electronics

Article

Applying Advanced Lightweight Architecture DSGSE-Yolov5 to Rapid Chip Contour Detection

Bao Rong Chang [1], Hsiu-Fen Tsai [2,*] and Fu-Yang Chang [1]

[1] Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 81148, Taiwan; brchang@nuk.edu.tw (B.R.C.); m1115509@mail.nuk.edu.tw (F.-Y.C.)
[2] Department of Fragrance and Cosmetic Science, Kaohsiung Medical University, Kaohsiung 80708, Taiwan
* Correspondence: sftsai@kmu.edu.tw

Abstract: Chip contour detection aims to detect damaged chips in chip slots during IC packaging and testing using vision facilities. However, the operation speed of the new chip transportation machine is too fast, and the current chip contour detection models, such as Yolov5, M3-Yolov5, FGHSE-Yolov5, and GSEH-Yolov5, running on the embedded platform, Jetson Nano, cannot detect chip contours in a timely manner. Therefore, there must be a rapid response for chip contour detection. This paper introduces the DSGSE-Yolov5s algorithm, which can accelerate object detection and image recognition to resolve this problem. Additionally, this study makes a performance comparison between the different models. Compared with the traditional model Yolov5, the proposed DSGSE-Yolov5s algorithm can significantly promote the speed of object detection by 132.17% and slightly increase the precision by 0.85%. As a result, the proposed approach can outperform the other methods.

Keywords: Yolov5s; ghost convolution; depthwise separable convolution; object detection; image recognition; attention mechanism

check for updates

Citation: Chang, B.R.; Tsai, H.-F.; Chang, F.-Y. Applying Advanced Lightweight Architecture DSGSE-Yolov5 to Rapid Chip Contour Detection. *Electronics* **2024**, *13*, 10. https://doi.org/10.3390/electronics13010010

Academic Editors: Hyeonjoon Moon and Lien Minh Dang

Received: 30 November 2023
Revised: 15 December 2023
Accepted: 16 December 2023
Published: 19 December 2023

1. Introduction

IC packaging and testing often uses automated optical inspection (AOI) [1] to examine whether the appearance of chips is defect-free. As the automatic production line delivers chips in the fab and puts the chips into the slots for transportation, the AOI checks the chip contours as quickly as possible to ensure that each chip has no appearance defects. The primary purpose of chip contour detection is to detect damaged chips in chip slots during manufacturing using vision facilities. If any damage is detected, the machine will immediately generate a warning and mark the exact location of the damaged chip in the chip slot. In chip packaging and testing, the devices operate at very high speeds, and chip contour damage detection must respond quickly.

Regarding object detection technology, the YOLVO-related algorithms have the characteristics of faster speed and high precision and have developed quite maturely in recent years [2]. Therefore, our previous work [3] introduced the LWMG-Yolov5 algorithm to detect objects quickly. Moreover, our previous article [4] delivered the FGHSE-Yolov5 and GSEH-Yolov5 algorithms to further speed up object detection by using the advanced convolution techniques developed in an earlier paper [5]. However, the operation speed of the new chip transportation machine in the IC packaging and testing process is much faster than that of the old one, and the algorithms mentioned above cannot provide faster chip contour detection. In this case, the speed of chip contour detection is more critical than precision in AOI. Therefore, the main contribution of this study is to propose the DSGSE-Yolov5s algorithm to boost the execution speed significantly in detecting small-scale automotive electronic control chips used in car computers, which can resolve the speed limitation problem to provide a rapid response to chip contour detection.

2. Related Work

2.1. Literature Review

Techniques for rapid object detection have involved the development of advanced visual algorithms in the following papers in the last few years. Rajaram et al. [6] intended to position object boxes precisely and explained restriction using a CNN model, an iterative region-of-interest pooling fabric. In a paper concerning MobileNetV2, Sandler et al. [7] explored the impact on convolutional neural networks with an improved residual structure that uses inverted residuals and linear bottlenecks. Their study delivered a few ideas for convolutional techniques, such as Ghost convolutional layers and Ghostbottleneck. Chollet et al. [8] gave the Xception CNN architecture, which leverages depthwise separable convolutions to replace standard convolutions, improving performance. This architectural choice shows model enhancement by utilizing depthwise separable convolutions. In a work on MobileNets, Howard et al. [9] developed a fabric designed for efficient and lightweight image processing on mobile devices. This method conducts depthwise separable convolutions to attain high efficiency in resource-constrained environments. Tan et al. [10] proposed an efficient model scaling approach, introducing depthwise separable convolutions to upgrade overall system performance. This method shows the benefit of using depthwise separable convolutions to obtain better model scalability and efficiency. Lin et al. [11] proposed an approach based on region segmentation search to distinguish defect contours from edges of structures. They simulated different illumination situations to verify the robustness of the proposed method. Zheng et al. [12] introduced a hybrid algorithm of the R-CNN, SDD, and YOLO methods based on geometric computation and a convolutional neural network for LED chip defect detection. The experimental results showed that this algorithm has an average precision (AP) of 96.7% for large-scale chip detection with a low defect rate.

On the other hand, the following articles have recently mentioned enhancing the precision of object detection in visual algorithms. Dahai et al. [13] devised a lightweight, improved YOLOX network with an attention mechanism (GhostC-YOLOX) in defect recognition and classification for Si3N4 ceramic chip substrate surfaces with an accuracy of 96.47%; this is better than the Faster RCNN model, with an improvement of 18.08%. Li et al. [14] proposed a modified Yolov8s, introducing Bi-PAN-FPN to improve the missed detection of small targets, a GhostblockV2 structure to replace some C2f modules to suppress the loss of information, and WiseIoU loss to evaluate the overall performance of anchor boxes. Aboah et al. [15] used few-shot learning to obtain a robust fewer-annotations Yolov8 model for detecting real-time helmet violations with an mAP of 0.5861. Wu et al. [16] explored the application of SqueezeDet in self-driving to improve accuracy by 6% in object detection at 0.22 s per frame with each 1242×375 image. Object detection using a fully convolutional neural network (CNN) ensures prompt car control with rapid inferences, compact model size, and energy-efficient embedded platform compatibility. In object detection, another SqueezeDet+ model fulfilled a commendable 32.1 FPS frame rate and showed an mAP of 80.4% in precision. Chang et al. [17] modified two models of Yolov4-tiny together with Resnet18 for self-driving control to detect objects quickly and handle the steering wheel precisely using an NVIDIA GTX 1080Ti. In addition, Cai et al. [18] demonstrated a fabric, Yolov4-5D, to promptly detect objects at 66 FPS, running on an NVIDIA GTX 2080Ti, where it was able to increase precision using a real-time Yolov4 model. We successfully tested the combination of two models to obtain 56.1 FPS and a Mean Square Error of 0.0683.

2.2. Chip Contour Detection Models

This study explored how to make the Yolo-related models perform better in applications with the Jetson Nano embedded platform [19,20]. According to the limitations of hardware specifications for the Jetson Nano, we need a lightweight model that can detect the chip contours quickly and precisely. Thus, the latest Yolov5 meets both needs. Modifying the model depth and the number of kernels can downsize the Yolov5 model. Therefore, we chose the most miniature model among the Yolov5-related models—Yolov5s.

GhostNet [21] delivered an insight into the issue of a downsized model for chip contour detection. Our previous paper [3] showed that although the feature maps obtained by the neural network are diversified, similar feature maps inevitably appear among all the obtained feature maps. Similar feature maps are annotated with the same color, as shown in Figure 1. Thus, we can produce similar feature maps using fast and easy operations with low cost to obtain similar feature maps, instead of traditional convolution, as shown in Figure 2. In our previous paper [3], we introduced the LWMG-Yolov5 model based on the Yolov5 and M3-Yolov5 models, as shown in Figures 3 and 4, respectively. LWMG-Yolov5 can boost chip contour detection due to its concurrent reductions in model parameters and computing load. However, the detection speed of this model cannot catch up with the operation speed of the new chip transportation machine.

