Multispectral Feature Fusion for Deep Object Detection on Embedded NVIDIA Platforms

Thomas Kotrba*†‡, Martin Lechner†‡, Omair Sarwar‡ and Axel Jantsch† *Senior Member, IEEE*
†Christian Doppler Laboratory for Embedded Machine Learning,
Institute of Computer Technology, TU Wien, Vienna, Austria
‡Mission Embedded GmbH, Vienna, Austria
*Corresponding author: Email: tko@mission-embedded.com

***Abstract*—Multispectral images can improve object detection systems' performance due to their complementary information, especially in adverse environmental conditions. To use multispectral image data in deep-learning-based object detectors, a fusion of the information from the individual spectra, e.g., inside the neural network, is necessary. This paper compares the impact of general fusion schemes in the backbone of the YOLOv4 object detector. We focus on optimizing these fusion approaches for an NVIDIA Jetson AGX Xavier and elaborating on their impact on the device in physical metrics. We optimize six different fusion architectures in the network's backbone for the TensorRT framework and compare their inference time, power consumption, and object detection performance. Our results show that multispectral fusion approaches with little design effort can benefit resource usage and object detection metrics compared to individual networks.**

***Index Terms*—multispectral fusion, deep object detection, embedded hardware, NVIDIA Jetson**

I. Motivation

Driven by use cases like autonomous driving and surveillance, neural network-based object detection has become an active research field in recent years. Dealing with safety-critical use cases makes it mandatory for such object detection systems to perform as accurately as possible within given time constraints since a misinterpretation of a scene could cause potential danger, risk, or injury to human life. Hence, much research has gone into improving object detection and making it more robust against real-world environmental influences.

Multispectral input images can improve object detectors by providing additional information in normal conditions and by giving alternative information for processing when individual spectra are disturbed. Although multispectral fusion is a hot research topic, experiments on embedded hardware are rarely conducted. Furthermore, state-of-the-art papers in multispectral fusion architectures often include significant design efforts in architecture and training.

II. Main Ideas

A recently published comprehensive survey [1] on deep multi-modal (including multispectral) sensor fusion methods lists the lack of general design methodologies and the lack of real-time considerations on standard devices as two challenges of fusion approaches.

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

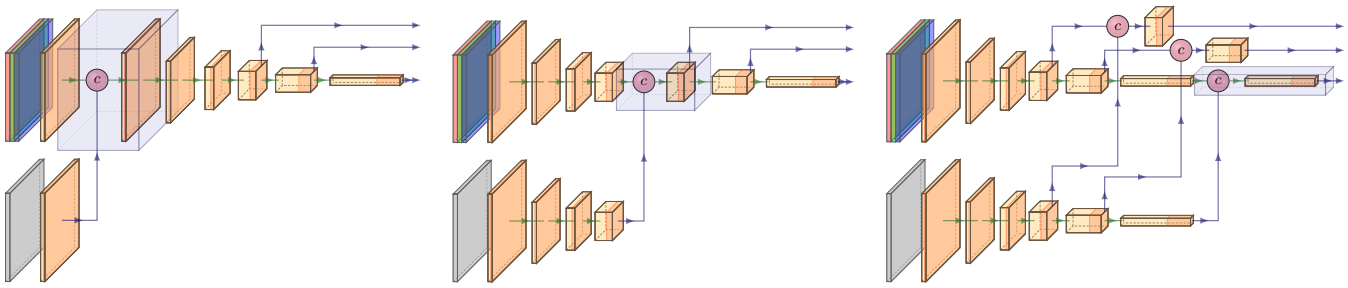
The question arises whether even simple fusion approaches can achieve measurable advantages in a real-world application on an embedded device or if more sophisticated methods are mandatory. To address the question, we propose six basic fusion architectures in the backbone of the YOLOv4 [2] architecture and optimize them for the TensorRT framework and measure their impact in physical metrics on an NVIDIA Jetson AGX Xavier.

We deliberately choose architectures that are easy to implement and do not require excessive design effort. We select six fusion positions in the backbone of YOLOv4 (three of them are shown in Fig. 1[1]) and arbitrarily group two of each into early, mid, and late fusion according to the fusion position in the backbone. A single fusion operator is sufficient for early and medium fusion architectures to unify the feature streams. For the late fusion approaches, additional fusion operators are required for fusing the streams going to the neck of the network. Because we focus on the fusion schemes themselves and eliminate an additional degree of freedom in our comparison, we choose to select the same fusion operator for all experiments. We select a relatively simple operator called Network in Network (NIN) [3]. The fusion is realized by concatenating the individual feature streams followed by a 1×1 convolutional layer to reduce the dimensions to match the original ones. After the NIN, the remaining network layers can be implemented unmodified.

