

Swin transformer adaptation into YOLOv7 for road damage detection

Riyandi Banovbi Putera Irsal¹, Fitri Utaminingrum¹, Kohichi Ogata²

¹Department of Computer Science, Faculty of Computers Science, Brawijaya University, Malang, East Java, Indonesia

²Faculty of Advanced Science and Technology, Kumamoto University, Kumamoto, Japan

Article Info

Article history:

Received Sep 13, 2023

Revised Jan 12, 2024

Accepted Feb 12, 2024

Keywords:

Object detection

Road damage detection

Road damage detection dataset

Transformer

You only look once

ABSTRACT

Highways are an important component of any country. However, some highways in Indonesia endanger users while maintaining road safety. Crack detection early in the deterioration process can prevent further damage and lower maintenance costs. A recent study sought to develop a method for detecting road damage by combining the road damage detection (RDD) dataset with generative adversarial network technology and data augmentation to improve training. The current study aims to broaden the you only look once (YOLO) framework by incorporating the Swin Transformer into the chiral stationary phases (CSP) component of YOLOv7, with the goal of improving object detection accuracy in a variety of visual scenarios. The study compares the performance of various object detection models with varying parameters and configurations, such as YOLOv5l, YOLOv6l, YOLOv7-tiny, YOLOv7, and YOLOv7x. YOLOv5l has 46 million parameters and 108 billion floating point operations per second (FLOPS), whereas YOLOv6l has 59.5 million parameters and 150 billion FLOPS. With 31 million parameters and 140 billion FLOPS, the YOLOv7-swin model performs best with mean average precision (mAP), mAP_0.50 of 0.47. and mAP_0.5:0.95 of 0.232. The experimental results show that our YOLOv7-swin model outperforms both YOLOv7x and YOLOv7-tiny. The proposed model significantly improves object detection accuracy while keeping complexity and performance in balance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fitri Utaminingrum

Department of Computer Science, Faculty of Computers Science, Brawijaya University

Veteran Road, Ketawanggede, Subdistrict Lowokwaru, Malang City, East Java, Indonesia

Email: f3_ningrum@ub.ac.id

1. INTRODUCTION

Almost every country in the world uses highways to connect their regions. In fact, the total length of all roads worldwide is up to 64 million kilometers. Indonesia has a road length of nearly 550 thousand kilometers, which may appear insignificant when compared to the owner of the longest highway, the United States, which has 6.8 million kilometers of highways. However, some highways in Indonesia are dangerous, even endangering their users. Furthermore, maintaining road safety remains a major concern, particularly in countries such as Indonesia, where the prevalence of traffic accidents is alarmingly high, reaching 103,000 cases [1]. Early identification of cracks during the degradation process can avert subsequent harm and malfunction [2], as well as reduce maintenance expenses. One of the most crucial objectives for a strong pavement management system is to have a rapid, resilient, and cost-efficient algorithm for identifying pavement surface faults [3].

Currently, there are three methods for detecting road defects: manual inspection, automatic inspection, and image processing techniques. Manual inspection is obviously time-consuming and expensive. The use of sensor equipment in automated inspection may complicate matters [4]. The use of image processing techniques to detect road defects is the best option because it saves time and money. As a result, some researchers [5] have used it to detect road damage. Conventional methods for image processing usually involve segmenting pavement faults by manually selecting criteria like color, texture, and geometric characteristics. Subsequently, machine learning algorithms are employed for classification and matching in order to identify pavement damage. recent research on road defects can be found in the work of Roy and Bhaduri [6], who used the swin transformer on prediction head on you only look once (yolov5) and compared it with the transformer on prediction head on YOLOv5 which we also compared [7]. The current study aimed to create a method for detecting road damage using the road damage detection (RDD) dataset using YOLOv7 and using swin transformer to the chiral stationary phases (CSP) part. There are nine classes in the dataset, including wheel mark part, construction joint part, equal interval, construction joint part, partial pavement, pothole, crosswalk blur, white line blur, and utility hole [8]. The dataset is supplemented with generative adversarial network technology. Data augmentation is used to multiply the data, which makes the data more varied and thus improves training [9].

It is critical to review the relevant literature in order to comprehend the current landscape. Several notable works have made significant contributions to this field. Previous research, for example, has investigated methods such as YOLO for real-time object detection [10]. The YOLO series is important in the object detection task when it comes to one-stage detectors [11]. YOLO detects objects using three methods: image grid division, bounding box regression, and intersection over union (IoU). The image's grid was used to detect every object that appeared within it. The bounding box is an outline that draws attention to an object in an image. IoU is a technique used to avoid box overlap, resulting in only one box in one object. This project's expected output is the accuracy, computing time, and frame per second (FPS) of each YOLO algorithm's performance. Researchers previously compared the YOLO method to faster region-based convolutional neural network (RCNN) and solid state drive (SSD) [12]. The YOLO used is YOLOv5, which produces the best results of the three methods, with an accuracy of 93%. Another study compared three different YOLO algorithms: YOLOv3, YOLOv4, and YOLOv5. YOLOv5 has the highest accuracy, but YOLOv4 has a higher FPS than the other two methods [13]. The swin transformer is a novel visual recognition architecture that combines the strengths of transformers and convolutional neural networks (CNNs) [14]. The swin transformer, unlike traditional transformers that use fixed-size patches, employs a hierarchical design with a series of stages to process the input image at multiple scales. Each stage includes a self-attention mechanism for capturing global dependencies as well as a window partitioning strategy based on shifts for efficient computation. The swin transformer achieves competitive results on a variety of computer vision benchmarks while remaining efficient and scalable by leveraging the power of self-attention and CNNs' ability to model local context.

