

Article

Efficient Eye State Detection for Driver Fatigue Monitoring Using Optimized YOLOv7-Tiny

Gwo-Ching Chang ^{1,*}, Bo-Han Zeng ² and Shih-Chiang Lin ³¹ Department of Information Engineering, I-Shou University, Kaohsiung City 84001, Taiwan² Center for General Education, Nanhua University, Chiayi County 62249, Taiwan; bohan@nhu.edu.tw³ Department of Electrical Engineering, I-Shou University, Kaohsiung City 84001, Taiwan; sclin@cloud.isu.edu.tw

* Correspondence: cgc@isu.edu.tw

Abstract: This study refines the YOLOv7-tiny model through structured pruning and architectural fine-tuning, specifically for real-time eye state detection. By focusing on enhancing the model's efficiency, particularly in environments with limited computational resources, this research contributes significantly to advancing driver monitoring systems, where timely and accurate detection of eye states such as openness or closure can prevent accidents caused by drowsiness or inattention. Structured pruning was utilized to simplify the YOLOv7-tiny model, reducing complexity and storage requirements. Subsequent fine-tuning involved adjustments to the model's width and depth to further enhance processing speed and efficiency. The experimental outcomes reveal a pronounced reduction in storage size, of approximately 97%, accompanied by a sixfold increase in frames per second (FPS). Despite these substantial modifications, the model sustains high levels of precision, recall, and mean average precision (mAP). These improvements indicate a significant enhancement in both the speed and efficiency of the model, rendering it highly suitable for real-time applications where computational resources are limited.

Keywords: eye state detection; driver fatigue monitoring; YOLOv7-tiny; structured pruning; neural network optimization



Citation: Chang, G.-C.; Zeng, B.-H.; Lin, S.-C. Efficient Eye State Detection for Driver Fatigue Monitoring Using Optimized YOLOv7-Tiny. *Appl. Sci.* **2024**, *14*, 3497. <https://doi.org/10.3390/app14083497>

Academic Editor: Laura Cercenelli

Received: 14 March 2024

Revised: 14 April 2024

Accepted: 20 April 2024

Published: 21 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Research conducted by the AAA Foundation for Traffic Safety reveals that between 6% and 11% of all motor-vehicle accidents reported to police, and from 16% to 21% of fatal accidents, may involve drowsy driving [1]. This statistic underscores the urgent need for the implementation of appropriate systems designed to detect driver fatigue. Such systems, by monitoring fatigue indicators and issuing alerts in real time, play an essential role in reducing the risk of drowsy driving, with the potential to significantly lower the incidence of traffic accidents and related fatalities.

In the domain of vehicular safety, the current research is focused on reducing accidents caused by driver somnolence through three primary detection methodologies: physiological metrics (including electroencephalography (EEG) [2,3], electrooculography [4], and multi-modal algorithms [5]); vision-based methods (tracking eyelid movement, yawning, and head orientation) [6–10]; and vehicular dynamics analysis (such as lateral position and speed variation) [11,12]. Although vehicular dynamics offer an indirect measure of alertness, physiological and optical methods provide a more direct assessment. Physiological markers, especially EEG patterns indicating alpha wave dominance, are sensitive but may be considered invasive. In contrast, image analysis of eye, mouth, and head movements is non-invasive and increasingly favored for its practicality in real-life applications, striking a balance between effectiveness and user comfort.

In the specialized area of driver drowsiness detection through image analysis, research has concentrated on eye blink detection, facial landmark analysis, and facial expression

analysis using classifier models. Various deep learning models have been successfully employed in eye blink detection [13,14]. Facial landmark analysis has also received attention, with technologies like MTCNN (Multi-task Cascaded Convolutional Neural Network) for face detection and classification systems achieving high accuracies in identifying drowsiness [15,16]. The exploration of 3D-CNN (Three-dimensional Convolutional Neural Network) integrated with LSTM (Long Short-term Memory) and other CNN architectures for facial expression analysis has yielded high accuracy rates on various datasets [17–20].

However, image analysis-based methods for drowsiness detection face challenges, including variability in ocular surface texture across different ethnicities, influenced by anatomical, dermal, and external factors like lighting conditions and eyewear. Additionally, the need for real-time processing of complex visual data demands substantial computational resources, posing an obstacle to wider adoption.

This study introduces a real-time driver fatigue detection system utilizing an optimized You Only Look Once version 7-tiny (YOLOv7-tiny) model. YOLOv7-tiny, chosen for its real-time processing capabilities and efficiency, is ideal for in-vehicle systems without high computational demands [21–24]. This model was selected based on its superior performance in diverse and challenging visual environments, demonstrating robust adaptability to various lighting conditions and head poses. These attributes are critical in dynamic driving contexts where changes in environmental light and driver movement are frequent. Furthermore, the implementation of structured pruning and parameter adjustments within the YOLOv7-tiny framework has been strategically employed to refine the model's operational efficiency, thus ensuring that performance optimization does not come at the expense of detection accuracy. Performance metrics such as precision, recall, and mean average precision (mAP) are used to evaluate the model's effectiveness post-pruning, ensuring the balance between accuracy and computational efficiency for deployment in resource-constrained settings.

