Universiteit Leiden
The Netherlands

Master Computer Science

Patching the Patches: SAM2-Refined Labels and a DINOv2 Backbone for Robust Pavement-Crack Segmentation on noisy supervisory signals

Name:	Rajiv D.V. Jethoe
Student ID:	s3490750
Date:	27/01/2025
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Hazel R. Doughty
2nd supervisor:	Dr. Ir. Pieter J. Piscaer (TNO)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

ABSTRACT

Routine pavement maintenance increasingly relies on van-mounted sensor fleets that inspect roads at highway speed, yet the crack masks delivered by proprietary detection pipelines are often coarse, inconsistent, and empirically unreliable. At TNO, the Dutch research organisation responsible for analyzing collected sensor data on the national highway network, vendor-supplied laser-line heightmaps face problems of *inconsistent label quality* which results in unreliable training data for downstream tasks. This thesis therefore pursues two complementary goals: *(i)* upgrade those masks at scale *without* manual pixel annotation, and *(ii)* leverage strong pretrained vision features to segment thin, noisy cracks more robustly than current practice.

First, we curate a reproducible dataset of ~ 3.8k laser strips and refine their noisy masks by prompting SegmentAnything2 (SAM2) with the low-resolution originals, producing sharper supervision fully automatically. Second, we embed a self-supervised DINOv2 ViT-B/14 backbone in a lightweight TransUNet decoder that retains CNN skip connections for sub-pixel detail while exploiting global Transformer context.

A new expert benchmark, with an *Unknown* label to ignore ambiguous pixels and distance-tolerant metrics, shows that SAM2 refinement lifts in-domain F1 from 0.732 to 0.765, and that the DINOv2 hybrid attains the best out-of-domain score on the hand-labeled set (F1=0.376 vs. 0.342 baseline). The results demonstrate that promptable foundation models can bootstrap label quality and that self-supervised ViTs offer robustness when perfect ground truth is unavailable.

Acknowledgments

Completing this thesis would not have been possible without the support, guidance, patience, and kindness of many people. I am deeply grateful to all who contributed–scientifically, practically, and personally–during what proved to be an intense and often unpredictable journey.

Academic supervision. My sincere thanks go first to **Prof. Hazel R. Doughty** (LIACS, Leiden University) for outstanding supervision. Her clear feedback in our regular meetings, steady encouragement, and constructive critiques shaped the direction and quality of this work in countless ways. In close succession, I thank **Dr. Pieter J. Piscaer** (TNO) for day-to-day advising, technical sparring sessions that regularly ran long (in the best possible way), and a rare ability to help re-focus the project whenever progress stalled. Both Hazel and Pieter also provided personal understanding and flexibility when life outside the lab became difficult; I am sincerely grateful for that support.

TNO & the DOS project. I thank **TNO** for hosting my Master's thesis research and for access to the road-surface laser data that made this study possible. Within TNO's DOS project I am especially indebted to **Willem L. C. van Aalst**, whose project leadership and strategic input–often relayed through Pieter–helped keep the work aligned with real operational needs. I also wish to thank **Arthur L. van Rooijen** for hands-on help with deep learning implementation details, model evaluation, and for frequently catching my reasoning errors before they became experiments. His willingness to discuss topics well beyond the formal project scope materially improved the technical depth of the thesis.

Family. My family carried me through difficult periods during the thesis. Thank you to my mother **Kamla Jethoe-Khusial**, my father **Chandrikapersad Jethoe**, and my oldest brother **Ashwin Jethoe** for unwavering support and motivation when things were hard. A special word for my brother **Danish Jethoe**, whose serious illness just before the start of this work put many things in perspective; this thesis is finished in no small part because of the strength I draw from my family.

Friends. My heartfelt thanks to my best friend **Daniel Kurpershoek**–your belief in me never wavered–and to **Vivian Dingelhoff**, whose kindness and encouragement mattered more than you know. I am also grateful to close friends **Saleem Sarwar**, **Romano Badal**, **Melissa Halley**, **Zia Steinbach**, and **Andy Vos** for making space to unwind, for mental support, and for cheering me on when motivation dipped.

Online crew. Strangely enough, a few people I mostly know through Discord helped keep morale up during tough times even though they did not know: **PaperNick**, **OvercookedNoodle** (Amanda), **Jacob**, **jurassicplayer** and **MacQui**–thank you for the banter and company. You may never read this, but it meant a lot.

To everyone named–and to those I have inadvertently omitted–thank you. Any remaining errors are entirely my own.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Context

Roads form the backbone of modern transportation networks, serving as vital arteries for commerce, commuting, and emergency services. Ensuring their structural integrity is therefore essential for multiple reasons. First, **public safety** is directly impacted by the presence of cracks and other road defects, which can lead to vehicular damage, accidents, and increased liability for the government. Second, **cost-efficiency** in infrastructure management benefits significantly from early detection of surface damage: timely interventions prevent small cracks from developing into extensive damage, thereby reducing the need for more expensive repairs. Third, **maintenance scheduling** can be optimized through accurate condition assessment, transport authorities can plan repairs or refurbishments during off-peak hours or integrate them with other projects to minimize traffic disruptions and public inconvenience.

Neglecting systematic road maintenance has substantial **societal and economic consequences**. Deteriorated road networks may harm local economies by increasing travel times, accelerating vehicle wear-and-tear, and dissuading commerce or tourism. At a broader scale, unreliable infrastructure can undermine regional growth and strain public budgets with high repair costs. In this context, **reliable crack detection** emerges as a crucial enabling factor: by pinpointing surface defects at their earliest stages, it supports more efficient resource allocation, reduces safety hazards, and helps maintain the essential mobility that underpins social and economic well-being.

1.2 Problem Statement and Research Questions

The current method employed for road crack detection in our specific context relies on an opaque, black-box algorithm provided by an external service. This algorithm lacks transparency regarding its internal processes, feature extraction techniques, and whether it uses machine learning or traditional image processing methods. As a result, the reliability of its crack detections is often questionable. Apart from its close source nature, the segmentation mask generated using the LCMS system (see Chapter 3) are subpar. Additionally, after experimentation, to the best of our knowledge, the current asphalt crack detection field supplies unreliable models where the results are not reproducible.

To address these limitations, this thesis proposes the development of an automated, semantic segmentation–based approach specifically designed to detect visible cracks on heightmap data captured by a laser system.

Our objectives focus primarily on two key areas. First, we aim to integrate DINOv2 ViT-B/14, a cutting-edge, self-supervised Vision Transformer (ViT), into an existing segmentation architecture to leverage its powerful, attention-driven feature extraction capabilities. DINOv2 surpasses its competitors, self-supervised or not in the fields of semantic/panoptic segmentation, classification (ImageNet-1k) and even video classification even though it was not trained for it (Oquab et al., 2023). A DINOv2 ViT shows strong generalization and performance which could translate very well to the use-case of this study. Second, recognizing that ground-truth labels in our dataset are imperfect due to algorithm-generated annotations, we seek to develop training and evaluation strategies robust enough to handle incomplete or noisy annotations effectively. In Kheradmandi & Mehranfar (2022) the issue of valid data sources with noisy annotations is shown to be a ongoing issue in the field. We intend to use SAM2 developed by Ravi et al. to refine current ground truth masks for finer grained masks ultimately produced by SAM2. We use SAM2 because of its top-1 performance in image segmentation and because of its strong promptable architecture.

Specifically, this research addresses the following questions:

1. How can DINOv2 be effectively integrated into existing segmentation backbones to enhance semantic segmentation performance, specifically for detecting fine-grained structures such as cracks?
2. What training and evaluation methodologies can be implemented to improve performance when labels do not accurately represent all crack pixels?

This research represents a novel integration of DINOv2 into crack segmentation tasks, an approach that, to our knowledge, has not been explored previously (Dosovitskiy et al., 2020; Zhu et al., 2024a; Gong et al., 2024).. It builds upon DINOv2's demonstrated state-of-the-art capabilities in semantic segmentation (Oquab et al., 2023). Additionally, it directly addresses the prevalent challenge of noisy annotations common in automated crack detection datasets, an issue that conventional supervised methods often struggle to overcome (Zheng et al., 2024; Benz & Rodehorst, 2024).

By explicitly focusing on visible crack detection, this thesis intentionally excludes subsurface structural analysis, detection of filled cracks or asphalt raveling, long-term pavement performance modeling, and comprehensive assessments of environmental impacts. These aspects, while relevant to broader pavement management, lie outside the defined scope of our study. Ultimately, our goal is to improve immediate crack detection accuracy and reliability, thereby enhancing proactive road maintenance strategies.

Contribution statement.

1. **Backbone innovation**: first empirical study to couple self-supervised DINOv2 features with a UNet-style crack decoder in the pavemetn crack detection field to the best of our knowledge.
2. **Automatic mask cleaning**: novel, fully automatic DOS $\rightarrow$ SAM2 prompting workflow that potentially upgrades millions of noisy labels in a single, run once off-line step.
3. **Benchmarking protocol**: Introduction of an *internal* expert-annotated 127-image test set and group-wise distance-based disk metric fusion code, enabling fair comparison under real-world label imperfections.

Together these choices position the thesis at the intersection of scalable industrial inspection and cutting-edge self-supervised vision, addressing the key limitations of both CNN-only and Transformer-only predecessors.

1.3 Thesis Structural Overview

This thesis is organized as follows. Chapter 2 reviews prior work in pavement condition assessment, classical and deep crack-segmentation methods (CNN, Transformer, hybrid), and positions the present study. Chapter 3 details the laser-line acquisition pipeline, DOS auto-labels, patch extraction, SAM2-refined label variant, and the expert hand-labelled benchmark. Chapter 4 describes the proposed DINOv2-augmented TransUNet architecture, SAM2 prompting strategy, training regimes, and the evaluation protocol for noisy / incomplete ground truth. Chapter 5 reports quantitative and qualitative experiments across all datasets, including ablations and headline comparisons. Chapter 6 discusses findings with respect to the research questions, limitations, and avenues for future work. Chapter 7 covers immediate operational changes that can be made to improve current methods and goes over more fundamental, deeper research directions needed to be taken to improve the model and data on a more fundamental level.

2 Background & Related Work

This chapter situates the present thesis within the broader evolution of pavement–crack inspection, highlighting both *convolutional* and *Transformer* based segmentation approaches and clarifying the open problems our method addresses. Low-level tutorials on computer–vision fundamentals are deliberately omitted; the focus is squarely on state-of-the-art crack-detection research.

2.1 From Manual Inspection to Vision Automation

For much of the twentieth century pavement agencies relied on *walking surveys*: inspectors visually graded ride quality and surface distress, often using the Pavement-Condition-Index (PCI) used by Joint Departments of the Army and the Air Force, USA (1989) as part of large scale manual inspection methods. These clipboard audits are slow, labor-intensive, and, because they require lane closures, expose crews to substantial traffic risk.

The advent of van-mounted *laser-line* profilers in the early 2010s radically changed this landscape. Systems such as Pavemetrics' LCMS capture sub-millimeter texture and depth while traveling at highway speed, producing roughly 2×10^7 Greyscale–LIDAR pixels per $100\,\mathrm{m}$ strip (Pavemetrics Systems Inc., 2024a). Comprehensive reviews confirm that these optical rigs deliver centimeter-scale crack-width accuracy and centimeter-level geolocation when combined with differential GPS (Chu et al., 2022). A raveling[1] detection algorithm was developed by Aalst et al. (2015) that uses 3-D laser triangulation to detect asphalt type, raveling and determine remaining service life of porous asphalt in The Netherlands. In addition van Aalst (2021) presented the current system in use for road-surface inspection in The Netherlands developed at TNO.

Yet the "big-data" boon created a new bottleneck: annotating *pixel-perfect* ground truth for supervised learning is prohibitively expensive, evident by various studies trying to automate dataset generation (Zhang et al., 2021; Figueira & Vaz, 2022; Lu et al., 2023) . State-of-the-art crack detectors therefore rely on convolutional or Transformer backbones trained on carefully curated, often hand labeled datasets, e.g. DeepCrack by Zou et al. (2018) or CrackU-Net by Huyan et al. (2020) leaving open questions about scalability and generalization (Gong et al., 2024). Our work addresses precisely this gap: we combine the speed and coverage of laser imaging with a label-refinement pipeline that reduces human effort while preserving metric-level fidelity, ultimately enabling safer, data-driven maintenance scheduling of road inspections.