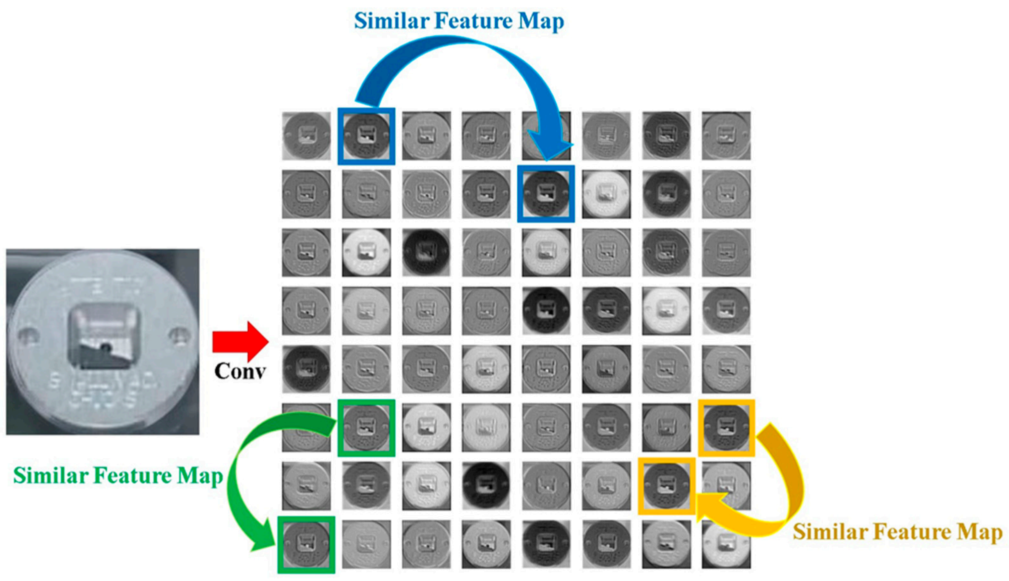


Figure 1. Many similar results in the feature maps using traditional convolution.

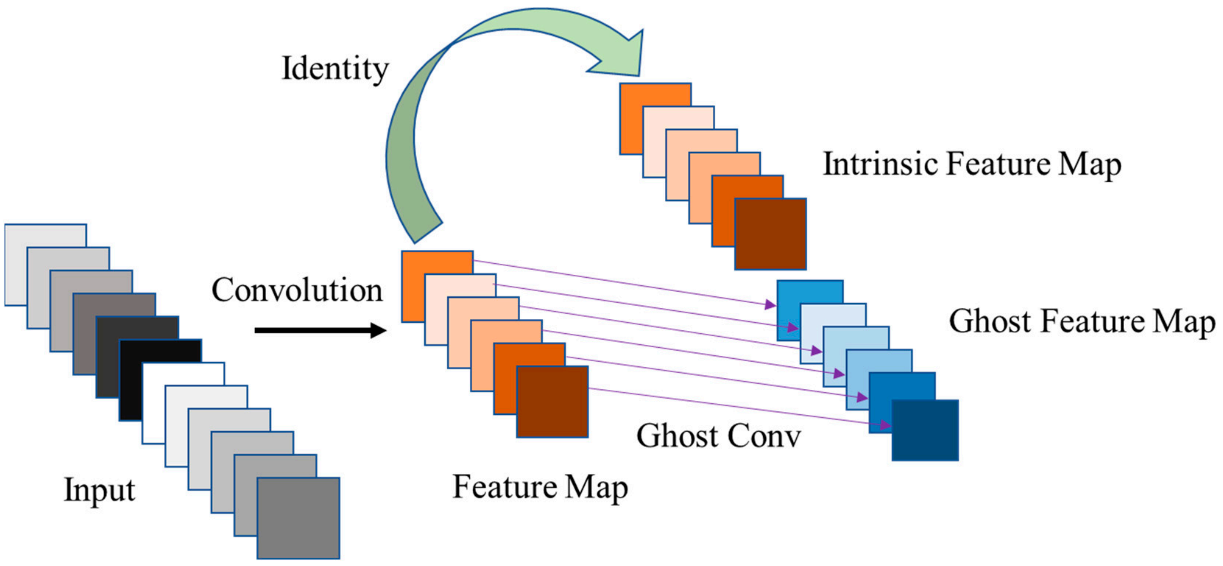


Figure 2. Ordinary Ghost Module.

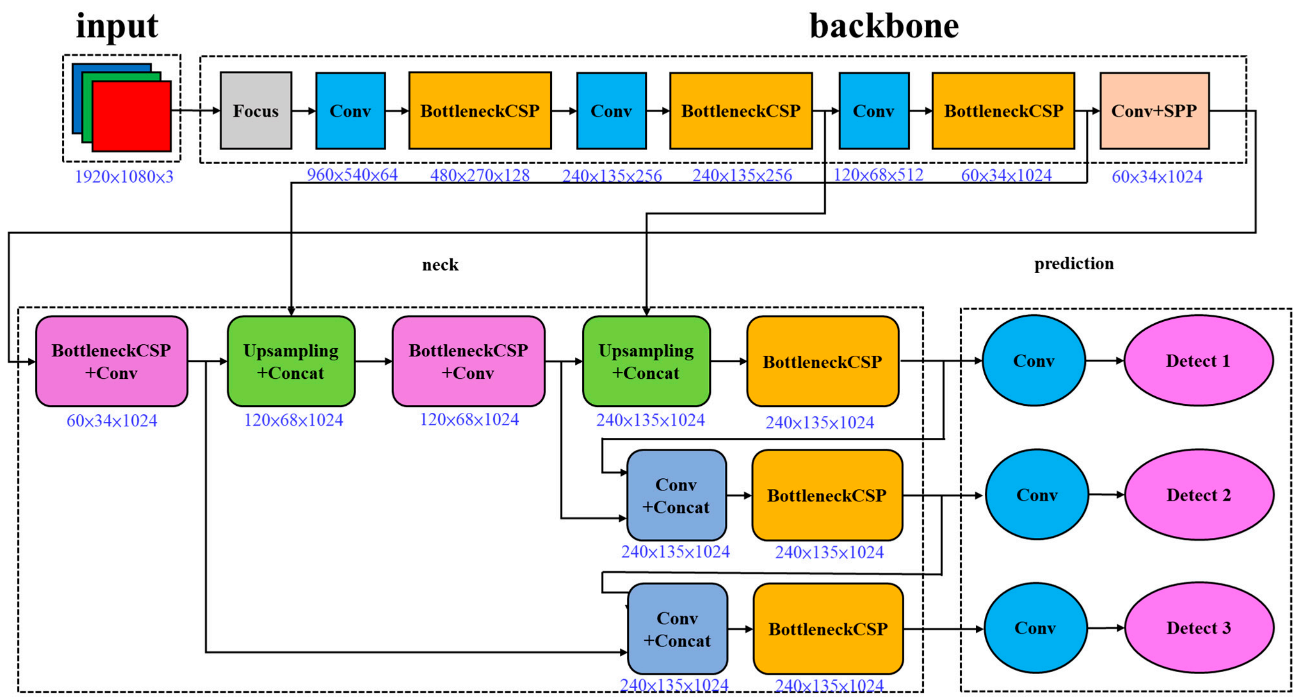


Figure 3. Yolov5 architecture.