2. Materials and Methods

2.1. Principles of YOLOv7-Tiny Network Structure

The YOLOv7-tiny model is a state-of-the-art neural network for object detection tasks, offering a compact and efficient alternative to its larger counterparts [25]. The “tiny” version is specifically designed to balance performance with computational efficiency, making it suitable for deployment on devices with limited computational resources.

The architecture of YOLOv7-tiny is divided into two primary components: the backbone and the head, as elucidated in Figure 1. The backbone is crucial for extracting feature maps from input images, which are then utilized by the head to determine object classifications and locations. A notable innovation within the backbone is the integration of ELAN (Enhanced Efficient Layer Aggregation Network) blocks, which significantly improve the extraction and integration of multi-scale features. This is vital as edge information plays a critical role in defining object boundaries, a key factor in accurate object detection.

Furthermore, the backbone architecture is augmented by the inclusion of the SPPC-SPC (Spatial Pyramid Pooling—Cross Stage Partial Connections) block, a hybrid module that amalgamates the Spatial Pyramid Pooling (SPP) and Cross Stage Partial Connections (CSPC) methodologies. The SPP facet of this block guarantees the preservation of feature map spatial dimensions, thereby facilitating the detection of objects varying in size by amalgamating features pooled at disparate scales into a rich, spatially invariant feature representation. Concurrently, the CSPC component innovates upon traditional layer connectivity paradigms by implementing partial connections between stages, thus curtailing computational overhead whilst maintaining the flow of essential information. This strategic design choice significantly contributes to the model's streamlined profile, eschewing superfluous computations in favor of efficiency and effective learning.

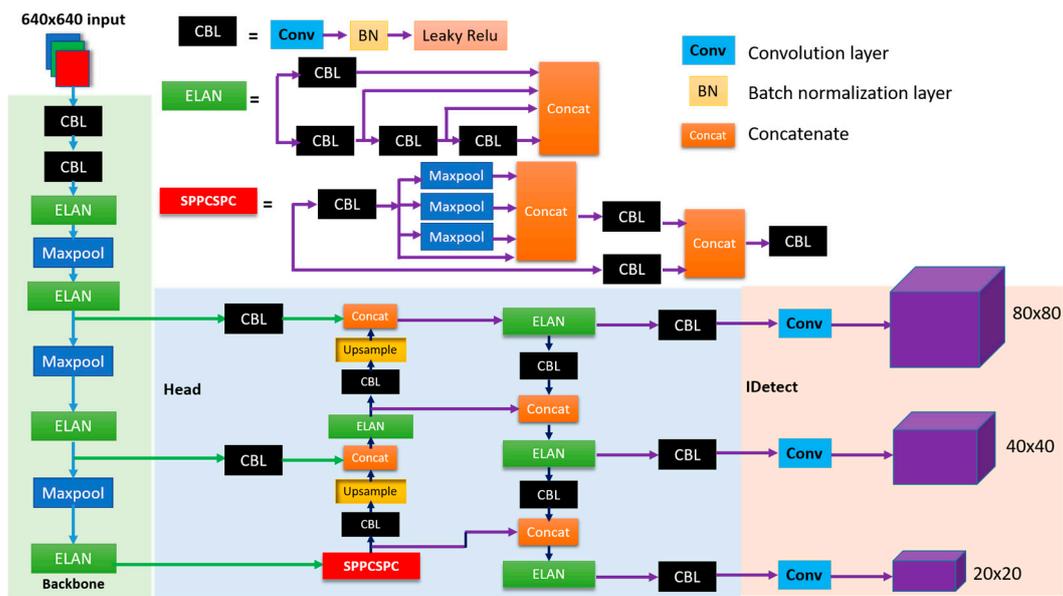


Figure 1. Architectural overview of YOLOv7-tiny: integrating ELAN and SPPCSPC for enhanced object detection.

Positioned at the terminus of the YOLOv7-tiny network, the head component employs detection heads operational at diverse scales (notably, 80×80 , 40×40 , and 20×20), exploiting the multi-scale feature maps synthesized by the backbone. This multi-resolution processing framework enables the adept detection of objects across a spectrum of sizes. Notably, the 80×80 feature maps, with their finer granularity, are indispensable for the detection of minuscule objects, providing an enhanced pixel density per object for capturing minute details. Conversely, the 40×40 feature maps offer a harmonious equilibrium between granular detail and broader context, apt for objects of intermediate size. The 20×20 feature maps, with their broader receptive fields, are adept at encapsulating the necessary context for detecting larger objects. This stratified approach to feature map scaling is indispensable for ensuring robust object detection across varying scales and aspect ratios, underlining the YOLOv7-tiny model’s versatility and precision in object detection endeavors.