Despite the fact that previous approaches such as YOLO have demonstrated potential in real-time object recognition applications, the current study intends to expand the capabilities of the YOLO framework by incorporating the swin transformer into the CSP part of YOLOv7. The CSP's ability to model local settings and the swin transformer's ability to capture global relationships are expected to significantly improve object detection accuracy in a variety of visual scenarios. To compare the efficiency and scalability of this model to previous models, important performance measures such as mAP and FLOPS will be used.

2. RELATED WORK

2.1. Convolutional neural network

CNN is divided into two architectural components: feature extraction and fully-connected layers. Convolution and pooling layers are included in the extraction layer feature [15]. Convolution reduces the complexity of calculations by using the sliding window and weight sharing principles. The feature extraction layer retrieves extracted features as numbers, which are then entered into a fully-connected layer. There are convolution and pooling layers in this layer. Convolution layers operate on the sliding window and weight sharing principles. The CNN architecture is shown in Figure 1.

A filter length (pixel) and height (pixel) will be formed by the convolution layer. The first layer, for example, contains a convolution layer that is 3 pixels long, 3 pixels high, and 3 pixels thick. That is, the layer contains three filters drawn from the convolution layer's thickness. By operating a dot between the input and the filter, these three filters will be shifted to all parts of the image, resulting in output in the form of a feature mAP or activation mAP. Each convolution layer result will have an activation function. The activation function is a node at the end or between the Neural Network that determines whether or not the neuron will be activated. The most common activation is rectified linear unit (ReLU). ReLU is commonly used in neural

networks because it does not activate neurons at the same time, making it computationally efficient. ReLU will not activate all negative inputs, as shown in (1) and Figure 1.

$$ReLU = \max(0, x) \quad (1)$$

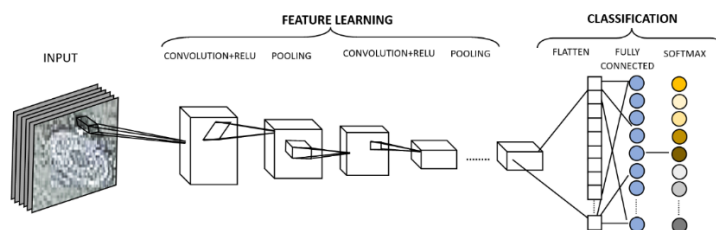


Figure 1. CNN architecture

The result of the convolution layer will go into the pooling layer. This layer consists of filters of a certain size and stride that will shift over the entire feature mAP area generated from the convolution layer. The pooling used is usually max pooling and average pooling. Max pooling will take the maximum value on each filter shift, while average pooling will select the average value of the filtered feature mAP area. The resulting dimensions of the feature mAP can be reduced when using layer pooling, so it can speed up computation because there are fewer parameters to update and overcome overfitting.

The feature mAP produced on the feature extraction layer is still in the form of a multidimensional array, so it must be flattened or reshaped into a vector so that it can be used as input from a fully-connected layer. The results of the flatten will be connected to a dense layer consisting of several nodes to find the classification results.

2.2. You only look once

YOLO is a real-world object detection algorithm. YOLO divides the image into grids by applying a neural network to the entire image. Each grid will have a confidence score, as well as bounding box predictions, which will be analyzed based on that score [16]. At the end of the process, the bounding box's final score is calculated; if the confidence score is less than 30%, the bounding box is discarded.

YOLO offers numerous advantages when compared to other conventional methods. YOLO uses CNN for object classification and localization. YOLO is quite fast, capable of processing images at rates ranging from 40 to 90 frames per second. It is faster than fast R-CNN which is faster than R-CNN [17]. YOLO has many versions as major versions. Each major version is released as a complete model in a smaller version with a reduced number of layers and is usually faster than the full version [18].

2.3. Swin transformer

The swin transformer architecture is revolutionary in the field of visual recognition. It incorporates the benefits of the transformer [19] and the CNN to enhance image object recognition performance. Transformer-based models have large receptive fields and have superior performance on data by large amounts [13]. The window partitioning stage of the swin transformer is where the majority of calculations are performed. The windows are generated by slicing the input image into segments of constant size. Then, each block undergoes a transformation using the self-attention mechanism. This mechanism enables the swin transformer to extract the image's global dependencies, i.e., the information that is distributed throughout the image. By observing the interactions between the blocks, the swin transformer is able to identify interconnected patterns in the image. Next, the calculation continues with the CNN mechanism processing each block. CNN enables the swin transformer to model the local context of each block by utilizing convolution operations that are highly efficient.