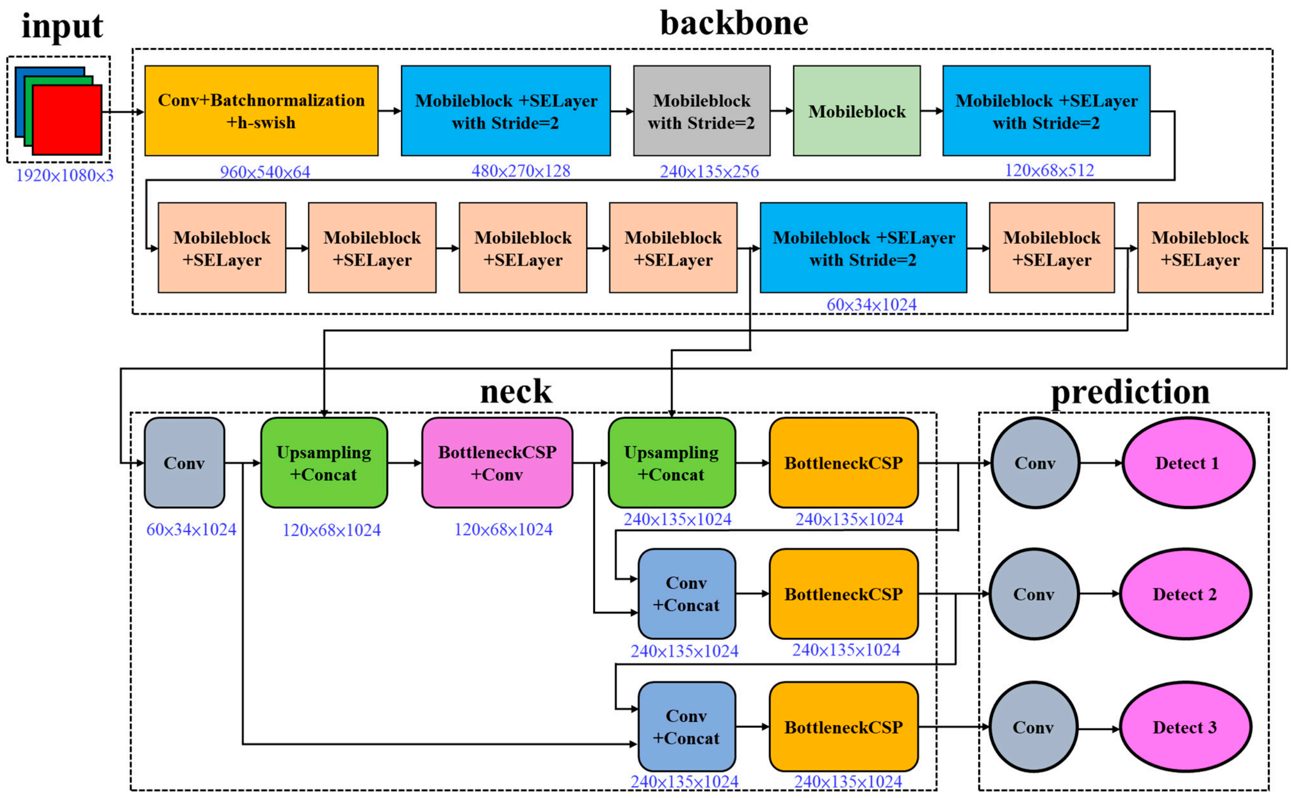


Figure 4. M3-Yolov5 architecture.

In our previous work [4], we added the Ghost module [3] and Ghostbottleneck [5] to the traditional Yolov5 structure to construct the FGHSE-Yolov5 and GSEH-Yolov5 models, as shown in Figures 5 and 6, respectively. The primary goal in this case is to acquire the features that the model can obtain initially with less calculation. The most critical technique is to reduce the number of convolution kernels used so that this also reduces the output of feature maps using traditional convolution. In Figure 2, on an array of intrinsic feature maps, Ghost convolution performs simple linear transformations to yield their corresponding similar feature maps, significantly reducing the amount of calculation [21].

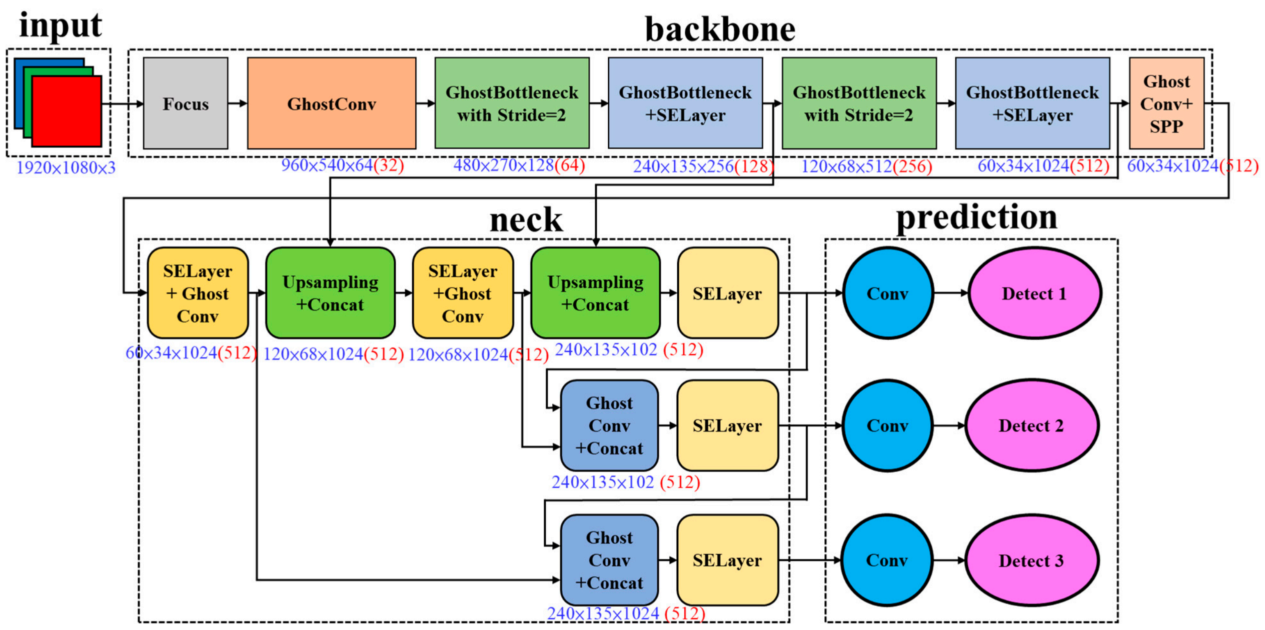


Figure 5. FGHSE-Yolov5 architecture.

