

Article

Not peer-reviewed version

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

[Gwo-Ching Chang](#)*, Bo-Han Zeng, Shih-Chiang Lin

Posted Date: 15 March 2024

doi: 10.20944/preprints202403.0925.v1

Keywords: Eye State Detection; Driver Fatigue Monitoring; YOLOv7-tiny; Structured Pruning; Neural Network Optimization



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 ³

¹ Department of Information Engineering, I-Shou University; cgc@isu.edu.tw

² Center for General Education, Nanhua University; bohan@nhu.edu.tw

³ Department of Electrical Engineering, I-Shou University; sclin@cloud.isu.edu.tw

* Correspondence: cgc@isu.edu.tw

Abstract: This study demonstrates the efficacy of structured pruning and architectural fine-tuning on the YOLOv7-tiny model for eye state detection, emphasizing optimization for real-time applications. Structured pruning significantly reduced the model's complexity and storage size while maintaining high detection accuracy, as evidenced by stable precision, recall, and mAP@.5 metrics across iterations. Further fine-tuning adjusted the model's width and depth, optimizing efficiency and processing speed without compromising performance. These optimizations yielded YOLOv7-tiny variants that are both computationally efficient and accurate, suitable for resource-constrained environments. The findings highlight the critical role of model optimization in deploying effective neural networks for specific detection tasks in real-time scenarios.

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

1. Introduction

Data from the U.S. National Highway Traffic Safety Administration (NHTSA) indicate that drowsy driving resulted in 633 fatalities in 2020, with the actual figures potentially higher due to underreporting. Crashes involving fatigue often feature a single vehicle without evasive maneuvers, highlighting a lapse in driver alertness. Advanced detection systems that monitor signs of fatigue and provide real-time alerts could mitigate the risk associated with drowsy driving, potentially preventing an estimated 21% of fatal collisions related to driver drowsiness [1].

In the domain of vehicular safety, current research is focused on reducing accidents caused by driver somnolence through three primary detection methodologies: physiological metrics (including 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 for face detection and classification systems achieving high accuracies in identifying drowsiness [15,16]. The exploration of 3D-CNN integrated with LSTM and other CNN architectures for facial expression analysis has yielded high accuracy rates on various datasets [17–19].

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. The model's adaptability to various lighting conditions and head poses, coupled with structured pruning and parameter adjustments, aims to optimize performance without compromising 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 architectural essence of YOLOv7-tiny bifurcates into two fundamental components: the backbone and the head, as elucidated in Figure 1. The former plays a pivotal role in feature map extraction from input imagery, while the latter leverages these extracted features to ascertain object classifications and locales.

