Lightning YOLOv4 for a Surface Defect Detection System for Sawn Lumber

Fityanul Akhyar
Department of Electrical Engineering
Yuan Ze University – Taiwan
School of Electrical Engineering
Telkom University – Indonesia
fityanul@telkomuniversity.ac.id

Ledya Novamizanti
School of Electrical Engineering
Telkom University – Indonesia
ledyaldn@telkomuniversity.ac.id

Trianusa Putra
School of Electrical Engineering
Telkom University – Indonesia
trianusaputra@student.telkomuniversity.ac.id

Elvin Nur Furqon
Artificial Intelligence –
High Performing Computer Division,
Nova Global Pte Ltd - Singapore
elvin@novaglobal.com.sg

Ming-Ching Chang
Department of Computer Science,
University at Albany – USA
mchang2@albany.edu

*Chih-Yang Lin
Department of Electrical Engineering,
Yuan Ze University – Taiwan
andrewlin@saturn.yzu.edu.tw

Abstract

Lumber is a primary material for the production of various types of wood products. However, many industries still carry out the lumber quality inspection process manually, relying on human sight and instinct to compare many similar objects. To streamline the inspection process, this study developed a deep learning-based surface defect detection system with a proposed "lightning YOLOv4" model. Specifically, to improve the model's performance speed, we simplify CSPDarknet53 and path aggregation network (PANet) for the feature extraction stage of YOLOv4 by reducing the convolution layers. Moreover, we introduce the simplification technique to reduce the number of channels in CSPDarknet53 by multiplying it with the scaling coefficient. In addition, we add spatial attention module (SAM) to the structures, which can improve whole system performance on two types of lumber datasets (pine and rubber lumber). According to the experimental results, the proposed detection system improves the average precision of defect localization with the highest gap of 1.3%, as well as improves the frames per second (FPS) by 10.8 points over the baseline.

Keywords: Surface defect inspection system; YOLOv4, CSPDarknet53, PANet, SAM.

1. Introduction

Even in modern times, lumber cannot be entirely replaced as a building material, and continues to be used, especially in wood products. In general, companies that engage in lumber production still carry out the quality inspection process manually, relying on the sense of sight and instinct to compare many similar objects. However, observations with human eyes can only give accuracy between 50-60% [8], which makes the accuracy and efficiency of production time variable and subjective, and hinders the performance of the lumber industry. The deep learning field poses a potential solution for the lumber industry to automatically inspect processed products with consistent quality standards.

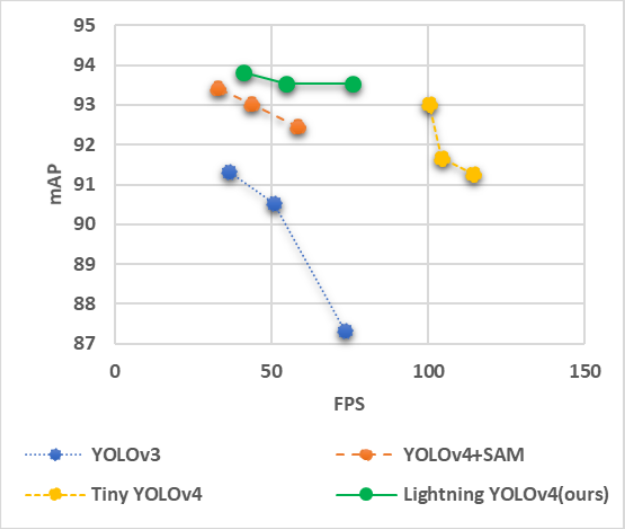


Figure 1. The comparison on rubber lumber dataset.

The proposed works in [1] included an experimental study of defects in lumber surface datasets. This method applied the state-of-the-art deep learning approach, YOLOv3. In the results on the rubber lumber dataset in Figure 1, YOLOv4 [1] achieved higher accuracy, but had a slower processing speed. That is because the YOLOv4 architecture is more complex than that of YOLOv3. The detection system of YOLOv4 consists of 167 processing layers; consequently, the system detection process takes longer than it does with YOLOv3. On the other hand, the tiny version of YOLOv4, or "tiny YOLOv4[1]", proposes compressing the convolution layers down to 53 to enhance the processing speed. However, this slim version delivers suboptimal performance as the complexity of the dataset increases.