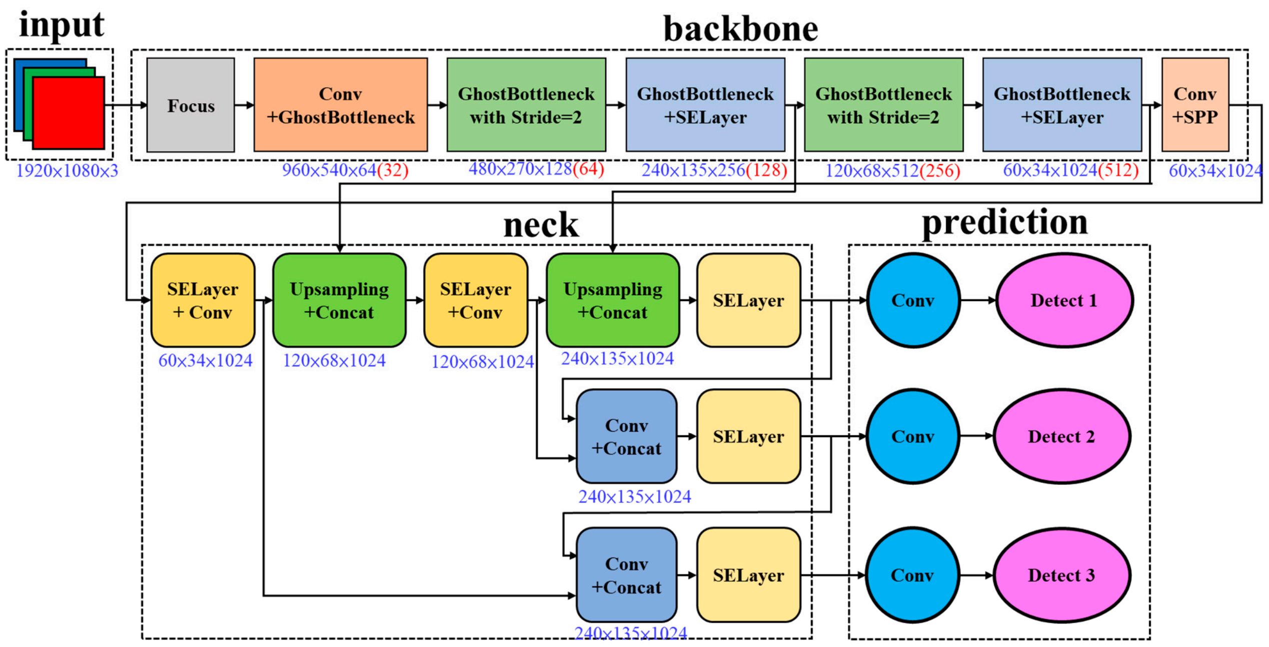


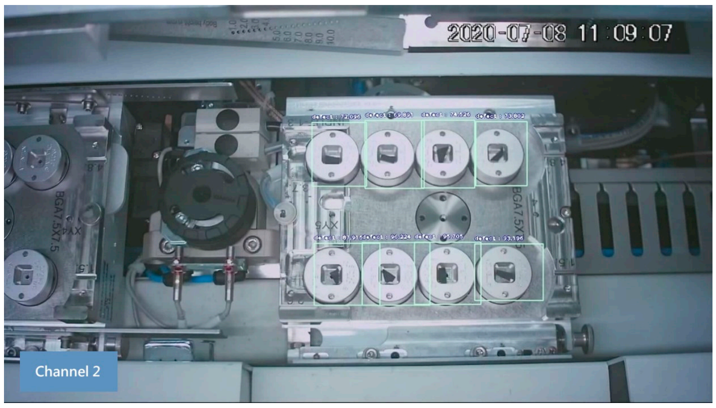
Figure 6. GSEH-Yolov5 architecture.

In our previous article [5], we applied a Ghostbottleneck comprising two Ghost modules, and its connection method is similar to that of ResNet [22]. The features are output by performing the operation of the Ghost module twice and summing up the input features before the process. Furthermore, another idea to improve the Yolov5 model comes from SENet [23], given in our previous article [5]. The function of the SElayer [5] is to infer the relationship between feature vectors so that the system can learn the importance of features between different feature vectors.

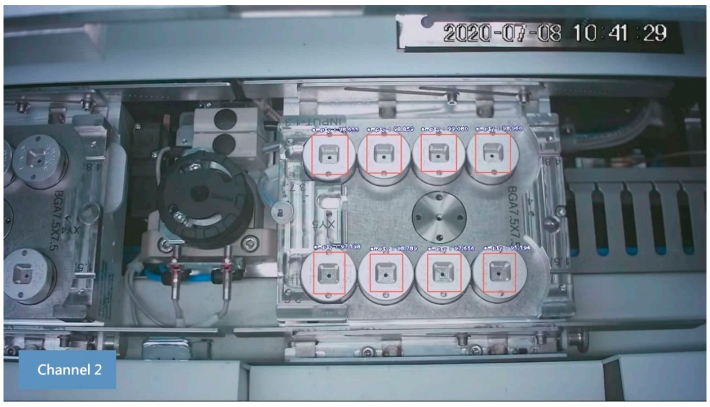
In previous research, modifications to the network structure of Yolov5 were able to reduce computation in feature extraction and decrease model parameters while enhancing the extraction of valuable features. The critical modules employed in this endeavor included the Ghostbottleneck from GhostNet, the SElayer from SENet, and the network structure inspired by MobileNet, all used to replace the traditional CSP module within the CSPNet framework. Figures 3–6 illustrate this modification, which aims to optimize the Yolov5 architecture for improved object detection performance.

3. Method

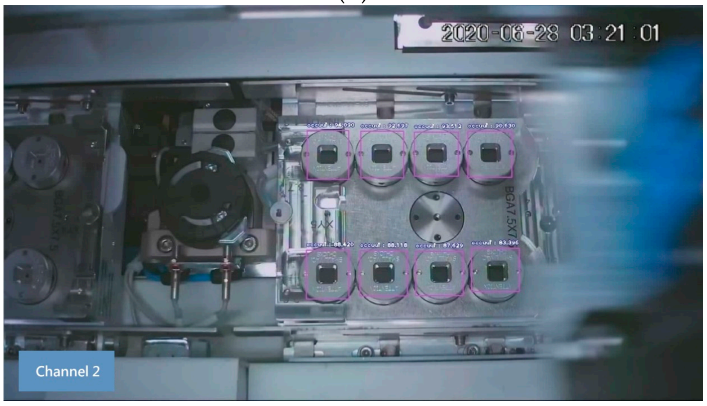
The Jetson Nano is an artificial intelligence platform designed by NVIDIA for embedded systems, and its computing power is quite excellent. The recognition result is output after the chip contour detection has been performed on the Jetson Nano platform. Figure 7 shows the suitability of the model based on the chip contour detection and recognition results [4].



(**a**)



(**b**)



(**c**)

Figure 7. Classification of chip contour detection: (**a**) defective status, (**b**) empty status, and (**c**) occupied status.

In our previous work [4], the GSEH-Yolov5 model replaced Bottleneck_CSP with GhostBottleneck and added SElayer to GhostBottleneck to upgrade its precision. In addition, the FGHSE-Yolov5 model used ghost convolution (Ghost Conv) to obtain a good execution speed. However, these models cannot handle the fast operation of new chip transportation machines. This study aims to amend the network architecture to escalate the execution speed while maintaining a certain level of precision. This can also cut convolution operations and power consumption to achieve an energy-efficient system. Accordingly, we propose an improved architecture of the FGHSE-Yolov5 model to establish a high-performance model, abbreviated as DSGSE-Yolov5s, as shown in Figure 8. Figure 9 explains depthwise separable convolution (DS Conv) [24]. This convolution replaces traditional convolution for generating an array of intrinsic feature maps. The advantage of DS Conv is that it can decrease the computing load and uphold execution performance [5]. In Figure 9, we propose the Randomized Mish (RMish) [25] activation function as a substitute for ReLU in DS Conv to enhance the system performance. Equations (1) and (2) compute RMish, where α_{ji} is a coefficient, $\aleph$ denotes a normal distribution, μ represents the mean, σ stands for variance, ε indicates the upper bound, a_{ji} is the input, p_{ji} denotes intermedia output, and q_{ji} signifies output.

$$p_{ji} = a_{ji} \cdot tanh(ln(1 + e^{a_{ji}})), \quad j = 1, 2, \ldots, \beta; \quad i = 1, 2, \ldots, \gamma \tag{1}$$

$$q_{ji} = \begin{cases} p_{ji} & if \ \ p_{ji} \geq 0 \\ \alpha_{ji} \cdot p_{ji} & if \ \ p_{ji} < 0 \end{cases}, \quad where \ \alpha_{ji} \sim \aleph(\mu, \sigma), \quad \mu = 0, \quad \sigma \in [0, \varepsilon) \tag{2}$$

