CNN-based Traffic Sign Detection on Embedded Devices*

Aleksandra Kos[1,2][0000−0001−9726−4472] and Karol Majek[1][0000−0002−1351−8496]

[1] Cufix, 05-825 Grodzisk Mazowiecki, Poland
[2] Poznan University of Technology, 60-965 Poznan, Poland
aleksandra.kos@doctorate.put.poznan.pl

Abstract. Traffic sign detection is a key task in autonomous driving. In addition to high accuracy, the algorithm must operate in real-time on an embedded device. Traffic signs are often found occupying a small area of a high-resolution image and can be easily confused with other signs and billboards. We analyze the aforementioned challenges, using the YOLOv4 model, which we train on the Mapillary Traffic Sign Dataset (MTSD) with a designed data augmentation method and weighted loss function. We achieve $AP_{50} = 59.0\%$ on the validation dataset. The contribution of this work is a quantized YOLOv4 traffic sign detector with an input resolution of 960×960px. The proposed model is optimized to achieve better performance on devices with limited computational resources. Our model runs at 11.2 FPS on Nvidia Jetson Xavier AGX.

Keywords: Traffic sign detection · Mapillary Traffic Sign Dataset · YOLOv4.

1 Introduction

Robotic applications of object detection algorithms often require implementation on devices with limited computational resources and memory. To run the existing object detectors in real-time, lightweight models and limited input resolution are required, which can lead to poor detector accuracy [1, 9]. To alleviate these problems, techniques for optimizing the model inference performance, such as quantization and pruning [4, 5], are used. Our goal is to analyze the challenges of real-time traffic sign detection and suggest a method that deals with this problem on embedded devices. In this work, we use MTSD to train the YOLOv4 model, then optimize our network with the tkDNN [9] library for inference on an embedded device, and assess the detection quality with 12 COCO [6] metrics. The contributions of this work are as follows: a detailed analysis of a large traffic-signs detection dataset (MTSD), a trained model capable of detecting signs belonging to 314 classes ($AP_{50} = 59.0\%$), and analysis of the results and suggestions on how to improve the system.

* The research was supported by the Ministry of Education and Science as part of the "Doktorat Wdrożeniowy" program (DWD/5/0203/2021).

2 Related Work

In recent years, many real-time object detection methods, such as [1], as well as autonomous driving datasets [11, 3] have been published. These datasets provide a variety of data from different sensors, but lack detail in road sign classes. Until recently, a thorough examination of detectors in traffic sign detection was practically impossible, due to the absence of a large dataset that would contain realistic data [2, 8, 13]. Traffic sign detection can be solved using single-stage object detection methods with large input image size, as in TSingNet [7], which achieves 20.6 FPS using the desktop Nvidia GeForce GTX 1080 GPU. Object detection networks can be optimized to achieve high performance on embedded systems when moving to embedded GPUs, such as Nvidia Jetson series models [12].

3 Dataset Analysis

In Tab. 1 we demonstrate the main features of selected traffic sign datasets. MTSD is characterized by the largest number of object instances and classes, as well as the highest variability in image and object size. It has 41909 labeled, and

Table 1. Comparison of traffic sign datasets. Sizes were calculated as geometric means $s = \sqrt{w \cdot h}$

Dataset	Images	Objects	Classes	Image size	Object size	Country	Year
RTSD [8]	59188	104358	198	1052.02±188.95	38.76±22.85	Russia	2016
TT100K [13]	16811	26349	182	2048.00±0.00	45.85±31.62	China	2016
MTSD [2]	41909	206388	314	2837.99±911.44	63.10±71.62	Global	2019

10544 unlabeled images. Each object is annotated with an axis-aligned bounding box and an identifier of one of the 314 classes. The classes are grouped into 5 main categories: *information*, *complementary*, *regulatory*, *warning* and *other*. *Other* is both a category and a class, and accounts for about 70% of objects. In addition

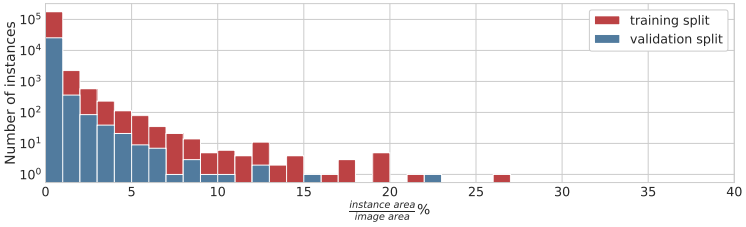

Fig. 1. Histogram of relative object areas in the MTSD dataset.

to the uneven class distribution, many images are larger than 10 MPx and most objects take up less than 1% of that area (see Fig. 1).

