Research and Development of a Traffic Sign Recognition Module in Vietnam

Pham Xuan Tung
University of Science and Technology of Hanoi, Vietnam | Vietnam Academy of Science and Technology, Vietnam
pham-xuan.tung@usth.edu.vn

Nguyen Luong Thien
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
nlthien@sti.vast.vn

Ngoc Pham Van Bach
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
pvbngoc@sti.vast.vn

Minh Hung Vu
PetroVietnam University, Vietnam
hungvm@pvu.edu.vn

Received: 22 November 2023 | Revised: 4 December 2023 | Accepted: 6 December 2023

ABSTRACT

Automatic traffic sign recognition is essential in researching and developing driver assistance systems and autonomous vehicles. This paper presents the research and development of an automated traffic sign recognition module in Vietnam. The recognition model is developed based on the deep learning model YOLOv5 and incorporates architectural modifications to reduce computational complexity, increase inference speed, and meet real-time requirements for embedded system applications. The model is trained using a custom dataset collected by the research team from real-world street environments in Vietnam, encompassing diverse locations, times, and weather conditions. The trained recognition model is deployed on the Jetson embedded system, yielding high-quality recognition results and meeting real-time recognition needs.

Keywords-deep learning; traffic sign recognition; YOLOv5; embedded system; image processing

I. INTRODUCTION

Researching and developing traffic sign recognition is fundamental in developing driver assistance and autonomous systems. The information obtained from traffic sign recognition can aid drivers in reacting correctly to traffic conditions, reduce the risk of accidents, and improve overall driving safety. Nonetheless, developing a traffic sign recognition module is often faced with two primary challenges: customized traffic sign datasets and the trade-off between accuracy and processing speed on embedded systems. Firstly, traffic signs vary in structure and shape in each country and region, which poses difficulties when applying a pre-existing traffic sign recognition system from one country to another. A few research groups in Vietnam have conducted studies and solutions for traffic sign recognition based on the German Traffic Sign Recognition Benchmark (GTSRB) dataset [1]. However, the signs in this dataset differ significantly in terms of shape, color, and type when compared to the Vietnamese traffic signs. Moreover, since this dataset has been preprocessed and had its background removed, applying the model to real-world street scenarios results in lower recognition accuracy. Secondly, the trade-off between accuracy and processing speed on embedded systems is another crucial factor in the research and development of traffic sign recognition modules. Studies on traffic sign recognition typically focus on detecting regions with traffic signs and extracting distinguishing features represented by sign patterns [2, 3]. The current methods can be divided into two main groups: traditional and deep learning-based. Standard algorithms are often built based on color and shape information extracted from sample images, followed by segmenting and labeling signs through pattern classification [4]. Traditional methods yield lower accuracy due to variations in traffic signs, environmental factors, lighting conditions, image angles, and partially occluded signs. However, they offer fast processing speeds due to their low computational complexity. Recently, with the advancement of computational hardware, research on traffic sign recognition systems has been built using object recognition algorithms based on deep learning models with multilayer neural networks for feature extraction and learning [5]. Nonetheless, most Deep Neural Networks (DNNs) have millions of parameters, and they demand significant resources, memory, and computational costs, which pose challenges related to the overall system performance, particularly real-time application requirements.

Considering the aforementioned challenges, it becomes evident that researching and developing a real-time traffic sign recognition module explicitly designed for Vietnam is necessary and highly practical.

II. MATERIALS AND METHODS

A. Traffic Sign Recognition Model

Deep learning-based object recognition algorithms can be divided into two main types: single-stage object recognition algorithms and two-stage object recognition algorithms. Single-stage algorithms, such as YOLO [6, 7] and Single Shot Multibox Detector [8], can extract features from the network for immediate classification, recognition, and object localization. On the other hand, two-stage detection methods are built on the proposal region principle, where regions of interest are proposed and then classified and refined. R-CNN [9], Fast R-CNN [10], and Faster R-CNN [11] are representative algorithms in this group that rely on Convolutional Neural Network (CNN) models.

Single-stage recognition models based on YOLO have speed and processing efficiency advantages compared to two-stage recognition models like R-CNN and its variants. This makes them suitable for deployment on embedded systems. YOLOv5 [12] consists of three main components: backbone, neck, and head. The backbone serves as the foundation of the model architecture and is responsible for extracting essential features from the input images. It uses the CSPDarkNet53 architecture, which consists of consecutive convolutional layers capable of extracting features at different scales. The neck connects the backbone and the head, processing and transmitting the extracted features from the backbone to the head. The neck utilizes PANet (Pyramid Attention Network) and FPN (Feature Pyramid Network) architectures, which play a role in fine-tuning object detection accuracy. The head is the final component responsible for object recognition based on the extracted features. Some studies have successfully used Yolo deep learning models to solve real-time application problems, such as smoke detection for trajectory planning and navigation of a mobile robot [13] and recognition of road surface anomalies [14].

