no obvious speed decline. The speed of four models running the same dataset is tested under the hardware condition that CPU is i7-10700K and GPU is RTX3060 with CUDA 10.2 and the platform is Windows 10 Pytorch 1.11.0. TABLE II shows the comparison of parameters and speed between our model and other mainstream networks.

Compared with the original EfficientNet, our model has a slight increase in parameters but a small decrease in speed. Nevertheless, our model detection speed remains below 60s which is acceptable in practical application. One reason is the lightweight and high-speed characteristic of EfficientNet, even if we replace the original feature extraction part with a more complex Self-Attention module, with our network parameters to be increased significantly, our whole network can still maintain a fast-running speed. Another reason is that although the Self-Attention module will increase the number of parameters, it will filter out useless information before the pre-trained model starts to learn, meaning the pre-trained model only needs to learn data reduced on dimension to achieve the same result.

TABLE II. COMPARISON OF PARAMETERS AND TEST SPEED

Detection models	Parameters	Speed
ResNet	28.6M	40s
SENet	147.0M	115s
Original EfficientNet-B4	19.2M	32s
Our model	**74.6M**	**49s**

C. Comparison with Recent Researches

We selected four models that were published recently in the field of SHCC CT for comparison. The model based on U-Net proposed by Duc et.al. in 2020 reaches an accuracy of 92.1% [7]. HCCNet was reviewed in 2021, by Wang et.al. which average accuracy can reach 95% [3].The Lim et.al. approach can reach 96% [13].Our model is close in accuracy with those mentioned state-of-the-art methods. In addition, a MSAN-CNN model was introduced to SHCC detection by Diao et.al. in 2022 to solve detection tasks in whole-slide pathologic images (WPI). Their model uses multi-scale attention which achieves an average accuracy of 98.9% in WPI datasets [4]. The average accuracy of our model can be as high as 98.4%, which is almost the same as the MSAN-CNN network using multiscale attention mechanism in lower computational complexities. Obviously, the high accuracy of both attention-based models benefits from the pertinence of attention mechanism for small difference recognition. Although the average accuracy of our model is slightly lower than that of the mentioned MSAN-CNN model, our Self-Attention mechanism has the advantages of easier parallel computing and lower complexity which enable our improved model to have faster training speed and shorter detection period. The TABLE III intuitively shows the comparison of several recent models used for liver cancer detection.

TABLE III. COMPARISON OF AVERAGE ACCURACY OF RECENT MODELS FOR LIVER CANCER DETECTION

Model Name	Average Accuracy (%)	Year
U-Net [7]	92.1	2022
HCCNet [3]	95.0	2021
DCNN in LTP[13]	96.0	2022
MSAN-CNN [4]	**98.9**	**2022**
Our model	**98.4**	**2022**

VI. DISCUSSION

In this research, we used the improved EfficientNet with Self-Attention mechanism to detect small hepatocellular carcinoma (SHCC). From experimental results, it is found that our model has great advantages in dealing with the detection of SHCC. The cooperation of our Self-Attention mechanism and EfficientNet network can accurately identify the lesions of small regions. However, our network also needs to be improved. The main disadvantage is the insufficient stability of our network, which is mainly revealed in the performance fluctuation on different dataset volumes. For CT images with high background similarity in the same group, our network can better pay attention to the location of lesions. Nevertheless, if tested on CT data with varying backgrounds, our network may pay more attention to other irrelevant locations, resulting in a significant reduction in accuracy. These confinements will be improved in our future work.

VII. CONCLUSION

In this paper, we introduce the high-precision detection method of small hepatocellular carcinoma (SHCC) by using improved EfficientNet with Self-Attention model. Through the analysis of pathological data of SHCC, we find that its small area and unclear boundary are the main reasons for the inaccurate detection of SHCC from CT examination. Our model makes targeted optimization for the above challenges. Our Self-Attention mechanism at the front of CNN backbone can directly extract the subtle changes of tumour early occurrence and boundary blurs by comparing the differences of images. Meanwhile, the pre-training network of our improved backbone network has been adapted from

EfficientNet which not only ensures high overall accuracy, but also guarantees the speed of our model. After comprehensive experiments and comparisons, our network accuracy can achieve satisfactory results, which is over 98% much higher than other CNN frameworks by 10% - 30% in the same dataset. In our work, the Self-Attention mechanism is innovatively introduced to SHCC detection, providing a novel effective solution to the problem of accurate detection of SHCC. In recent years, the possibility of SHCC patients being diagnosed or cured has increased continuously. Our work enlightens a new path to solving medical imaging problems.