2.2 Convolutional Architectures for Crack Segmentation

Over the past decade fully–convolutional networks (CNNs) have dominated the field of pavement–distress mapping, steadily evolving from simple encoder-decoders to multi-branch generative models.

2.2.1 Early Fully–Convolutional Baselines (2015–2018)

FCN (Long et al., 2015), **SegNet** (Badrinarayanan et al., 2017) and **U-Net** (Ronneberger et al., 2015) were the first architectures to be re-trained on grayscale road imagery. Their symmetric encoder-decoder design yields dense predictions at a fraction of the time required by hand-crafted operators, yet the limited effective receptive field causes *fragmented* outputs once a crack exceeds a few hundred pixels in length, a problem already reported in comparative studies of 2017–2018 (Ragnoli et al., 2018).

[1] Raveling is the concept where in porous asphalt, over time, small stones start to wear off and detach from the asphalt layer. This causes damage to the road surface and requires timely maintanence

2.2.2 Multi–Scale and Boundary-Aware CNNs

Subsequent work tackled the context deficit by fusing features at multiple resolutions. **DeepCrack** introduces five side-outputs that are edge-supervised and later aggregated to obtain continuous crack skeletons, improving Average Precision (AP) on Crack500 by 3-4% over plain U-Net (Zou et al., 2018). **MFPANet** grafts a pyramid attention module on top of a ResNet encoder so that global context guides the localization of fine cracks; the network outperforms DeepCrack by 8% F1 on the DeepCrack dataset (Jiang et al., 2022). Most recently, **CT-CrackSeg** couples dilated convolutions with a lightweight convolution-transformer head and an explicit boundary branch; on the CFD benchmark it attains a 8% gain in F1 compared to DeepCrack (Tao et al., 2023).

2.2.3 Generative & Diffusion CNNs

Although discriminative CNNs excel at crack/background separation, they do not *model* the physical formation of cracks. **CrackDiff** reframes segmentation as a denoising-diffusion process: a multi-task U-Net learns to predict both the crack mask and the noise that corrupted it while reverse-sampling from random Gaussian input (Zhang et al., 2024). CrackDiff achieves state-of-the-art scores on Crack500 but at the price of a much slower training and inference speed per image, significantly slower than feed-forward CNNs, mirroring the sampling overhead seen in generic DDPM models (Song et al., 2021).

2.2.4 Lightweight Encoders for Embedded Deployment

Industrial road scanners often demand on-board execution, motivating networks that trade parameters for speed. **RHA-CrackNet** compresses the encoder with depthwise separable convolutions and inserts hybrid channel-spatial attention blocks in the decoder; despite using only 3.4 M parameters it reports state-of-the-art F1 scores on the CamCrack789 dataset (Zhu et al., 2024a). Attempts to reproduce the claimed Crack500 numbers revealed a 10–20% gap, suggesting either missing training tricks or metric mis-alignment. In this thesis we therefore discarded this model fairly quick.

Key Take-aways. CNNs remain the backbone of most pavement-crack detectors thanks to their computational efficiency and mature tooling and their inductive bias towards imagery input data, yet (i) long-range connectivity, (ii) label noise robustness, and (iii) reproducibility of reported gains on more niche architectures are ongoing challenges (that we faced during the literature review) that motivate hybrid and generative alternatives explored in later sections (or less niche architectures which are more stable and have a solid foundation in the field).

2.3 Vision Transformers and Hybrid Architectures

Motivation. Pure CNNs, despite progressive tricks such as dilated kernels and pyramid attention, struggle to capture *global* context along meandering long-range cracks (§2.2). Vision Transformers (ViTs), whose self-attention operates across the *entire* token sequence, offer an attractive remedy; the challenge is to retain pixel-level precision for hair-line cracks while taming the memory footprint on megapixel road imagery.

2.3.1 CNN–ViT Hybrids

TransUnet Chen et al. fuse a ResNet stem with a ViT encoder and a UNet decoder, showing that even a *shallow* Transformer (L=12) markedly improves organ boundary continuity in 2-D CT; the skip-connections restore lost detail and keep parameters modest at 105 M (Chen et al., 2021).

SegFormer Xie et al. replace heavy UNet-style decoders with a three-layer MLP head fed by multi-resolution Transformer features, yielding a 150 FPS stream on 512^2 images while matching HRNet on ADE20K; the B5 variant is now a popular backbone in crack papers (Xie et al., 2021).

2.3.2 PURE/LOCAL VITS FOR ASPHALT

CrackFormer Liu et al. introduce windowed self-attention and deformable tokens to balance receptive field and efficiency on 544×384 laser scans, outperforming DeepCrack by 2.0% Average Precision on CrackTree260 dataset (Liu et al., 2021).

Swin U-Net By sliding Swin Transformer blocks inside a UNet ladder, Cao et al. achieve state-of-the-art dice scores on Synapse with $\sim 15\times$ fewer FLOPs than TransUNet, evidence that hierarchical, shifted-window attention scales gracefully to dense prediction (Cao et al., 2022).

2.3.3 EMERGING TREND

Recent literature converges on *hybrid* designs: a low-level CNN stem for crisp edges, a mid-level Transformer for contextual reasoning, and a lightweight decoder (e.g., MLP) to fuse multi-scale cues. Such architectures consistently report $3-8\%$ gains in F1/IoU/AP on public asphalt sets while halving parameter counts compared with plain UNets, underscoring the synergy between local texture and global attention.

Limitations in prior work. Despite their promise, published hybrids share several shortcomings:

1. **Label assumptions.** Most methods train and report on hand-curated datasets with comparatively clean, isotropic annotations; performance under *noisy*, over-dilated, or incomplete masks is rarely studied. (Liu et al., 2021; Zhu et al., 2024b; Tao et al., 2023)
2. **Modality gap.** Nearly all benchmarks use perspective RGB (or shallow texture) imagery; few address the anisotropic sampling and speckle/noise characteristics of laser-line profiles typical of highway-speed survey vehicles. (Li et al., 2017)
3. **Evaluation bias.** Reported gains often reflect train-test overlap in acquisition conditions; robustness across label qualities (original vs. refined) or across annotation protocols (auto vs. expert) is seldom quantified.
4. **Outdated ViT backbones.** All previous works thusfar use architectures that are based off the original Vision Transformer by Dosovitskiy et al. (2020). In recent years more modern architectures with self-supervised training regimes have emerged like DINOv2 ViTs by Oquab et al. (2023).

To this end we make use of a DINOv2 trained ViT to help in integrating a better backbone and introducing various techniques like label refinement and more intelligent metric calculations to build a more robust model that can deal with noisy data.

2.4 STATE-OF-THE-ART SEGMENTATION MODELS

Recent progress in large-scale pretraining and promptable vision architectures has reshaped semantic segmentation well beyond the domain-specific crack literature reviewed in §2.2§-2.3. This section briefly situates two families of models that directly inform our design choices in Chapter 4: (i) *foundation / promptable segmenters* (SAM, SAM2) and (ii) *large self-supervised ViT representations* (DINOv2) that downstream decoders can adapt to dense prediction. We also note complementary universal decoders (Mask2Former, MaskDINO) that demonstrate how rich pretraining plus lightweight task heads transfer to pixel labeling.

2.4.1 PROMPTABLE FOUNDATION SEGMENTERS

Segment Anything (SAM). Kirillov et al. introduced SAM as a class-agnostic model trained on $\sim$1B masks spanning 11M images (Kirillov et al., 2023). A powerful image encoder (ViT-H/L/B) feeds a promptable mask decoder that accepts points, boxes, or low-resolution masks and returns one or multiple high-quality segment hypotheses in $\lesssim$50 ms per prompt on a GPU. Zero-shot generalization across domains (medical, satellite, document, materials) is a key strength; however, SAM processes each frame independently and can under-segment thin, low-contrast structures unless prompt placement is carefully curated (Kirillov et al., 2023; Cheng et al., 2023).

Segment Anything 2 (SAM2). Ravi et al. extend SAM to images *and* videos with a streaming memory mechanism that propagates mask information across frames, improves small-object recall, and supports iterative refinement from mixed prompt types (Ravi et al.). Of particular interest here, SAM2's mask-prompt pathway accepts low-resolution binary masks that seed the decoder, exactly the interface required to refine noisy, coarse crack masks produced by an upstream system (our DOS labels; see Chapter 3). Ravi et al report stronger boundary fidelity than SAM on elongated/fragmented objects when mask prompts provide coarse structure.(Ravi et al.)

2.4.2 Large Self-Supervised ViT Representations

DINO & DINOv2. Self-distillation with no labels (DINO) demonstrated that ViTs trained self-supervised on unlabeled internet-scale images learn surprisingly semantically aligned patch embeddings. (Caron et al., 2021) Oquab et al. scaled this recipe to curated multi-billion-image corpora, improved training stability, and released DINOv2 backbones (ViT-S/B/L/G) whose frozen features transfer strongly to downstream dense tasks, including semantic, panoptic, and instance segmentation, often rivaling supervised pretraining (Oquab et al., 2023). DINOv2 tokens preserve fine texture and mid-range context, traits desirable for hair-line crack detection where labeled data are scarce.

Masked Autoencoding (MAE) family. Mask-token reconstruction (MAE) and derivatives learn spatially aware ViT features that benefit dense prediction when fine-tuned with lightweight decoders (He et al., 2022; Peng et al., 2022). Although not used directly in this thesis, MAE results support the broader claim that large unlabeled corpora can yield transferable pixel representations, motivating our adoption of a self-supervised encoder.

2.4.3 Universal Decoders for Dense Prediction

Mask2Former. A unified transformer decoder architecture for semantic, instance, and panoptic segmentation; queries attend to multi-scale pixel features and emit class-agnostic masks plus class scores (Cheng et al., 2022). Mask2Former's separation of representation (backbone) from lightweight mask decoding illustrates how strong pretrained features (ViT/ConvNeXt) can be reused across domains.

MaskDINO. Li et al. fuse DETR-style detection queries with dense mask prediction, achieving competitive panoptic and semantic results when coupled with self-supervised backbones (Li et al., 2023). The framework is tolerant to varying annotation granularity and can leverage weak masks, relevant to our noisy DOS supervision scenario.

2.4.4 Implications for This Thesis (preview)

Three lessons emerge:

1. **Promptable refinement scales noisy labels.** SAM/SAM2 demonstrate that coarse masks can bootstrap higher-quality segmentation without dense manual editing. We exploit this by feeding our auto-generated DOS crack masks as low-resolution prompts to SAM2 to produce refined training targets at scale (Chapter 4).
2. **Frozen self-supervised ViTs transfer.** DINOv2 encoders provide rich spatial embeddings even without task-specific supervision; we integrate a DINOv2 ViT-B/14 into a TransUNet-style architecture to compensate for limited, noisy crack labels (§4.5).
3. **Light decoders suffice with strong features.** Inspired by Mask2Former/MaskDINO, we retain a relatively lightweight decoder head; modeling effort is invested in robust feature reuse and label cleaning rather than depth/width scaling of the head.

These observations directly motivate the design decisions detailed next in §2.5, where we explain how SAM2-refined labels and a DINOv2-powered TransUNet variant are combined for crack segmentation on laser-line road imagery.

2.5 Design Choices Specific to This Thesis

We deliberately departed from the backbones and label–refinement strategies explored in prior pavement-crack literature (§2.2 - §2.3) and instead combined two recent *foundation* models whose capabilities had, at the time of writing, seen little or no evaluation on road-surface imagery. The section below summarizes the motivation and empirical evidence that guided these decisions.

