Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Hybrid Detector–ViT-LSTM Approach for Pothole Recognition in Road Video Sequences

Zahra ELMI

zahraelmi@beykoz.edu.tr

Beykoz Üniversitesi

Mustafa TOKAT

Research Article

Keywords:

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11578620/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Hybrid Detector–ViT-LSTM Approach for Pothole Recognition in Road Video Sequences

Zahra Elmi[1*] and Mustafa Tokat[2]

[1*]Beykoz University, İstanbul, Türkiye.
[2]Corneural Artificial Intelligence and Deep Technologies, Inc., İzmir, Türkiye.

*Corresponding author(s). E-mail(s): zahraelmi@beykoz.edu.tr;
Contributing authors: mustafatokat@corneural.com;

Abstract

This paper presents a hybrid deep learning approach for sequence-level pothole recognition in autonomous vehicles by combining a YOLOv8s detector with Vision Transformer (ViT) and Long Short-Term Memory (LSTM) networks. The unique characteristics of potholes, varying shapes, sizes, depths and lighting conditions, make their detection particularly challenging. To represent the short temporal evidence available from a moving vehicle, each road segment is represented by a non-overlapping 16-frame sequence corresponding to 0.53 s at 30 fps. Our proposed detector–ViT-LSTM pipeline leverages the strengths of each component: the detector's ability to localise the pothole in every frame, ViT's ability to capture global spatial relationships in the proposed region, and LSTM's capability to process how that region evolves over 16 consecutive frames. Using road video recorded from a moving vehicle, segmented into non-overlapping 16-frame sequences with frame-level box annotations and evaluated under a sequence-level method with five folds and three seeds, we show that the hybrid pipeline improves on YOLOv8 alone and on single-frame ViT classification. The fused model achieves a PR-AUC of 0.85, an F1 score of 0.85 and a recall of 0.63 at 90% precision, compared with 0.75, 0.78 and 0.39 for YOLOv8 alone and 0.67, 0.66 and 0.39 for a single-frame ViT. The results indicate that temporal verification is more effective when applied to detector-proposed regions than to whole-strip representations. These results indicate the potential of the proposed framework for sequence-level road-surface perception in autonomous-driving pipelines, where false-positive detections could lead to unnecessary speed or trajectory adjustments and a missed pothole affects safety and driving comfort.

Keywords: pothole recognition, road video sequences, sequence-level evaluation, object detection, vision transformer, LSTM, autonomous driving

1 Introduction

Driving automation is commonly classified into six levels, from Level 0 to Level 5 [1]. Regardless of the automation level, reliable environmental perception remains a prerequisite for safe vehicle operation.

More broadly, autonomous vehicle technology has the potential to transform transportation by improving both passenger safety and overall traffic safety. Abdel-Aty et al. [2] analyzed data from 2,100 advanced driving and driver assistance system-equipped vehicles together with 35,113 accident records involving human-driven vehicles. Their findings suggest that, under similar scenarios, the accident risks of vehicles equipped with autonomous driving technologies are lower than those of human-driven vehicles. A similar conclusion was also reported in [3], further reinforcing the safety potential of autonomous systems.

However, achieving safer and more reliable autonomous driving requires accurate perception not only of surrounding vehicles, pedestrians, and traffic signs, but also of the road surface itself. Among road-surface defects, potholes are one of the most widespread and persistent infrastructure problems worldwide. They pose serious risks to road users and cause substantial economic losses through vehicle damage and increased maintenance costs. Therefore, timely pothole detection is essential for improving road safety and enabling efficient maintenance operations.

The severity of this problem has been reflected in recent statistics from various countries. According to [4], potholes account for 22% of accidents caused by road defects in Turkey in 2023. A similar pattern has been reported in South Korea [5], where pothole-related traffic issues have persisted in recent years. Between 2015 and 2019, approximately 47,200 potholes were recorded along 4,767 km of expressways, corresponding to an average of nearly 10 potholes per kilometer. Furthermore, 221 pothole-related traffic accidents occurred between 2017 and 2019, while in the Seoul Metropolitan Area alone, 33,885 potholes were reported, with 36.5% of them emerging during the wet season in July and August. Potholes also negatively affect pedestrian safety, vehicle integrity, and driving comfort. In the United Kingdom, councils across England and Wales have paid out more than £32 million in compensation over the past five years for pothole-related injuries [6]. Between 2017 and 2021, 157 councils settled 5,596 personal injury claims caused by potholes, with an average payout of £5,745 per case. Taken together, these findings clearly highlight the serious safety and economic consequences of inadequate road maintenance and underscore the urgent need for efficient pothole detection and repair systems.

In parallel with this need, vision-based road-distress monitoring has been rapidly evolving toward transformer-based and multi-modal pipelines. For example, a 2024 *Scientific Reports* study demonstrated that Vision Transformer-based classifiers can improve robustness in simultaneous pothole and traffic-sign detection under complex

illumination and occlusion conditions [7]. Similarly, large-scale evaluations have shown that combining image detection with complementary geometric cues, such as point-cloud features, can improve quantification quality and operational reliability for field deployment [8]. In addition, recent review studies in the civil-infrastructure literature emphasize that data quality, annotation methods, and cross-domain generalization are now as critical as model architecture for building sustainable pavement-monitoring systems [9, 10].

Contemporary studies further highlight the practical importance of robust road-damage perception for intelligent and autonomous transportation. Improved YOLO-based models have reported meaningful gains in road-defect localization accuracy under complex road conditions, highlighting the importance of architecture-level refinements for detecting small and irregular defects [11–13]. Complementary field-oriented studies have further shown that vision-based defect detection can be integrated with road-condition indicators such as the International Roughness Index (IRI) to support maintenance prioritization [14]. At the system level, autonomous-driving safety analyses indicate that reliability under rare and edge-case conditions remains a central challenge, which further motivates the development of pothole detection frameworks with high recall and low false-alarm rates [15].

Within autonomous driving systems, sensors such as cameras, LiDAR, and radar are used to capture information about traffic scenes and road environments. These data are then processed by object-recognition models, whose outputs support vehicle decision-making. In this context, accurate road-surface perception is especially important for detecting potholes, which remain challenging targets due to their highly variable visual characteristics. Unlike many conventional road objects, potholes differ substantially in size, shape, color, texture, illumination, and apparent depth. Variations in lighting and shadowing further complicate their recognition, making pothole detection a difficult task for existing models.