Two training models, YOLOv5n (nano) and YOLOv5s (small), have been chosen to meet the deployment requirements on embedded systems. Additionally, the CSPDarkNet network architecture in the backbone of the YOLOv5 model is replaced with the MobileNet [15] network to reduce computational complexity and increase computational speed on embedded systems. Based on the training and inference speed results, the recognition models will be considered and selected for application on embedded systems.

B. A Dataset of Vietnamese Traffic signs

The dataset used to train the model was meticulously curated to encompass 16 essential traffic signs frequently encountered while driving, as depicted in Figure 1. Some signs share similarities in color and shape. It is imperative to note that if the model is not performing optimally during training, it may incorrectly identify these similar sign pairs. Therefore, besides selecting the appropriate model and training parameters, the quality of the data source for the sign is also a crucial factor. Data from multiple sources were gathered, taking into account several locations, times, weather conditions, and sign angles.

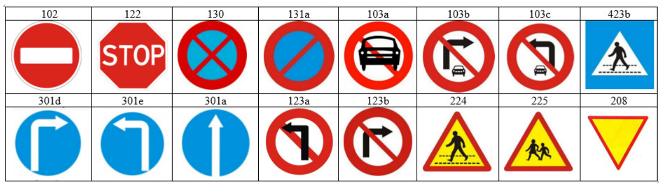


Fig. 1. Vietnamese traffic signs considered in this study.

In order to ensure a diverse range of data sources, we collected information from various devices, including cameras, phones, and dash cams installed on cars and motorbikes traveling on different routes across various provinces and cities throughout Vietnam. The collection of traffic sign samples took place at different times of day and in varying weather conditions. Data augmentation algorithms were applied to address potential imbalances within the dataset that adjust the detector's brightness, dimming, and angle. Additionally, the data were enhanced by superimposing signs onto realistic street scene backgrounds. We also incorporated other traffic sign images from the Vietnamese road traffic sign set into the training set to help the model differentiate between easily confused signs, ultimately improving the quality of the training model.

Fig. 2. The custom dataset.

The data set was split into two sets: a training set accounting for 80% of the images (24896 images) and a test set accounting for the remaining 20% (6224 images).

C. Recognition Model Deployment on the Embedded System

The advanced traffic sign recognition model was successfully implemented on NVIDIA's powerful Jetson NX (8GB) embedded system.. The Jetson NX has a GPU of 48 tensor cores, providing an AI computing capacity of 21 TOPS. The trained model was optimized using TensorRT to further accelerate its inference on embedded systems. Adjusting the accuracy from FP32 to FP16 greatly reduces the number of model calculations without sacrificing precision. This precision correction streamlines the computation process and eliminates the need to re-calibrate the model after training and optimization. The optimized model was deployed on Jetson and developed into a module with camera equipment and a display screen (Figure 3).

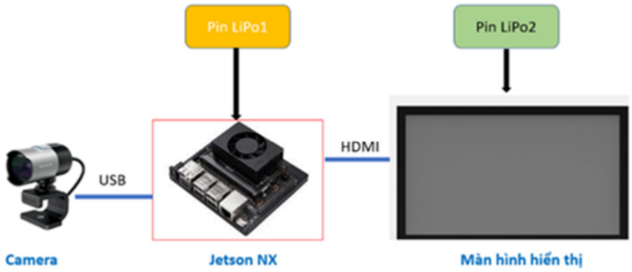


Fig. 3. Schematic of the traffic sign recognition module.

III. TRAINING, EXPERIMENTAL RESULTS, AND DISCUSSION

A. *Training Result*

When assessing the effectiveness of a trained model, several key metrics were considered, such as IoU (Intersection over Union), Precision, Recall, and Mean Average Precision. IoU denotes the proportion of the predicted area that intersects with the actual object image area. Precision represents the ratio of True Positive (TP) predictions to the total number of positive predictions made by the model, encompassing both true and False Positives (FP). The model's mean accuracy (mAP) is computed across all confidence thresholds. However, only mAP 0.5:0.95 derived from IoU ranging from 0.5 to 0.95 significantly contributes to evaluating the prediction model.