2.2. Structured Pruning for Eye State Detection

To optimize the YOLOv7-tiny model for the specialized task of detecting eye states, a method known as structured pruning is employed. This approach elaborately targets and eliminates less-essential scales of feature maps that do not significantly contribute to the accuracy of detecting small to medium-sized objects such as eyes, which are critical in the context of image resolution and camera proximity.

Structured pruning involves the strategic removal of layers or channels associated with these non-essential scales. For example, coarser 20×20 feature maps, which typically provide less detail required for detecting eye states, can be pruned to reduce redundancy. This reduction reallocates computational resources to more relevant feature map scales, thereby enhancing both the detection speed and accuracy.

This study begins with the baseline configuration of the YOLOv7-tiny model. The primary objective here is to identify fine details such as the open or closed state of eyes, necessitating higher resolution feature maps like the 80×80 scale. These maps are more capable of capturing the intricate details essential for accurate eye state analysis. Consequently, we introduce the YOLOv7-tiny-First-Prune architecture, which eschews the model’s original components tied to the 20×20 feature map scale, as depicted in Figure 2. This strategic decision is predicated on the understanding that the task of eye state detection does not necessitate the discernment of less detailed spatial information, often associated

with the detection of larger objects. Progressing further, the YOLOv7-tiny-Second-Prune architecture streamlines the network even more by also excising elements contributing to the 40×40 feature map scale, as illustrated in Figure 3. This refinement suggests that the network can forgo processing details of larger and medium-sized objects, which are extraneous for the detection of eye states. The impact of these pruning interventions on the model’s performance is meticulously evaluated to ascertain that the optimized model maintains its efficacy for the intended detection task. This approach underscores a concerted effort to balance model complexity with performance efficiency, ensuring the YOLOv7-tiny model remains a viable and effective tool for real-time eye state detection applications.

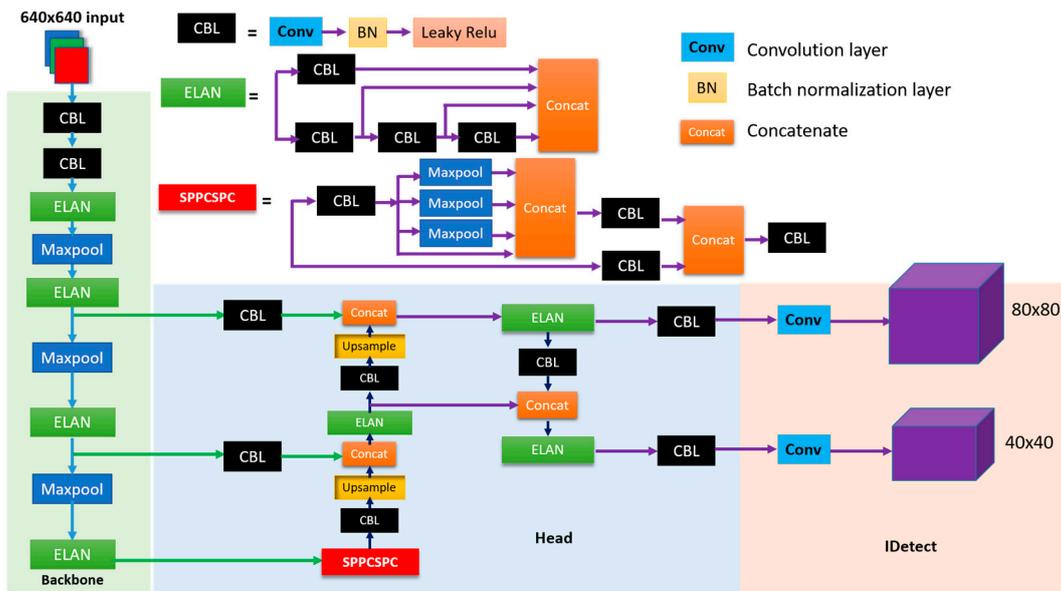


Figure 2. YOLOv7-tiny-First-Prune architecture: eliminating the 20×20 feature map scale for optimized eye state detection.