2.5.1 DINOv2 ViT-B/14 as TransUNet Encoder

From random to self-supervised weights. Early experiments with **TransUNet** (Chen et al., 2021) showed that the original ImageNet-1k pre-training left the ViT encoder ill-adapted to the grey, low-contrast textures of laser road imagery. Instead of fine-tuning from scratch we adopted **DINOv2** (Oquab et al., 2023), a self-supervised ViT that learns general-purpose dense features from ~ 142 M images without any manual labels. DINOv2 exhibits state-of-the-art zero-shot transfer on semantic and panoptic segmentation benchmarks such as ADE20k and Cityscapes (Oquab et al., 2023) and therefore promised stronger inductive bias for thin crack patterns than purely supervised alternatives.

2.5.2 Label Refinement with Segment Anything 2

Why SAM2? The **Segment Anything 2 (SAM2)** model extends the original promptable SAM architecture with stronger mask quality and markedly better generalization to out-of-distribution textures (Ravi et al.). Its *low-resolution mask* prompt is a perfect match for our setup: each 256×256 auto-generated DOS mask can be fed directly as a seed without down-sampling artifacts. Early benchmarks showed SAM2 improving mean IoU by 7-10% over none-SAM2 improved labels. suggesting that the model could plausibly "repair" the omissions and over-dilation described in §3.4.

Prompt selection. Among the five prompting modes offered by SAM2: *bounding box*, *points*, *mask*, *box+points* and *text*, we opted for **mask prompting**:

1. Single-point prompts require a heuristic to find a crack seed pixel midst heavy asphalt noise and therefore do not scale to millions of patches.
2. Bounding boxes suffer the same localization dilemma which made it easy to discard this notion as well.
3. Text prompts fit an application where distinct well recognized objects are being segmented, which is far from the case for this study, immediately disqualifying this technique.
4. Mask prompts leverage the *existing* DOS raster as prior knowledge; SAM2 then refines boundaries and hallucinates missing hair-line branches, producing markedly crisper ground truth.

The fully automated pipeline (Fig. 4.3) therefore proceeds as

$$\text{DOS mask}_{256^2} \xrightarrow[\text{low-res mask}]{\text{prompt}} \text{SAM2} \longrightarrow \text{refined mask}_{256^2},$$

yielding supervision that is both *denser* and *cleaner* than the original black-box output yet incurs no manual labor.

2.6 Positioning of This Thesis

Existing pavement–crack detectors fall into two partially complementary camps. Pure CNN pipelines (e.g. DeepCrack, MFPANet, CT-CrackSeg) excel at tracing fine textures thanks to hierarchical local filters, yet their limited receptive field causes fragmented predictions once a crack traverses more than a few hundred pixels and they remain sensitive to illumination artifacts and shadows that mimic edges. Vision-Transformer (ViT) variants such as TransUNet, CrackFormer and SegFormer-B5 remedy the long-range issue with global self-attention, but at the price of considerable data hunger and a marked performance drop when the training masks are coarse or over-dilated. Almost all recent benchmarks therefore rely on small, hand-curated datasets whose pixel labels are painstakingly cleaned, an assumption that does not hold for industrial road-survey pipelines where ground truth is produced by opaque black-box heuristics (in the case of TNO specifically, but the Pavemtrics system is widely used according to them).

This thesis targets precisely that neglected corner case: large-scale but *noisy* training corpora. First, we adopt a self-supervised **DINOv2 ViT-B/14** encoder whose rich pre-training on 142 million images empowers it with strong zero-shot segmentation skills and robustness to label noise. The ViT is embedded in a lightweight TransUNet-style decoder so that global context from attention is fused with crisp boundary cues from the CNN stem. Second, we introduce a **SAM2-based label-refinement** stage: each noisy DOS mask is fed to SAM2 as a low-resolution prompt, letting the foundation model extrapolate cleaner crack contours without any human clicks, an approach made possible by SAM2's promptable design and its state-of-the-art zero-shot performance. Finally, a custom evaluation pipeline aggregates sub-patch predictions back to their mother image and ignores pixels the expert flagged as "unknown", thus reporting metrics that are both tolerant to annotation uncertainty and free from train-test leakage, the former being specifically relevant when we see small curated benchmark datasets which do not represent the real world of road surveying.

2.6.1 Summary of Architecture and Data Pipeline

1. **Backbone:** TransUNets ViT encoder replaced by DINOv2 ViT-B/14 for its proven zero-shot performance on dense prediction tasks and for reducing domain-gap between natural and laser-scan imagery.
2. **Label cleaning:** SAM2 with mask prompting converts noisy DOS annotations into high-quality pseudo-labels at virtually no additional cost.
3. **Synergy:** A self-supervised ViT backbone benefits from the richer supervision produced by SAM2, while the latter, in turn, relies on the coarse yet readily available DOS masks; no human in the loop.

Together these design choices form a training recipe that is, *to our knowledge, the first to couple DINOv2 features with SAM2-refined labels for pavement crack segmentation*.

3 Data

In this chapter the data used for this study is introduced. We go over how the data is acquired, how we process it for downstream tasks and how we try to cope with noisy, inaccurate labeling done by a black box algorithm. Furthermore we introduce a hand labeled test set we use as a proper validation of our models to gauge true effectiveness. Lastly, for evaluation of this hand labeled test set a custom evaluation process was needed compared to the auto-generated test sets. This method will also be described in this chapter.

IMPORTANT: *Create a Data acquisition/processing pipeline as asked for in 1*

3.1 Existing Dataset Discussion

Public research on pavement–distress detection has produced several *camera–based* crack–segmentation corpora[1]. All of them were acquired with **area RGB cameras** a few centimeters above the surface and therefore differ fundamentally from our *laser-line* profilometer supplied by Pavemetrics, a Canadian company (§3.2).

Modality and geometry mismatch. Camera images exhibit perspective distortion and a ground–sampling distance (GSD) between $0.2\text{mm}\,\text{px}^{-1}$ and $1.7\text{mm}\,\text{px}^{-1}$. Our Pavemetrics sensor, in contrast, delivers an *orthorectified* strip at $5\text{mm}\,\text{px}^{-1}$ (drive direction) $\times$ $1\text{mm}\,\text{px}^{-1}$ (cross-lane) with depth encoded in a separate channel. Attempting to train on perspective RGB and test on laser data would induce a severe domain shift.

Pre-processing pipeline. Open sets contain only minimal photo corrections, whereas the Pavemetrics chain applies various algorithms to clean up the images. Afterwards our one pipelines further cleans up the images and generates the masks resulting in JPEG/PNG strips. These proprietary transforms further alter the appearance statistics and invalidate naïve transfer learning.

No open laser-strip benchmark. A literature and web search (keywords "*laser-line pavement dataset*", "*3-D road profilometer*") revealed *no* pixel-annotated laser-strip repositories. The few 3-D road–surface sets that exist are either from completely different domains (autonomous driving (Zhao et al., 2024)) or they are in proprietary commercial databases.

Consequence for this thesis. Because (i) no comparable laser-strip dataset exists, (ii) camera benchmarks differ in viewpoint, resolution and pre-processing, and (iii) extensive training volume is mandatory, we **deliberately rely on the noisy but abundant DOS masks** produced and refine them with SAM2 (§4.4.2). Our expert hand-labeled benchmark (§3.6) then provides an unbiased evaluation of both the refined labels and the proposed model.

3.2 Data acquisition

The imagery analyses in this thesis is captured by a van-mounted *laser-line* system operated by Pavemetrics called LCMS (Pavemetrics Systems Inc., 2024a). For every 100 meter road segment the on-board computer outputs one rectified strip image $\mathbf{I} \in \mathbb{R}^{19995\times 4160}$ an additional dimension is potentially available if you make use of the intensity, but this was not used in this study; the physical pixel pitch therefore equals 1mm longitudinally and 5mm laterally. Raw sensor signals (pose, roll/pitch, illumination) are stabilized by TNOs pipeline, discussed in §3.3; the resulting JPEG (images), PNG (masks) strips are treated as the "raw" input for all experiments in this work (after patchifying the segments, see §3.5).

[1] See, e.g., *Crack500* (Yang et al., 2019), *CrackTree* (Zou et al., 2012), *DeepCrack537* (Zou et al., 2018), *CamCrack789* (Zhu et al., 2024a).

3.3 DOS SOFTWARE PIPELINE

The strip images used throughout this thesis are *not* raw sensor dumps; they are produced by the data processing software from laser-line scans collected at highway speed (§3.2). The Pavemetrics-class laser crack measurement system (LCMS) acquires dense transverse surface profiles (depth + co-registered intensity/texture) which the software assembles into rectified lane-width strips suitable for downstream distress analytics.(Pavemetrics Systems Inc., 2024b; Li et al., 2017)

Before export, several conditioning steps are applied to stabilize the road surface and suppress artifacts induced by vehicle motion and pose drift. A low-frequency longitudinal trend (meter-scale undulations) is removed to "flatten" the surface; depth outliers are clipped to a narrow elevation band (on the order of $\pm$1cm around the estimated pavement plane) to reduce spikes from debris and specular returns. (Li et al., 2017)

After stabilization of the images, the conditioned depth/texture channels feed a proprietary distress classifier that outputs geometric crack descriptors (polyline centers with local width estimates). These vectors are rasterized by the DOS pipeline into coarse binary masks, the *DOS* labels introduced in §3.4, which serve as the starting point for all subsequent label-refinement and learning experiments in this thesis. Following this some filtering operations are done with a custom build CNN in-house to determine actual cracks based on this data.

3.4 ANNOTATION FORMAT

For each strip we receive a JSON file that stores crack center-lines as piece-wise poly-lines with a local brush width:

```
[
  {"x":[1466.26,1477.00, ...],
   "y":[15207.70,15214.15, ...],
   "width":[19.1,22.3, ...],
   "crack_class":1}
]
```

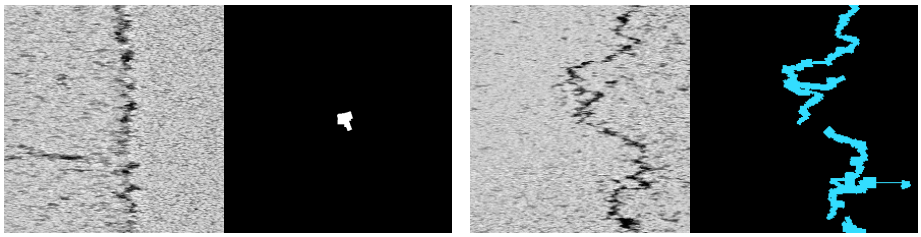
- (x_i, y_i) – sub-pixel coordinates in the (row, col) plane of the 100 meter image.
- `width` – local crack width in pixels (2–25 px observed).
- `crack_class`=1 – longitudinal / transverse crack (other distress classes are ignored in this study)

Rasterization. Successive vertices are connected by straight segments; each segment is drawn where the width is defined by the "width" in the JSON file, yielding a binary mask $G \in \{0,1\}^{H \times W}$, which from now will be referred to as a *DOS mask*, which were labeled algorithmically, since it is the end of the DOS software pipeline. This *DOS mask* covers the physical crack. But in practice it can be observed that the mask *over-covers* partial crack segments *sometimes*, but also does not succeed in fully covering cracks in a lot of occurrences (Fig. 3.1a-b). Fig. 3.1c-d on the other hand shows an example of the *DOS masks* covering the original crack representatively. These masks are used as:

1. baseline supervision,
2. prompts for later label-refinement experiments (see Chapter 4), and
3. the *DOS-orig* evaluation split in §5.

Limitations. The DOS masks suffer from two systematic issues:

1. *Over-segmentation*: the brush radius inflates true crack width, penalizing thin, precise predictions; and
2. *Omissions*: hair-line cracks and complex junctions are often not annotated at all.



(a) Raw (weak) (b) Mask (weak) (c) Raw (good) (d) Mask (good)

Figure 3.1: Two example image–mask pairs illustrating the varying quality of algorithmically generated DOS labels.

3. Since this data acquisition pipeline is a product there is no insight into its workings. How do these labels get generated? We do not know, so we cannot spot shortcomings or strengths of this method.