Swin transformer can optimize object detection in a variety of visual scenarios by combining a self-attention mechanism that captures global dependencies with CNN's ability to model local context [20]. Swin Transformer has attained competitive performance on a variety of image object recognition benchmarks thanks to its innovative approach. The strengths of the swin transformer are its capacity to combine global and local representations of each image block, as well as its scalability and computational efficiency. This architecture has created new possibilities for visual recognition and is a significant contribution to the advancement of object detection systems. The architecture of the swin transformer can be seen in Figure 2.

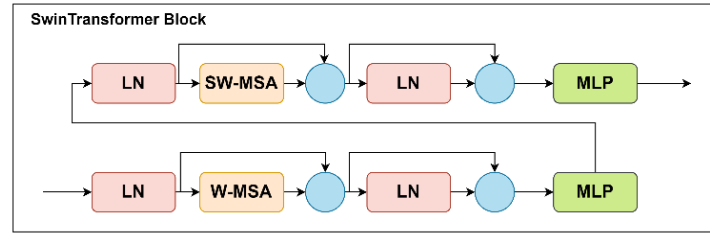


Figure 2. Swin transformer block

3. METHOD AND MATERIALS

3.1. Dataset

This study used the RDD dataset consisting of nine classes, including wheel track sections, construction joint sections, equal intervals, damaged road sections, potholes, blurred pedestrians, blurred white lines, and utility pits [8]. Generative adversarial network technology is used for data augmentation, which helps to increase the variety of data for a more effective and accurate training process [9]. The dataset used is obtained from Roboflow which consists of a total of 9,888 images [21]. Of these, 70% (6,921 images) are used for the training process, 20% (1,978 images) for validation, and the remaining 10% (989 images) are used for testing.

3.2. Proposed swin transformer adaptation into YOLOv7

The current study intends to expand the capabilities of the YOLO framework by adding the swin transformer into the CSP part of YOLOv7, even though previous approaches like. It is anticipated that the skill of the CSP in modelling local settings and the swin transformer's capacity to capture global relationships would considerably improve the accuracy of object detection in a variety of visual scenarios.

3.2.1. YOLOv7

The YOLOv7 model for single-stage object detection was introduced in 2022 [22]. It is based on the YOLOv3 architecture, but includes the following enhancements. A new, more accurate and efficient network backbone, a novel neck network that employs pyramid attention to enhance precision. A new head network that predicts image object bounding boxes, confidence scores, and class probabilities. On multiple object detection datasets, including COCO and VOC, YOLOv7 has been shown to be more accurate than YOLOv3. It is also more efficient, making it suitable for real-time object detection applications. YOLOv7 is a single-stage object detection model, which means it can estimate object bounding boxes, confidence scores, and class probabilities in a single pass. As a result, YOLOv7 outperforms two-stage object detection models such as Faster R-CNN [23].

Figure 3 shows the architecture of original YOLOv7 with spatial pyramid pooling with cross stage partial concatenation (SPPCSPC) on the head. SPPCSPC is a kind of neural network block found in YOLOv7 [24]. SPPCSPC is a hybrid of superficially porous particles (SPP) and CSP. It works by first using an SPP layer to pool the input tensor with different kernel sizes. The SPP layer's output is then routed through a CSP block. The CSP block's output is then passed through a final convolutional layer. SPPCSPC is used in YOLOv7 to improve network accuracy and speed. The SPP layer contributes to network accuracy by pooling input tensors with different kernel sizes. This aids in capturing features at various scales. The CSP block contributes to network speed by sharing weights between the two branches of the block. This reduces the number of parameters in the network, making training and inference faster.

The CSPDarknet53 architecture, a modified variant of the Darknet53 architecture, serves as the foundation of YOLOv7 for backbone [25]. The goal of the CSPDarknet53 architecture is to be more precise and efficient than the original Darknet53 architecture. CSPDarknet53's architecture is made up of 18 convolutional layers and 5 max-pooling layers. The convolutional layers extract features from the input image, which are then down sampled by the max-pooling layers. The collar of YOLOv7 is path aggregation network (PANet), a pyramidal attention network [26]. The PANet is designed to improve YOLOv7 precision by identifying long distance dependencies between features. The PANet is made up of three modules: the pyramid module, the attention module, and the fusion module. The pyramid module is in charge of creating a feature pyramid from the backbone. The attention module is in charge of comprehending features' long-term dependencies. The pyramid module and the attention module are combined in the fusion module. YOLOv7's leader is also the leader of YOLOv3. The YOLOv3 head is intended to estimate the bounding boxes of image objects, confidence scores, and class probabilities [27]. The YOLOv3 head is made up of two convolutional

layers and one completely connected layer. Convolutional layers are in charge of extracting features from the features of the neck. The fully connected layer is in charge of predicting bounding boxes, confidence scores, and class probabilities.