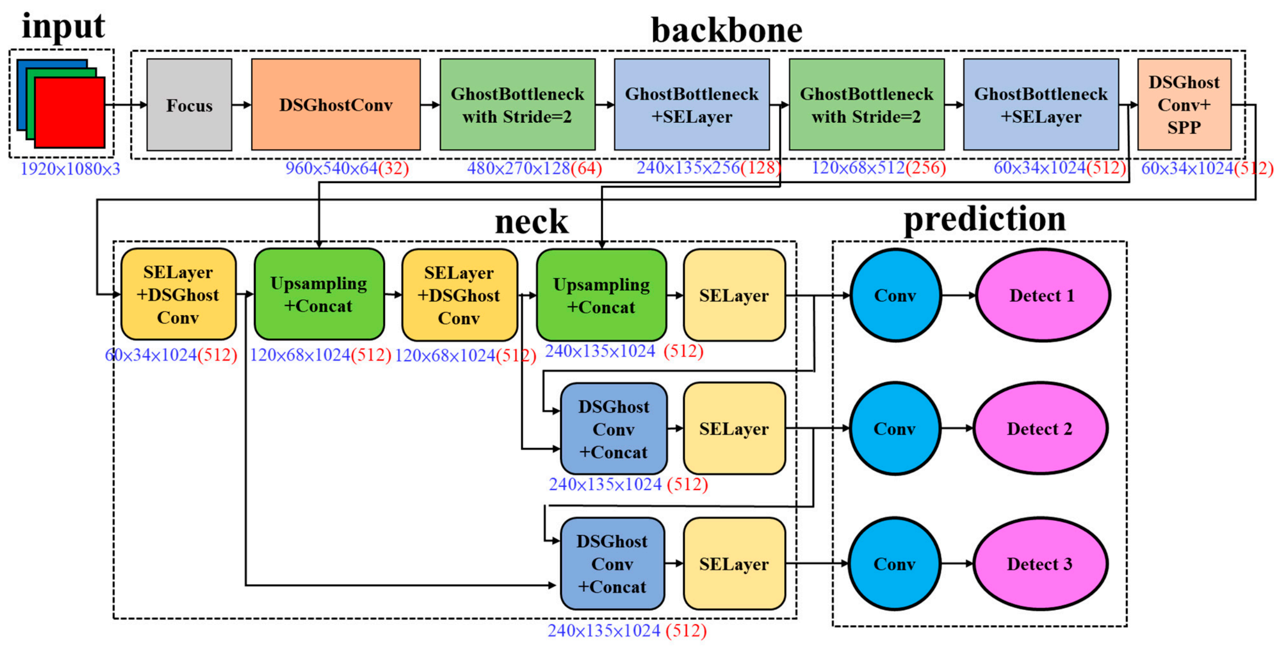


Figure 8. Proposed DSGSE-Yolov5s architecture.

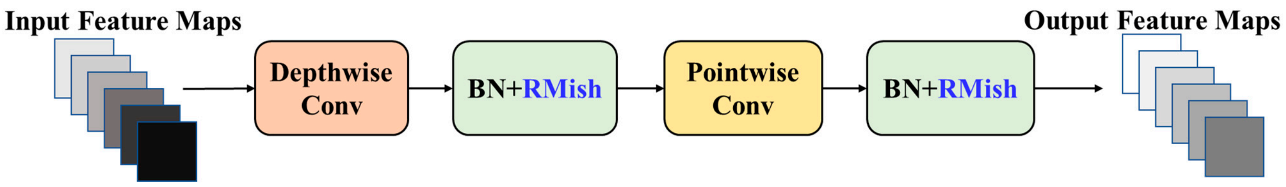


Figure 9. Depthwise separable convolution (DS Conv) with RMish.

Our previous work [5] demonstrated the depthwise and pointwise convolutions that make a depthwise separable convolution (abbreviated as DS Conv). In depthwise convolution, convolution filters are applied independently to each input channel, effectively capturing spatial relationships in a picture. That is, the convolution highlights feature interactions within individual channels. On the other hand, in pointwise convolution, the intrinsic feature maps in the output channels then execute linear combinations to yield the extra feature maps in the output channels. With relatively few parameters, this convolution can perform cross-channel combinations to give the model suitable feature representation. DS Conv uses fewer parameters in the convolution, achieving computational efficiency. Technically, DS Conv, followed by a ghost convolution, can form a Depthwise Separable Ghost Conv (DSGhost Conv), as shown in Figure 10. In Figure 10, DS Conv produces intrinsic feature maps, and Ghost Conv generates ghost feature maps by means of a simple linear transformation. Compared with the GSEH-Yolov5 and FGHSE-Yolov5 models, DSGSE-Yolov5s substitutes a pure Ghost Conv for a Depthwise Separable Ghost Conv (DSGhost Conv), replaces Bottleneck_CSP with GhostBottleneck, and adds SElayer to GhostBottleneck. With DSGhost Conv, the DSGSE-Yolov5s model can accelerate the inference speed. Nevertheless, DSGSE-Yolov5s uses GhostBottleneck plus SElayer to uphold its precision.

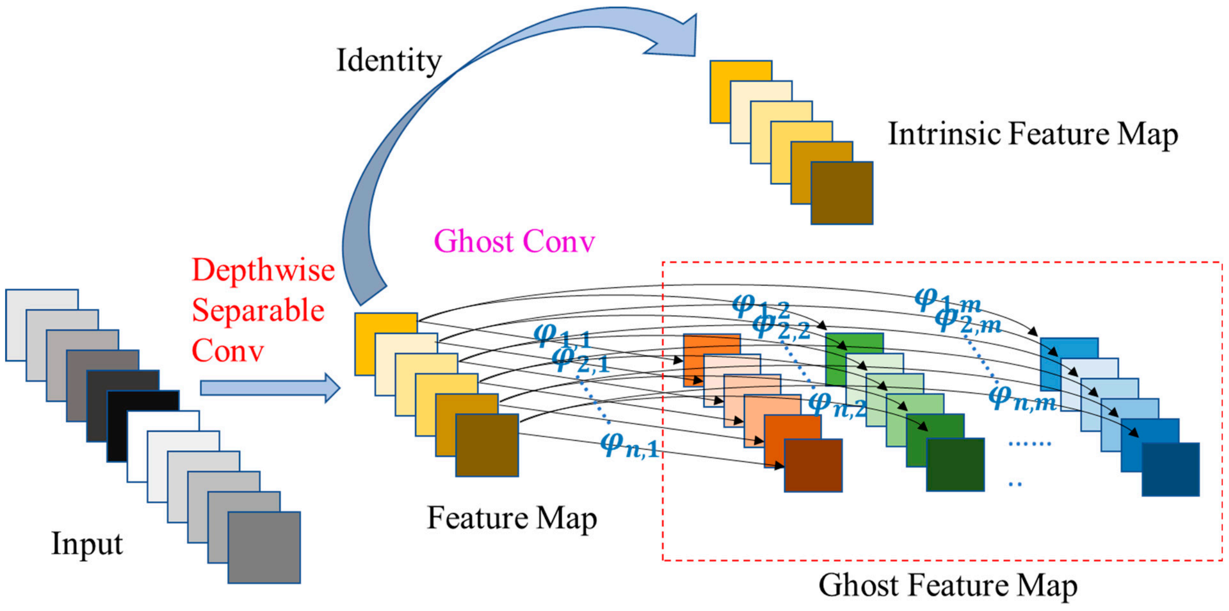


Figure 10. DSGhost Conv.

4. Experiment Results and Discussion

A GPU workstation and Jetson Nano were used as the test environments. The specifications of the GPU workstation consisted of an NVIDIA GeForce GTX 1080 Ti GPU, an Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz CPU, 32 GB of 64-bit LPDDR4 memory (1600 MHz, 25.6 GB/s), a 256 GB SSD, and a 1 TB SATA drive. On the other hand, the specifications of the Jetson Nano comprised a GPU with NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores, a Quad-core ARM Cortex-A57 MPCore CPU, 4 GB of 64-bit LPDDR4 memory (1600 MHz, 25.6 GB/s), and a 16 GB eMMC 5.1 storage module. The packages LabelImg 1.8, Anaconda® Individual Edition 4.9.2, Jupyter Notebook 6.1.4, TensorFlow 2.2, TensorRT 7.2.3, PyTorch 1.6, and JetPack 4.5 were used.