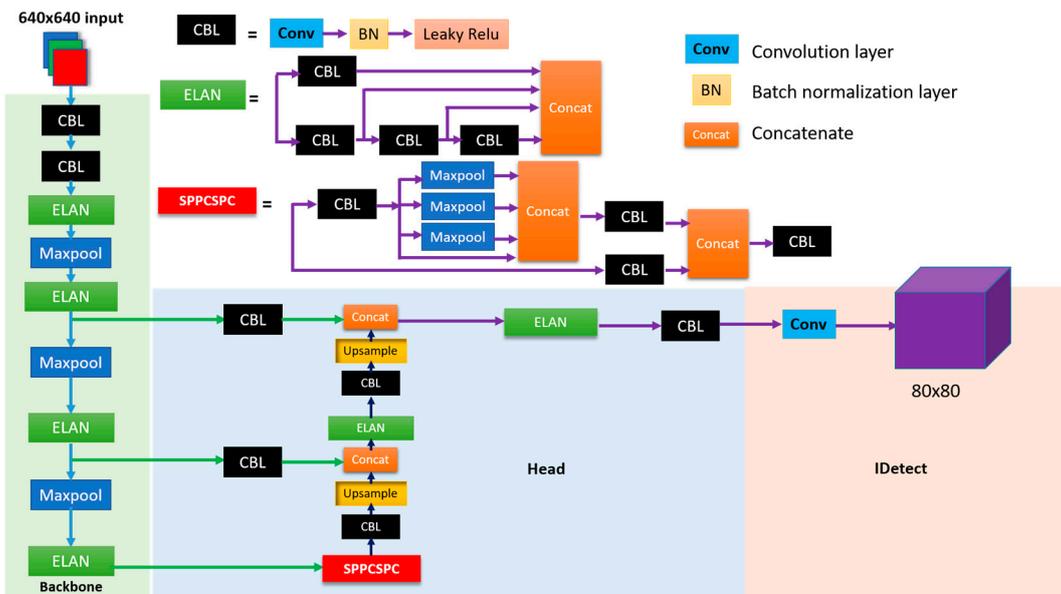


Figure 3. YOLOv7-tiny-Second-Prune architecture: further refinement by removing the 40×40 feature map scale.

2.3. Fine-Tuning Model Architecture

Initially, the depth_multiple parameter is adjusted, which alters the number of convolutional layers in the network. This reduction in layers is designed to decrease the

model's complexity and enhance inference speed without significantly compromising the accuracy of eye state detection. Similarly, the `width_multiple` parameter is refined to control the number of neurons in each layer, effectively reducing the number of computational operations required for each forward pass through the network.

These fine-tuning steps are systematically implemented in stages, with depth and width parameters being scaled down from initial values of 0.5 to more compact configurations of 0.33, 0.125, and finally, 0.0625. This structured approach not only ensures incremental optimization of the model's efficiency, but also maintains a balance between performance and speed, essential for deployment in systems with limited computational capabilities.

The empirical validation of these fine-tuning endeavors is conducted with utmost rigor. We subject the modified model architectures to a series of evaluations, utilizing established metrics such as precision, recall, and mAP. These metrics serve as benchmarks to assess the model's efficacy in accurately detecting eye states under varying conditions. This methodological approach to fine-tuning allows for a granular analysis of the impact each adjustment has on the model's overall performance, ensuring that the final architecture is not only optimized for eye state detection, but is also viable for real-time application within constrained computational environments.

2.4. Dataset

This study utilizes the Nighttime–Yawning–Microsleep–Eyeblink–Distraction (NITYMED) dataset [26]. This dataset is a valuable resource containing video recordings of male and female drivers showing signs of drowsiness through eye and mouth movements during nighttime vehicle operations. The importance of this dataset is heightened by its focus on nighttime driving scenarios, a time during which the likelihood of drowsiness-induced vehicle accidents increases significantly. The dataset carefully collects examples of drivers operating actual vehicles under nighttime conditions, thereby enhancing the ecological validity of the study.

Drivers represented within the NITYMED dataset comprise 11 Caucasian males and 8 females, carefully chosen to represent a broad spectrum of physical attributes (including variations in hair color, the presence of facial hair, and the usage of optical aids), thereby ensuring a diverse sample that could potentially affect the detection of drowsiness-related symptoms.

For the purpose of this study, video recordings featuring 14 drivers (consisting of 8 males and 6 females) were selected to generate a substantial pool of training and validation data, amassing a total of 3440 images. This study adopts a binary classification approach, delineating the data into two distinct categories: "Open Eyes" and "Closed Eyes". From the total dataset, 2955 images (constituting 86% of the dataset) were allocated for training purposes, with the remaining 485 images (representing 14% of the dataset) set aside for validation. Furthermore, video content featuring an additional 5 drivers was incorporated as test data to rigorously evaluate the model's efficacy.

A critical aspect of this research involves the analysis of ocular activities to infer the driver's state of alertness. A driver is considered to be in an awake state if the duration of eye closure ranges between 100 and 300 milliseconds; should this duration extend beyond 300 milliseconds, the driver is deemed to be in a state of drowsiness [27,28]. Given that the NITYMED dataset operates at a frame rate of 25 frames per second, a detection of the driver's eyes remaining closed for more than 8 frames is indicative of drowsiness.