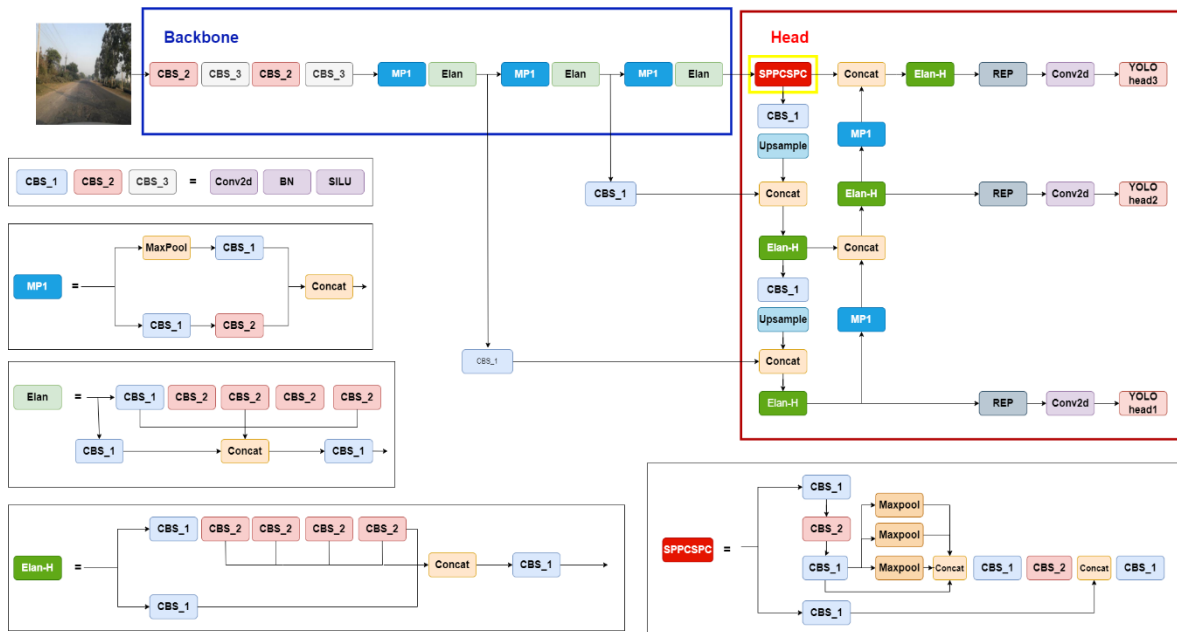


Figure 3. Original YOLOv7 architecture

3.2.2. Proposed STCSPC on YOLOv7

The suggested approach involves integrating the YOLOv7 model with the swin transformer by a modification of the SPPCSPC component in the YOLOv7 head, which is transformed into STCSPC, as shown in Figure 4 in the yellowbox. The YOLOv7 model incorporates the SPPCSPC technique to effectively integrate spatial information derived from the features produced by the preceding layer. Nevertheless, with the substitution of this particular segment with STCSPC, we may exploit the inherent capabilities of swin transformers to effectively capture more robust spatial and contextual linkages.

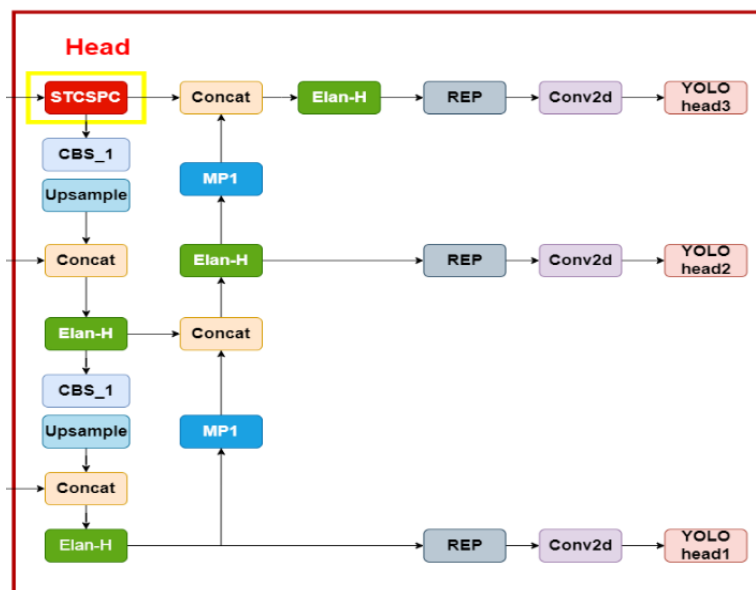


Figure 4. YOLOv7-swin head

The swin transformer is an architectural framework that utilises self-attention methods with computationally efficient properties. In the proposed methodology, following the extraction of features using a convolutional layer, the obtained features will undergo processing via the swin transformer layer in order to generate a more robust representation. The swin transformer layer incorporates self-attention mechanisms to effectively integrate spatial input from all elements, hence enhancing the contextual representation.