In the experiment, we used Anaconda 3 to establish the run-time environment for the Yolov5 model and collect the data. The idea was first to modify the traditional Yolov5 model, which is suitable for implementing rapid object detection, running on the Jetson Nano embedded platform. Next, we deployed the improved Yolov5 models on the Jetson Nano in this application. Finally, the experiment tested applications on the Jetson Nano to examine the difference in execution speed and precision between Yolov5 and its improved versions.

Initially, we labeled the data set provided by a semiconductor manufacturing company in Kaohsiung, Taiwan through LabelImg [26], as shown in Figure 1. From the video, with 1805 frames recorded at the chip slots, we collected the data set, including training data for 1444 images and testing data for 361 images. We captured a part of the video frames in individual images and then manually marked the detected object boxes in parts of the image. Each picture has eight objects, and AOI can classify them into their respective categories as defective, empty, or occupied, as shown in Figure 7.

The experiment recorded the time required for detecting objects and calculated the inference time needed for an image. Equation (3) calculates the average inference time (*AIT*) required for a picture by the traditional and improved Yolov5 models given the same video stream. In Equation (3), *IN* signifies the number of frames in the video stream, and IT_{fgh} denotes the time for inferring an image to complete classification.

$$AIT_{fgh} = \frac{IT_{fgh}}{IN}, where \quad f = 1, 2, \ldots, \vartheta; \quad g = 1, 2, \ldots, \pi; \quad h = 1, 2, \ldots, \tau \tag{3}$$

This study mainly used the number of recognized frames per second and the average precision to evaluate the object detection performance. In chip contour detection, this study also presents a performance evaluation executed in the Jetson Nano embedded platform. Equation (4) calculates how many frames per second (FPS) the traditional and improved Yolov5 models can reach for rapid object detection in AOI, where $RAIT_{fgh}$ represents the time required for detecting an image from the video stream. Equation (5) calculates the mean average precision (mAP) of object detection by the traditional and improved Yolov5 models in AOI, where *cfgh* denotes the number of classes that the model needs to recognize, and AP_{cfgh} indicates the respective average precision in the number of classes.

$$FPS_{fgh} = \frac{1}{RAIT_{fgh}}, where \quad f = 1, 2, \ldots, \vartheta; \quad g = 1, 2, \ldots, \pi; \quad h = 1, 2, \ldots, \tau \tag{4}$$

$$mAP_{fgh} = \frac{APc_{fgh}}{c_{fgh}}, where \quad f = 1, 2, \ldots, \vartheta; \quad g = 1, 2, \ldots, \pi; \quad h = 1, 2, \ldots, \tau \tag{5}$$

This study applied the traditional and improved Yolov5 models to evaluate their respective performance in chip contour detection. According to Equation (2), this study used the Jetson Nano for rapid object detection at an image resolution of 480 × 640 and a specific frame rate. According to Equation (3), we can calculate the mean average precision of all classes obtained from the traditional and improved Yolov5 models trained with the same data set. The trained models were constructed for chip contour detection, and Table 1 lists the model parameters and computation quantity used by the traditional and improved Yolov5 models.

Table 1. Parameters and Flops in Chip Contour Detection.

Specification	Yolov5	M3-Yolov5	FGHSE-Yolov5	GSEH-Yolov5	DSGSE-Yolov5s
Parameter	7,251,912	3,205,296	2,606,944	4,182,136	2,213,726
Flop (Gflops.)	16.8	6.0	4.6	6.9	4.3

The experiment realized operations for saving the parameters of the best-performing model by using a callback function and visualizing the training process using a tool. In the training phase of the DSGSE-Yolov5s model, loss curves presenting its accuracy during training were drawn in six plots during 70 training epochs, as shown in Figure 11. In Figure 11, the first, second, and third columns are the positioning loss, the confidence level loss, and the matching loss between the prediction and the valid target, respectively. Next, the first and second rows signify the training and verification losses, respectively. In the second row, the abbreviation “val” means validation.

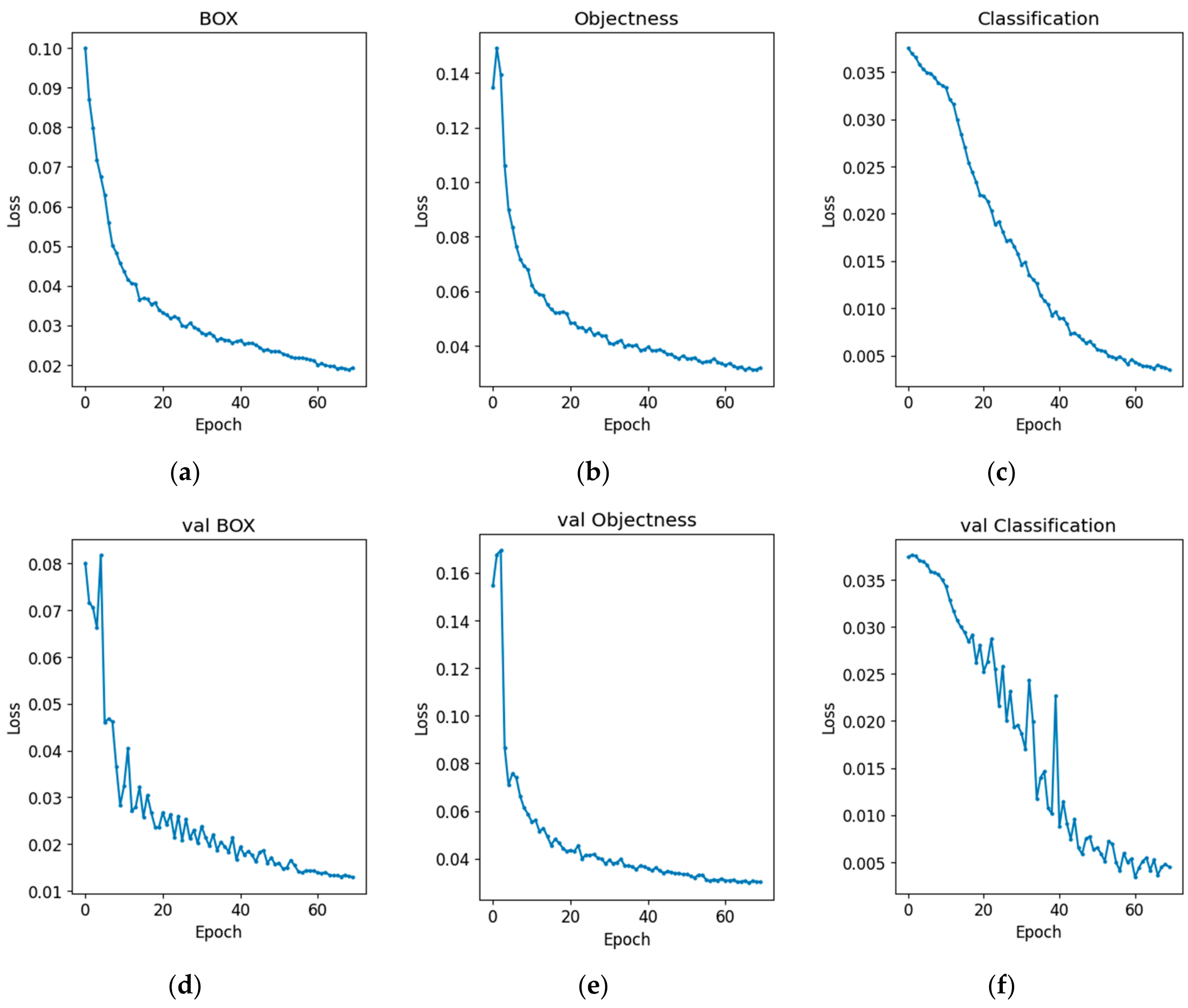


Figure 11. Loss plots in training DSGSE-Yolov5s. (**a**) Loss in box training. (**b**) Loss in objectness training. (**c**) Loss in classification training. (**d**) Loss in box validation. (**e**) Loss in objectness validation. (**f**) Loss in classification validation. Each plot indicates that the x-axis represents the error value and the y-axis stands for the number of epochs.