4 Results

We trained two YOLOv4 [1] object detectors with 960×960 input resolution and the SGD optimizer with momentum and weight decay. The first training process involved minor color and geometric transformations and allowed us to achieve $AP_{50} = 41.5\%$. The second model was trained on a dataset obtained with a more complex data augmentation method, which together with the use of the weighted cost function, increased the AP_{50} to 59.0%. The model with higher AP_{50} was optimized (FP32, FP16, INT8) using tkDNN [9] to enable inference on Nvidia Jetson Xavier AGX. In Table 2 we show the impact of quantization on the quality and speed of the detector.

Table 2. COCO metrics [6] and speed (defined as frames per second) of YOLOv4 on the MTSD validation set.

Precision	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
FP32	4.2	**34.4**	**59.0**	**35.6**	**31.9**	51.3	56.7	**44.9**	53.5	**53.6**	50.2	**64.3**	**68.5**
FP16	9.0	**34.4**	58.9	**35.6**	**31.9**	51.3	56.4	**44.9**	**53.6**	**53.6**	**50.3**	**64.3**	67.9
INT8	**11.2**	33.4	56.1	35.3	29.2	**52.7**	**58.9**	43.9	51.0	51.1	47.0	64.0	68.4

Considering the reduced input resolution and the large number of small instances, we decided to recalculate the metrics, discarding tiny ($s < 8^2$) and very tiny ($8^2 \leq s < 16^2$) objects [10]. The new validation datasets had 19 419 and

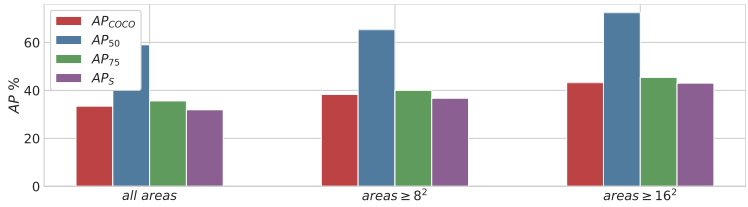

Fig. 2. The impact of discarding tiny and very-tiny objects on the average precision.

11 518 instances, respectively, compared to 26 101 objects in the original dataset. Fig. 2. shows the resulting changes in AP.

5 Conclusion and Future Work

The results presented in Fig. 2 show that the tiny and very tiny objects decrease the average precision (AP) of traffic signs detection (by up to 9.8% in our experiments). Optimizing the model does not degrade the quality, but allows for 3 times faster inference, as shown in Tab 2. For the INT8 model, an increase

in AP_M and AP_L was observed. An in-depth analysis of the effect of sample selection on quality is necessary, and we consider it to be our future work. In addition, we plan to investigate techniques such as two-step detection (utilizing Region of Interest proposals), detection in original resolution, validation on additional datasets, and analysis of AP for tiny and very tiny objects.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Ertler, C., Mislej, J., Ollmann, T., Porzi, L., Neuhold, G., Kuang, Y.: The mapillary traffic sign dataset for detection and classification on a global scale. In: European Conference on Computer Vision. pp. 68–84. Springer (2020)
3. Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A.S., Hauswald, L., Pham, V.H., Mühlegg, M., Dorn, S., et al.: A2d2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320 (2020)
4. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
5. Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X.: Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing **461**, 370–403 (2021)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
7. Liu, Y., Peng, J., Xue, J.H., Chen, Y., Fu, Z.H.: Tsingnet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild. Neurocomputing **447**, 10–22 (2021)
8. Shakhuro, V.I., Konouchine, A.: Russian traffic sign images dataset. Computer optics **40**(2), 294–300 (2016)
9. Verucchi, M., Brilli, G., Sapienza, D., Verasani, M., Arena, M., Gatti, F., Capotondi, A., Cavicchioli, R., Bertogna, M., Solieri, M.: A systematic assessment of embedded neural networks for object detection. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 937–944. IEEE (2020)
10. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 3791–3798. IEEE (2021)
11. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2636–2645 (2020)
12. Zhang, X., Lu, H., Hao, C., Li, J., Cheng, B., Li, Y., Rupnow, K., Xiong, J., Huang, T., Shi, H., et al.: Skynet: a hardware-efficient method for object detection and tracking on embedded systems. Proceedings of Machine Learning and Systems **2**, 216–229 (2020)
13. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2110–2118 (2016)