Following the passage via the swin transformer layer, the resultant feature representation will undergo processing by the STCSPC layer. The STCSPC aims to integrate spatial information using a space pyramid similar to the SPPCSPC, but by leveraging the representations acquired via the swin transformer. Therefore, the integration of STCSPC will enable the fusion of more robust spatial and contextual data, thereby augmenting the object detection capabilities of YOLOv7. The anticipated outcome of integrating the swin transformer and STCSPC into the YOLOv7 head is an enhancement in object detection capabilities, namely in capturing intricate objects and boosting the accuracy of detection.

The STCSPC module, which shown in Figure 5, gets input in the form of a tensor with shape (batch_size, c1, height, width). The input will enter into two CBS_1 which is a 1x1 convolution layer. CBS_1 will reduce the number from input channel c1 to hidden channel size c_ so that it has the output shape (batch_size, c_, height, width). The second CBS_1 works like the first one so that it has the same output.

The output of the first CBS_1 will enter the swin transformer block. This block performs a series of computations, including self-attention and feed-forward neural network layers. The output of the swin transformer block is also a shape tensor (size_batch, c_, height, width). The swin transformer block output tensor, $y1'$, is further processed by the same 1x1 convolution layer as the previous convolution, CBS_1. This layer helps in refining and transforming the features generated by the swin transformer block. The resulting tensor is denoted as $y1''$ and has the same shape as the original (batch_size, c_, height, width). Meanwhile, the original input tensor x is processed by the convolution layer $cv2$, resulting in the previously mentioned tensor $y2$.

Finally, the tensors $y1''$ which are the output of CBS_1 of the swin transformer block result and $y2$ which is the second CBS_1 are combined along the channel dimension (dimension 1), resulting in a tensor of the form (batch_size, $2*c_$, height, width). This combined tensor is then processed by the convolution layer of CBS_1, which uses 1x1 convolution to reduce the channel dimension from $2*c_$ to the desired output channel size $c2$. The output tensor form is (batch_size, $c2$, height, width). In summary, the input tensor x is convolved, self-attention operation in swin transformer block, and combined to produce the final output tensor output. By utilising partial connections between different parts of the network, the CSPC bottleneck structure improves information flow and feature representation.

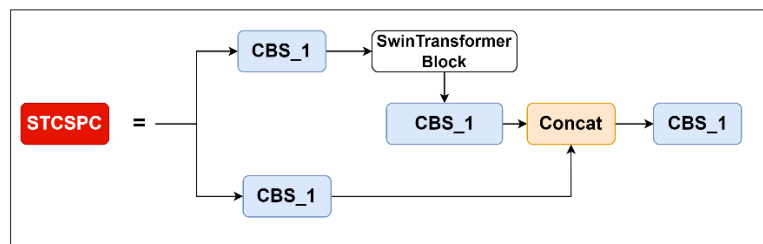


Figure 5. STCSPC module

4. RESULTS AND DISCUSSION

In this section, the proposed model's performance is compared to the YOLOv5 large, YOLOv5 with swin transformer, and YOLOv7. The confusion matrix is used to analyze the performance of both methods in classifying fault types in order to evaluate their performance. The confusion matrix is a tabular representation that provides the count of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), allowing the classification accuracy and method performance to be visualized.

Precision is a measure of the accuracy of a model's positive predictions. It is the proportion of TP predictions (objects correctly detected) to the sum of true positive and FP predictions (objects incorrectly detected). In (2) is used to calculate precision.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall, also known as sensitivity or true positive rate, measures the completeness of a model's positive predictions. It is the ratio of correct positive predictions to the sum of correct positive and incorrect negative FN predictions (missed objects). In (3) is used to calculate recall.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

IoU is a measure of overlap between the predicted and true bounding boxes. The area of intersection between the two bounding boxes is divided by the area of their union. The mAP evaluates an object detection model's overall performance by combining precision and recall across different IoU thresholds. It is calculated as the mean of the precision values at different recall levels. In (4) is used to calculate mAP score, which is calculated by averaging the AP over all classes and/or the total IoU thresholds, depending on the detecting problems.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (4)$$