The cause of the *over-segmentation* is hard to determine since we do not have access to the pavemetrics system. But most likely they make use of classical computer vision methods like; edge-detection, thresholding etc. And apparently their system *over-estimates* mask size for some examples. Because of this classical approach, the finer details are probably less likely to be discovered because they most likely perform some *morphological opening* operations to reduce noise. This in turn will remove finer-grained detail.

Figure 3.1 shows two examples of a 512×512 pixels crop from the larger 100 meter strip. This crop is centered around a potential crack. These images showcase an example of where the proprietary system potentially falls short of expectations and one that shows good crack coverage. The left pair shows weak masks because usually in the *DOS masks* dark pixels represent road surface damage. The dark pixels here are shaped in the form of a crack in the driving direction, which is common. But a lot of the crack is not represented with the DOS mask. While the converse can be seen in the right pair.

These shortcomings motivate the label-refinement strategy introduced later in Chapter 4; however, the present chapter restricts itself to describing the *original* data delivered by the acquisition pipeline.

3.5 Patch extraction & filtering

Rationale. Each 100 meter strip (§3.2) contains $\approx 2 \times 10^7$ pixels, far exceeding GPU memory limits and the input resolution expected by the network and by SAM2. We therefore slice every strip into *non-overlapping* square tiles while enforcing three additional constraints:

1. exactly one tile per annotated crack center-line,
2. no duplicate tiles (high IoU overlap) across a strip, and
3. deterministic train/val/test assignment to avoid spatial leakage.

Algorithm. Algorithm 1 sketches the Python/OpenCV implementation

Only tiles with at least 30% crack foreground survive; the remainder are discarded to keep the class balance manageable during training.

Outcome. The procedure yields 3079 training tiles, 387 validation tiles, and 387 test tiles. Because the de-duplication and split assignment happen *before* any sub-sampling or SAM2 refinement, no pixel from a given physical location can appear in more than one split, ensuring strict evaluation integrity.

Algorithm 1 IoU–filtered patch extractor (sketch)

Require: RGB strip I, DOS mask M, JSON cracks $\mathcal{C}$
$\mathcal{P} \leftarrow \varnothing$ {accepted patch bboxes}
for all $c \in \mathcal{C}$ **do**
  $(x_{\text{mid}}, y_{\text{mid}}) \leftarrow \text{midpoint}(c)$
  bbox $\leftarrow 512{\times}512$ window centred at $(x_{\text{mid}}, y_{\text{mid}})$
  bbox $\leftarrow$ bbox $\oplus$ 50 px margin
  if $\exists\, p \in \mathcal{P} : \text{IoU}(\text{bbox}, p) \geq 0.5$ **then**
    continue{skip duplicate}
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\text{bbox}\}$
  Save (I, M) crop; down-sample according to Fig. **??**
Hash base-filenames into {`train`,`val`,`test`} splits =0

3.6 Expert Hand-Labeled Benchmark

Motivation and Annotation Protocol The fully–automatic DOS masks inherit the same limitations as the on-van detection pipeline that generated them. Relying on such labels for *both* training *and* testing would therefore prevent any fair assessment of the proposed method's true capacity to surpass the baseline. To break this circular dependency we commissioned an **expert-annotated test set**. Each pixel is assigned to one of three semantic classes

Table 3.1: Semantic classes defined in the hand-labelled benchmark.

Colour	Definition
Green	**Unknown**: visually ambiguous regions that are omitted from metric computation
Blue	**Crack**: pixels that the expert confidently assigns to a pavement crack
Black	**Background**: all remaining pixels (implicit class)

Pixels tagged *Unknown* are ignored during loss/metric computation via the `ignore_index=255` mechanism discussed in §4.6. The annotator worked with a broad 10–20 px brush: fine-grained tracing proved prohibitively time-consuming. While this introduces mild over-dilation, it is still vastly cleaner than the legacy DOS masks and, crucially, *human-verified*.

3.6.1 From 100 meter Strips to "Mother Patches"

For the expert study we re-sampled the original 100 m laser strips into *rectangular* tiles of size 256×1280 px ($H \times W$). The width (1280 px) gives annotators five times more *horizontal* context than the square patches used for training, which was necessary to properly judge road damage the expert found.

The process outlined in this subsection was designed so that;

1. The expert labeler had more visual context for labeling in the lane-width dimension.
2. The model could handle multiple square patches ($5 \times 224 \times 224$) pixels, that belonged to a bigger *mother patch* during test time, when the model would perform inference on the hand labeled test set.

Sensor-aspect correction. The road-scanner records one pixel per 5 mm in the drive direction and 1 mm per pixel across the lane. To display an *undistorted* view during labeling the expert interface first *vertically* repeats each row: every scan line was repeated five times, yielding a square with a $1\,\text{mm} \times 1\,mm$ per pixel aspect ratio image that matches the real-world metric. After annotation the duplicate rows were removed so that the raw data fed to our network (and all subsequent processing) retained the native $1{\times}5$ mm/px geometry, consistent with the auto-generated DOS masks described earlier.

The resulting 256×1280 "mother patches" therefore

- preserve full horizontal context for reliable manual segmentation;
- remain perfectly aligned with the underlying sensor grid after de-stretching; and
- are later subdivided into 224×224 sub-patches (§3.6.2) without any further spatial warp.

3.6.2 SUB-PATCH LABEL MATCHING

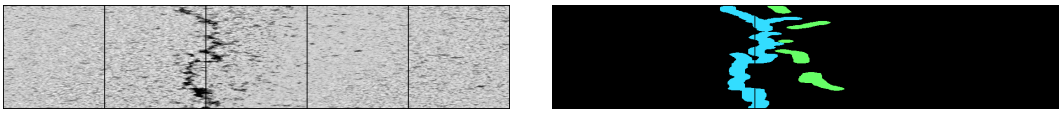
Why split? The segmentation network ingests fixed 224×224 inputs, whereas the expert benchmark consists of **mother patches** sized 256×1280 px. Each mother patch is therefore tiled into five contiguous 256×256 crops and subsequently rescaled to 224^2 using Lanczos interpolation (OPENCV `INTER_LANCZOS4`), a kernel that best preserves the hair-line crack texture compared with bilinear or bicubic (OpenCV Team (moukthika), 2025).

Dataset wrapper. To keep track of which sub-patches originate from the same mother patch we package each crop into a `.npz` archive containing the RGB image, its label mask, and a string `group_id`. This way, during inference we can keep track of which sub-patch belongs to what *mother patch*. We build up a stacked matrix of sub-patches belonging to each *mother patch* and when all patches are ran through the model we calculate the metrics for every *mother patch* as described in §4.6.

Group-wise metric fusion. During inference every tile produces a four-component confusion vector $\mathbf{c} = (\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN})$ using the distance-tolerant counting rules of §4.6. Vectors belonging to the same `group_id` are *summed before* computing precision, recall, F1 and IoU ensuring that evaluation reflects complete cracks rather than arbitrary tile borders:

$$\mathbf{C}_{\text{mother}} = \sum_{i=1}^{5} \mathbf{c}_i.$$

Visual re-stitching. After inference each 256×1280 *mother patch* is split into five non-overlapping 224×224 sub-patches for the network. For qualitative inspection the five predictions are concatenated back into a single 224×1120 mosaic. The same re-stitching is applied to the expert label so that all three share the exact pixel grid.



(a) RGB mosaic

(b) Expert mask mosaic

Figure 3.2: Re-stitched mosaics for a single hand-labeled mother patch. The top row shows the RGB context and the expert annotation (crack / unknown), Blue/Green respectively

This pipeline yields a compact yet rigorous benchmark: every prediction is scored against a *human-verified* reference while maintaining compatibility with the square-patch training interface used throughout the thesis.

3.7 SAM2–REFINED CRACK MASKS

Motivation. The legacy *DOS* masks introduced in §3.4 over–dilate true crack width and miss hair-line cracks *on a regular basis*. Rather than resorting to labor-intensive manual clean-up, we employ the **Segment Anything 2** (SAM2) foundation model as an *automatic label–refinement engine* (details in Chapter 4).

Generation pipeline. For every 512×512 patch produced by Algorithm 1 we feed the 256×256 DOS mask to SAM2 as a low-resolution *mask prompt*. SAM2 returns a crisper 224^2 prediction

whose thin boundaries more closely trace the visible crack skeleton. No extra human input is required; the entire corpus is processed offline. So, this is a one-time step that needs to be performed.

Resulting dataset. The images are *identical* to the DOS corpus, but each now has an additional *SAM2 label*. Henceforth we refer to these two supervision variants as

Table 3.2: Datasets used in this thesis: the original noisy DOS masks (*DOS-orig*) and the refined SAM2 masks (*SAM2*).

Image patches	**Label variant**
same RGB crop	*DOS-orig* (noisy, over-dilated) *SAM2* (refined, thinner)

Both label sets inherit the spatially consistent train/val/test assignment described next in §3.8; models can therefore be trained and evaluated under two noise regimes without risk of data leakage.

3.8 Train/Val/Test Splits

3.8.1 Automatic DOS/SAM2 dataset

The full DOS/SAM2 dataset comprises **3,853** crack patches after all filtering (§3.5). Patches are assigned to splits with a deterministic `MD5` hash of the parent strip ID (Algorithm 1):

The hash-based allocation guarantees *zero* strip overlap between splits, thereby preventing spatial leakage of very similar cracks.

3.8.2 Expert hand-labelled benchmark

For the hand-labeled benchmark (§3.6) we keep the same 70 %/15 %/15 % partitioning of strip IDs but manually review the **387** test patches. After removing blank or duplicate views the final test set contains **127** expertly annotated "mother patches" (Table 3.3).

Table 3.3: Dataset sizes after cleaning.

	Automatic DOS/SAM2		**Hand-labelled**	
Split	Patches	%	Mother patches	%
Train	3,079	70	–	–
Val.	387	15	–	–
Test	387	15	127,(subset)	100

The hand-labeled test subset is used *only* for final reporting; no model selection or hyper-parameter tuning touches this data, ensuring an unbiased assessment of generalization beyond auto-generated masks.

3.9 CHAPTER SUMMARY

This chapter detailed the full data pipeline that underpins the remainder of the thesis. We began with the *van–mounted line-laser* acquisition system, describing the 1×5 mm native sensor pitch and the resulting 19995×4160 pixel strip images. The DOS algorithm converts poly-line crack descriptions into raster masks, but these suffer from systematic over-segmentation and omissions. To create network-ready inputs we devised an IoU-filtered patch extractor that (i) yields exactly one 512^2 tile per crack, (ii) removes near-duplicate crops, and (iii) assigns tiles to deterministic 70% / 15% / 15% train–val–test splits, ultimately producing 3 079 / 387 / 387 patches. Recognizing the limitations of auto-labels, we commissioned an **expert-annotated benchmark**: 127 "mother patches" (256×1280 px) with three classes: *crack*, *unknown*, *background*. These are de-stretched, tiled into 224^2 crops, and evaluated with group-wise confusion-matrix fusion so that metrics reflect whole cracks rather than arbitrary tile boundaries. Together, the automatic DOS/SAM2 dataset and the hand-labeled benchmark provide a rigorous, non-overlapping foundation for the methodological and experimental chapters that follow.

4 Methodology

4.1 Problem statement

Given a monocular RGB road patch[1] $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ ($H = W = 224$), the task is to predict a binary crack mask $\widehat{\mathbf{y}} \in \{0,1\}^{H \times W}$ such that

$$f_{\boldsymbol{\theta}} : \mathbf{x} \longmapsto \widehat{\mathbf{y}} = \arg\max_{c \in \{0,1\}} p_{\boldsymbol{\theta}}(c \mid \mathbf{x}),$$

where $f_{\boldsymbol{\theta}}$ is a deep network with parameters $\boldsymbol{\theta}$ and $p_{\boldsymbol{\theta}}(1 \mid \mathbf{x})$ denotes the foreground (crack) probability at each pixel. Ground-truth masks $\mathbf{y} \in \{0,1\}^{H \times W}$ are available only through *noisy* annotation pipelines; therefore the loss is computed with respect to *soft* targets $\tilde{\mathbf{y}}$ that are first denoised §4.4.2. Training minimizes a hybrid Dice (see §4.5.2) + Cross-Entropy objective

$$\mathcal{L}(\boldsymbol{\theta}) = \tfrac{1}{2}\, \mathcal{L}_{\mathrm{CE}}\big(p_{\boldsymbol{\theta}}, \tilde{\mathbf{y}}\big) \;+\; \tfrac{1}{2}\, \mathcal{L}_{\mathrm{Dice}}\big(p_{\boldsymbol{\theta}}, \tilde{\mathbf{y}}\big).$$

Why is this hard? Road cracks are (i) *fine-grained*: one-two-pixel branches surrounded by strong texture clutter; (ii) *label-noisy*: Expert labels, in this study, are painted with a broad brush, missing hairline fractures, whereas automatic DOS masks are coarse and fragmented; and (iii) *class-imbalanced*: foreground pixels often represent a small minority in the dataset distribution.