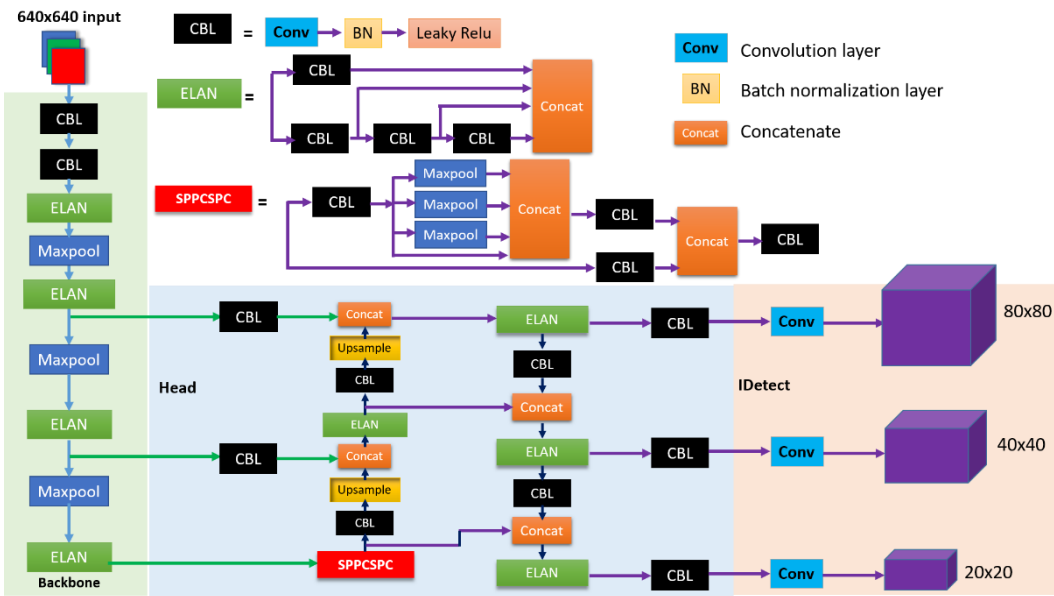


Figure 1. Architectural Overview of YOLOv7-Tiny: Integrating ELAN and SPPCSPC for Enhanced Object Detection.

A critical innovation within the backbone's design is the integration of ELAN (Enhanced Efficient Layer Aggregation Network) blocks. These blocks represent a seminal advancement in enhancing the extraction and amalgamation of features across multiple scales, a capability of paramount importance given the edge information's critical role in demarcating object boundaries – a cornerstone of precise object detection.

Furthermore, the backbone architecture is augmented by the inclusion of the SPPCSPC (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.

Positioned at the terminus of the YOLOv7-tiny network, the head component employs detection heads operational at diverse scales (notably, 80x80, 40x40, and 20x20), 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 80x80 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 40x40 feature maps offer a harmonious equilibrium between granular detail and broader context, apt for objects of intermediate size. The 20x20 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 refine the YOLOv7-tiny model for the specialized task of detecting the state of eyes, employing a method known as structured pruning. This technique meticulously targets and eliminates specific scales of feature maps deemed non-essential for accurately detecting objects of particular sizes—in this case, the eyes, which in the context of image resolution and camera proximity, generally manifest as small to medium-sized entities. It is imperative that the selected scales of feature maps are congruent with the dimensional attributes of the eyes to ensure their states are captured with utmost precision.

The YOLOv7-tiny model, by design, generates feature maps across a multitude of scales. The process of structured pruning thus entails the strategic removal of certain layers or channels corresponding to scales that hold minimal significance for the detection of eye states. For example, if the coarser 20x20 feature maps yield negligible improvements to the detection accuracy of eye states, it stands to reason that these can be pruned. This action reallocates computational resources towards processing scales that are more pertinent to the task at hand, potentially augmenting both the detection speed and accuracy.

This study commences with the original YOLOv7-tiny model as a foundational benchmark. The overarching goal herein is to discern minute features, such as the open or closed states of eyes, which necessitates the 80x80 feature maps for their superior resolution capabilities. These high-resolution maps are adept at capturing the intricate details crucial for a precise analysis of eye states. Consequently, we introduce the YOLOv7-tiny-First-Prune architecture, which eschews the model's original components tied to the 20x20 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 40x40 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.