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

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

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \quad (3)$$

The training model was run on a Core (TM) i7-12700F 2.10 GHz computer with 32 GB RAM and NVIDIA GeForce RTX 3070 Ti GPU, CUDA 11.7. The models were trained for 100 epochs. The training process results are presented in Figure 3 and Table I. The training results indicate that the backbone correction model using MobileNet has more parameters than the original model, but has better computational performance due to its reduced complexity. The identification accuracy is proportional to the model's complexity. However, the difference in accuracy between the models is not significant.

The YOLOv5n model (MobileNet) has a computational performance of 3.1 GFLOPs for 91.8% mAP_0.5:0.95. The YOLOv5s (origin) model has a computational performance of 16.1 GFLOPs for 93.7% mAP_0.5:0.95.

In addition to the previously mentioned evaluation metrics, we should consider the energy consumption required for training when creating modules, particularly for applications with limited energy resources. As illustrated in Figure 5, the energy consumed by the GPU during model training is directly proportional to its complexity. Out of the four models tested, the YOLOv5n model (MobileNet) exhibited the lowest energy consumption during exercise, averaging 60 Wh. On the other hand, the YOLOv5s (origin) model consumed the highest amount of energy (averaging more than 70 Wh) and took longer to train.

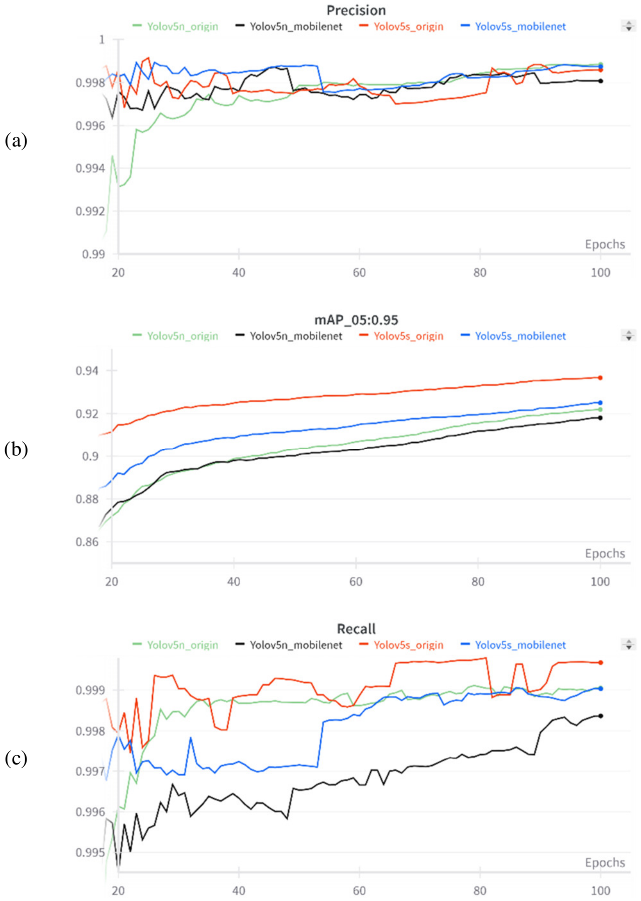


Fig. 4. Training results. (a) Precision, (b) mAP, (c) Recall.

TABLE I. TRAINING RESULTS

Model	Layers	Parameters	GFLOPs	mAP 0.5:0.95	Recall
YOLOv5n	214	1785565	4.3	0.927	0.999
YOLOv5n (MobileNet)	287	2156253	3.1	0.918	0.999
YOLOv5s	214	7062781	16.1	0.937	0.999
YOLOv5s (MobileNet)	287	4667901	7.4	0.925	0.998

The models were optimized using TensorRT and were deployed on Jetson NX. The research team conducted speed tests on an embedded system using videos from dashcams with resolutions of 1920×1080 and 1280×720. The inference process involves three main steps: frame pre-processing, prediction, and duplicate prediction removal algorithm (NMS). The frame pre-processing task involves separating frames from the video and resizing them to 640×640 to match the prediction model. The preprocessed frames were then passed into the training model for prediction. Finally, a duplicate removal algorithm was applied to remove any duplicate findings. Table II provides the inference speed of the proposed recognition models.

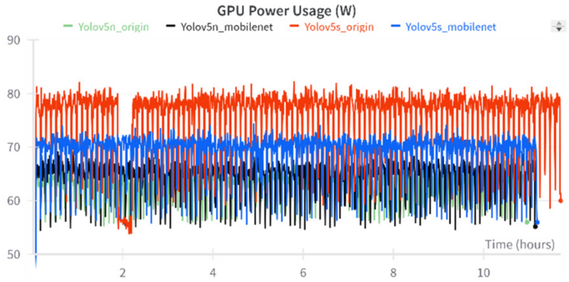


Fig. 5. GPU power consumption during training.