Most of the pothole literature, including the studies above, works with still images collected from the web or photographed on foot and reports frame-level accuracy or mAP under random image splits [16–18]; this is the established and appropriate setting for image-based classification and detection. A moving vehicle raises two further questions. First, in the present dataset short road-surface events are represented using 16-frame sequences, corresponding to 0.53 s at the acquisition rate of 30 fps, so the natural evidence is a short sequence rather than a single frame. What aggregating frames contributes, and what a learned sequence model adds over simple averaging, has not yet been measured separately on the same features and the same split. Second, detection and classification are mostly evaluated as separate tasks [18], whereas a driving system uses the location of the pothole and the decision about the road segment ahead together. A false positive leads to an unnecessary braking or steering reaction and a false negative to an unnoticed pothole, so the costs of the two error types have to be weighed together when the operating point is chosen.

This paper addresses both questions with a two-stage pipeline evaluated on road video sequences recorded from a moving vehicle. The contributions are the following.

- **A two-stage pipeline that decides at the sequence level.** A YOLOv8s detector on a 768×192 road strip proposes the pothole location in each frame; a

frozen ViT-B/16 encodes the proposed region across the 16 frames; a ViT-LSTM head (four-layer bidirectional LSTM with multi-head attention pooling) verifies the sequence. Detector confidence and temporal score are combined into one score for the road segment ahead.

- **An ablation that separates three effects**: looking at the proposed region instead of the whole strip, looking at 16 frames instead of one, and learning the temporal model. The ablation compares whole-strip and detector-crop representations using matched single-score and recurrent aggregation settings, evaluated on the same sequence unit, under a declared split method with five folds and three seeds.
- **An operating point for driving decisions**: one decision per road segment, chosen at recall for precision ≥ 0.9, so that the vehicle reacts to potholes it is confident about and does not react to patches, manhole covers and shadows.

Section 2 reviews related work. Section 3 describes the dataset and the annotation method, Section 4 the pipeline and the ablation heads, and Section 5 the evaluation method. Results are reported in Section 6 and discussed in Section 7; Section 8 concludes.

2 Related Work

2.1 Pothole and pavement-distress detection

Three dominant directions in pothole and pavement-distress detection are evident in the recent literature: (i) migration from purely CNN-based pipelines to hybrid Transformer-CNN designs, (ii) emphasis on real-time deployability through lightweight backbones and model compression, and (iii) stronger focus on robustness across illumination, weather, and road-material shifts [7–9]. These trends are closely aligned with autonomous-driving requirements, where both detection accuracy and latency stability are safety-critical.

Another recurring finding is that many models report strong benchmark performance but remain vulnerable to distribution shift, particularly in night-time scenes, wet surfaces, and geographically diverse road textures [10]. Therefore, current research increasingly emphasizes generalization-aware design choices, including richer augmentation strategies, multi-source datasets, and architecture-level mechanisms that preserve global context while maintaining efficient inference. This motivation is directly relevant to the ViT-LSTM strategy adopted in this study, which aims to combine global feature modeling and sequence-aware refinement for more stable pothole recognition under real-world driving conditions.

Evidence from recent studies further indicates that road-damage detection performance is increasingly shaped by two key factors: fine-grained defect representation and deployment-oriented robustness. Studies based on improved YOLO families have shown that attention-enhanced and feature-fusion strategies can improve the detection of small-scale and heterogeneous road defects [11–13]. In addition, research integrating pothole detection with roughness-related indicators suggests that combining defect detection with condition-assessment outputs can better support maintenance decision workflows [14]. From an autonomous-systems perspective, the rarity of safety-critical

events remains a major limitation in real-world validation, highlighting the need for models that maintain stable behavior beyond average-case scenarios [15].

Earlier studies also established the practical feasibility of vision-based pothole detection under challenging environmental conditions. For example, Prabhakar et al. [19] evaluated Faster R-CNN on KITTI and other datasets collected under normal and wet-weather conditions in India, using labels such as bus, car, motorcycle, and human, and reported AP values of 0.98, 0.92, 1.00, and 0.98, respectively. Although this study was not limited to potholes, it demonstrated the effectiveness of deep object-detection frameworks in complex road environments.

More directly related to pothole detection, Khan et al. [20] compared YOLOv8 with YOLOv5small and YOLOv5large for pothole detection and reported that YOLOv8 outperformed the compared alternatives. Similarly, Jakubec et al. [21] evaluated both CNN-based two-stage detectors and YOLO-based one-stage detectors using images captured under diverse weather and lighting conditions, including clear, rainy, sunset, evening, and night scenes. Their results, summarized in Table 1, show a substantial performance decline under adverse conditions, especially at night, confirming that environmental variability remains a major obstacle for reliable pothole detection.

In addition, [22] investigated several real-time deep learning architectures for pothole detection using a dataset composed of windshield-mounted cellphone images and additional internet-sourced pothole images. Their experiments compared SSD-TensorFlow, YOLOv3-Darknet53, and YOLOv4-CSPDarknet53, and showed that YOLOv4 achieved 81% recall, 85% precision, and 85.39% mAP while operating at approximately 20 FPS with an image resolution of 832×832. The proposed system was also reported to detect potholes at distances of up to 100 meters. Likewise, in [23], the SPFPN-YOLOv4-tiny model was developed by integrating spatial pyramid pooling and a feature pyramid network with CSPDarknet53-tiny. Using a dataset expanded to 2665 samples through augmentation techniques such as gamma adjustment, horizontal flipping, and scaling, the authors reported that SPFPN-YOLOv4-tiny outperformed YOLOv2, YOLOv3, and YOLOv4-tiny, achieving 79.6% mAP at IoU = 0.5.

Alternative lightweight and hybrid classification-oriented approaches have also been explored. In [24], features extracted by MobileNetV2 were further reduced using PCA, LDA, and t-SNE, after which several machine-learning classifiers, including SVM, Logistic Regression, Random Forest, Elastic Network, and Decision Tree, were evaluated. Among these methods, SVM yielded the best reported classification performance, reaching 99% accuracy. While such results are promising, they are not directly comparable to object-detection metrics such as precision, recall, or mAP, which are more informative for real-world pothole localization tasks.