2.5. Evaluation Metrics

To ensure the empirical analysis was both rigorous and objective, we adopted three principal metrics: precision, recall, and mAP. Precision is meticulously defined as the proportion of accurately detected eye states relative to the aggregate number of eye states identified by the model. Conversely, recall is determined by the ratio of detected eye states to the aggregate actual eye states present. These metrics are mathematically represented in Equations (1) and (2), where TP (True Positive) symbolizes the accurately identified eye

states, FP (False Positive) denotes instances erroneously classified as eye states, and FN (False Negative) signifies eye states that were not correctly detected.

$$\text{precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (2)$$

The formula for calculating mAP is outlined in Equations (3) and (4). The area under the precision–recall curve (PR curve) is known as AP (average precision), and mAP represents the mean of AP values across different categories. In this study, with the detection of drivers' eyes being open or closed, the N value, indicating the total number of categories tested, is 2.

$$AP = \int_0^1 P(R)dR \quad (3)$$

$$mAP = \frac{\sum_1^N \int_0^1 P(R)dR}{N} \quad (4)$$

Additionally, to provide a comprehensive evaluation of the model's efficiency and effectiveness, we included the following metrics: GFLOPS (Billion Floating-Point Operations Per Second), number of parameters, and FPS.

GFLOPS measures the computational complexity of the model, indicating the number of billion floating-point operations required to process a single input. It is crucial for understanding the model's demands on computational resources. The number of parameters quantifies the total number of trainable parameters within the model. A lower number of parameters typically implies a more lightweight and potentially faster model. FPS assesses the model's processing speed, specifically how many frames it can analyze per second. Higher FPS rates are indicative of better suitability for real-time applications.

3. Experimental Results and Analysis

3.1. Quantitative Analysis of Structured Pruning on YOLOv7-Tiny for Eye State Detection

For the purposes of our research, we have availed ourselves of the computational resources provided by Google Colab's Central Processing Unit (CPU) environment. The specifications of the CPU runtime allocated by Google Colab for our investigative processes are anchored by an Intel Xeon Processor (Santa Clara, CA, USA). This processor features a dual-core configuration, each core operating at a clock speed of 2.30 GHz. Complementing this processing capability is a memory allocation of 13 GB RAM (Random Access Memory), facilitating the efficient handling of data-intensive tasks inherent to our research activities. This computational setup has been chosen to strike an optimal balance between processing power and memory capacity, ensuring that our analytical models and algorithms perform with the requisite efficiency and reliability.

Our research begins a systematic exploration of structured pruning applied to the YOLOv7-tiny model, aiming to remove less important features or layers to streamline the model for specific tasks, in this case, eye state detection. The objective was to reduce the model's computational demands while maintaining or improving its detection accuracy.

Table 1 contains a detailed comparison of four models: the original YOLOv7, YOLOv7-tiny, and two iterations of YOLOv7-tiny after structured pruning (First-Prune and Second-Prune). The data includes several key metrics for each model: Number of Parameters, Number of Network Layers, Precision, Recall, mAP@.5 (mean Average Precision at 50% Intersection over Union), and Storage Size of Weight File. As seen in Table 1, the progression from the original YOLOv7 model to the YOLOv7-tiny-Second-Prune demonstrates a significant reduction in both the number of parameters and the network layers, which directly correlates to a decrease in the model's storage size and, presumably, its computational complexity.

Table 1. Comparative analysis of YOLOv7 iterations: assessing the impact of structured pruning on model efficiency and eye state detection accuracy.

Model	Number of Parameters	Number of Network Layers	Precision	Recall	mAP@.5	Storage Size of Weight File (MB)
Yolov7	36,487,166	314	0.995	0.984	0.996	74.8
Yolov7-tiny	6,010,302	208	0.991	0.992	0.996	12.3
Yolov7-tiny-First-Prune	3,963,128	182	0.999	1	0.996	8.1
Yolov7-tiny-Second-Prune	3,449,878	156	0.999	1	0.996	7.1

Remarkably, despite these reductions, the precision, recall, and mAP@.5 metrics remain consistently high across all iterations. The maintained or even improved precision and recall highlight the effectiveness of structured pruning in removing redundant or less important features without compromising the model's ability to accurately detect eye states.

The transition from the original model to its highly pruned version showcases an impressive reduction in storage size from 74.8 MB to 7.1 MB, which is particularly beneficial for deployment in resource-constrained environments. This size reduction does not adversely affect the model's detection capabilities, as evidenced by the consistent mAP@.5 score of 0.996 across all iterations.

This structured pruning process effectively reduced the YOLOv7-tiny model's complexity and size without compromising detection accuracy. The reduction in parameters and layers contributes to enhanced computational efficiency, crucial for deployment in environments with limited hardware resources. The consistency in precision, recall, and mAP@.5 metrics across iterations demonstrates that the model retains its ability to accurately detect eye states, even with significant reductions in model size and complexity. This balance between efficiency and performance underscores the potential of structured pruning for optimizing neural networks for specific tasks, particularly those requiring real-time processing capabilities.

3.2. Analyzing the Impact of Fine-Tuning the YOLOv7-Tiny-Second-Prune Architecture on Eye State Detection

Following the second structured pruning of the YOLOv7-tiny model, resulting in the YOLOv7-tiny-Second-Prune architecture, we embarked on fine-tuning the model architecture to optimize its performance for eye state detection.