Motivated by the issues highlighted above, this research sought to develop a defect detection system for sawn lumber by simplifying the YOLOv4 model. We call the proposed model "lightning YOLOv4". The experimental results show that the proposed approach significantly increases speed and performance compared to the previous methods [1, 12]. The remaining sections of this paper will present in detail the design of the proposed method.

2. Related works

The objective of this work is to improve the performance of the state-of-the-art model YOLOv4 [1] or proposed lightning YOLOv4. This section overviews the detection baseline, activation function, and attention module, which are the three main components of our proposed method.

2.1. Detection baseline

YOLOv4 [4] was developed from the YOLOv3 [10] model, and has shown improvement in Accuracy of Performance (AP) and Frames Per-Second (FPS). There are five essential components of the YOLOv4 architecture. The first is a backbone, which serves as the foundation of the YOLOv4 method. The second is the neck, which functions as a connector between the foundation and the detection process. And lastly, dense prediction with a sparse prediction function is used as an object detection algorithm.

Most of the architecture of YOLOv4 is nearly identical to that of YOLOv3; however, there are key differences between the two versions. The first distinction is that YOLOv3 uses Darknet-53 as a backbone, while YOLOv4 uses CSPDarknet-53 [13]. This new extractor has a more complex arrangement that includes 29 convolutional layers with 3 x 3 layers and a 725 x 725 receptive field, and 27.6 million parameters. With the improvements described in this section, the model is able to offer a better backbone because it runs more effectively. The next distinction lies in the detection enhancement methods. YOLOv3 only uses Feature Pyramid Networks (FPN) [4] to extract the multiscale features, while YOLOv4 uses many more methods, such as mosaic data augmentation, IoU-loss [11], cross mini-Batch Normalization (CmBN), DropBlock regularization, self-adversarial training, elimination of grid sensitivity, multiple anchors for single ground truth, cosine annealing scheduler [6], optimal hyper-parameters, and random training shapes. Therefore, YOLOv4 can produce Frames Per Second (FPS) faster and more accurately.

2.2. Activation Function

The purpose of using activation is to find the optimal accuracy value and reduce errors. The model's weights should be changed to minimize the loss function during the training process to make predictions as accurate as possible. Activation unifies the loss function and model parameters by updating the model to obtain the loss function's output. In short, the optimizer will use its weights to shape the model we have into the most appropriate form [21].

The comparison experiments in [7] show that the Mish activation function was more effective than the leaky ReLU activation function in neural networks for improving classification accuracy. Mish uses the self-gating property, where the non-modulated input is multiplied by the output of a non-linear input function. In order to retain a small amount of negative information, Mish eliminates some of these negative values by designing an activation like the ReLU [2] phenomenon. This property helps achieve better expressivity and flow of information. Being infinite, Mish avoids saturation, which generally causes training to slow down drastically as the gradient approaches zero. Unlike Leaky-ReLU [15], inspired by Swish [9], Mish is continuously distinguishable, which is an attractive property because it avoids singularities, which are undesirable side effects when performing gradient-based optimizations.

2.3. Attention Module

SAM, or Spatial Attention Module, is a module for spatial attention in convolutional neural networks [14]. SAM generates a spatial map of attention by utilizing the inter-spatial relationship of features. In contrast to the attention channel [3, 5], spatial attention focuses on which part is informative. In the original SAM, maximum and average pooling are applied separately to the input feature maps to generate feature maps. The results are fed into a convolution layer to generate spatial attention, then

[1] https://github.com/AlexeyAB/darknet

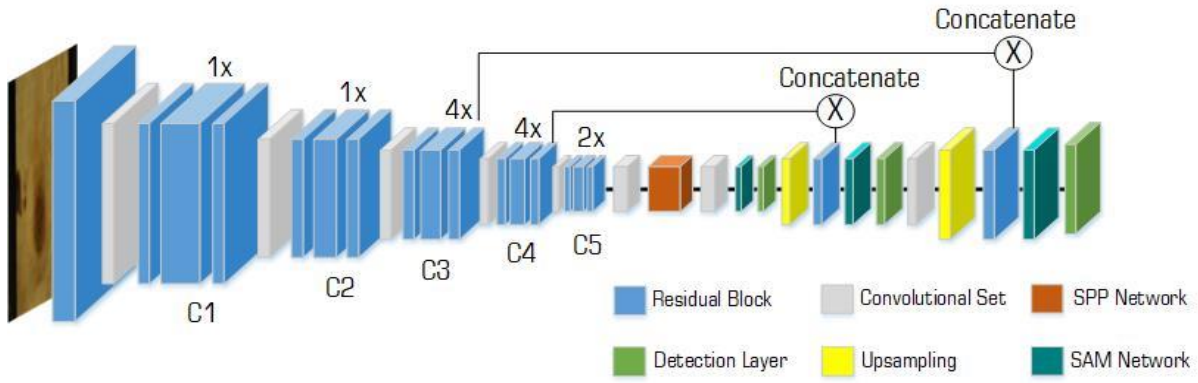


Figure 2. Proposed lightning YOLOv4 Architecture.