AP_k = the AP of class K
 n = the number of classes

The mAP is computed by evaluating the IoU at various thresholds for each class k. The overall mAP for the test data is obtained by averaging the mAP values for each class. Table 1 shows the performance comparison among several detection models. We compared the results of the previous model, which was trained with 400 epochs and a similar image size. This refers to the number of parameters used in the model, which is related to the model's complexity. FLOPS which describes how quickly the model can perform computational operations. The greater the value, the greater the number of computational operations the model can perform in one second. The YOLOv5l, YOLOv5l-tph-plus, YOLOv6l, YOLOv7-tiny, YOLOv7, and YOLOv7x models, which are the baseline models, have been tested with various parameters and configurations to evaluate the object detection performance in various scenarios, as shown in the table. The YOLOv5l model has 46 million parameters and 108 billion FLOPS, with mAP_0.50 reaching 0.457 and mAP_0.5:0.95 reaching 0.216. Meanwhile, YOLOv6l had 59.5 million parameters and 150 billion FLOPS, with mAP_0.50 increasing to 0.459 and mAP_0.5:0.95 increasing to 0.232. With 6.2 million parameters and 13 billion FLOPS, YOLOv7-tiny has a mAP_0.50 similar to YOLOv5l, but a slightly lower mAP_0.5:0.95 of 0.202. With 36 million parameters and 105 billion FLOPS, YOLOv7 increases mAP_0.50 to 0.462 and mAP_0.5:0.95 to 0.226. YOLOv7x, on the other hand, has 71 million parameters and 189 billion FLOPS, as well as mAP_0.50 of 0.46 and mAP_0.5:0.95 of 0.219. With 31 million parameters and 140 billion FLOPS, the proposed model, YOLOv7-swin, performs best, with mAP_0.50 of 0.47 and mAP_0.5:0.95 of 0.232. These findings demonstrate that the proposed model is capable of significantly improving object detection accuracy while maintaining a good balance between model complexity and detection performance. The RDD dataset shows a variety of detection outcomes. Indeed, the obtained results might seem modest compared to the performance of DenseSPH-YOLOv5, which achieved mAP of 0.85 [6]. This difference is primarily attributed to the utilization of the same dataset, albeit with different partitions and number of classes, where they use eight classes instead of nine. As a result, the outcomes we obtained were not as significant as those documented in DenseSPH-YOLOv5. Additionally, we conducted a comparative analysis using the same model employed in DenseSPH-YOLOv5, namely YOLOv5l-tph [7], achieving mAP of 0.459, while their reported value was 0.77. The photographs used to depict the test results are representative of the overall findings. We have conducted an experiment to try our model YOLOv7-swin (Figure 6(a)) and compared it with some other models such as YOLOv7x (Figure 6(b)) and YOLOv7-tiny (Figure 6(c)). The experimental outcome depicted in image indicates that our model (Figure 6(a)) achieves the detection result, while the other two models fail to do so (refer to the image contained within the red box). Moreover, with regard to the image depicted in the yellow box, our model demonstrates a higher level of detection capability than both the YOLOv7x (Figure 6(b)) model (one detection) and the YOLOv7tiny (Figure 6(c)) model (no detection).

Table 1. The performance of detection models

Method	Parameter	FLOPS	Img_Size	mAP_0.50	mAP_0.5:0.95
YOLOv5l	46M	108G	640	0.457	0.216
YOLOv5l-tph-plus	41M	160G	640	0.459	0.215
YOLOv6l	59.5M	150G	640	0.458	0.232
YOLOv7-tiny	6.2M	13G	640	0.457	0.202
YOLOv7	36M	105G	640	0.462	0.226
YOLOv7x	71M	189G	640	0.46	0.219
YOLOv7-swin (Ours)	31M	140G	640	0.47	0.232