Table 2 presents a detailed overview of the YOLOv7-tiny-Second-Prune model's performance across four different configurations, each with varying width and depth multiples (0.5, 0.33, 0.125, and 0.0625), respectively named YOLOv7-tiny-OptimaFlex-0.5, YOLOv7-tiny-OptimaFlex-0.33, YOLOv7-tiny-OptimaFlex-0.125, and YOLOv7-tiny-OptimaFlex-0.0625. As seen in Table 2, the fine-tuning involved adjusting the width_multiple and depth_multiple parameters of the YOLOv7-tiny-Second-Prune structure, maintaining 156 layers across all configurations but varying the number of parameters.

Table 2. Evaluating the efficacy of fine-tuning YOLOv7-tiny-Second-Prune architecture across varied configurations for enhanced eye state detection.

Model	Width Multiple/Depth Multiple	Number of Parameters	Number of Layers	Precision	Recall	mAP@.5	Storage Size of Weight File (MB)
Yolov7-tiny-Second-Prune	0.5	864,174	156	0.984	0.992	0.996	1.9
	0.33	358,878	156	0.997	1	0.996	0.9
	0.125	55,742	156	0.997	1	0.996	0.3
	0.0625	20,174	156	0.958	0.982	0.975	0.2

The results demonstrate a clear trend of improving the model's efficiency as we decrease the number of parameters by adjusting the width and depth multiples. Notably, even with substantial reductions in complexity, the model maintains high performance levels, as indicated by precision, recall, and mAP@.5 metrics.

The configurations with width/depth multiples of 0.33 and 0.125 offer an exceptional balance, achieving nearly perfect precision and recall with significantly reduced parameter counts and storage sizes. This highlights the effectiveness of fine-tuning in optimizing model performance while reducing computational demands.

The configuration with a width/depth multiple of 0.0625, while demonstrating the highest efficiency in terms of parameter count and storage size, shows a slight drop in precision and recall. This suggests there is a limit to how much the model can be simplified without affecting its ability to accurately detect eye states. The substantial decrease in storage size, from 1.9 MB in the 0.5 configuration to 0.2 MB in the 0.0625 configuration, illustrates the benefits of fine-tuning for applications where storage capacity is limited.

The experimental analysis underscores the significance of fine-tuning the YOLOv7-tiny-Second-Prune model architecture for specific tasks such as eye state detection. By systematically adjusting the `width_multiple` and `depth_multiple` parameters, we can significantly enhance the model's efficiency and suitability for real-time applications without substantially compromising accuracy. The study demonstrates the potential for neural network optimization through structured pruning followed by strategic fine-tuning, offering insights into developing lightweight yet effective models for deployment in resource-constrained environments.

3.3. The Impact of Structured Pruning and Architecture Fine-Tuning

The experimental results showcased in Table 3 not only highlight the efficiency and performance of various configurations of the YOLOv7 model, but also underscore the critical importance of utilizing structured pruning and fine-tuning of the model architecture in achieving significant improvements in model efficiency and speed.

Table 3. Efficiency and performance enhancements in YOLOv7 models through structured pruning and architecture fine-tuning.

Model	GFLOPS	FPS
YOLOv7	51.584816	1.61
YOLOv7-tiny	6.5150944	7.83
YOLOv7-tiny-First-Prune	5.689472	8.01
YOLOv7-tiny-Second-Prune	4.86528	8.69
YOLOv7-tiny-OptimaFlex-0.5	1.2399104	19.63
YOLOv7-tiny-OptimaFlex-0.33	0.7261696	21.09
YOLOv7-tiny-OptimaFlex-0.125	0.1383936	41.84
YOLOv7-tiny-OptimaFlex-0.0625	0.1109504	51.00

Initially, the YOLOv7 model, serving as the baseline, required a substantial quantity of computational resources, reflected by its high GFLOPS. However, the structured pruning process, which began with YOLOv7-tiny and progressed through two pruning stages, resulted in a significant reduction in GFLOPS, suggesting a decrease in computational load. Notably, this reduction did not compromise the model's ability to detect eye states, as demonstrated by the consistent mAP scores across iterations.

Fine-tuning further enhanced these results. The YOLOv7-tiny-OptimaFlex variants demonstrate how fine-tuning, combined with structured pruning, can lead to models that are not only significantly more efficient, but also maintain a high level of accuracy, as evidenced by their improved FPS rates. The YOLOv7-tiny-OptimaFlex variants, with their drastically reduced GFLOPS and significantly increased FPS, showcase an exemplary case of how structured pruning combined with model architecture fine-tuning can lead to breakthroughs in achieving the dual goals of high efficiency and speed.

From the experimental results, the importance of structured pruning and fine-tuning in model optimization is clear. These strategies allow for the creation of highly efficient models that are well-suited for deployment in environments where computational resources are limited, such as on edge devices. By meticulously reducing the model's complexity while carefully adjusting its parameters to retain accuracy, researchers and practitioners can develop lightweight models capable of real-time inference, opening up new possibilities for the application of deep learning models in various real-world scenarios.