The trained model was tested for chip contour detection of 361 images to obtain the frame rate and precision. The precision–recall curve, abbreviated as the PR curve, can be used to estimate the precision of object detection during the testing phase, as shown in Figure 12. In Figure 12, every coordinate exemplifies a particular pair of recall and precision. In Table 2, Equation (2) gives the frames per second (FPS), called the frame rate, and Equation (3) provides the mean average precision (mAP). In summary, DSGSE-Yolov5s obtained the best performance, and Yolov5 gave the worst-case result. What is noteworthy about the PR curve here is how to accomplish the marginal improvement in precision. The proposed model, DSGSE-Yolov5s, can speed up the inference significantly by using DSGhost Conv as a substitute for traditional convolution, and it can uphold the detection precision by using GhostBottleneck and SElayer to replace Bottleneck_CSP. In Figure 9, we replaced ReLU with RMish in depthwise separable convolution (DS Conv) to increase the precision slightly.

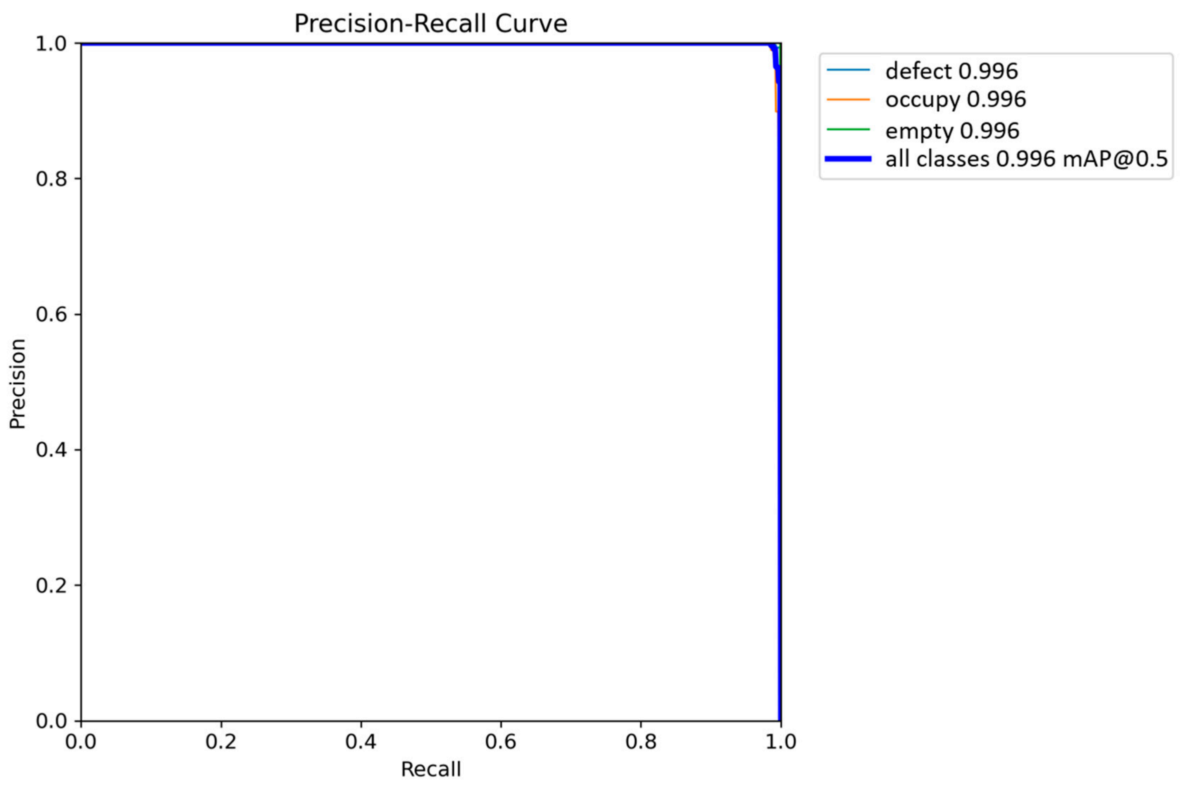


Figure 12. Precision–recall curve in testing DSGSE-Yolov5s.

Table 2. FPS and Precision in Chip Contour Detection.

Metrics	Yolov5	M3-Yolov5	FGHSE-Yolov5	GSEH-Yolov5	DSGSE-Yolov5s
FPS	5.75	6.46	10.13	8.47	13.39
Precision (%)	98.57	98.94	99.29	99.53	99.56
Recall (%)	98.01	98.27	98.86	99.14	99.22
F1-score	0.985	0.986	0.991	0.993	0.995
Accuracy (%)	98.91	99.13	99.33	99.51	99.55

After the chip contour detection test on the Jetson Nano, Table 2 shows the experimental results. According to the performance evaluation in Table 2, the proposed approach GSEH-Yolov5s achieved the best performance and outperformed Yolov5, M3-Yolov5, FGHSE-Yolov5, GSEH-Yolov5s, and DSGSE-Yolov5. Compared with Yolov5, GSEH-Yolov5s has improved FPS, precision, recall, F1-score, and accuracy, increasing by 132.87%, 1.00%, 1.23%, 1.02%, and 0.65%, respectively. The primary goal of this study is to detect damaged chips in chip slots during IC packaging and testing using on-site cameras. However, the current chip contour detection models, such as Yolov5, M3-Yolov5, FGHSE-Yolov5, and GSEH-Yolov5, running on the embedded platform, Jetson Nano, cannot keep up with the speed of operation of the new chip transportation machine. Therefore, this study proposed a faster object detection algorithm to respond to chip contour detection rapidly. The resilient detection speed of GSEH-Yolov5s reached up to 13.39 FPS, which can catch up with the operation speed of new chip transportation machines.

Improving the speed of object detection will inevitably involve changing the model's architecture, significantly improving it toward lightweight architecture. This paper focuses on lightweight architecture proposed to achieve high-speed object detection. The method proposed in earlier papers [8–12] was to make the model more lightweight and improve the speed of inference. These works are consistent with the purpose of this study, but we can provide better performance and accelerate inference. On the other hand, the object detection model must also consider the accuracy of image recognition. Otherwise, loss of precision will cause unexpected failures. Therefore, in our proposed method, we also specifically aim to maintain the accuracy of image recognition. The accuracy specifically improved in certain architectural blocks. The purpose discussed in earlier papers [13–17] is the same idea as our proposal to ensure the detection precision. Fortunately, the proposed approach can maintain the detection precision above a certain level.

The following experiment estimated chip contour detection success and failure cases to ensure that the proposed object detection and image recognition is effective and efficient. Assuming that the chip conveying machine can send eight chip bases through the camera in front of the conveyor per second, the video equipment can detect an image of each chip base within 0.125 s (i.e., the retrieval time per image from a camera). The access time per image through the model is 0.05 s. According to the FPS value of each detection model, as listed in Table 2, we can next calculate the inference time of each model for each image. The access time plus the inference time gives the processing time per image through the model. Then, the retrieval time divided by the processing time calculates the success rate per image the visual device processes. According to the accuracy, as listed in Table 2, Table 3 finally shows the yield of chip contour detection by multiplying it and the success rate. As a result, the proposed model DSGSE-Yolov5s obtained the best yield rate in the experiment.

Table 3. Yield in Chip Contour Detection.