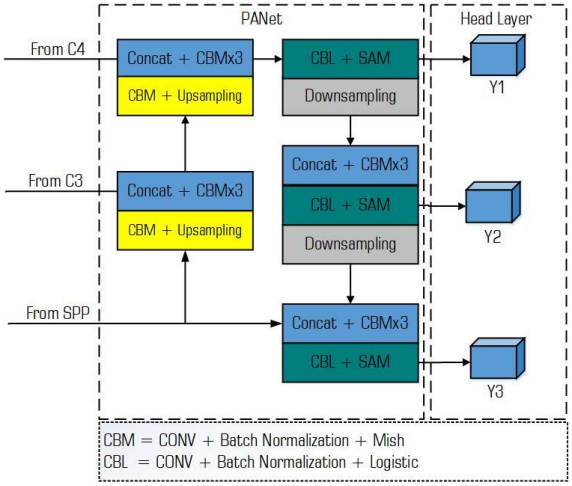


Figure 3. Proposed lightning YOLOv4 head layer.

followed by a sigmoid function. In YOLOv4, SAM is modified so that maximum and average pooling are not applied.

3. Proposed work

This study designed a lumber defect detection system using lumber image data from rubber and pine trees as the inputs. The system model was designed with a modified YOLOv4 model as its baseline architecture. A modified YOLOv4 architecture compressed YOLOv4 to make it simpler and faster. There are three main modifications in the modified YOLOv4. First, we simplified CSPDarknet53 as shown in Table 1. Not only did our model reduce the residual block network, but it also carried out the simplification method by reducing the number of specific channels from the original backbone through multiplication with the scaling coefficient α. This new technique uses an α between 0.6 up to 1 so that the number of channels will decrease according to this coefficient. Next, the residual block network of PAN is reduced as shown in Figure 3, repeated from five to three times. Both of these architectural changes were made to speed up the training and testing processes. Second, Mish activation was substituted in place of the Leaky-ReLU used in the original PANet convolution layers, to increase accuracy. This approach also allows information to better penetrate the neural networks' structure and allow the gradient to propagate more efficiently so that the neural networks will achieve better classification accuracy and generalization. Third, the Spatial Attention Module was added to the prediction head to increase the localization performance.

Figure 2 gives a comprehensive view of the complete proposed framework architecture, which is referred to as lightning YOLOv4. It consists of 83 convolution layers, 12 shortcut layers, 1 SPP layer, 21 route layers, 3 detection layers, and 2 up-sampling layers. Lightning YOLOv4 yields three feature maps in 68x68, 32x32, and 16x16 sizes, if the input image size is set to 512 × 512.

4. Ablation and results

Tests were conducted under three scenarios to obtain the best experimental results in a single Tesla P100 16GB GPU. The sawn lumber dataset[2] for each scenario contained 832 images of pine lumber and 1,553 images of rubber lumber, where each category was partitioned 70:10:20 for training, validation, and testing, respectively. The pine lumber category had two classes: dead knot and intergrown knot. In contrast, the rubber lumber category had four classes: dead knot and intergrown knot, growth shake, and inbark. The default input image size used in these scenarios was set to 320x320. Scenarios one, two, and three, (or backbone simplification, activation function selection and attention module selection), used mean average precision (mAP) as a reference to get the best configuration. The comparison

[2] https://github.com/WenHe-Hnu/sawn-lumber-dataset

of results from our proposed system versus the baseline methods are shown at the end of this section.

4.1. Backbone network simplification

The simplification of channels in the proposed lightning YOLOv4 architecture in the first scenario is shown in Table 2. This simplification was done by reducing the number of layers in the modified CSPDarknet53 by multiplying by the specified scale coefficient α. The scale ranged from 0.6 up to 1. This operation aims to determine the effect of simplification on speed performance or FPS as compared to the original architecture. The testing results show that the simplification with a coefficient of 0.8 obtains the best performance among all the numbers tested.