A significant limitation of the current study is its reliance on a specific computing environment provided by the Google Colab CPU, which may not fully replicate the diverse range of real-world operational conditions in which such a system might be deployed. Additionally, while the structured pruning and fine-tuning approaches demonstrate promising results in controlled settings, the variability in naturalistic driving environments—including varying lighting conditions, camera angles, and unpredictable driver behaviors—may impact the generalizability of the findings. It is also important to acknowledge that while the dataset used provides a solid foundation for initial testing, it may not encompass all possible scenarios of eye states, particularly under different levels of occlusion or fatigue levels. Thus, further validation with more extensive, varied datasets would be crucial for confirming the robustness and applicability of the optimized model in real-world conditions. These considerations are essential for guiding future research directions and for realistic assessments of the system's operational effectiveness.

4. Conclusions

This research demonstrates the significant benefits of structured pruning and architectural fine-tuning on the YOLOv7-tiny model for eye state detection, highlighting the essential balance between model efficiency and detection accuracy crucial for real-time applications. Through structured pruning, we effectively reduced the model's complexity, computational demands, and storage size while maintaining high accuracy, as evidenced by consistent precision, recall, and mAP@.5 metrics across various model iterations. Fine-tuning, particularly adjusting `width_multiple` and `depth_multiple` parameters, further optimized the model, enhancing processing speed without compromising performance. The YOLOv7-tiny-OptimaFlex variants illustrate the potential of these optimization techniques in developing highly efficient and accurate models suitable for deployment in resource-constrained environments. This study underscores the importance of model optimization for real-time object detection, providing valuable insights for future developments in neural network efficiency and application.

Author Contributions: Conceptualization, methodology and software, G.-C.C. and B.-H.Z.; validation and formal analysis, G.-C.C., B.-H.Z. and S.-C.L.; resources, B.-H.Z.; data curation, S.-C.L.; writing—original draft preparation, G.-C.C. and B.-H.Z.; writing—review and editing, S.-C.L.; supervision, G.-C.C.; project administration, G.-C.C.; funding acquisition, G.-C.C. and S.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study was exempt from ethical review and approval as informed consent was obtained from all individuals included in the NITYMED dataset, which is a previously collected public resource. The study was conducted in accordance with the Declaration of Helsinki.

Informed Consent Statement: Informed consent was obtained from all subjects participating in the study, confirming their voluntary participation and understanding of the purpose and procedures of the study.

Data Availability Statement: The public database used in this paper is NITYMED (Nighttime–Yawning–Microsleep–Eyeblink–Driver distraction), which can be found in <https://datasets.esdalab.ece.uop.gr/> (accessed on 12 September 2023).