4.2 Design rationale

To tackle the above challenges we combine three orthogonal ingredients:

1. **DINOv2 backbone.** Self-supervised Vision Transformers (ViTs) pre-trained with DINOv2 (Oquab et al., 2023) capture long-range structural cues vital for filamentary objects. Replacing the original ViT-B/16 in TransUNet with a stronger ViT-B/14 (DINOv2) injects rich mid-level representations without requiring extra labels (§4.5).
2. **Hybrid CNN + ViT encoder.** A shallow ResNetV2 stem preserves high-frequency edge details while the transformer encodes global context; skip connections fuse the two (§4.5.3).
3. **Offline label refinement with SAM2.** Promptable SAM2 masks (Ravi et al.) replace the coarse DOS masks, yielding a cleaner training set $\mathcal{D}_{\mathrm{SAM2}} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}$ (§4.4.2).

4.3 High-level pipeline

Figure 4.1 gives an overview of the full research pipeline:

1. **Data acquisition**: raw DOS frames + coarse masks.
2. **Offline label improvement**: SAM2 prompts $\Rightarrow$ refined $\tilde{\mathbf{y}}$.
3. **Network modifications**: integrate DINOv2, ResNetV2, positional-embedding tweaks, skip-connection variants.
4. **Training**; hybrid Dice/CE loss, cosine LR decay, tolerant-metric monitoring (§4.5.4).
5. **Evaluation**: tolerant disk-kernel metrics ($r = 10$ px) (See. §4.6) evaluated on three test splits + qualitative inspection (Chapter 5).

[1] Captured by a laser system mounted on the back of a van at 224×224 px after cropping and normalization.

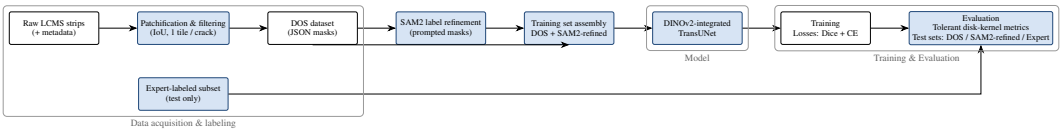


Figure 4.1: End-to-end pipeline. Blue blocks highlight novel contributions of this thesis: SAM2 label refinement and the DINOv2-augmented TransUNet.

Connecting back to the research questions. The methodological choices outlined above are not ad-hoc; each one targets a specific research question stated in §1.2.

- **RQ1** *How can DINOv2 be integrated to improve fine-grained segmentation?* This is tackled by **(i)** replacing the original ViT-B/16 backbone with a stronger self-supervised DINOv2 ViT-B/14, **(ii)** reinstating a shallow ResNetV2 stem to recover high-frequency edges, and **(iii)** fusing the two representations via skip connections. These architectural modifications (§§4.5–4.5.3) are therefore the concrete hypotheses tested under RQ1.
- **RQ2** *How can we train and evaluate robustly under noisy or incomplete labels?* We address this through a two-pronged strategy: **(a)** offline *label refinement* with SAM2 to produce a cleaner train/val split ($\mathcal{D}_{\mathrm{SAM2}}$; §4.4.2), and **(b)** a *tolerant disk-kernel metric* ($r = 10\,\mathrm{px}$) plus morphological opening during evaluation (§4.6) so that minor annotation misalignment does not dominate the loss/metrics. Together, these steps constitute the experimental answer to RQ2.

Hence, the architectural upgrades (**RQ1**) and the label-noise mitigation pipeline (**RQ2**) jointly form the proposed solution to the overarching problem of accurate, reproducible crack segmentation under imperfect supervision.

The next sections zoom into each numbered block: data curation (§4.4), architectural changes (§4.5), training protocol (§4.5.4), and evaluation strategy (§4.6).

4.4 Data pre-processing and Preparation

This section will discuss two topics:

1. **Naive label generation:** to try and generate better labels naively a thresholding technique was devised to try and quickly generated mask to see if further development of more novel measures was necessary.
2. **SAM2 label refinement:** A novel label refinement pipeline is introduced to generate better masks using the already available *DOS masks* as prompting material for SAM2 to generate higher quality labels.

These two topics outlined are explored and further elaborated upon in the following sub-sections.

4.4.1 Label generation using Thresholding techniques

As mentioned previously in this chapter and as described in Chapter 3, the current dataset uses auto-generated labels. The decision was made to use these generated labels as a starting point to improve the data using advanced methods, namely Segment Anything 2 (SAM2) by Ravi et al.. But this brings up a valid question, is there a more naive method that could possibly generated solid results with low effort?

To this end a more traditional method was employed to try and generate decent groundtruth labels by thresholding the raw images on certain pixel values to see if this would produce a representable mask. To get rid of small noise which is prevalent in the data, a morphological opening operation was performed after the thresholding operation (OpenCV Team, 2025).

1. Thresholding was performed because the characteristics of the data suggested that darker pixels were part of anomalies in the road, that includes cracks. The hypothesis was that

thresholding on a certain pixel value can keep most crack pixels whilst removing most of the unimportant background data.

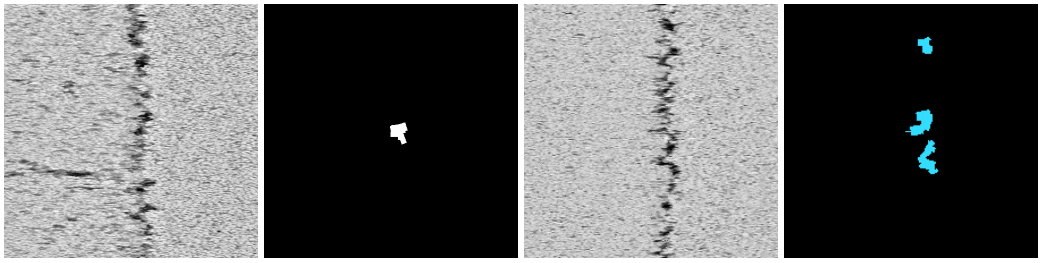
2. A morphological opening operation was performed to remove noise that was left in the data. This operation erodes the remaining pixels and dilates the remaining pixels back up using the same structural erosion used before.

in §5.4 the results of this approach will briefly be discussed as a precursory approach to the deep learning method that will be elaborated upon in the coming subsections.

4.4.2 Label Refinement with SAM2

Groundtruth issue and solution proposition

As discussed in Chapter 3 the groundtruth segmentation masks currently used are algorithmically generated using a black-boxed algorithm. They are somewhat representative of a crack as described by experts, but they miss nuanced crack structures. Very obvious cracks are highlighted, but some more nuanced cracks are being ignored or not fully mapped by the current algorithm, see Fig 4.2. To this end this research proposes a groundtruth refinement methodology using the vision foundation model (VFM) Segment anything 2 (SAM2). Using its promptable architecture the proposed system can feed in the image patches of areas with their respective raw segmentation masks as a prompt to generate a potentially better mask that can be used in the training process to make a better generalizing model. This section will go over this approach to tackle sub-question 2 highlighted in §1.2 of this thesis and show preliminary results.



(a) Image (b) Mask (c) Image (d) Mask

Figure 4.2: Visual illustration of two crack-patch excerpts and their associated algorithmically generated ground-truth masks. Notice that some crack structures (e.g. fine branching or thin cracks) are often absent or only partially captured in the existing labels and usually with over dialated masks, motivating the SAM2-based refinement proposed in §4.4.2.

Segment Anything 2 as an offline groundtruth refinement tool-set

Preliminary trials showed that SAM2 could substantially densify our legacy ground-truth masks: a domain expert at *TNO* confirmed that the additional pixels corresponded to genuine crack structure in a random sample of patches. Independently, Ravi et al. report that SAM2 attains top-1 segmentation accuracy on most benchmarks evaluated in their study, demonstrating strong zero-shot generalization and a flexible promptable interface.

SAM2 includes multiple prompting strategies, one of these methods being low resolution mask prompting. Since we have noisy masks already we opted to use the mask prompting strategy available in SAM2 as a first stage refinement step we can potentially build up later on in the research.

These factors motivated our choice to run SAM2 as an offline pre-processing step for ground-truth refinement.

Integration pipeline

In Figure 4.3 the pipeline is shown that describes how the entire process is setup of inserting patch images into the system, SAM2 being prompted with the auto-generated groundtruth and the output being processed to get ready for the training stage.

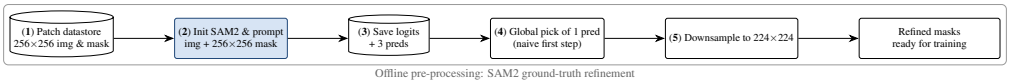


Figure 4.3: Integration pipeline for SAM2 refinement. Pairs of 256×256 images and noisy masks are fetched, SAM2 is prompted with the mask, outputs (logits and three predictions) are stored, a naive global selection is made, and the chosen mask is downsampled to 224×224 for training.

Figure 4.3 contains a few key steps in the pipeline, namely:

1. The data-store of the 256 x 256 image and original mask pair which are fetched from storage.
2. SAM2 being initialized, the images are fed into SAM2 with the original auto-generated mask provided as a low-resolution mask prompt. The mask needs to be exactly 256 x 256 pixels.
3. The output from SAM2, raw logits and three predictions, are saved to storage.
4. In a naive first step a simple selection is made out of the three predictions globally. This will be elaborated on in §4.4.2.
5. Final SAM2 masks re downsampled to 224 x 224 pixels, this resolution is required by our network

These steps together largely form the steps to tackle the second research question of trying to improve the training and evaluation when working with noisy labels in a supervised learning setting. In §4.6 the additional feature will be discussed to also help aid in improving the training and evaluation steps specifically targeting the specific training and evaluation challenges in this research.

CONSTRUCTING STRONGER TRAINING DATA FROM NOISY LABELS USING SAM2

Segment Anything 2 (SAM2) is used in conjunction with its low-resolution mask prompting ability to try and create richer segmentation masks for more representable training data. As mentioned previously in Figure 4.4.2 the system saves the raw logits and three predictions SAM2 makes. This data can be used to create richer segmentation masks that better represent cracks in their partner image, but how can this be done? A first naive method was developed after examining the results of generating richer segmentation masks over the dataset.

IoU Filtering of SAM2 predictions SAM2 produces three predictions based on the input given. These predictions are given an intersection over Union (IoU) score by SAM2 itself. The predictions are named their IoU score

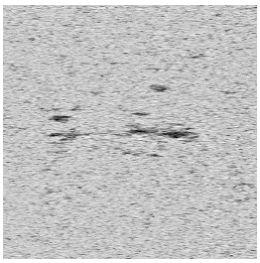
$$IoU = \frac{A \cap B}{A \cup B}$$

In Eq. 4.4.2 the IoU formula is shown where A, B are the groundtruth and the prediction respectively.

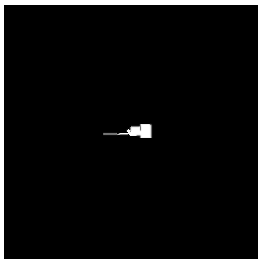
In Figure 4.4 an example is shown how SAM2 can improve label quality. Five images are shown, Figs. 4.4a, 4.4b are the original image and mask patch. The rest are the SAM2 predictions made thresholded at different IoU values. This example shows that to even none experts it might seems like Fig. 4.4d best represents the crack that is shown in the original image. As mentioned previously, during experimentation with SAM2 it was clear that taking the second highest IoU value mask that SAM2 predicted continuously represented the cracked pixels in the image patches the best. The naive method developed for improving training data was thusly straightforward. The system fed the 256 x 256 patches extracted through SAM2, the system filtered out the masks with the second highest IoU values and these new image and patch pairs are the new training/validation set.