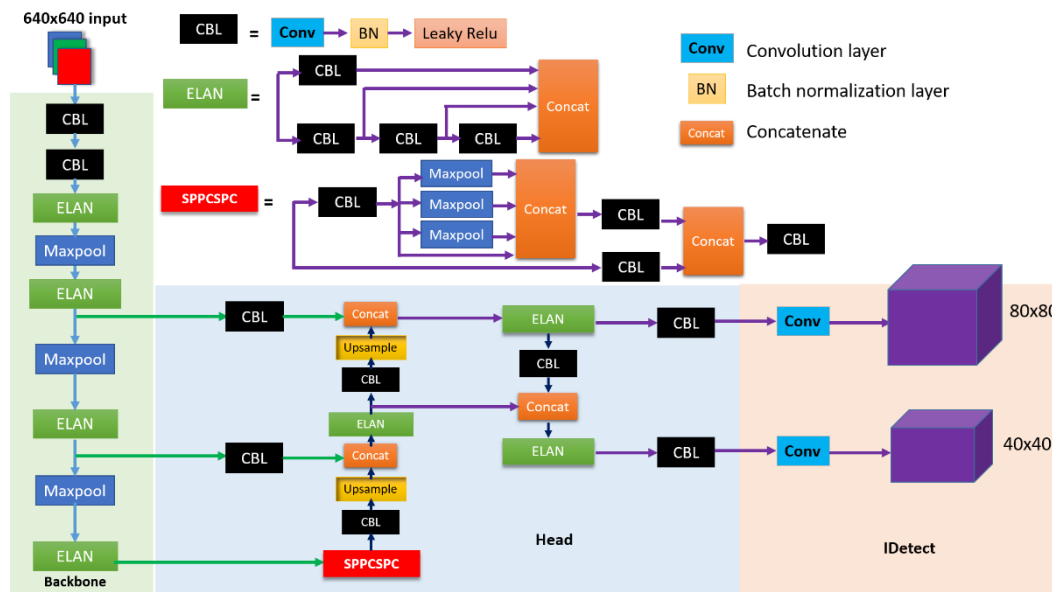


Figure 2. YOLOv7-Tiny-First-Prune Architecture: Eliminating the 20x20 Feature Map Scale for Optimized Eye State Detection.

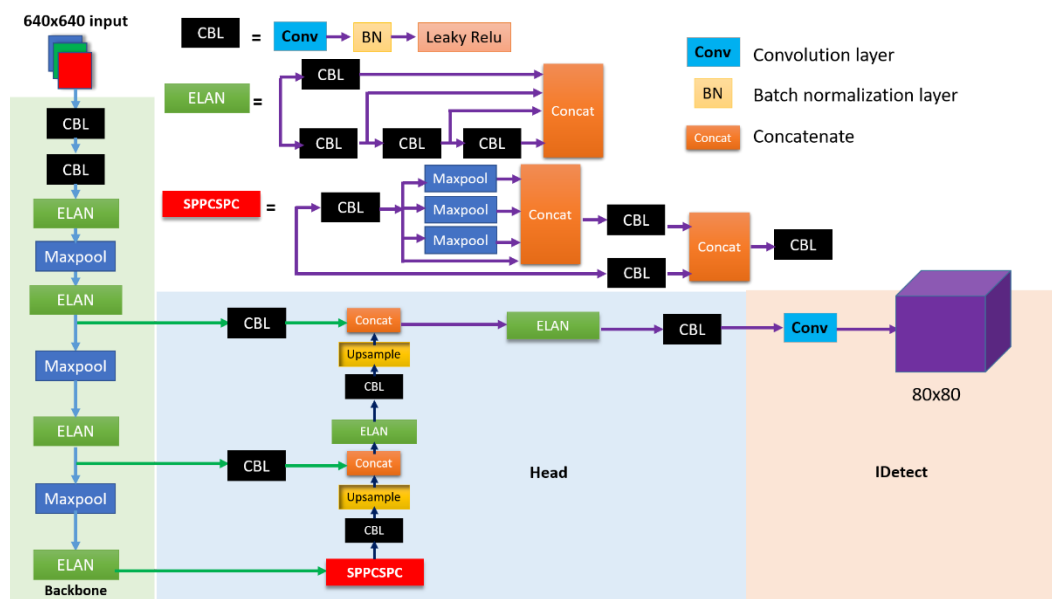


Figure 3. YOLOv7-Tiny-Second-Prune Architecture: Further Refinement by Removing the 40x40 Feature Map Scale.

2.3. Fine-tuning Model Architecture

The process of fine-tuning the model architecture helps further improve its performance in detecting the state of eyes. This fine-tuning process entails a judicious adjustment of the model's architectural parameters, specifically the 'depth_multiple' and 'width_multiple'. These parameters are pivotal in dictating the model's structural complexity, influencing both the number of layers and the dimensions of the convolutional kernels. By delicately modifying these parameters, we endeavor to sculpt a model that harmonizes the trifecta of speed, precision, and computational efficiency.

This refinement process systematically scales down the model's complexity, transitioning the 'depth_multiple' and 'width_multiple' values through a sequence from 0.5 to 0.33, then to 0.125, and

ultimately to 0.0625. Each stage of this transition is designed to incrementally streamline the model, ensuring that each modification contributes towards an enhanced balance of model performance and computational demand.

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 also viable for real-time application within constrained computational environments.