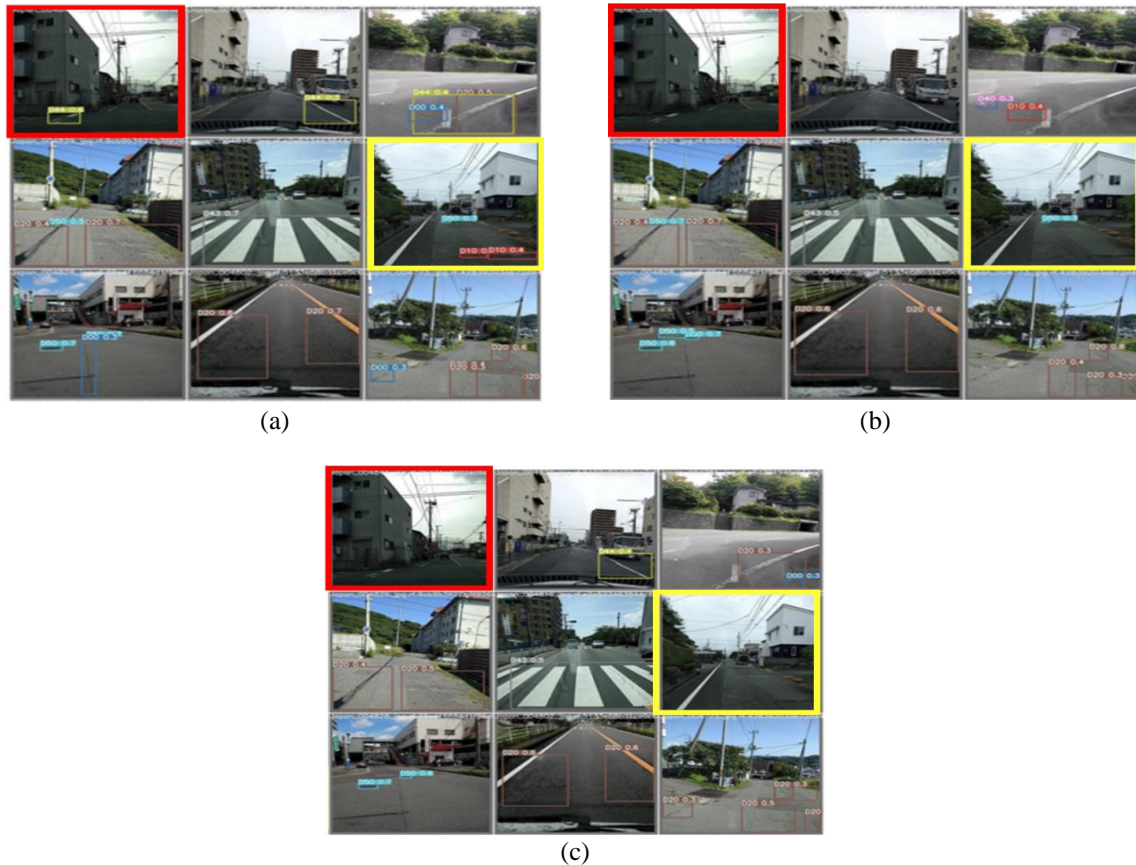


Figure 6. Some visualization results from YOLOv7-Swin; (a) our YOLOv7x, (b) YOLOv7-tiny, and (c) on RDD testset, different categories use colored bounding boxes

5. CONCLUSION

The YOLOv7-Swin approach, which integrates the YOLOv7 model with the swin transformer, demonstrates superior performance compared to its competitors. The proposed methodology entails modifying the SPPCSPC component of the YOLOv7 head architecture to STCSPC. Based on the examination of the obtained findings, it can be inferred that the object detection performance of the YOLO method is influenced by the parameters size and FLOPS. Typically, detection performance is enhanced by employing approaches characterised by greater parameter sizes and a higher number of FLOPS. Nevertheless, there exist certain cases when the YOLOv7-Swin (ours) approach deviates from the norm. Despite having comparatively smaller parameter sizes, it surpasses alternative methods in terms of its ability to accurately recognise objects. The results obtained from this approach demonstrate the highest mAP values at IoU thresholds of 0.50 and 0.5 to 0.95, with respective values of 0.47 and 0.232. These findings indicate that this method outperforms other approaches in terms of accuracy and precision. The experimental results show that our YOLOv7-swin model outperforms both YOLOv7x and YOLOv7-tiny. In comparison to DenseSPH-YOLOv5, which achieved an mAP of 0.85, our obtained results appeared relatively modest, likely due to variations in dataset partitioning and the number of classes. Furthermore, future work should focus on refining the dataset for more accurate and comprehensive evaluations.

ACKNOWLEDGEMENTS




We would like to acknowledge Brawijaya University for providing the necessary resources, facilities, and conducive research environment that have facilitated the completion of this project. The academic and research opportunities provided by Brawijaya University have been essential in fostering my intellectual growth and nurturing my passion for knowledge. Together, the Faculty of Computer Science, Universitas Brawijaya, and the Graduate School of Science and Technology, Kumamoto University, have played a pivotal role in shaping our academic and personal development. We carry the lessons and experiences garnered from both institutions as we step into the future, inspired to make a positive impact in

our respective fields and society as a whole. In addition, we would like to recognize and convey our deep appreciation to Professor Research Grant Faculty of Computer Science (FILKOM), Universitas Brawijaya Number: 2118/UN10.F15/PN/2023, for their wonderful assistance. The funding has facilitated our pursuit and exceptional performance in our research pursuits, so contributing to the progress of knowledge in the field of computer science.