Beyond pothole-specific studies, the broader autonomous-driving literature provides useful context for understanding perception-system requirements. Al-Qizwini et al. [25] proposed a lane-following method based on angular deviation estimation with respect to lane boundaries and compared Clarifai, VGGNet, and GoogleNet, reporting the lowest RMS error with GoogleNet. Ciberlin et al. [26] investigated object detection and tracking for autonomous-driving applications and found that YOLOv3 outperformed Viola-Jones in detection performance, while Median Flow provided higher processing efficiency in tracking. Lo Bianco et al. [27] developed a

semantic-segmentation framework based on ERFNet for detecting road contours and surrounding road participants using RGB input, showing effective performance on the large-scale BDD100K dataset. Collectively, these studies underline that robust visual perception is central to tasks such as road understanding, collision avoidance, and driving-path planning.

The effectiveness of such perception systems also depends on the sensing modality. As noted in [28], cameras are cost-effective and widely used, but their performance can degrade under low- or high-illumination conditions. LiDAR sensors provide stable operation through three-dimensional scanning and accurate distance estimation, but they impose considerable computational demands. Radar sensors can also contribute to perception, although their efficiency may be reduced by unnecessary activity and sensitivity to electromagnetic interference from other radar signals or wireless devices. For this reason, vision-based methods remain especially attractive for scalable pothole detection, provided that sufficient robustness can be achieved under real-world operating conditions.

Overall, the literature indicates that pothole detection has progressed from conventional CNN-based approaches toward more robust, efficient, and context-aware architectures. However, limitations related to adverse weather, night-time visibility, domain shift, and deployment constraints remain only partially resolved. These gaps motivate the present study, which investigates a ViT-LSTM-based approach to improve pothole detection stability under diverse and realistic driving scenarios.

Three recent studies are closest to the present work in aim. Baek and Chung [16] convert road images to grey scale, extract pothole contours by edge detection and then detect and classify potholes with YOLO, reporting an AUC of 0.9. Ahmad et al. [17] classify photographs of the road surface into small pothole, large pothole and normal pavement with pretrained CNNs and report 98% accuracy for pothole presence. Kaisar et al. [18] benchmark pretrained CNN classifiers and a custom lightweight CNN together with YOLOv8 detectors on merged Kaggle and Roboflow collections split at the image level, and report classification accuracy above 98% with real-time detection. All three work with individual images and evaluate classification and detection as separate tasks; the sequence-level pipeline in this paper starts from the same components and adds the temporal dimension that a moving vehicle provides.

2.2 Temporal and video-based road surface analysis

Video action recognition has long asked how frame evidence should be combined. Temporal segment networks showed that averaging predictions over a few sparsely sampled frames is already a strong baseline [29]. In road-surface work, consecutive frames have mostly been used for stereo or depth reconstruction [30, 31], not for the pothole decision itself, and when a recurrent component is used its contribution is rarely isolated by ablation. Our ablation is built around that gap. The same aggregation rules, a single frame, the mean over frames, and an LSTM [32] with attention pooling [33], are compared on identical features, so the contribution of the recurrent head is measured rather than assumed.

Table 1 Performance of different CNN and YOLO variants under various weather conditions, as reported by Jakubec et al. [21].

Model	Weather	Precision	Recall	mAP@[0.5:0.95]
R-CNN	Clear	64.2%	45.5%	26.9%
	Rain	49.4%	42.4%	12.7%
	Sunset	51.2%	45.9%	15.8%
	Evening	47.5%	41.6%	10.4%
	Night	31.6%	24.3%	8.1%
Fast R-CNN	Clear	67.9%	48.4%	29.2%
	Rain	51.3%	44.0%	14.8%
	Sunset	53.2%	46.7%	17.5%
	Evening	48.8%	43.3%	12.8%
	Night	32.7%	31.6%	9.2%
YOLOv5m	Clear	84.0%	81.2%	35.7%
	Rain	66.3%	56.1%	21.5%
	Sunset	75.0%	53.6%	21.0%
	Evening	80.2%	52.2%	19.7%
	Night	38.9%	17.0%	6.7%
YOLOv5l	Clear	86.1%	85.4%	36.6%
	Rain	67.9%	57.5%	22.0%
	Sunset	76.9%	54.9%	21.5%
	Evening	82.2%	53.5%	20.2%
	Night	39.9%	17.4%	6.9%

3 Data

3.1 Acquisition and annotation

Video was recorded with a VIOFO A119 Mini 2 camera mounted behind the windscreen of a passenger car on urban and peri-urban roads in daytime and dry conditions, at 1920×1080 and 30 fps. Bounding boxes were drawn by one annotator with a purpose-built browser tool on the full-resolution frames: for each positive sequence, boxes were placed on the first and last frame in which the pothole is identifiable and linearly interpolated for the intermediate frames. Licence plates and faces are blurred in the released frames.

3.2 Sequence construction

The unit of annotation and of decision is a 16-frame sequence: 0.53 s at 30 fps. Sequences do not overlap.

3.3 Region of interest and OSD removal

At the distances that matter for pothole recognition, only the lower part of the frame shows road surface. We therefore crop a fixed region of interest (ROI) that spans the full frame width and the vertical range 0.66–0.945 of the original height, and resize it to 768×192. The 768×192 ROI produces a 48×12 patch grid (576 patches), matching

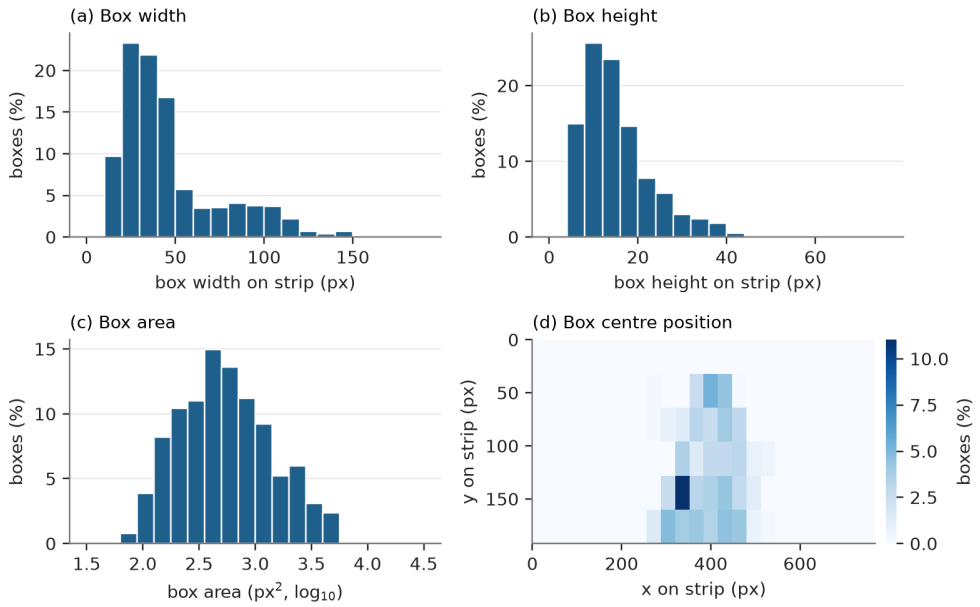


Fig. 1 Box statistics on the 768×192 strip: (a) width, (b) height, (c) area and (d) centre position of the annotated boxes, as percentages of all boxes. The median box is 38×14 pixels, which places the task in the small-object regime.