REFERENCES

[1] Global Cancer Observatory, World Health Organization, https://gco.iarc.fr/. Accessed Apr. 20th , 2022.

[2] Azer SA. Deep learning with convolutional neural networks for identification of liver masses and hepatocellular carcinoma: A systematic review. *World J Gastrointest Oncol. 2019*;11(12):1218-1230. doi:10.4251/wjgo.v11.i12.1218, 2019.

[3] Wang, M., Fu, F., Zheng, B. *et al.* Development of an AI system for accurately diagnose hepatocellular carcinoma from computed tomography imaging data. *Br J Cancer 125* 1111–1121 (2021).

[4] Diao, S., Tian, Y., Hu, W., Hou, J., Lambo, R., Zhang, Z., ... & Qin, W. (2022). Weakly supervised framework for cancer region detection of hepatocellular carcinoma in whole-slide pathologic images based on multiscale sttention convolutional neural network. *The American journal of pathology*, *192* (3), 553-563.

[5] Bi WL, Hosny A, Schabath MB, Giger ML, Birkbak NJ, Mehrtash A, et al. Artificial intelligence in cancer imaging: Clinical challenges and applications. CA Cancer J Clin. 2019;69:127–57.

[6] Cui, H., Wang, K. Y., Li, W. Y., Zhu, H. B., Xiao, L. S., & Liu, L. Artificial Intelligence and Informatics CT images-based 3D convolutional neural network to predict early recurrence of solitary hepatocellular carcinoma after radical hepatectomy, unpublished.

[7] Duc, V. T., Chien, P. C., Chau, T. L. M., Chanh, N. D. T., Soan, D. T. M., Huyen, H. C., ... & Uyen, M. T. T. (2022). Deep learning model with convolutional neural network for detecting and segmenting hepatocellular carcinoma in CT: a preliminary study. *Cureus*, *14* (1).

[8] Jiménez Pérez, M., & Grande, R. G. (2020). Application of artificial intelligence in the diagnosis and treatment of hepatocellular carcinoma: A review. *World journal of gastroenterology*, *26* (37), 5617–5628.

[9] Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10076-10085).

[10] Mingxing Tan, Quoc Le, EfficientNet: rethinking model scaling for convolutional neural networks, proceedings of the 36th International Conference on Machine Learning, PMLR 97:6105-6114, 2019.

[11] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chlebusr et.al., The liver tumor segmentation benchmark (LiTS), arXiv cs.CV 1901.04056, 2019.

[12] Eldad S. Bialecki, Adrian M. Di Bisceglie,Diagnosis of hepatocellular carcinoma, HPB, Volume 7, Issue 1, 2005, Pages 26-34, ISSN 1365-182X.

[13] Sanghyeok Lim, YiRang Shin & Young Han Lee, Arterial enhancing local tumor progression detection on CT images using convolutional neural network after hepatocellular carcinoma ablation: a preliminary study, Scientific Reports (2022) 12:1754 .

[14] Heimbach JK, Kulik LM, Finn RS, Sirlin CB, Abecassis MM, Roberts LR, Zhu AX, Murad MH, Marrero JA. AASLD guidelines for the treatment of hepatocellular carcinoma. Hepatology 2018; 67: 358-380 [PMID: 28130846 DOI: 10.1002/hep.29086].

[15] Kaul V, Enslin S, Gross SA. History of artificial intelligence in medicine. Gastrointest Endosc 2020 [PMID: 32565184 DOI: 10.1016/j.gie.2020.06.040].

[16] Li W, Jia F, Hu Q. Automatic segmentation of liver tumor in CT Images with deep convolutional neural networks. Computer Communnications 2015; 3: 146-151 [DOI: 10.4236/jcc.2015.311023].

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. Attention is all you need. Advances in neural information processing systems, 30 (2017).

[18] Yuxiang Wang. An improved algorithm of EfficientNet with Self-Attention mechanism. In proceedings of the 2021 International Conference on Internet of Things and Machine Learning, in press.

[19] Bruntha, P.M., Dhanasekar, S., Jency, J.G., Pandian, S.I., Pillai, P., Steven, Pramod, & Malani, V. Performance analysis of certain classifiers for liver CT images, 2019.

[20] Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933, 2016.

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

[22] Hu, J., Shen, L., & Sun, G. Squeeze‐and‐excitation networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.