4.2. Activation function selection

In the second scenario, Mish activation was used. To demonstrate the efficiency, five types of activation functions were applied in the proposed modified YOLOv4 network: original (Mish + ReLU), ReLU, Leaky-ReLU (L-ReLU), Swish, and Mish. The network architecture employed in this stage was modified according to the best accuracy (mAP) on the results from the first scenario. This configuration aimed to determine which activation function could improve accuracy on our proposed lightning YOLOv4 architecture.

According to Table 3, the best accuracy was achieved using a Mish activation of 91.95% for pine lumber, and 90.7% for rubber lumber. In the Mish activation function: the positive values could go to any height, and it will avoid saturation due to capping. In theory, the small allowance for negative values should permit better gradient flow vs. a complex zero bound as in ReLU. Due to this, Mish contributes to the highest accuracy across the different categories for both datasets, among other activation functions.

4.3. The attention module selection

In the third scenario, an attention module, spatial attention module (SAM), was added to modified YOLOv4 (Mod-YOLOv4). The module was added before entering the detection head. In addition, comparisons were made to several other attention modules, including squeeze and excitation network (SENet), adaptively spatial feature fusion (ASFF), and a combination of SENet and SAM. The architecture used in this test was the architecture with the best mAP value based on the comparison test results in the second scenario. This operation aimed to achieve the most optimal optimization in detection accuracy.

Table 4 and Table 5 show that the modified (Mod) YOLOv4 network architecture with SAM attention modules achieved the best mAP on both types of wood. Moreover, the FPS value generated by the network architecture remained the highest with the inclusion of SAM channel attention. Therefore, the network architecture with the SAM attention module is the best choice for the proposed lightning YOLOv4.

Table 1. The simplification transformation of CSPDarkne53 architecture.

	Type	Filters		Repeat		Size
		Before	After	Before	After	
C1	Conv	32	32			3x3
	Conv	64	64			3x3/2
	Conv	32	32			1x1
	Conv	64	64	1x	1x	3x3
	Residual					
C2	Conv	128	128			3x3/2
	Conv	64	α			1x1
	Conv	128	α	2x	1x	3x3
	Residual					
C3	Conv	256	α			3x3/2
	Conv	128	α			1x1
	Conv	256	α	8x	4x	3x3
	Residual					
C4	Conv	512	α			3x3/2
	Conv	256	α			1x1
	Conv	512	α	8x	4x	3x3
	Residual					
C5	Conv	1024	α			3x3/2
	Conv	512	α			1x1
	Conv	1024	α	4x	2x	3x3
	Residual					

Table 2. Comparison of simplification by reducing CSPDarknet53 channels based on a scale coefficient.

α	Pine Lumber		Rubber Lumber	
	mAP	FPS	mAP	FPS
1.0	91.00	80.1	87.73	76.9
0.9	91.51	80.2	88.58	77.1
0.8	**91.95**	81.3	**89.49**	80.3
0.7	91.73	84.4	89.25	84.3
0.6	91.04	90.3	88.69	88.1

Table 3. Performance comparison of the selection activation function.

Activation Function	Pine Lumber		Rubber Lumber	
	mAP	FPS	mAP	FPS
Original	91.95	81.3	89.49	80.3
ReLU	91.16	82.9	89.95	80.9
L-ReLU	91.23	80.7	88.80	80.7
Swish	92.20	82.2	89.84	79.2
Mish	**92.40**	81.0	**90.70**	78.4

Table 4. Performance comparison with the addition of an attention module in the pine lumber dataset.

Method	mAP	FPS
YOLOv4	91.68	60.1
YOLOv4 + SAM	92.45	58.5
Mod-YOLOv4 + SENet	93.48	74.0
Mod-YOLOv4 + ASFF	93.50	75.7
Mod-YOLOv4 + SAM (proposed)	**93.55**	76.1

Table 5. Performance comparison with the addition of an attention module in the rubber lumber dataset.

Method	mAP	FPS
YOLOv4	89.39	60.2
YOLOv4 + SAM	90.19	60.0
Mod-YOLOv4 + SENet	89.42	74.2
Mod-YOLOv4 + ASFF	90.02	76.6
Mod-YOLOv4 + SAM (proposed)	**91.52**	77.1

4.4. The comparison results

To demonstrate the efficiency of the proposed lightning YOLOv4, we ran the proposed method on the same GPU hardware and compared it against existing baselines.