Measures	Yolov5	M3-Yolov5	FGHSE-Yolov5	GSEH-Yolov5	DSGSE-Yolov5s
FPS	5.75	6.46	10.13	8.47	13.39
Inference time	0.1739	0.1548	0.0987	0.1181	0.0747
Processing time	0.2239	0.2048	0.1487	0.1681	0.1247
Success Rate	0.5583	0.6104	0.8405	0.7438	1.0025
Accuracy (%)	98.91	99.13	99.33	99.51	99.55
Yield (%)	55.22	60.50	83.49	74.01	99.80

In our previous work [3], the LWMG-Yolov5 algorithm was applied to chip contour detection to detect objects quickly. Furthermore, our previous article [4] delivered two advanced models, FGHSE-Yolov5 and GSEH-Yolov5, to accelerate chip contour detection, because the frame rate is more important than the precision in AOI. Furthermore, this study proposed the DSGSE-Yolov5s model, which is better than both models in boosting the execution speed and concurrently maintaining high precision. DSGSE-Yolov5s can escalate the FPS by up to 32.18% and slightly increase the precision by 0.27% when compared with FGHSE-Yolov5. Compared with GSEH-Yolov5, DSGSE-Yolov5s can promote the FPS by up to 58.09% and slightly increase the precision by 0.03%. Therefore, the proposed approach can achieve the acquired rate of fast AOI operations with rapid response.

In the chip packaging and testing process, the chip transportation speed of the new machines is fast. The operation may not significantly improve the IC packaging and testing yield rate if the AOI cannot produce high-resolution frames of live video streaming. However, the Jetson Nano embedded platform cannot produce higher-resolution images for real-time object detection. Dealing with this hardware limitation is a big issue. Therefore, we must find an advanced embedded platform, e.g., the Jetson AGX Xavier, that promptly resolves higher-resolution video streaming. On the other hand, supplying enough battery power is requisite for a high-performance embedded platform. Thus, we need a power-efficient embedded platform in the AOI application.

In this application, a carrier contains eight chip slots to accommodate eight small chips for transportation. Our proposed approach can detect eight chips simultaneously. Our proposed method can also perform AOI well for large-scale chip contour detection applications. Still, we need a larger carrier and larger chip slots to accommodate large chips and adjust the viewing distance of the camera for appropriate video capture. Besides chip contour detection, the proposed approach can extend to other applications. As long as many objects are in a picture, the classifier has to classify many objects exactly. Especially in everyday life, many applications require object detection, such as object tracking, street scene analysis, mask-wearing detection, gesture recognition, and obstacle detection.

5. Conclusions

This study's main contribution is introducing DSGSE-Yolov5s to provide a rapid object detection and image recognition response to boost AOI during IC packaging and testing. This approach can keep up with the fast operation of the new chip transportation machines. The experiment showed that DSGSE-Yolov5s can outperform M3-Yolov5, FGHSE-Yolov5, and GSEH-Yolov5. As a result, DSGSE-Yolov5s achieved the best performance in chip contour detection.

In future works, we seek a high-performance embedded platform like the Jetson AGX Xavier to outperform the Jetson Nano and Xavier NX in running high-speed chip object detection. On the other hand, we are still working on improving the visual algorithms to increase the speed of object detection. In other words, we will adopt a recently developed model, e.g., Yolov9, and use an advanced convolution method, e.g., dilated convolution, to further speed up the inference and maintain high precision in object detection. Next, we want to know whether this study's results can be used to gain a better manufacturing advantage in increasing chip yields and reducing losses in production costs during the IC packaging and testing process. In other words, we will check each visual algorithm's efficiency and calculate their respective operational times, factory overheads, chip yields, and manufacturing costs. Examining the chip yields and production costs during chip packaging and testing would be of greater benefit in terms of annual income.

Author Contributions: B.R.C. and F.-Y.C. conceived and designed the experiments; H.-F.T. collected the data set and proofread the manuscript; and B.R.C. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The Ministry of Science and Technology fully supported this work in Taiwan, Republic of China, under grant numbers NSTC 112-2622-E-390-001 and NSTC 112-2221-E-390-017.

Data Availability Statement: The Sample Programs for Sample Program.zip data used to support the findings of this study are available at https://drive.google.com/drive/folders/1HibUtORzDRl7taHIC0JOfiPAiobc3q95?usp=drive_link (accessed on 11 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, Y.L.; Chiang, Y.M.; Hsu, H.C. Capacitor Detection in PCB Using Yolo Algorithm. In Proceedings of the 2018 IEEE International Conference on System Science and Engineering, New Taipei, Taiwan, 28–30 June 2018; pp. 1–4.
2. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
3. Chang, B.R.; Tsai, H.-F.; Hsieh, C.-W. Location and Timestamp Based Chip Contour Detection Using LWMG-Yolov5. *Comput. Ind. Eng.* **2023**, *180*, 109277. [CrossRef]
4. Chang, B.R.; Tsai, H.-F.; Chang, F.-Y. Chip Contour Detection and Recognition Based on Deep-Learning Approaches. In Proceedings of the 2023 5th International Conference on Emerging Networks Technologies, Okinawa, Japan, 22–24 September 2023.
5. Chang, B.R.; Tsai, H.-F.; Chang, F.-Y. Boosting the Response of Object Detection and Steering Angle Prediction for Self-Driving Control. *Electronics* **2023**, *12*, 4281. [CrossRef]
6. Rajaram, R.N.; Ohn-Bar, E.; Trivedi, M.M. Refinenet: Refining Object Detectors for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2016**, *1*, 358–368. [CrossRef]
7. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
8. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21 July 2017; pp. 1800–1807.
9. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
10. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.

11. Lin, B.; Wang, J.; Yang, X.; Tang, Z.; Li, X.; Duan, C.; Zhang, X. Defect contour detection of complex structural chips. *Math. Probl. Eng.* **2021**, *2021*, 5518675. [CrossRef]
12. Zheng, P.; Lou, J.; Wan, X.; Luo, Q.; Li, Y.; Xie, L.; Zhu, Z. LED Chip Defect Detection Method Based on a Hybrid Algorithm. *Int. J. Intell. Syst.* **2023**, *2023*, 4096164. [CrossRef]
13. Dahai, L.; Zhihui, C.; Xianqi, L.; Qi, Z.; Nanxing, W. A lightweight convolutional neural network for recognition and classification for Si_3N_4 chip substrate surface defects. *Ceram. Int.* **2023**, *49*, 35608–35616. [CrossRef]
14. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
15. Aboah, A.; Wang, B.; Bagci, U.; Adu-Gyamfi, Y. Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and Yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5349–5357.
16. Wu, B.C.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 129–137.
17. Chang, B.R.; Tsai, H.F.; Chou, H.L. Accelerating the Response of Self-Driving Control by Using Rapid Object Detection and Steering Angle Prediction. *Electronics* **2023**, *12*, 2161. [CrossRef]
18. Cai, Y.; Luan, T.; Gao, H.; Wang, H.; Chen, L.; Li, Y.; Sotelo, M.A.; Li, Z. Yolov4-5D: An effective and efficient object detector for autonomous driving. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 4503613. [CrossRef]
19. Marco, V.S.; Taylor, B.; Wang, Z.; Elkhatib, Y. Optimizing Deep Learning Inference on Embedded Systems through Adaptive Model Selection. *arXiv* **2019**, arXiv:1911.04946. [CrossRef]
20. Sun, Y.; Wang, C.; Qu, L. An Object Detection Network for Embedded System. In Proceedings of the 2019 IEEE International Conferences on Ubiquitous Computing & Communications and Data Science and Computational Intelligence and Smart Computing, Networking and Services, Shenyang, China, 21–23 October 2019; pp. 506–512.
21. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016; pp. 770–778.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
24. Ding, W.; Huang, Z.; Huang, Z.; Tian, L.; Wang, H.; Feng, S. Designing Efficient Accelerator of Depthwise Separable Convolutional Neural Network on FPGA. *J. Syst. Archit.* **2019**, *97*, 278–286. [CrossRef]
25. Misra, D. Mish: A Self Regularized Non-monotonic Activation Function. *arXiv* **2020**, arXiv:1908.08681v3.
26. Saqlain, M.; Abbas, Q.; Lee, J.Y. A Deep Convolutional Neural Network for Wafer Defect Identification on an Imbalanced Dataset in Semiconductor Manufacturing Processes. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 436–444. [CrossRef]