UDC 004

INTEGRATION OF LIGHTWEIGHT MACHINE LEARNING MODELS FOR MOBILE DIAGNOSTICS OF DEFECTS IN THE FIELD OF AUTO SERVICE

Kotvytska A., Hrama M.
National University of Food Technologies, Kyiv, Ukraine
E-mail: kotvyckaaa@nuft.edu.ua

Automated visual inspection of vehicle defects is a rapidly growing area in the automotive service industry. This report explores the deployment of lightweight deep learning models directly on mobile devices (Edge AI) for real-time detection of body damage, corrosion, and fluid leaks. We analyze the computational efficiency of Depthwise Separable Convolutions and present a comparative analysis of optimized architectures like MobileNetV3 and YOLOv8-nano converted to TensorFlow Lite format. The proposed approach eliminates server latency, protects user privacy, and allows offline operation at service stations.

The current stage of development of the car service industry (STOR) is characterized by the implementation of the Industry 4.0 concept, where automation of routine processes is an absolute priority. Fast and accurate initial assessment of the condition of the car - detection of scratches, dents, corrosion, glass cracks or leaks of technical fluids, without the involvement of expensive stationary equipment, is an extremely urgent task. Traditional cloud solutions of computer vision (Computer Vision) have significant drawbacks: they require stable high-speed Internet, create delays in the transmission of high-resolution images and significantly load the server infrastructure. An alternative is the concept of On-Device ML (Edge AI) – execution of machine learning models directly on the mobile device of the master or client [1]. The main obstacle to the deployment of deep neural networks on smartphones are the strict limitations of hardware resources: the amount of RAM, battery capacity and processor computing power. To overcome these limitations, optimized architectures based on Depthwise Separable Convolutions [2] are used instead of standard convolutional layers.

The key advantage of this approach is a drastic reduction in the number of computational operations. Instead of performing a full-fledged three-dimensional convolution for each filter, this method breaks the process into two stages: first, each input channel is processed separately, and then a lightweight 1×1 pointwise convolution is performed, which mixes the resulting features. Due to this, when using standard 3×3 kernels, the computational cost is reduced by about 8–9 times compared to traditional convolutional layers, and the accuracy loss does not exceed 1–2% relative to full-size counterparts [3]. This makes such models ideal for real-time operation on mobile devices.

As part of the study, we conducted a comparative analysis of four lightweight architectures optimized for mobile platforms and converted to TensorFlow Lite (TFLite) format using 8-bit quantization (INT8). The results of testing on the central and neural processors of a medium-power mobile device are presented in Table 1.

Table 1.

Comparative analysis of mobile architectures for car defect detection

Model architecture	File size (TFLite INT8), MB	Accuracy (mAP@0.5 / Top-1 Accuracy)	Latency (Android CPU), ms	Latency (Android NPU/GPU), ms
MobileNetV3-Small	2.1	67.4%	14	4
MobileNetV3-Large	5.4	75.2%	32	9
EfficientNet-Lite0	4.5	75.1%	28	8
YOLOv8-nano	3.2	78.4% (mAP)	45	12

Analysis of the data in the table shows that for defect classification tasks (for example, determining the type of leaking fluid), the balance of the MobileNetV3-Large model is optimal. For body damage localization tasks (Object Detection) in the image, YOLOv8-nano demonstrates the best results due to its high mAP (Mean Average Precision) index with a response time on a graphics accelerator (GPU) or neuroprocessor (NPU) of only 12 milliseconds, which allows for real-time diagnostics (Real-time Video Stream) [4].

Software integration into a native mobile application in the Kotlin language is implemented through the TensorFlow Lite Task Vision API or Firebase ML Kit library. Using a hardware delegate (NNAPI Delegate) allows you to redirect calculations from the smartphone's central processor to specialized neurochips, which further reduces energy consumption during long-term work of the master-receiver in the shop. Dynamic model updates are performed using the Firebase Custom Model Deployment service, which allows you to retrain the neural network on new data specific to a specific service station, without the need to re-release the mobile application on Google Play or the App Store [5].

The integration of lightweight machine learning models directly into mobile car service systems provides full diagnostic autonomy (including work in underground boxes without Internet access), instant interface response, and high accuracy of defect detection. This not only significantly increases customer trust, but also significantly speeds up the service process.

References

1. Khan A. (2023) Edge AI: Convergence of Machine Learning and Edge Computing. IEEE Transactions on Cybernetics, vol. 53, no. 4, pp. 2104–2115.
2. Howard A., Sandler M., Chu G. (2019) Searching for MobileNetV3, *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324.
3. Google Developers (2024) *TensorFlow Lite for Mobile and Edge Devices* [online]. URL: https://www.tensorflow.org/lite.
4. Jocher G. (2023) *Ultralytics YOLOv8 Docs: Real-time Object Detection Framework* [online]. URL : https://docs.ultralytics.com.
5. Firebase Team (2025) *Firebase Machine Learning: Firebase Custom Model Deployment* [online]. URL : https://firebase.google.com/docs/ml/use-custom-models.