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

References

- Bayne, A.; Trivedi, N.; Liotta, M.; Siegfried, A.; Gaspar, J.; Carney, C. *Countermeasures to Reduce Drowsy Driving: Results of a Literature Review and Discussions with Experts (Technical Report)*; AAA Foundation for Traffic Safety: Washington, DC, USA, 2022.
- Min, J.; Xiong, C.; Zhang, Y.; Cai, M. Driver Fatigue Detection Based on Prefrontal EEG Using Multi-Entropy Measures and Hybrid Model. *Biomed. Signal Process. Control* **2021**, *69*, 102857. [[CrossRef](#)]
- Yang, Y.; Gao, Z.; Li, Y.; Cai, Q.; Marwan, N.; Kurths, J. A Complex Network-Based Broad Learning System for Detecting Driver Fatigue from EEG Signals. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 5800–5808. [[CrossRef](#)]
- Beles, H.; Vesselenyi, T.; Rus, A.; Mitran, T.; Scurt, F.B.; Tolea, B.A. Driver Drowsiness Multi-Method Detection for Vehicles with Autonomous Driving Functions. *Sensors* **2024**, *24*, 1541. [[CrossRef](#)]
- Karuppusamy, N.S.; Kang, B.-Y. Multimodal System to Detect Driver Fatigue Using EEG, Gyroscope, and Image Processing. *IEEE Access* **2020**, *8*, 129645–129667. [[CrossRef](#)]
- Dong, B.-T.; Lin, H.-Y.; Chang, C.-C. Driver Fatigue and Distracted Driving Detection Using Random Forest and Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 8674. [[CrossRef](#)]
- Anber, S.; Alsaggaf, W.; Shalash, W. A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features. *Electronics* **2022**, *11*, 285. [[CrossRef](#)]
- Zheng, H.; Wang, Y.; Liu, X. Adaptive Driver Face Feature Fatigue Detection Algorithm Research. *Appl. Sci.* **2023**, *13*, 5074. [[CrossRef](#)]
- Zhu, T.; Zhang, C.; Wu, T.; Ouyang, Z.; Li, H.; Na, X.; Ling, J.; Li, W. Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences. *Appl. Sci.* **2022**, *12*, 2224. [[CrossRef](#)]
- Florez, R.; Palomino-Quispe, F.; Coaquira-Castillo, R.J.; Herrera-Levano, J.C.; Paixão, T.; Alvarez, A.B. A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification. *Appl. Sci.* **2023**, *13*, 7849. [[CrossRef](#)]
- Iskandarani, M.Z. Relating Driver Behaviour and Response to Messages through HMI in Autonomous and Connected Vehicular Environment. *Cogent Eng.* **2022**, *9*, 2002793. [[CrossRef](#)]
- Charissis, V.; Falah, J.; Lagoo, R.; Alfalah, S.F.M.; Khan, S.; Wang, S.; Altarteer, S.; Larbi, K.B.; Drikakis, D. Employing Emerging Technologies to Develop and Evaluate In-Vehicle Intelligent Systems for Driver Support: Infotainment AR HUD Case Study. *Appl. Sci.* **2021**, *11*, 1397. [[CrossRef](#)]
- Park, S.; Pan, F.; Kang, S.; Yoo, C.D. Driver Drowsiness Detection System Based on Feature Representation Learning Using Various Deep Networks. In Proceedings of the Computer Vision—ACCV 2016 Workshops, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2017; pp. 154–164.
- Hashemi, M.; Mirrashid, A.; Beheshti Shirazi, A. Driver Safety Development: Real-Time Driver Drowsiness Detection System Based on Convolutional Neural Network. *SN Comput. Sci.* **2020**, *1*, 289. [[CrossRef](#)]
- Zhao, Z.; Zhou, N.; Zhang, L.; Yan, H.; Xu, Y.; Zhang, Z. Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN. *Comput. Intell. Neurosci.* **2020**, *2020*, 7251280. [[CrossRef](#)] [[PubMed](#)]
- Phan, A.C.; Nguyen, N.H.Q.; Trieu, T.N.; Phan, T.C. An Efficient Approach for Detecting Driver Drowsiness Based on Deep Learning. *Appl. Sci.* **2021**, *11*, 8441. [[CrossRef](#)]
- Alameen, S.A.; Alhothali, A.M. A Lightweight Driver Drowsiness Detection System Using 3DCNN with LSTM. *Comput. Syst. Sci. Eng.* **2023**, *44*, 895–912. [[CrossRef](#)]
- Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A Deep Learning Approach to Detect Driver Drowsiness. *Int. J. Eng. Res. Technol.* **2021**, *10*, 183–189.
- Gomaa, M.W.; Mahmoud, R.O.; Sarhan, A.M. A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction. *J. Eng. Res.* **2022**, *6*, 59–70. [[CrossRef](#)]
- Dua, M.; Shakshi; Singla, R.; Raj, S.; Jangra, A. Deep CNN Models-Based Ensemble Approach to Driver Drowsiness Detection. *Neural Comput. Appl.* **2021**, *33*, 3155–3168. [[CrossRef](#)]
- Jing, J.; Zhai, M.; Dou, S.; Wang, L.; Lou, B.; Yan, J.; Yuan, S. Optimizing the YOLOv7-Tiny Model with Multiple Strategies for Citrus Fruit Yield Estimation in Complex Scenarios. *Agriculture* **2024**, *14*, 303. [[CrossRef](#)]
- Zhang, L.; Xiong, N.; Pan, X.; Yue, X.; Wu, P.; Guo, C. Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery. *Algorithms* **2023**, *16*, 520. [[CrossRef](#)]
- Gu, B.; Wen, C.; Liu, X.; Hou, Y.; Hu, Y.; Su, H. Improved YOLOv7-Tiny Complex Environment Citrus Detection Based on Lightweighting. *Agronomy* **2023**, *13*, 2667. [[CrossRef](#)]
- Wang, Q.; Zhang, Z.; Chen, Q.; Zhang, J.; Kang, S. Lightweight Transmission Line Fault Detection Method Based on Leaner YOLOv7-Tiny. *Sensors* **2024**, *24*, 565. [[CrossRef](#)] [[PubMed](#)]
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
- Petrellis, N.; Voros, N.; Antonopoulos, C.; Keramidas, G.; Christakos, P.; Mousouliotis, P. NITYMED. IEEE Dataport 2022. Available online: <https://iee-dataport.org/documents/nitymed/> (accessed on 12 September 2023).

27. Poudel, G.R.; Innes, C.R.; Bones, P.J.; Jones, R.D. The Relationship Between Behavioural Microsleeps, Visuomotor Performance and EEG Theta. In Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, Argentina, 31 August–4 September 2010; pp. 4452–4455.
28. Malla, A.M.; Davidson, P.R.; Bones, P.J.; Green, R.; Jones, R.D. Automated Video-Based Measurement of Eye Closure for Detecting Behavioral Microsleep. In Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, Argentina, 31 August–4 September 2010; pp. 6741–6744.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.