the number of patches of a conventional 384×384 ViT input while preserving the aspect ratio of the road strip. The camera burns an on-screen display (OSD) into the bottom 3.4% of every frame; the ROI ends above the OSD band, and the band itself is removed from all released frames by cutting the rows below $0.96H$. The same ROI is used for every drive; nothing was tuned per drive.

3.4 Annotation

Frame-level boxes were annotated in the positive sequences on the full-resolution frames and mapped to the strip. Figure 1 summarises their size and position on the strip: the median box is 38×14 pixels (10th–90th percentile 20–93 by 7–26 pixels), and most boxes lie in the lower half of the strip, close to the vehicle.

4 Method

4.1 Problem formulation

A sequence is $X = (x_1, \ldots, x_T)$ with $T = 16$ ROI frames $x_t \in \mathbb{R}^{192\times768\times3}$. The task is a binary decision for the sequence, $\hat{y} \in \{0, 1\}$ ("this road segment contains a pothole"), with a confidence $s \in [0, 1]$ and, when available, the position of the pothole in the frame. We decide per sequence because a driving system acts on the road segment ahead, not on a single frame, and because a decision made from one frame is easily

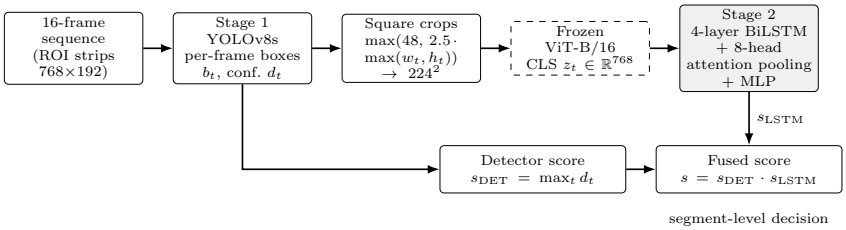


Fig. 2 Two-stage sequence-level pipeline. Stage 1 proposes the pothole region in each of the 16 frames; stage 2 encodes the proposed region with a frozen ViT-B/16 and verifies the sequence with a bidirectional LSTM and attention pooling. Detector confidence and temporal verification are multiplied into one segment-level score.

triggered by a patch or a shadow. All models below, including the single-frame ones, are therefore compared on the same unit: a sequence score obtained by aggregating frame-level outputs. Figure 2 gives an overview.

4.2 Background: Vision Transformer and LSTM

Vision Transformer.

ViT-B/16 [34] partitions an input image into 16×16 non-overlapping patches, projects them into token embeddings, adds positional information and processes the token sequence with stacked self-attention blocks. Each block computes

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where Q, K and V are linear projections of the input embeddings, followed by a feed-forward network with residual connections and layer normalisation. A learnable class token is prepended to the patch sequence; in this study its final representation is used as the frame-level feature vector.

Long Short-Term Memory.

Long Short Term Memory (LSTM) is a special type of Recurrent Neural Network (RNN) developed by [32]. It was designed to address the "vanishing gradient" problem that standard RNNs encounter when learning long-term dependencies. The fundamental building block of LSTM is an LSTM cell, which incorporates three gating mechanisms:

- Forget Gate: Determines which information should be discarded from the cell state.
- Input Gate: Controls which new information should be added to the cell state.
- Output Gate: Determines which portions of the cell state should be produced as output.

LSTMs offer significant advantages when working with time series data. For instance, through their cell state, they can preserve information over extended periods and establish relationships between historical and incoming data. They can evaluate

each element in a sequence within the context of preceding elements. The network learns which information should be retained and discarded through the forget and input gates. Additionally, the cell state and gating mechanisms mitigate the issues of vanishing or exploding gradients. While pothole detection is typically performed on static images in real-world applications (e.g., video streams from vehicle cameras), pothole detection assumes a temporal dimension. In this context, LSTMs provide the following advantages:

- Modeling Relationships Between Sequential Frames: A pothole may appear across multiple video frames. LSTM can correlate these sequential appearances to produce more reliable detections.
- Noise Filtering: It can filter out transient false positives. For example, a shadow resembling a pothole in one frame would be eliminated upon examination of consecutive frames.
- Utilizing Motion Information: it can model temporal changes in the pothole's apparent size, position and visual appearance as the vehicle approaches.
- Enhancing Prediction Consistency: by learning dependencies across successive frame representations, the LSTM can reduce sensitivity to transient frame-level fluctuations.

4.3 Stage 1: lightweight detector

Stage 1 is a YOLOv8s detector [35] with 11.2 M parameters, initialised from COCO weights and fine-tuned on the 768×192 ROI strips. The training images are the annotated positive frames, one frame from every negative sequence as background, and all 16 frames of a small number of hard-negative sequences: sequences that were flagged as pothole candidates during collection but reclassified as pothole-free at annotation. No model or confidence threshold is involved in this selection, and these sequences are used only when their drive lies in the training partition of a fold. During training, ROI strips are letterboxed into a 768×768 input canvas. We use mosaic augmentation (switched off for the last 8 epochs), horizontal flip with $p = 0.5$, scale ± 0.4 and translation ± 0.1, and no rotation. Training runs for 40 epochs with batch size 32, the library's automatic optimiser and learning-rate schedule, and early stopping with patience 15 on validation mAP@0.5. During inference, rectangular preprocessing is used: the 768×192 strip is fed at its native size, which is a multiple of the network stride (32), so no padding is applied; non-maximum suppression uses IoU 0.6. Rectangular batching was not used because, in the employed implementation, it disabled shuffling and resulted in unstable optimisation: validation mAP@0.5 remained at 0.00 throughout training.