TABLE II. SPEED TRAINING RESULTS

Model	Preprocessing (ms)	Prediction	NMS (ms)
YOLOv5n	2	14.2	5.6
YOLOv5n (MobileNet)	2	13.6	4.9
YOLOv5s	2	17.3	5.2
YOLOv5s (MobileNet)	2	15.3	4.9

According to the experimental findings, it is clear that the MobileNet backbone correction model offers quicker prediction results compared to the corresponding models of YOLOv5s and YOLOv5n. The sign recognition results are notably impressive, particularly when the weather conditions are favorable. Furthermore, the model performs exceptionally well even in scenarios where the signs are captured from tight angles or are partially obscured, as demonstrated in Figure 6.

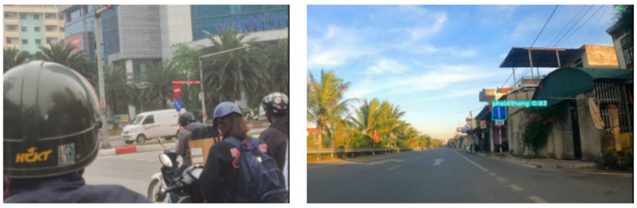
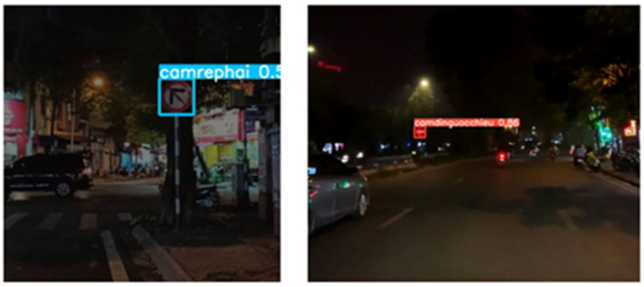


Fig. 6. Traffic sign recognition results under normal conditions.

Therefore, recognition models are not always equally effective under all lighting conditions. It is essential to consider the type of lighting that will be present in the environment where the recognition model will be utilized. The findings of this study highlight the importance of carefully selecting and testing recognition models in various lighting conditions to ensure optimal performance and accuracy.

The effectiveness of recognition models can vary under certain lighting conditions, particularly those in a low-light environment. In Figure 7, we can see that despite the poor lighting conditions, the YOLOv5s, YOLOv5n, and YOLOv5s (MobileNet) models correctly identified the target. However, it is worth noting that in this particular instance, the YOLOv5n (MobileNet) model failed to recognize the target sign.



(a) (b)

Fig. 7. Results of sign recognition in low light conditions.

Upon analyzing Figure 9(b), it is evident that the complexity of the sign recognition escalates with small size and varying surrounding color, e.g. the lights of vehicles on the road at night. In this scenario, only the YOLOv5s model can accurately detect signs that prohibit driving in the opposite direction. The precision of the model corresponds to its level of intricacy during training, which is apparent from the results.

B. Experimental Results on Streets

Through road experiments, it was noted that the module possesses a remarkable ability to detect traffic signs, even with the camera positioned behind a car's windshield (Figure 8). Upon establishing a recognition model with a prediction threshold of 0.5, the outcomes demonstrate that the module can efficiently and accurately identify signs from an average distance under 30 m (Figure 9). The YOLOv5s (original) model has an impressive average FPS (Frames Per Second) processing speed of 54, while the YOLOv5n (MobileNet) model offers an impressive processing speed of 70 FPS. These models provide accurate recognition rates and are highly suitable for real-time deployment. Additionally, users can switch between the two models during deployment based on their accuracy and recognition speed requirements, providing greater flexibility.

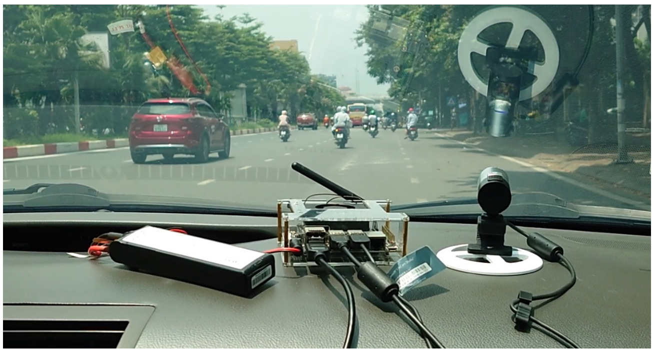


Fig. 8. The module set on a car.