4.5 MODEL ARCHITECTURE

This section describes the design decisions behind the segmentation network architecture used in this research, starting from baseline selection, continuing with architectural modifications, and concluding with variant designs used for ablation studies.



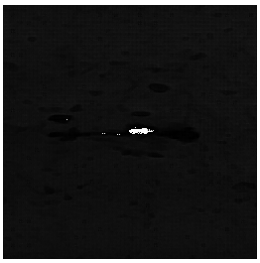
(a) Original patch



(b) Original mask

(c) SAM2 mask IoU of 0.14

(d) SAM2 mask IoU of 0.40



(e) SAM2 mask IoU of 0.56

Figure 4.4: Example of SAM2 improving a patch example with three returned predictions, the original patch and mask

4.5.1 Baseline selection

Initial experimentation with domain-specific crack detection networks such as CrackDiff (Zhang et al., 2024) and RHACrackNet (Zhu et al., 2024b) revealed limited reproducibility and suboptimal generalization. These models, during the literature study, showed strong performance against the competitor networks. When implementing these networks the codebase was lacking key code to reproduce their claimed results for CrackDiff. RHACrackNet's code was fully available but we were not able to reproduce their results they claimed in their paper with metric deltas in the 20% range. Consequently, the search was expanded to the medical image segmentation field, where architectures exhibit strong performance on fine-grained, irregular structures, characteristics shared with asphalt cracks.

TransUNet emerged as a compelling candidate due to its hybrid architecture that combines convolutional feature extraction with Transformer-based global context modeling (Chen et al., 2021). Its encoder-decoder structure, enriched by skip connections, enables effective segmentation of elongated, fine details, properties highly relevant for crack detection.

TU's design aligns well with our task requirements and research question, particularly in evaluating architectural generalization under noisy supervision. Its proven performance across medical benchmarks, wide adoption, and publicly available implementation further motivated its selection as the baseline architecture (Xiao et al., 2023)

4.5.2 Loss definition

To alleviate the extreme foreground–background imbalance ($\approx 1\%$ crack pixels) while preserving thin-structure continuity, we adopt *unchanged* the hybrid loss from the reference implementation of the original method by Chen et al. (2021), (GitHub[2]). No novel loss engineering was performed in this work; the contribution here is limited to applying the authors' publicly released code within our training pipeline.

[2]Public repository of the original authors, `https://github.com/Beckschen/TransUNet`; code reused verbatim.

$$\mathcal{L}(\boldsymbol{\theta}) = \tfrac{1}{2}\,\mathcal{L}_{\mathrm{CE}}\big(p_{\boldsymbol{\theta}}, \tilde{\mathbf{y}}\big) \;+\; \tfrac{1}{2}\,\mathcal{L}_{\mathrm{Dice}}\big(p_{\boldsymbol{\theta}}, \tilde{\mathbf{y}}\big), \tag{4.1}$$

where $p_{\boldsymbol{\theta}}$ denotes the softmax output of the network and $\tilde{\mathbf{y}}$ the one-hot ground truth.

Cross-Entropy. The pixel-wise term is the standard `torch.nn.CrossEntropyLoss` module provided by PyTorch.

Soft Dice. To maximize the spatial overlap between prediction and ground truth the authors minimized the complement of the differentiable Sørensen–Dice coefficient:

$$\mathcal{L}_{\mathrm{Dice}} = \frac{1}{K}\sum_{c=1}^{K} w_c\Big(1 - \frac{2\sum_{x\in\Omega} p_c(x)\,\tilde{y}_c(x) + \varepsilon}{\sum_{x\in\Omega} p_c(x)^2 + \sum_{x\in\Omega} \tilde{y}_c(x)^2 + \varepsilon}\Big), \tag{4.2}$$

where $p_c(x) \in [0,1]$ is the model's probability for pixel x to belong to class c, $\tilde{y}_c(x) \in \{0,1\}$ the corresponding one-hot ground truth, w_c an optional class weight (default $w_c = 1$), and $\varepsilon = 10^{-5}$ guarantees numerical stability when a class is absent.

During each update step we compute the two losses and back-propagate their arithmetic mean, mirroring the authors' training script without modification. This reuse ensures comparability to the baseline reported in the original paper; performance-related results presented later in §5.2 therefore reflect *model* and *data* changes only, not alterations to the loss function.

4.5.3 Why Upgrade the ViT Backbone? Limitations of ViT-B/16 and the Case for DINOv2

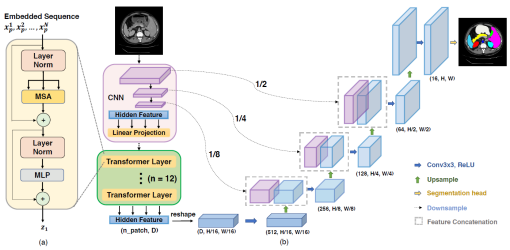


Figure 4.5: TransUNet architecture as described in Chen et al. (2021)

Figure 4.5 reproduces the original TransUNet encoder–decoder. The shaded block marks the *Vision-Transformer encoder*, implemented in TransUNet (Chen et al., 2021) as a **ViT-B/16** pre-trained on ImageNet under a *supervised* objective (Dosovitskiy et al., 2020). Since 2021, the ViT landscape has advanced rapidly: **DINOv2** (Oquab et al., 2023), a self-supervised **ViT-B/14**, now sets the state-of-the-art on COCO panoptic, ADE20k, and a plethora of zero-shot transfer tasks. Our work therefore *replaces* the legacy ViT-B/16 weights and network with the more recent DINOv2 backbone while leaving the CNN stem, skip connections, and decoder unchanged (with a minor change to the decoder).

Why swap the backbone?

- **Finer token grid.** ViT-B/14 uses a 14×14 patch stride (vs. 16×16), preserving higher spatial resolution in the tokens supplied to the decoder, crucial for delineating sub-pixel-wide cracks.
- **Self-supervised contour awareness.** The DINOv2 momentum-teacher objective encourages attention heads to lock onto object boundaries. Visualizing its intermediate maps shows crack-like ridges even *before* fine-tuning, giving the network a "head-start" on boundary localization.

- **Robustness to noisy labels.** Because its representations are learned without class labels, DINOv2 is less prone to over-fitting the artifacts that plague our auto-generated ground truth.
- **Cross-domain evidence.** DINOv2 has demonstrated strong transfer on satellite, medical, and documentary imagery, suggesting that its features generalize beyond the natural images used for pre-training.

Implementation details. The upgrade is architectural only in patch stride; the Transformer *block structure* remains the vanilla ViT. We initialize with the public `dinov2_vitb14` weights, keep the ResNetV2 stem for early local features, and funnel its activations to the UNet-style skip connections exactly as in the original TransUNet. Choosing the *Base* (B) size keeps parameter count comparable to ViT-B/16, ensuring that any performance gains can be attributed to richer representations rather than sheer scale.

Hypothesis. Replacing TransUNet's supervised ViT-B/16 with the contour-aware DINOv2 ViT-B/14 will yield crisper crack masks, especially under noisy or incomplete supervision. §5.3 quantifies this hypothesis through controlled ablations.

ARCHITECTURAL MODIFICATIONS

To replace ViT-B 16 with DINOv2 modifications of several key model layers was necessary.

1. ViT-B 16 has patch sizes of 16x16 and DINOv2 has patch sizes of 14x14.
2. DINOv2 expects RGB images as input, not high dimensional feature maps (ResNetV2 output).
3. Due to patch size mismatches between the original and new Transformer downsampling of the final output from 256 x 256 to 224 x 224 pixels so that metrics can be computed.

Since the main architecture of TransUNet has not changed majorly there is no architecture overview figure. We replaced the existing ViT-B/16 with a DINOv2 ViT-B/14 and made the necessary changes to accommodate this new model as outlined in subsequent sections. Finally, we added a downsampling layer before outputting the masks since the original network upsamples to 256 $\times$ 256 pixels. To match the ground truth we down sample to 224 $\times$ 224.

Implementation Challenges and Solutions The integration of DINOv2 into the TransUNet framework required addressing several architectural and dimensional mismatches between the original ViT-B/16 backbone and DINOv2's structure.

First, DINOv2 uses a patch size of 14×14 compared to the 16×16 patch size in ViT-B/16. Given input images of size 224×224 (the patch size used in this study), DINOv2 outputs a grid of 16×16 tokens (i.e., $\frac{224}{14} = 16$), while ViT-B/16 would yield 14×14 tokens. This affects the spatial resolution of features forwarded into the decoder and required adaptations in the upsampling path to ensure spatial alignment with skip connections and prediction heads.

Second, the original TransUNet architecture feeds high-dimensional CNN feature maps (e.g., 768 or 1024 channels) into the Transformer. In contrast, DINOv2 expects 3-channel RGB images as input. To reconcile this, two strategies were used:

- In the **ViT-only** variant, the ResNetV2 encoder was removed entirely and raw 3-channel 224×224 images were directly passed into DINOv2.
- In the **hybrid** variant, a 1×1 convolutional projection layer was inserted to map the high-dimensional output of ResNetV2 (shape: [B, 1024, 14, 14]) into a 3-channel representation ([B, 3, 224, 224]) suitable for DINOv2 input. Note that this also required spatial upsampling from 14×14 to 224×224, which was performed using bilinear interpolation prior to the projection.

Third, since SAM2-generated masks and the DINOv2 input images were both fixed at 224×224 resolution, the final prediction maps were also resized back to this resolution post-decoding. This

Table 4.1: Detailed tensor shape overview when integrating DINOv2 ViT-B/14 into TransUNet

Stage	Operation / Component	Tensor Shape [B, ...]
Input Image	Input patch image	[B, 3, 224, 224]
Patch Embedding	14×14 non-overlapping patch embedding + linear projection	[B, 256, 768] (16 × 16 patches)
Transformer Encoder	12 transformer blocks (ViT-B)	[B, 256, 768] (positional encodings added internally)
Reshape for Decoder	Reshape tokens back to spatial map	[B, 768, 16, 16]
Upsampling Block 1	ConvTranspose2d (2× upsample)	[B, C, 32, 32] (C: decoder channels)
Skip Connection (Low-Level)	ResNetV2 feature skip connection (e.g., conv3$_x$)	[B, C, 32, 32]
Decoder Block 2	Up-conv + fusion with skip features	[B, C, 64, 64]
Decoder Block 3	Up-conv + fusion with skip features	[B, C, 128, 128]
Decoder Block 4	Up-conv + fusion with skip features	[B, C, 256, 256]
Final Conv Layer	1×1 conv to logits	[B, 1, 256, 256] (binary segmentation mask)
Reshape for Evaluation	Resize to match groundtruth	[B, 1, 224, 224]
Groundtruth Alignment	GT mask shape	[B, 1, 224, 224]

ensured that pixelwise comparisons for computing segmentation metrics (e.g., IoU, F1) remained consistent.

The details of these changes made to the network are showcased in Table 4.1

Fair model selection. Early backbone ablations showed that replacing the ImageNet-initialised **ViT-B/16** with a self-supervised **DINOv2ViT-B/14** boosts mean IoU by roughly $+\mathbf{5}\%$ at similar parameter count. Scaling up to larger DINOv2 variants (ViT-L/14, ViT-g/14) could have offered possible performance gains. But to keep the comparison between models fair we chose to select the DINOv2 ViT-B/14 model to try and match model size as closely as possible. Independent studies in aerial and biomedical segmentation support our hypothesis and findings findings, reporting consistent gains of DINOv2B/14 over CLIP, MAE, and supervised ViT backbones (Shah, 2024; Ayzenberg et al., 2024), underscoring the fairness and domain-agnostic strength of our backbone choice.