On its own the detector gives the sequence score $s_{\mathrm{DET}} = \max_t d_t$, where d_t denotes the confidence of the highest-scoring detection in frame t and is zero if the detector returns no box.

4.4 Stage 2: crop-level temporal verification

For each frame we take the highest-confidence box b_t. If a frame has no detection, the nearest detection in time within the sequence is propagated. If a sequence contains no

detection in any frame, a fixed region at the bottom centre of the strip (100×64 pixels) is cropped at all 16 positions, so that every sequence yields 16 crops. A square crop centred on b_t with side $\max\big(48,\ 2.5 \cdot \max(w_t, h_t)\big)$ pixels keeps the pothole together with some road context. The crop is resized to 224×224 and encoded by a frozen ViT-B/16 [34] (ImageNet-21k pre-training, 86 M parameters, `timm` implementation [36]) into its class-token embedding $z_t \in \mathbb{R}^{768}$. With the backbone frozen, stage 2 has a few hundred trainable parameters in the linear variant and 8.0 M in the recurrent one, and the ablation below compares aggregation rules rather than feature extractors. As a variant we also unfreeze the last two transformer blocks and the final normalisation and fine-tune them together with the head, after the head has converged on frozen features (Section 6.3).

Three heads are trained on $Z = (z_1, \ldots, z_{16})$ with the sequence label.

- **VIT-1 (single crop).** A linear layer $f(z_t) = w^\top z_t + b$ scores every crop and the sequence score is $s_{\mathrm{VIT1}} = \sigma(\max_t f(z_t))$. There are no temporal parameters.
- **ViT-LSTM (learned temporal aggregation).** A four-layer bidirectional LSTM with hidden size 256 and dropout 0.3 processes Z. An 8-head attention layer with a learned query pools the hidden states $H = (h_1, \ldots, h_{16})$ into one vector, $p = \mathrm{MHA}(q, H, H)$, followed by a two-layer MLP. The gates follow the standard formulation,

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, z_t] + b_i), & f_t &= \sigma(W_f[h_{t-1}, z_t] + b_f), \\ o_t &= \sigma(W_o[h_{t-1}, z_t] + b_o), & \tilde{c}_t &= \tanh(W_c[h_{t-1}, z_t] + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, & h_t &= o_t \odot \tanh(c_t), \end{aligned} \tag{1}$$

  applied in both directions and stacked over four layers.
- **DET × LSTM (fused decision).** $s = s_{\mathrm{DET}} \cdot s_{\mathrm{LSTM}}$: a sequence has to be supported by the detector and by the temporal verifier.

4.5 Whole-strip baselines

To separate the effect of looking at the pothole (the stage-2 crops) from the effect of looking at more frames, the same aggregation rules are also trained on class-token embeddings of the whole 768×192 strip (ViT-B/16 with 576 patches, position embeddings resampled to the 48×12 grid): **A**, a single frame (maximum over frames of a linear score); **B**, mean pooling of the 16 embeddings followed by a linear layer; and **C**, the ViT-LSTM head above. None of A, B and C sees box information. As a CNN reference we also train an EfficientNet-B0 [37] end to end on the whole strip with max pooling over frames (**D**).

All heads minimise binary cross-entropy with a positive-class weight equal to the negative-to-positive ratio of the training set. Heads are trained with AdamW (weight decay 0.01), a cosine schedule, batch size 64 and gradient clipping at 1.0, for 60 epochs (linear heads) or 40 epochs (ViT-LSTM). The operating threshold on s is the value that maximises F1 on the validation split.

5 Experimental Setup

5.1 Split method

Five folds are formed at the drive level: the drives are partitioned into five groups, and in each fold one group is held out in its entirety as the test set, so that no drive contributes sequences to more than one fold. Within the remaining drives, sequences are divided at random, stratified by class, into a training set and a validation set; consecutive sequences of the same drive are not grouped in this inner split, so the validation set shares road surface, illumination and camera geometry with the training set and, in some cases, adjacent views of the same pothole. The validation results reported in Section 6 should therefore be read as a known-route setting, in which the vehicle repeatedly travels roads seen during training. Such a setting is not without practical counterpart: shuttle, service and logistics vehicles repeatedly operate on fixed routes, and a model deployed on such a route would likewise encounter road surfaces seen during training. Even there, however, newly formed potholes must be recognised without having been seen before, and the present evaluation does not measure that ability. Across the five folds, the training set contains 3,085–3,952 sequences, the validation set 544–697 sequences and the held-out test set 313–1,333 sequences. Checkpoints and thresholds are selected on the validation set. All ViT-based heads are run with three seeds per fold; the detector and the EfficientNet-B0 baseline are run with one seed per fold. Results are given as mean $\pm$ standard deviation over folds and, where applicable, seeds.

5.2 Metrics

The primary metric is sequence-level PR-AUC (average precision). We also report ROC-AUC, F1 at the validation-selected threshold, and recall at precision ≥ 0.9 (R@P90), the operating point that matters for a driving system, because in a downstream planning system each false-positive detection could lead to an unnecessary speed or trajectory adjustment. PR-AUC and ROC-AUC summarise ranking performance across all thresholds. R@P90 is the maximum recall achievable while precision is at least 0.90; F1 is computed at the threshold selected on the validation set. Detector quality is additionally reported at frame level as mAP@0.5.

5.3 Implementation and compute

Frozen ViT-B features are extracted once and cached in fp16, after which a head trains in minutes on one GPU. The detector trains in about 22 min per fold on an NVIDIA A100 (40 GB). Extracting and encoding the stage-2 crops for all sequences takes about 10 min per fold. Split definitions and box annotations are publicly available with the dataset. Table 2 reports the inference cost of the trained pipeline on the same GPU. End-to-end, one 16-frame sequence is processed in 34 ms, excluding crop extraction on the CPU, against 9 ms for the end-to-end EfficientNet-B0 baseline on the same 16 frames; the 0.53 s observation window of the bidirectional verifier, not computation, therefore bounds the decision latency (Section 7).

Table 2 Inference cost on one NVIDIA A100 (40 GB), batch size 1, mean ± SD over 100 runs after 20 warm-up runs. Detector and EfficientNet-B0 in fp32 as used in the experiments; ViT in fp16. GFLOPs count a multiply–add as two operations. Crop extraction and resizing on the CPU are not included.