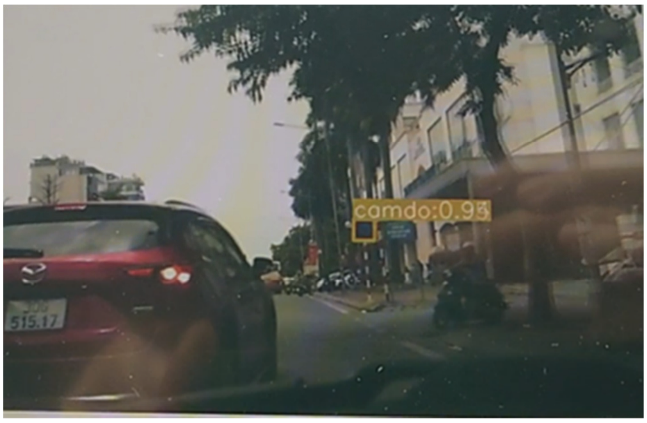


Fig. 9. Experimenting on real conditions.

IV. CONCLUSION

The current study aimed to develop an automatic identification method for common traffic signs in Vietnam. Detection models were created based on YOLOv5, with a modified backbone using the MobileNet model to reduce the amount of computation and speed up the inference model to respond to real-time requirements.

To train the models, a diverse dataset was used, collected from various sources under varying conditions (location, weather, lighting, and shooting angles). Data augmentation techniques were employed to prevent data imbalance, which aided in improving the models' identification ability and prediction accuracy. After training, the modified models gave relatively good results, i.e. mAP 0.918 (Yolov5n+MobileNet model) and mAP 0.925 (Yolov5s+MobileNet).

After training, the optimized model was deployed on the Jetson NX embedded system. The results from actual tests conducted on the streets of Hanoi demonstrated that the module can recognize traffic signs accurately and reliably. The module can detect traffic signs in normal weather conditions and low light conditions at a distance and with a high FPS rate. The module is suitable for real-time deployment.

ACKNOWLEDGMENT

This research is funded by the University of Science and Technology of Hanoi (USTH) under grant number USTH.EN.02/23.

REFERENCES

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, Jul. 2011, pp. 1453–1460, https://doi.org/10.1109/IJCNN.2011.6033395.

[2] W. Sun, H. Du, S. Nie, and X. He, "Traffic Sign Recognition Method Integrating Multi-Layer Features and Kernel Extreme Learning Machine Classifier," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 147–161, 1970, https://doi.org/10.32604/cmc.2019.03581.

[3] W. Ali, G. Wang, K. Ullah, M. Salman, and S. Ali, "Substation Danger Sign Detection and Recognition using Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 10051–10059, Feb. 2023, https://doi.org/10.48084/etasr.5476.

[4] C. Wang, "Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning," in *2018 International Conference on Robots & Intelligent System (ICRIS)*, Changsha, China, Feb. 2018, pp. 150–152, https://doi.org/10.1109/ICRIS.2018.00047.

[5] A. Avramović, D. Sluga, D. Tabernik, D. Skočaj, V. Stojnić, and N. Ilc, "Neural-Network-Based Traffic Sign Detection and Recognition in High-Definition Images Using Region Focusing and Parallelization," *IEEE Access*, vol. 8, pp. 189855–189868, 2020, https://doi.org/10.1109/ACCESS.2020.3031191.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525, https://doi.org/10.1109/CVPR.2017.690.

[8] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, 2016, pp. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, Jun. 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81.

[10] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Sep. 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, https://doi.org/10.1109/TPAMI.2016.2577031.

[12] G. Jocher *et al.*, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation." Zenodo, Aug. 22, 2022, https://doi.org/10.5281/zenodo.7347926.

[13] P. V. B. Ngoc, L. H. Hoang, L. M. Hieu, N. H. Nguyen, N. L. Thien, and V. T. Doan, "Real-Time Fire and Smoke Detection for Trajectory Planning and Navigation of a Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11843–11849, Oct. 2023, https://doi.org/10.48084/etasr.6252.

[14] D. D. Van, "Application of Advanced Deep Convolutional Neural Networks for the Recognition of Road Surface Anomalies," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10765–10768, Jun. 2023, https://doi.org/10.48084/etasr.5890.

[15] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, Apr. 16, 2017, https://doi.org/10.48550/arXiv.1704.04861.