Expectations and possible trade-offs After successful integration of DINOv2 into the backbone of TransUNet it is expected to predict more accurate segmentation mask due to DINOv2's better alignment with segmentation tasks in conjunction with a smaller patch size of 14 x 14 giving more fine-grained details in global context modeling which is of great importance in crack detection due to the nature of asphalt cracks. The original ResNetV2 + ViT-B/16 architecture was pre-trained together which allows a smooth learned CNN-feature to token representation. With DINOv2 integration this rich learned layer is lost and needs to be re-adapted which could cost the network in performance. Also, since DINOv2 is such a strong global context modeler, being able to generalize well to various applications such as segmentation or object detection, care must be taken to recombine it with ResNetV2. The two together may be redundant, but later experiments will try to answer this question.

Conclusion and Transition These targeted modifications allowed DINOv2 to be successfully embedded within the TransUNet architecture, replacing the original ViT-B/16 while preserving compatibility with the original image patch sizes, decoder expectations, and evaluation metrics. These steps were essential to fairly evaluate the effect of stronger backbone features in noisy supervised segmentation, which directly ties into the first sub-question of this research. The next section out-

lines ablation studies and model variants designed to isolate and evaluate the individual contributions of each architectural component.

4.5.4 Training Procedure

This section will go over the training approach during this study. First a high-level training overview is described on how the training process was implemented. Following this description, optimization and fine-tuning will be discussed in more detail and what software and hardware that was used in the course of this study.

Training pipeline

1. **Data ingestion**
   Each mini-batch is a stack of $B = 18$ Grayscale tiles (224×224); labels are down-sampled on-the-fly before auto-putting the final segmentation mask to 224×224 to match the decoder output.
2. **Forward pass**
   The image batch is fed through
   (a) an *optional* ResNetV2 stem,
   (b) the DINOv2 ViT-B/14 encoder (patch size 14),
   (c) Decoder-CUP and segmentation head.
3. **Loss computation**
   $\mathcal{L} = 0.5\,\mathcal{L}_{\text{CE}} + 0.5\,\mathcal{L}_{\text{Dice}}$ (binary cross-entropy and binary Dice; no class weighting). Just like in the original paper.
4. **Back-propagation & optimisation**
   AdamW, base LR $= 1\times10^{-5}$, Weight decay $= 0.01$.;
   Learning rate follows cosine annealing ($T_{\text{max}} = 120$epochs), min LR $= 1 \times 10^{-7}$.
5. **Validation loop**
   After each epoch we compute Dice, IoU and AUROC on the validation split; the best checkpoint (lowest val-loss) is retained.
6. **Early stopping & checkpointing**
   Training stops if val-loss fails to improve for ten consecutive epochs. Snapshots are also written at 20% / 40% / 60% / 80% of the run.

These steps outline in big strokes what the training process looks like from data-loading to the end of training.

Optimization and Fine-Tuning

Since ResNetV2 and DINOv2 are pre-trained models the training process was more fine-tuning these two models in the backbone than training a model from scratch. The learning rate was mostly adapted to these two models than for the decoder part of the UNet like network. In the original paper (Chen et al., 2021) it is not mentioned whether they fine-tune or not, but it seems logical to adapt the weights to the specific use case you intend to use, that is why we fine-tuned both these models instead of freezing the weights in place but no experimentation was done on this front.

The learning rate for the entire network was kept the same, 1×10^{-5} across all network components, as described above. According to the literature this is not the most optimal fine-tuning strategy when utilizing large pre-trained encoders (Dong et al., 2022)., (Touvron et al., 2022). The better option is to choose a lower learning rate for the pre-trained encoder to maintain its rich representation and to suitably adapt to the new downstream task. Due to only using a set learning rate across the entire network some performance is left on the table.

4.6 Customized Evaluation and Metrics

Motivation The original TransUNet work evaluates medical organ segmentation with *Dice* and *95 % Hausdorff distance*. For pavement-crack maps these metrics are overly sensitive to (i) coarse

or incomplete ground-truth masks and (ii) usually not used in pavement crack detection (specifically the 95% Hausdorff-distance). Therefore a new metrics suite is adopted that tries to balance false positives (FP) and false negatives (FN) while remaining interpretable by civil-engineering practice: *Intersection-over-Union (IoU)*, *Precision*, *Recall*, and *F1*. All are derived from a distance-tolerant confusion matrix described below.

Handling Incomplete Ground-Truth

Traditional medical–image work on TransUNet reports Dice and Hausdorff scores computed under the *exact* pixel match assumption (Chen et al., 2021). In asphalt-crack imagery this assumption is too strict: In the case of this study masks are painted with a broad brush or our auto-generated, hair–line cracks are frequently missed, and large swaths are deliberately marked `255` to denote "unknown" pixels, see Chapter 3. We therefore adopt a **relaxed, distance-aware evaluation protocol**.

Morphological pre–cleaning of predictions Before the tolerant disk dilation is applied symmetrically to *both* ground–truth and prediction, we subject the predicted binary mask $\hat{P} \in \{0,1\}^{H\times W}$ to a *morphological opening*

$$\hat{P}^{\star} \;=\; \underbrace{\mathrm{dilate}\big(\mathrm{erode}(\hat{P},\, K_{\text{open}}),\, K_{\text{open}}\big)}_{\text{opening with structuring element } K_{\text{open}}}$$

where K_{open} is a disk structuring element of a small radius $r_{\text{open}} = 1$–$2\,\text{px}$. The initial erosion removes single or small pixel count artifacts and thin "salt-and-pepper" noise; the subsequent dilation restores the support of legitimate crack regions larger than r_{open}. Empirically, this pre-cleaning suppresses *spurious* false positives that would otherwise be counted after the subsequent tolerance dilation, without harming genuine detections. This method is applied to the prediction specifically because the groundtruth labels are noisy and a robust method of evaluation is necessary.

After obtaining $\hat{P}^{\star}$ we proceed with the *symmetric* tolerance dilation:

$$\widetilde{P} = \mathrm{dilate}(\hat{P}^{\star}, K_{\text{tol}}), \qquad \widetilde{G} = \mathrm{dilate}(G, K_{\text{tol}}),$$

using an Euclidean disk kernel K_{tol} of radius r_{tol} (default $r_{\text{tol}} = 2\,\text{px}$). This grants a localization tolerance of $\pm r_{\text{tol}}$ pixels when computing the confusion-matrix entries $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}, \mathrm{TN}$, yielding a fairer evaluation for hair–line cracks whose manual annotations are often *not* pixel perfect.

Pixel-relax radius. Let $\mathcal{P}$ and $\mathcal{G}$ be the binary prediction and ground–truth masks, respectively. Instead of counting a pixel-wise TP only when $p_{ij} = g_{ij} = 1$, we allow a prediction at (i, j) to match any foreground pixel inside a closed Euclidean disk of radius r centred at (i, j). Concretely we *dilate* one operand with a disk structuring element $\mathcal{D}_r$ before forming the confusion matrix.

Unknown class. Pixels labeled `255` carry no supervisory signal; they are ignored in both the loss and the metric computation by masking them out before the operations above. This `ignore_index` paradigm is common in ADE20k, Cityscapes and other dense-labeling datasets (Zhou et al., 2018), (Cordts et al., 2016).

Distance-Tolerant Confusion Matrix

Denote binary masks $G,\ P \in \{0,1\}^{H\times W}$ for ground-truth and prediction; $\mathcal{D}_r(\cdot)$ is binary dilation with a disk of radius r:

$$\mathcal{D}_r(G) = G * K_r, \tag{4.3}$$

where $K_r(x, y) = \mathbb{1}\big(x^2 + y^2 \le r^2\big)$ is the disk kernel and $*$ denotes 2-D convolution. The relaxed counts are

$$\mathrm{TP}_r = \big|\mathcal{D}_r(G) \wedge P\big|, \qquad \mathrm{FN}_r = \big|G \wedge \neg\mathcal{D}_r(P)\big|, \tag{4.4}$$

$$\mathrm{FP}_r = \big|P \wedge \neg\mathcal{D}_r(G)\big|, \qquad \mathrm{TN}_r = \big|\neg G \wedge \neg P\big|. \tag{4.5}$$

The implementation follows Algorithm 2.

Algorithm 2 Disk–kernel tolerant confusion matrix (PyTorch)

Require: probability map p, GT mask g, radius r, threshold τ
$P \leftarrow (p \geq \tau)$
$G \leftarrow (g = 1)$ {ignore pixels where $g = 255$}
$K \leftarrow \textsc{DiskKernel}(r)$
$\tilde{G} \leftarrow \textsc{Dilate}(G, K)$
$\tilde{P} \leftarrow \textsc{Morphological Open}(P, K)$
TP $\leftarrow \|\tilde{G} \wedge P\|_1$
FP $\leftarrow \|P \wedge \neg\tilde{G}\|_1$
FN $\leftarrow \|G \wedge \neg\tilde{P}\|_1$
TN $\leftarrow \|\neg G \wedge \neg P\|_1$
return (TP, FP, TN, FN) =0

Implementation and Reproducibility

Metrics are computed *in-process* during training and re-computed during inference for the test set. All operations are deterministic; we fix seeds for *PyTorch*, *numpy*, *CuDNN* and *random*, and save every confusion matrix as a *NumPy* file for independent verification.

4.7 SUMMARY OF METHODOLOGY

- **Data pipeline.** Long hectometer-strip scans are cut into non-overlapping 224×224 tiles and paired with coarse DOS masks. More detail in Chapter 3 A naïve *threshold + opening* labeling baseline is reported for reference, but the main experiments use **SAM2** low-resolution mask prompting to yield the refined training set $\mathcal{D}_{\text{SAM2}}$.
- **Network design.** TransUNet is upgraded by (i) swapping the supervised ViT-B/16 encoder for a self-supervised **DINOv2** ViT-B/14, (ii) reinstating a shallow **ResNetV2** stem for high-frequency edges, and (iii) experimenting with optional UNet-style **skip connections**. These components are toggled to form the ablation variants.
- **Training regime.** Hybrid Dice / BCE loss, AdamW (1×10^{-5} base LR, cosine decay), batch $= 18$, early-stopping at 10 stagnant epochs. All encoders are *fine-tuned* rather than frozen, a parameter-efficient strategy that re-uses pre-trained weights instead of training from scratch.
- **Robust evaluation.** Noisy or incomplete masks are handled with a **disk-kernel tolerant** confusion matrix (radius $r = 10$px) and a 1-px morphological opening of predictions. Metrics reported are Precision, Recall, F1 and IoU, all computed after masking out `unknown` pixels.
- **Logic recap.** *Noisy labels* motivate tolerant metrics and SAM2 pre-cleaning; *fine-grained cracks* motivate a contour-aware DINOv2 backbone and skip fusion; *computational economy* is preserved by re-using pre-trained weights instead of heavy end-to-end training. Together, these choices operationalize **RQ1** (architecture under fine detail) and **RQ2** (robust learning under label noise).

5 Experiments

This chapter reports and analyses the empirical evidence supporting the contributions of the thesis. We first describe the experimental protocol, followed by a series of quantitative and qualitative evaluations that expose the strengths and limitations of the proposed approach under multiple architectural variants using ablation studies.

5.1 Experimental Setup

This section will discuss the setup done for the experiments, the data and its splits, hardware used and any other important configuration details.

Data splits this experiments used *train*, *validation* and *test* split setup where the *train* split is *70%* of the data. The *validation* set is *15%* of the data. The *test* set is a subset of the remaining *15%* of data, to be specific it consists of 127 samples.

Hardware/Software: All experiments were conducted on the same machine using the same virtual python environment with the same modules installed inside of a docker container.

Hardware:

1. GPU: NVIDIA RTX 2080TI 11GB VRAM
2. CPU: Intel i7 7820X 3.6GHz
3. RAM: 32 Gigabyte

Software:

1. Docker image: PyTorch 2.5.1 CUDA 12.4 CuDNN 9 runtime
2. Python version 3.12.3

Fixed seeds

1. python random: 1234
2. CuDNN/CUDA: 1234
3. NumPy: 1234
4. PyTorch: 1234

Training parameters:

1. Optimizer: AdamW
2. Learning rate $1e-5$
3. Weight decay: 0.01
4. Learning rate scheduler: Cosine annealing
5. Batch size: 18
6. Max Epochs: 100
7. Early stopping: After 10 epochs of no validation loss improvement stop training. Best model = model with lowest learning rate.