Component	Time (ms)	GFLOPs	Parameters
YOLOv8s, one 768×192 frame	8.9 ± 0.4	10.3	11.1 M
YOLOv8s, 16 frames batched	15.9 ± 1.0		
ViT-B/16, one 224×224 crop	13.4 ± 0.3	33.7	85.8 M (frozen)
ViT-B/16, 16 crops batched	14.2 ± 0.5		
ViT-LSTM head, one sequence	3.2 ± 0.03	0.02	8.0 M
End-to-end, one 16-frame sequence	34.1 ± 1.3		105 M total, 19.2 M trained
EfficientNet-B0 (baseline D), one 768×192 frame	8.9 ± 0.2	2.3	4.0 M
EfficientNet-B0 (baseline D), 16 frames batched	9.2 ± 0.4		

Table 3 Stage-1 detector (YOLOv8s, 768×192 strips) per fold on the test set. Sequence score $= \max_t d_t$.

Fold	mAP@0.5 (frame)	Seq. PR-AUC	Seq. R@P90
1	0.92	0.61	0.11
2	0.80	0.94	0.83
3	0.61	0.65	0.00
4	0.82	0.72	0.17
5	0.94	0.85	0.83
Mean	0.82 ± 0.12	0.75 ± 0.13	0.39 ± 0.37

6 Results

Unless stated otherwise, numbers are sequence-level and computed on the test set of each fold (Section 5.1), averaged over five folds (mean ± SD) and, for the heads, over three seeds.

6.1 Detector

Table 3 reports stage 1 on its own. The detector localises potholes in the 768×192 strip with a frame-level mAP@0.5 of 0.82, and its max-confidence sequence score reaches a PR-AUC of 0.75.

6.2 Whole-strip ablation

Table 4 compares the aggregation heads on whole-strip embeddings with all negatives kept. Averaging the 16 frame embeddings (B) improves on the maximum over single-frame scores (A) by 0.04, and learned temporal aggregation (C) adds a further 0.02 with a larger variance. For whole-strip inputs these gains are small compared with the fold-to-fold spread. Undersampling the negatives to 2:1 lowers PR-AUC to 0.44, 0.49 and 0.59 for A, B and C, so all negatives are kept in every other experiment. The

Table 4 Whole-strip heads on frozen ViT-B/16 embeddings of the 768×192 strip (no box information). Sequence-level, mean ± SD over 5 folds × 3 seeds (D: 5 folds, 1 seed).

Head	Aggregation	PR-AUC	ROC-AUC	F1	R@P90
A	single frame (max)	0.65 ± 0.08	0.95 ± 0.02	0.61 ± 0.09	0.25 ± 0.16
B	mean pooling	0.69 ± 0.08	0.95 ± 0.02	0.63 ± 0.07	0.32 ± 0.17
C	ViT-LSTM	0.71 ± 0.14	0.92 ± 0.05	0.65 ± 0.24	0.42 ± 0.29
D	EfficientNet-B0 end-to-end	0.95 ± 0.04	1.00 ± 0.00	0.93 ± 0.06	0.74 ± 0.22

end-to-end EfficientNet-B0 baseline (D), trained frame by frame on the whole strip with the sequence score taken as the mean of the 16 frame probabilities, reaches a PR-AUC of 0.95 ± 0.04, an F1 of 0.93 ± 0.06 and an R@P90 of 0.74 ± 0.22 with one seed per fold, above every frozen-feature head in Table 4 and above the fused two-stage decision of Table 5; the comparison is taken up in Section 7.

6.3 Two-stage pipeline

Table 5 reports the stage-2 heads on the detector's proposals. A single-crop ViT score is weaker than the detector confidence itself (0.67 against 0.75), and mean pooling over the 16 crops recovers part of the gap (0.70). Aggregating the 16 crops with the ViT-LSTM head exceeds all three (0.81), and it exceeds the single-crop head in every fold (per-fold PR-AUC, VIT-1 / ViT-LSTM: 0.66/0.71, 0.85/0.90, 0.52/0.64, 0.65/0.84, 0.68/0.97). Fusing the detector confidence with the temporal verifier gives the highest PR-AUC among the two-stage variants in four of the five folds (0.85 ± 0.10) and the largest gain at the high-precision operating point: recall at precision ≥ 0.9 rises from 0.39 with the detector alone to 0.63 (per-fold R@P90 of the frozen-backbone fused decision, DET / fused: 0.11/0.11, 0.83/0.58, 0.00/0.77, 0.17/0.75, 0.83/0.92). In fold 3 the detector alone recalls no positive at 90% precision while the fused decision recalls 77% of them, so the verifier's main effect is to suppress confident false alarms. Partially fine-tuning the backbone (the last two of the twelve transformer blocks) does not change the outcome beyond seed variance. In a paired comparison on the same folds and seeds, fine-tuning changes PR-AUC by less than 0.01 in either direction ($+0.005 \pm 0.075$ for the ViT-LSTM alone, -0.004 ± 0.029 for the fused decision), which is within seed variance; the rounded values in Table 5 accordingly differ by at most 0.01 between the frozen and the fine-tuned rows. We therefore keep the frozen backbone in the main configuration.

7 Discussion

What the temporal model contributes, and where.

On whole-strip embeddings the recurrent head is indistinguishable from mean pooling (0.71 against 0.69). When the input is the entire road strip, most of the 768-dimensional embedding describes the scene, and a sequence model has nothing pothole-specific to integrate. Once the input is the detector's proposal, the same head improves on the single-crop score in every fold (0.81 against 0.67) and, fused

Table 5 Two-stage pipeline: sequence-level results on the test set, mean ± SD over 5 folds (heads averaged over 3 seeds). ‡Two-phase training, three seeds: the head is first trained on frozen features (40 epochs), then the last two transformer blocks, the final normalisation and the head are fine-tuned jointly (4 epochs, learning rates 2×10^{-5} / 5×10^{-5}).