Since our proposed fusion approaches do not explicitly tackle the problem of spatial shifts between the individual spectra, we need a dataset with aligned image pairs. The KAIST multispectral dataset [4] is well-suited for this task. It contains 95 328 image pairs in the visible and infrared spectrum with a resolution of 640×512 pixels. For evaluation, we measure the log-average miss rate (LAMR) [5] values by using the definition of the *Reasonable* subset: only not heavily occluded pedestrian instances taller than 55 pixels in height, and the *All* subset: All occlusions and sizes [4].

All experiments are implemented in the *Darknet* framework [2], which was modified to allow multispectral input images. We use the TensorRT framework to achieve the maximum inference performance on the NVIDIA Jetson AGX Xavier. To convert the networks from the Darknet framework format to

[1]The network plots in this work are created with a modified version of https://doi.org/10.5281/zenodo.2526396



(a) (b) (c)

Fig. 1. YOLOv4: Selection of the investigated backbone fusion architectures: (a) early fusion I; (b) middle fusion II; (c) late fusion II;

TABLE I
FUSION ARCHITECTURE RESULTS: RELATIVE OPERATIONS REFER TO DARKNET, LAMR AND POWER CONSUMPTION TO TENSORRT

Fusion Architecture	**Relative Operations**	**Inference Time**		**LAMR (Reasonable)**			**LAMR (All)**			**Power Consumption**	
		Darknet	***TensorRT (relative)***	**All**	***Day***	***Night***	**All**	***Day***	***Night***	***GPU***	***System***
RGB Reference	1.00	128 ms	32.29 ms (1.00)	0.189	0.151	0.266	0.439	0.392	0.549	28.074 W	39.706 W
IR Reference	1.00	129 ms	32.50 ms (1.01)	0.192	0.244	0.080	0.380	0.435	0.257	27.191 W	38.908 W
Early Fusion I	1.02	145 ms	36.06 ms (1.12)	0.106	0.133	0.057	0.315	0.339	0.262	27.523 W	39.329 W
Early Fusion II	1.10	170 ms	41.18 ms (1.28)	0.108	0.130	0.065	0.311	0.334	0.254	27.986 W	40.240 W
Mid Fusion I	1.17	181 ms	43.48 ms (1.35)	0.104	0.132	0.050	0.295	0.326	**0.209**	27.861 W	40.448 W
Mid Fusion II	1.33	198 ms	48.18 ms (1.49)	0.091	**0.109**	0.053	0.285	**0.307**	0.235	28.062 W	40.875 W
Late Fusion I	1.50	214 ms	52.82 ms (1.64)	**0.090**	0.114	**0.044**	**0.284**	0.309	0.228	27.820 W	41.246 W
Late Fusion II	1.61	226 ms	55.49 ms (1.72)	0.105	0.134	0.052	0.291	0.321	0.219	27.831 W	41.406 W

TensorRT, we utilize the *tkDNN* library [6]. To further improve the performance of the Jetson, we instruct TensorRT to use FP16 calculations whenever possible.

III. RESULTS

The results in Tab. I show that all fusion architectures perform significantly better than the reference networks in LAMR evaluated on the *All* score in both subsets. Also, in the more detailed evaluation with day and night distinction, almost all architectures achieve better results than the reference networks.

The measured inference times in Tab. I show that the increase for the optimized TensorRT engine is nearly proportional to the network's operations. The optimized TensorRT engines are four times as fast as the corresponding Darknet models.

The power consumption is measured by averaging the current values over 10 minutes with the Jetson-specific tegrastats utility using onboard sensors. Results in Tab. I indicate no evident trend in the power values of the GPU, which likely indicates that the GPU utilization is maxed out for every architecture. The minor uptrend in overall system power can be explained by enhanced memory activity caused by the increasing number of operations.

Our experiments show that almost all tested fusion architectures achieve better object detection results than the reference networks trained on individual spectra. In our evaluation, the Early Fusion I approach can reduce the LAMR in the reasonable setting by 44 % while increasing the inference time by only 12 %. We conclude that a fusion architecture can improve an object detection system on an embedded device even with little design effort.

ACKNOWLEDGMENT

The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

REFERENCES

[1] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021, arXiv: 1902.07830 Publisher: Institute of Electrical and Electronics Engineers Inc.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, arXiv: 2004.10934. [Online]. Available: http://arxiv.org/abs/2004.10934

[3] J. Liu, S. Zhang, S. Wang, and D. N. Metaxas, "Multispectral deep neural networks for pedestrian detection," in *British Machine Vision Conference 2016, BMVC 2016*, vol. 2016-September. British Machine Vision Conference, BMVC, 2016, pp. 73.1–73.13, arXiv: 1611.02644.

[4] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2015, pp. 1037–1045. [Online]. Available: http://ieeexplore.ieee.org/document/7298706/

[5] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[6] M. Verucchi, G. Brilli, D. Sapienza, M. Verasani, M. Arena, F. Gatti, A. Capotondi, R. Cavicchioli, M. Bertogna, and M. Solieri, "A Systematic Assessment of Embedded Neural Networks for Object Detection," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Sep. 2020, pp. 937–944. [Online]. Available: https://ieeexplore.ieee.org/document/9212130/