Table 6 and Table 7 show the experimental results on the pine and rubber lumber datasets. Compared to YOLOv4 with SAM attention as the default configuration of YOLOv4, if the input system had 320×320, 512×512, 608×608 of input image sizes, the mAP of the proposed lightning YOLOv4 increased by 0.72, 0.51, and 0.41 on the pine lumber dataset, and by 2.2, 1.3, and 0.3 on the rubber lumber dataset.

Table 6. Performance comparison for the pine lumber category.

Method	Input size	mAP	FPS
YOLOv3	320x320	87.34	73.5
YOLOv4 + SAM		92.45	58.5
Tiny YOLOv4		91.28	114.3
Lightning YOLOv4 (our)		**93.17**	76.1
YOLOv3	512x512	90.54	50.7
YOLOv4 + SAM		93.04	43.5
Tiny YOLOv4		91.66	104.3
Lightning YOLOv4 (our)		**93.55**	54.8
YOLOv3	608x608	91.32	36.5
YOLOv4 + SAM		93.41	33.0
Tiny YOLOv4		93.02	100.2
Lightning YOLOv4 (our)		**93.82**	41.2

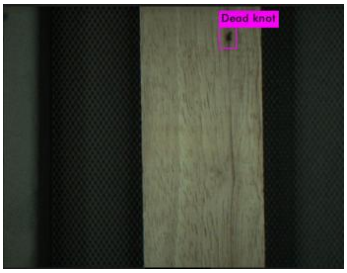
Table 7. Performance comparison for the rubber lumber category.

Method	Input size	mAP	FPS
YOLOv3	320x320	76.0	72.4
YOLOv4 + SAM		89.3	60.0
Tiny YOLOv4		87.3	113.8
Lightning YOLOv4 (our)		**91.5**	77.1
YOLOv3	512x512	82.6	50.8
YOLOv4 + SAM		91.7	43.6
Tiny YOLOv4		88.2	106.9
Lightning YOLOv4 (our)		**93.0**	54.4
YOLOv3	608x608	85.6	36.4
YOLOv4 + SAM		92.5	32.9
Tiny YOLOv4		88.3	100.8
Lightning YOLOv4 (our)		**92.8**	40.9

Aside from the improvement in accuracy, once the input system size was 608×608, the detection speed for both categories of the sawn lumber dataset increased by 4.7 and 4.5 from YOLOv3, thus meeting real-time time detection requirements in the industry. Although the proposed YOLOv4 modifications had a lower FPS than tiny YOLOv4 did, it should be noted that the mAP of the proposed lightning YOLOv4 modification was significantly higher for the complex defect category of the dataset or for the rubber lumber category.

To complement the findings, a subjective performance comparison is given in Figure 4. The localization samples illustrate that the proposed lightning YOLOv4 can achieve a better result than the original YOLOv4 structure does.

(a)



(b)

Figure 4. (a) and (b) are localization samples on rubber lumber of YOLOv4 and Lightning YOLOv4, respectively.

5. Conclusion

This work proposes a detection system for defects in sawn lumber based on simplifying YOLOv4 architecture by reducing the residual block on the CSPDarknet53 and PANet networks. Also, the number of channels on CSPDarknet53 was reduced by multiplying the number with a scale coefficient. This operation aimed to increase the detection speed of the system when compared to the original structure. However, the simplification dramatically and negatively affected the system's accuracy. Therefore, our proposed modified YOLOv4 uses Mish activation on the network and adds a spatial attention module to increase the system's accuracy. Additionally, this article defines a rubber lumber dataset that contains four defects, and a pine lumber dataset that contains two types of defects, for a total of 832 and 1,553 annotated images. The experimental results on the two datasets show that the proposed lightning YOLOv4 model can increase accuracy and detection speed, which can significantly enhance the effectiveness of a lumber cutting inspection system.

References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[2] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132-7141, 2018.

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

[5] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," *arXiv preprint arXiv:1911.09516*, 2019.

[6] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[7] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[8] W. Pölzleitner and G. Schwingshakl, "Real-time surface grading of profiled wooden boards," *Industrial Metrology*, vol. 2, no. 3-4, pp. 283-298, 1992.

[9] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[11] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658-666, 2019.

[12] Y. Tu, Z. Ling, S. Guo, and H. Wen, "An accurate and real-time surface defects detection method for sawn lumber," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-11, 2020.

[13] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390-391, 2020.

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," *Proceedings of the European conference on computer vision (ECCV)*, pp. 3-19, 2018.

[15] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.