REFERENCES

- [1] A. Karnadi, "The number of traffic accidents will increase to 103,645 in 2021, (in Indonesian: Jumlah Kecelakaan Lalu Lintas Meningkatkan Jadi 103.645 pada 2021)," 2022. <https://dataindonesia.id/otomotif-transportasi/detail/jumlah-kecelakaan-lalu-lintas-meningkat-jadi-103645-pada-2021> (accessed Oct. 20, 2023).
- [2] D. Dhital and J. R. Lee, "A Fully Non-Contact Ultrasonic Propagation Imaging System for Closed Surface Crack Evaluation," *Experimental Mechanics*, vol. 52, no. 8, pp. 1111–1122, Oct. 2012, doi: 10.1007/s11340-011-9567-z.
- [3] S. A. Hosseini and O. Smadi, "How prediction accuracy can affect the decision-making process in pavement management system," *Infrastructures*, vol. 6, no. 2, 2021, doi: 10.3390/infrastructures6020028.
- [4] M. E. Torbaghan, W. Li, N. Metje, M. Burrow, D. N. Chapman, and C. D. F. Rogers, "Automated detection of cracks in roads using ground penetrating radar," *Journal of Applied Geophysics*, vol. 179, Aug. 2020, doi: 10.1016/j.jappgeo.2020.104118.
- [5] N. Safaei, O. Smadi, A. Masoud, and B. Safaei, "An Automatic Image Processing Algorithm Based on Crack Pixel Density for Pavement Crack Detection and Classification," *International Journal of Pavement Research and Technology*, vol. 15, no. 1, pp. 159–172, Jan. 2022, doi: 10.1007/s42947-021-00006-4.
- [6] A. M. Roy and J. Bhaduri, "DenseSPH-YOLOv5: An automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism," *Advanced Engineering Informatics*, vol. 56, Apr. 2023, doi: 10.1016/j.aei.2023.102007.
- [7] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2778–2788, Oct. 2021, doi: 10.1109/ICCVW54120.2021.00312.
- [8] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, Jan. 2021, doi: 10.1111/mice.12561.
- [9] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding Data Augmentation for Classification: When to Warp?" in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–6, Nov. 2016, doi: 10.1109/DICTA.2016.7797091.
- [10] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *Journal of Physics: Conference Series*, vol. 1004, p. 012029, Apr. 2018, doi: 10.1088/1742-6596/1004/1/012029.
- [11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
- [12] J. Kim, J.-Y. Sung, and S. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," in *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, pp. 1–4, Nov. 2020, doi: 10.1109/ICCE-Asia49877.2020.9277040.
- [13] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs," *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.
- [14] Y. Lin *et al.*, "Swin transformer: hierarchical vision transformer using shifted windows," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9992–10002, 2021.
- [15] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
- [16] S. Shinde, A. Kothari, and V. Gupta, "YOLO based Human Action Recognition and Localization," *Procedia Computer Science*, vol. 133, pp. 831–838, 2018, doi: 10.1016/j.procs.2018.07.112.
- [17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020, doi: 10.48550/arXiv.2004.10934.
- [18] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms," *Agronomy*, vol. 12, no. 2, Jan. 2022, doi: 10.3390/agronomy12020319.
- [19] A. Dosovitskiy *et al.*, "an Image Is Worth 16X16 Words: Transformers for Image Recognition at Scale," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.
- [20] Z. Liu, Y. Tan, Q. He, and Y. Xiao, "SwinNet: Swin Transformer Drives Edge-Aware RGB-D and RGB-T Salient Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4486–4497, Jul. 2022, doi: 10.1109/TCSVT.2021.3127149.
- [21] R. Damage, "RDD Dataset," *Roboflow Universe*, 2022.
- [22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," pp. 7464–7475, 2023, doi: 10.1109/cvpr52729.2023.00721.
- [23] M. P. D. Cahyo and F. Utaminigrum, "Autonomous Robot System Based on Room Nameplate Recognition Using YOLOv4 Method on Jetson Nano 2 GB," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 1, pp. 117–123, 2022, doi: 10.30630/joiv.6.1.785.
- [24] S. Xu *et al.*, "A Locating Approach for Small-Sized Components of Railway Catenary Based on Improved YOLO With Asymmetrically Effective Decoupled Head," *IEEE Access*, vol. 11, pp. 34870–34879, 2023, doi: 10.1109/ACCESS.2023.3264441.
- [25] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "ICCV 2019 Open Access Repository," *ICCV 2019 Open Access Repository*, 2019.
- [26] M. Cheng *et al.*, "Hybrid Transformer and CNN Attention Network for Stereo Image Super-resolution," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2023-June, pp. 1702–1711, 2023, doi: 10.1109/CVPRW59228.2023.00171.
- [27] M. Elisisi, M.-Q. Tran, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings," *Sensors*, vol. 21, no. 4, Feb. 2021, doi: 10.3390/s21041038.




BIOGRAPHIES OF AUTHORS

Riyandi Banovbi Putera Irsal    was born on November 3, 2001. He is a graduate of Computer Engineering from Brawijaya University in 2023. Currently, he is pursuing a Master's degree in Computer Science at Brawijaya University. He specializes in the field of computer vision and artificial intelligence. Riyandi actively engages in research and development in the areas of computer vision and artificial intelligence technologies. For any inquiries or communication. He can be contacted at email: riyandiirsal1st@gmail.com.



Fitri Utamingrum    a renowned academic, was born in Surabaya, East Java, Indonesia. She completed her Bachelor's degree in Electrical Engineering at the National Institute of Technology. Building upon her foundation, she pursued a Master's degree in the same field at Brawijaya University in Malang, Indonesia. In addition, she obtained a Doctor of Engineering in the field of Computer Science and Electrical Engineering from Kumamoto University, Japan. She currently holds the esteemed position of Professor in the Faculty of Computer Science at Brawijaya University, Indonesia. With a passion for cutting-edge research, she's primary focus is in the field of smart wheelchairs, particularly in the development of computer vision algorithms. She can be contacted at email: f3_ningrum@ub.ac.id.



Kohichi Ogata    was born in Kumamoto, Japan, in 1967. He obtained the B.E., M.E., and Ph.D. degrees in Engineering from Kumamoto University, Kumamoto, in 1989, 1991, and 1994, respectively. He presently holds the position of Associate Professor in the Department of Computer Science and Electrical Engineering at Kumamoto University's Faculty of Advanced Science and Technology. His research focuses on signal processing, voice processing, and image processing. Specifically, he is interested in measuring the speech production process and developing applications related to it, creating eye-gaze interface systems, and designing augmented reality application systems. He can be contacted at email: ogata@cs.kumamoto-u.ac.jp.