Variant	PR-AUC	ROC-AUC	F1	R@P90
DET (stage 1 only, max box confidence)	0.75 ± 0.13	0.99 ± 0.01	0.78 ± 0.07	0.39 ± 0.37
VIT-1 (crop, single frame)	0.67 ± 0.11	0.97 ± 0.03	0.66 ± 0.09	0.39 ± 0.18
ViT-Mean (crop, mean pooling)	0.70 ± 0.11	0.98 ± 0.03	0.73 ± 0.08	0.29 ± 0.27
ViT-LSTM (crop, temporal)	0.81 ± 0.12	0.98 ± 0.02	0.80 ± 0.08	0.48 ± 0.30
DET × ViT-LSTM (fused; main configuration)	$\mathbf{0.85 \pm 0.10}$	0.99 ± 0.01	$\mathbf{0.85 \pm 0.07}$	$\mathbf{0.63 \pm 0.28}$
Last two ViT blocks fine-tuned‡				
ViT-LSTM (crop, temporal)	0.81 ± 0.09	0.98 ± 0.02	0.79 ± 0.07	0.58 ± 0.23
DET × ViT-LSTM (fused)	0.84 ± 0.09	0.99 ± 0.01	0.84 ± 0.07	0.65 ± 0.23

with the detector confidence, raises recall at 90% precision from 0.39 to 0.63. Fold 3 shows the mechanism: the detector alone recalls nothing at that precision, the fused decision recalls 77% of the positives. The results suggest that detector-guided crop representations benefit more from learned temporal aggregation than whole-strip representations, and that the main effect of the verifier is to reject the detector's confident mistakes rather than to find potholes the detector misses, which is consistent with the multiplicative fusion: it can lower a detector score but not raise it.

Why the single-crop ViT is weaker than the detector.

VIT-1 is trained with sequence labels on crops that, in positive sequences, sometimes do not contain the pothole because the detector proposed the wrong region, and that, in negative sequences, usually contain something the detector found plausible (when no frame has a detection, a fixed fallback region is used). A single crop therefore carries label noise that the detector confidence does not carry. Averaging the 16 crop scores recovers only part of the gap (0.70); it is the learned aggregation that exceeds the detector (0.81).

Operating point for a driving system.

In a downstream planning system, false-positive detections could lead to unnecessary speed or trajectory adjustments, whereas a missed pothole leaves the vehicle unprepared for it. At the test-selected threshold the fused system reaches an F1 of 0.85, and at precision above 0.9 it recovers 63% of the potholes. Thresholds are chosen on test sequences and should be re-estimated for each camera and mounting.

Comparison with the end-to-end CNN baseline.

The end-to-end EfficientNet-B0 baseline (D) reaches a higher PR-AUC than the fused two-stage decision on the test split (0.95 against 0.85, Tables 4 and 5) at about a quarter of the inference cost (Table 2). The two are evaluated on the same folds, the same sequences and the same negative set, with the threshold selected by the same rule on the test set; D is trained with one seed per fold. We therefore do not claim that the two-stage pipeline is the most accurate sequence classifier on this data. The

models differ in what they output and in what they can learn from. D is fine-tuned end-to-end on the whole strip and produces one score per road segment, so its score may draw on any part of the strip, including road texture and scene context that recur along a route; the two-stage pipeline scores only the region proposed by the detector and returns the position of the pothole in every frame, which a planning system needs to decide whether the pothole lies in the vehicle's path. A segment-level score alone does not tell a planner where in the lane the pothole lies or whether it can be avoided; the detector's boxes provide this at the pixel level (metric localisation would additionally require camera calibration or depth). The two-stage pipeline is therefore the configuration that yields both a verified segment decision and a per-frame location; whether the same could be obtained by pairing the end-to-end CNN score with detector boxes is a natural follow-up experiment. The comparison also shows that the frozen ViT-B/16 features are the limiting component of the pipeline: partially fine-tuning the last two blocks did not change the result, whereas D benefits from end-to-end training. Whether the accuracy advantage of D persists on routes not seen in training cannot be settled in the known-route setting of Section 5.1 and is left to a grouped evaluation.

Decision latency of the bidirectional verifier.

Because the temporal verifier is bidirectional, the current implementation requires the complete 16-frame window before producing a sequence-level decision. At 30 fps this corresponds to a temporal observation window of 0.53 s, excluding computational latency. A causal recurrent architecture, which we have not evaluated, should therefore be assessed for strictly online deployment.

8 Conclusion

We presented a two-stage pipeline for pothole recognition in road video sequences from a moving vehicle, in which a lightweight detector proposes the pothole region in each frame and a ViT-LSTM verifies the proposal over a 16-frame sequence. Under a sequence-level method with five folds and three seeds, the detector alone reaches a sequence-level PR-AUC of 0.75. Temporal verification of the detector's crops raises it to 0.81 and the fused decision to 0.85, and recall at 90% precision rises from 0.39 to 0.63. The ablation indicates that learned temporal aggregation provides a substantially larger benefit on detector-guided crop representations than on whole-strip embeddings. Three directions follow from the results: collecting data on more routes and under more conditions; estimating severity from box geometry and vehicle speed, in the spirit of roughness-index and risk-assessment work [14, 23]; and connecting the segment-level decision to the vehicle's planning stack, so that a confident pothole ahead can be translated into a speed or path adjustment.

Acknowledgements. This work was supported by TÜBİTAK under the 1002-A programme (project no. 125E491).

Declarations

Funding. TÜBİTAK 1002-A, project no. 125E491.

Conflict of interest. The authors declare no competing interests.

Data availability. The dataset, box annotations and split definitions generated and analysed during the current study are publicly available through Zenodo, https://doi.org/10.5281/zenodo.22643629.

Author contributions. Z.E.: conceptualisation, supervision, writing – review and editing. M.T.: data collection, annotation, methodology, software, experiments, writing – original draft.

References

[1] SAE International: SAE Levels of Driving Automation Refined for Clarity and International Audience (SAE J3016). https://www.sae.org/blog/sae-j3016-update. Accessed 10 September 2026 (2021)

[2] Abdel-Aty, M., Ding, S.: A matched case-control analysis of autonomous vs human-driven vehicle accidents. Nature Communications **15**(1), 4931 (2024) https://doi.org/10.1038/s41467-024-48526-4

[3] Bhatt, A.K., Raj, H., Sharma, V.B., Biswas, S., Singh, A., Silori, R., Pandey, M.: Advancements in pothole detection techniques: a comprehensive review and comparative analysis. Discover Artificial Intelligence **5**(1), 255 (2025)

[4] TUİK: Road Traffic Accident Statistics, 2023 (Corrected). https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2023-53479. Accessed: Mar. 18, 2025 (2023)

[5] Yang, C.H., Kim, J.G., Shin, S.P.: Road hazard assessment using pothole and traffic data in south Korea. Journal of Advanced Transportation **2021**, 1–10 (2021) https://doi.org/10.1155/2021/5901203

[6] Daily Mail: Councils pay £32m compensation in injuries caused by potholes over five years. Accessed 31 October 2025 (2022)