2.4. Dataset

This study utilizes the Nighttime-Yawning-Microsleep-Eyeblick-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 symbolizes the accurately identified eye states, FP denotes instances erroneously classified as eye states, and FN 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.

$$AP = \int_0^1 P(R) dR \tag{3}$$

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

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 (Frames Per Second).

GFLOPS measures the computational complexity of the model, indicating the number of billion floating-point operations required to process a single input. It's 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. 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, 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, 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	36487166	314	0.995	0.984	0.996	74.8
Yolov7-tiny	6010302	208	0.991	0.992	0.996	12.3
Yolov7-tiny - First-Prune	3963128	182	0.999	1	0.996	8.1

Yolov7-tiny - Second-Prune	3449878	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, 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)
	0.5	864174	156	0.984	0.992	0.996	1.9
Yolov7-tiny -	0.33	358878	156	0.997	1	0.996	0.9
Second-Prune	0.125	55742	156	0.997	1	0.996	0.3
	0.0625	20174	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 the 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

Structured pruning involves selectively removing less important neurons or connections within the network, effectively reducing the model's complexity without substantially degrading its performance. This technique is evident in the transition from the original YOLOv7 model to its pruned variants, including the YOLOv7-tiny first and second pruned versions. The gradual decrease in GFLOPS alongside an increase in FPS indicates that structured pruning successfully reduces computational overhead, enabling faster processing times essential for real-time applications.

The fine-tuning process is critical, especially after significant changes to the model's structure, to ensure that the model remains effective in detecting objects with high precision. 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.

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.

Data Availability Statement: The public database used in this paper is NITYMED (night-time yawning–microsleep–eyeblink–driver distraction), which can be found in <https://datasets.esdalab.ece.uop.gr/>.

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

References

1. National Highway Traffic Safety Administration. CAFE Compliance and Effects Modeling System. Available online: <https://www.nhtsa.gov/corporate-average-fuel-economy/cafecompliance-and-effects-modeling-system> (accessed on 4 March 2022).
2. 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.
3. 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.
4. 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, <https://doi.org/10.3390/s24051541>.
5. Karuppusamy, N.S.; Kang, B.-Y. Multimodal System to Detect Driver Fatigue Using EEG, Gyroscope, and Image Processing. *IEEE Access* 2020, 8, 129645–129667.
6. 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, <https://doi.org/10.3390/app12178674>.
7. Anber, S.; Alsaggaf, W.; Shalash, W. A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features. *Electronics* 2022, 11, 285.
8. Zheng, H.; Wang, Y.; Liu, X. Adaptive Driver Face Feature Fatigue Detection Algorithm Research. *Appl. Sci.* 2023, 13, 5074, <https://doi.org/10.3390/app13085074>.
9. 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.
10. 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, <https://doi.org/10.3390/app13137849>.

11. Iskandarani, M.Z. Relating Driver Behaviour and Response to Messages through HMI in Autonomous and Connected Vehicular Environment. *Cogent Eng.* 2022, 9, 1. <https://doi.org/10.1080/23311916.2021.2002793>.
12. 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, 1–28. <https://doi.org/10.3390/app11041397>.
13. 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.
14. 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.
15. 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.
16. 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.
17. Alameen, S.A.; Alhothali, A.M. A Lightweight Driver Drowsiness Detection System Using 3DCNN with LSTM. *Comput. Syst. Sci. Eng.* 2023, 44, 895–912.
18. Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A Deep Learning Approach to Detect Driver Drowsiness. *Int. J. Eng. Res. Technol.* 2021, 10, 183–189.
19. 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.
20. 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. <https://doi.org/10.1007/s00521-020-05209-7>.
21. 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. <https://doi.org/10.3390/agriculture14020303>.
22. 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. <https://doi.org/10.3390/a16110520>.
23. 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. <https://doi.org/10.3390/agronomy13112667>.
24. 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. <https://doi.org/10.3390/s24020565>.
25. 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. Available online: <https://arxiv.org/abs/2207.02696> (accessed on 4 March 2022).
26. Petrellis, N.; Voros, N.; Antonopoulos, C.; Keramidas, G.; Christakos, P.; Mousoulitis, P. NITYMED. *IEEE DataPort* 2022.
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 Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, 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 Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, 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.