[7] Satti, S.K., Rajareddy, G.N.V., Mishra, K., Gandomi, A.H.: Potholes and traffic signs detection by classifier with vision transformers. Scientific Reports **14**(1) (2024) https://doi.org/10.1038/s41598-024-52426-4

[8] Zhong, J., Kong, D., Wei, Y., Pan, B.: YOLOv8 and point cloud fusion for enhanced road pothole detection and quantification. Scientific Reports **15**(1) (2025) https://doi.org/10.1038/s41598-025-94993-0

[9] Shang, J., Zhang, A.A., Dong, Z., Zhang, H., He, A.: Automated pavement detection and artificial intelligence pavement image data processing technology. Automation in Construction **168**, 105797 (2024) https://doi.org/10.1016/j.autcon.2024.105797

[10] Safyari, Y., Mahdianpari, M., Shiri, H.: A review of Vision-Based pothole detection methods using computer vision and machine learning. Sensors **24**(17), 5652 (2024) https://doi.org/10.3390/s24175652

[11] Zeng, J., Zhong, H.: YOLOv8-PD: an improved road damage detection algorithm based on YOLOv8n model. Scientific Reports **14**(1) (2024) https://doi.org/10.1038/s41598-024-62933-z

[12] Wang, J., Meng, R., Huang, Y., Huo, L., Zhou, L., Qiao, Z., Niu, C.: Road defect detection based on improved yolov8s model. Scientific Reports **14**, 16758 (2024) https://doi.org/10.1038/s41598-024-67953-3

[13] Ma, L., Chen, M.: Road damage detection based on improved yolo algorithm. Scientific Reports **15**, 28506 (2025) https://doi.org/10.1038/s41598-025-14461-7

[14] Guerra, K., Raymundo, C., Silvera, M., Zapata, G., Moguerza, J.M.: Pothole detection and international roughness index (IRI) calculation using ATVs for road monitoring. Scientific Reports **14**(1) (2024) https://doi.org/10.1038/s41598-024-70936-z

[15] Liu, H.X., Feng, S.: Curse of rarity for autonomous vehicles. Nature Communications **15**, 4808 (2024) https://doi.org/10.1038/s41467-024-49194-0

[16] Baek, J.-W., Chung, K.: Pothole classification model using edge detection in road image. Applied Sciences **10**(19), 6662 (2020) https://doi.org/10.3390/app10196662

[17] Ahmad, C.F., Al-Sayegh, A.T., Cheema, A., Qayyum, W., Ehtisham, R., Saghir, S., Ahmad, A.: Classification of different size of potholes based on surface area using convolutional neural network. Discover Applied Sciences **6**, 492 (2024) https://doi.org/10.1007/s42452-024-06207-3

[18] Kaisar, R., Rahman, K.N., Kabir, H.: A lightweight deep learning framework for image-based pothole classification and real-time video-based pothole detection. Scientific Reports (2026) https://doi.org/10.1038/s41598-026-64199-z

[19] Prabhakar, G., Kailath, B., Natarajan, S., Kumar, R.: Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In: 2017 IEEE Region 10 Symposium (TENSYMP), pp. 1–6 (2017). https://doi.org/10.1109/TENCONSpring.2017.8069972

[20] Khan, M., Raza, M.A., Abbas, G., Othmen, S., Yousef, A., Jumani, T.A.: Pothole

detection for autonomous vehicles using deep learning: a robust and efficient solution. Frontiers in Built Environment **9** (2024) https://doi.org/10.3389/fbuil.2023.1323792

[21] Jakubec, M., Lieskovská, E., Bučko, B., Zábovská, K.: Comparison of CNN-Based models for pothole detection in real-world adverse conditions: Overview and evaluation. Applied Sciences **13**(9), 5810 (2023) https://doi.org/10.3390/app13095810

[22] Shaghouri, A.A., Alkhatib, R., Berjaoui, S.: Real-time pothole detection using deep learning. arXiv:2107.06356 (2021)

[23] Heo, D.-H., Choi, J.-Y., Kim, S.-B., Tak, T.-O., Zhang, S.-P.: Image-Based pothole detection using multi-scale feature network and risk assessment. Electronics **12**(4), 826 (2023) https://doi.org/10.3390/electronics12040826

[24] Jobayer Al Masud, A.K.M., Sharin, S.T., Shawon, K.F.T., Zaman, Z.: Pothole detection using machine learning algorithms. In: 2021 15th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–5 (2021). https://doi.org/10.1109/ICSPCS53099.2021.9660216

[25] Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., Radha, H.: Deep learning algorithm for autonomous driving using googlenet. In: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 89–96 (2017). https://doi.org/10.1109/IVS.2017.7995703

[26] Ciberlin, J., Grbić, R., Teslić, N., Pilipović, M.: Object detection and object tracking in front of the vehicle using front view camera. In: 2019 Zooming Innovation in Consumer Technologies Conference, ZINC 2019 (2019). https://doi.org/10.1109/ZINC.2019.8769367

[27] Bianco, L.C.L., Beltrán, J., López, G.F., García, F., Al-Kaff, A.: Joint semantic segmentation of road objects and lanes using convolutional neural networks. Robotics and Autonomous Systems **133** (2020) https://doi.org/10.1016/j.robot.2020.103623

[28] Javadi, S.H., Farina, A.: Radar networks: A review of features and challenges. Information Fusion **61**, 48–55 (2020) https://doi.org/10.1016/j.inffus.2020.03.005

[29] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal Segment Networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision (ECCV), pp. 20–36 (2016). https://doi.org/10.1007/978-3-319-46484-8_2

[30] Dhiman, A., Klette, R.: Pothole detection using computer vision and learning. IEEE Transactions on Intelligent Transportation Systems **21**(8), 3536–3550 (2020) https://doi.org/10.1109/tits.2019.2931297

[31] Fan, R., Ozgunalp, U., Hosking, B., Liu, M., Pitas, I.: Pothole detection based on disparity transformation and road surface modeling. IEEE Transactions on Image Processing **29**, 897–908 (2020) https://doi.org/10.1109/tip.2019.2933750

[32] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) https://doi.org/10.1162/neco.1997.9.8.1735

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

[34] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16×16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)

[35] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics (2023)

[36] Wightman, R.: PyTorch Image Models. https://github.com/huggingface/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861

[37] Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (ICML), pp. 6105–6114 (2019)