In addition to this CuDNN was set to *deterministic* to help reproducibility.

Furthermore for the experiments the new evaluation methodology was used introduced in §4.6 since this new methods should better represent real road cracks and their need to not be as tolerant as pixel perfect segmentation. Finally, all quantitative experimental results listed in Tables 5.1, 5.3, 5.4, 5.6 are obtained by averaging the results across three runs for a representative result.

5.1.1 METRIC DEFINITIONS

Let $\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN}$ be the counts after applying the radius-r tolerance.

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \epsilon}, \qquad \text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \epsilon},$$
$$\text{F1} = \frac{2TP}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \epsilon}, \qquad \text{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \epsilon}$$

Here ϵ is a small value of 10^{-8} to prevent possible division by zero scenarios.

5.2 MAIN QUANTITATIVE RESULTS

5.2.1 COMPARISON WITH BASELINES

Model	**Training split**	**DOS-orig test**		**SAM2 test**		**Hand-label test**	
		F1 ↑	IoU ↑	F1 ↑	IoU ↑	F1 ↑	IoU ↑
Baseline TransUNet	*Orig. train/val*	0.732	0.631	0.732	0.651	0.342	0.278
Baseline TransUNet	*SAM2 train/val*	0.622	0.512	0.747	0.663	0.225	0.171
Ours (best arch.)	*Orig. train/val*	**0.757**	**0.659**	0.746	0.668	**0.376**	**0.315**
Ours (best arch.)	*SAM2 train/val*	0.639	0.531	**0.765**	**0.684**	0.215	0.164

Table 5.1: Headline segmentation results across three evaluation splits. All scores use the tolerant disk–kernel metric ($r = 10$ px).

Table 5.1 reports F1 and IoU measured with the r=10 px tolerant kernel on the three evaluation splits introduced in Chapter 3. Results are grouped by *training split* (*Orig.* vs. *SAM2*) and by architecture (Baseline TransUNet vs. our **best arh.**: TransUNet equipped with a pure DINOv2 ViT-B/14 encoder).

Data quality dominates when train–test domains match. Training on the same label flavor that is used for testing outweighs most architectural differences:

- On the **DOS-orig** test set, models trained on *Orig.* masks outperform their SAM2–trained counterparts by ≈10 pp. F1 (rows 1/3 vs. 2/4).
- Conversely, on the **SAM2** test set the roles reverse: SAM2–trained networks gain roughly +0.02 F1/IoU over the ones trained on the thicker legacy masks (rows 2/4 vs. 1/3).

This confirms that the SAM2 prompts indeed yield cleaner supervision and that all pipelines can capitalize on that cleanliness *provided* the evaluation domain aligns with the training domain.

Architecture still matters once label noise is reduced. When comparing networks trained on the *same* split our DINOv2 backbone consistently outperforms the vanilla TransUNet:

$$\text{DOS-orig test: } \Delta\mathrm{F}_1 = +0.025, \ \text{SAM2 test: } \Delta\mathrm{F}_1 = +0.014$$

even under the forgiving r=10 px metric. The gap widens further on the stringent hand–labeled benchmark (§3.6): training on *Orig.* masks our best architecture raises F1/IoU from 0.342/0.278 to **0.376/0.315**, demonstrating that the self-supervised ViT features add resilience beyond what cleaner labels alone can provide.

Generalization to expert masks remains challenging. All models lose roughly half their F1 when evaluated on the 127 manually annotated images, indicating that neither label clean-up nor architectural upgrades fully bridge the domain shift from synthetic DOS strokes to real hair-line cracks. Nevertheless, the DINOv2 backbone trained on the legacy masks delivers the *best* headline score on this toughest split, narrowing the precision–recall gap and motivating the qualitative error analysis in §5.5.

In summary:

1. Matching the training labels to the test domain yields the largest absolute gain;
2. The proposed DINOv2 TransUNet provides an *additional* boost once label noise is under control; and
3. Significant headroom persists on the expert benchmark, setting the scene for the ablations in §5.3.

The next sections unpack how each architectural component (DINOv2 backbone, ResNet stem, skip connections) contributes under different training regimes (§5.3), and why certain variants generalize better than others across the three evaluation splits.

5.3 Ablation Study

To fairly assess the changes made to the original network, TransUNet, it is important to highlight the architectural variations on which experiments were ran to see what impact various parts had on the final evaluation metrics. In this section the architectural variations that were constructed will be explained.

1. Original TransUNet (*Baseline*)
2. TransUNet with only DINOv2 in the backbone (*No ResNetV2 nor skip connections*)
3. TransUNet with ResNetV2 and DINOv2 in the backbone (*No skip connecitons*)
4. TransUNet with ResNetV2 and DINOV2 in the backbone with Skip connections enabled.

Original TransUNet This is the original unmodified network that will serve as a baseline. In §4.6 new evaluation strategies are proposed to better and more fairly assess the predictions. This new method is also used in combination with the original network for fair comparison. But the core network and original weights are not modified nor re-trained. The new improved dataset discussed in §4.4.2 is also used in this network and all other variants.

TransUNet with DINOv2 backbone only In this network the new ViT-B/14 (DINOv2) has been integrated but without the CNN component originally present in TransUNet (ResNetV2). This will test how well DINOv2 on its own will compare to the full original backbone of TransUNet. This architectural variation will give early insights in the possible direction the research will be going in.

TransUNet with ResNetV2 and DINOv2 This will be the fully integrated architecture as used in the original paper with all adaptations made mentioned in **??**. This variation is two-fold, there will be two sub-variations of this network.

1. TransUNet with ResNetV2 and DINOv2 in the backbone without skip connections.
2. TransUNet with ResNetV2 and DINOv2 in the backbone with skip connections.

These two versions will show how skip connections contribute to the prediction quality. In the original paper it showed varying results on skip connection impact depending on what organ was segmented from the fMRI scan (Chen et al., 2021).

Table 5.2 shows a summary of the architecture variations. These will be experimented with to see the efficacy of the different configurations and which parts have the biggest impact on performance.

In addition to these model variations the model was also trained on two different dataset variants.

Table 5.2: Overview of architectural variants and their components

Model Variant	ResNetV2 (CNN)	ViT (Backbone)	Skip Connections
Baseline TU	✓	ViT-B/16	✓
TU+DINO+ResNet (skip conn.)	✓	DINOv2 (ViT-B/14)	✓
TU+DINO+ResNet (no skip conn.)	✓	DINOv2 (ViT-B/14)	X
TU+Pure DINO backbone	X	DINOv2 (ViT-B/14)	X

1. **Original DOS data**: This is the data which is not modified by SAM2 at all, this is the data coming straight out of the existing data pipeline from TNO (Chapter 3)
2. **SAM2 improved data**: This is the original data, but modified with SAM2 to try and generated more accurate ground truth labels.

This additional ablation step was added to see if the data has any impact to avoid drawing false conclusions on the impact of the architectural changes only.

5.3.1 Ablation study: networks trained on SAM2 data

Model variant	DOS-orig test				SAM2 test				Hand-label test			
	P ↑	R ↑	F1 ↑	IoU ↑	P ↑	R ↑	F1 ↑	IoU ↑	P ↑	R ↑	F1 ↑	IoU ↑
Baseline TU	0.745	0.650	0.622	0.512	0.915	0.709	0.747	0.663	0.292	0.246	0.225	0.171
TU+DINO+ResNet (no skip conn.)	0.744	0.670	0.633	0.522	0.902	0.714	0.747	0.662	0.365	0.258	0.269	0.207
TU+DINO+ResNet (skip conn.)	0.728	0.703	**0.649**	**0.537**	0.894	**0.751**	**0.767**	**0.684**	**0.366**	**0.277**	**0.283**	**0.221**
TU+Pure DINO backbone	**0.756**	**0.662**	0.639	0.531	**0.928**	0.723	0.765	**0.684**	0.276	0.241	0.215	0.164

Table 5.3: Ablation study with **all models trained on the SAM2 train/val split**. Each row shows performance on the three held-out test sets. Scores use our tolerant metric ($r = 10\,\text{px}$). Best per-column numbers are **bold**.

Table 5.3 compares four architectural variants, *all trained on the same SAM2 train/val split*, across the three evaluation sets. Three observations stand out:

Skip connections provide the largest single boost. Adding ResNet–to–ViT *and* enabling skip-fusion (TU+DINO+ResNet (skip)) lifts F1/IoU on every test split, with the strongest gains on **SAM2** itself (+0.020 F1, +0.022 IoU over the no-skip counterpart). The improvement stems mainly from higher *recall* (0.751 vs. 0.714), confirming that low-level CNN features help the decoder recover thin crack fragments that the ViT encoder alone misses.

Pure-ViT (DINO only) is precision-oriented. The "Pure DINO" backbone attains the *highest precision* on all splits (e.g. 0.928 on the SAM2 test) but lags behind the skip-enabled hybrid in recall and therefore F1/IoU. The model favors conservative, high-confidence masks; in the tolerant metric this yields fewer FP counts but leaves additional TP potential untapped.

Hybridization helps cross-domain generalization. When confronted with the out-of-distribution **DOS-orig** and **Hand-label** sets, the skip-enabled hybrid remains the best overall (F1/IoU 0.649/0.537 and 0.283/0.221, respectively), whereas the Pure-DINO variant slips in recall, and the baseline TransUNet, despite having seen the same SAM2 labels, cannot match either hybrid in any metric. We attribute this robustness to (i) multi-scale cues injected by the ResNet stem and (ii) the U-shaped skip pathway, both of which ease adaptation when texture statistics deviate from the training distribution.

Interim summary. For models trained on the cleaner SAM2 labels, *architectural upgrades matter.* The best configuration combines a ResNet stem, a DINOv2 ViT encoder, and full-resolution skip connections, achieving state-of-the-art performance on in-domain data while remaining the most reliable choice when the evaluation domain shifts. The next subsection repeats the analysis for models trained on the noisier **DOS-orig** data (§5.3.2) and shows how the relative importance of each component changes when label noise dominates.

5.3.2 Ablation Study: networks trained on DOS data

Model variant	DOS-orig test				SAM2 test				Hand-label test			
	P	**R**	**F1**	**IoU**	**P**	**R**	**F1**	**IoU**	**P**	**R**	**F1**	**IoU**
Baseline TU	0.803	0.744	0.732	0.631	0.894	0.709	0.732	0.651	0.454	0.300	0.342	0.278
TU + DINO + ResNet *(no skip conn.)*	0.801	**0.776**	0.748	0.644	0.896	**0.735**	**0.753**	**0.671**	0.438	0.283	0.325	0.251
TU + DINO + ResNet *(skip conn.*	0.785	0.754	0.729	0.621	0.887	0.721	0.739	0.653	0.428	0.267	0.309	0.251
TU + *pure* DINO backbone	**0.823**	0.770	**0.757**	**0.659**	**0.908**	0.722	0.746	0.668	**0.470**	**0.341**	**0.376**	**0.315**

Table 5.4: Ablation study with **models trained on the *DOS* train/val split**. All numbers are reported with the tolerant disk–kernel metric ($r = 10$ px).

Table 5.4 shows the same ablation as done in (§5.3.1) but now the models are trained on the original DOS data to see how the architectural elements are impacted when noisy labels prevail. A few take-aways can be observed.

Pure DINO dominates under noisy supervision. The ViT-only backbone attains the highest F1/IoU on *two out of the three* test splits, including the challenging hand-label benchmark, despite having no CNN stem or skip pathway. We argue that the global self–attention helps to *average out* annotation noise, while the absence of low-level fusion avoids propagating label errors to the decoder.

Skip connections amplify label noise. Introducing ResNet features and skip fusion *hurts* performance relative to the no-skip hybrid (row 3 vs. row 2). When trained on thick, coarse DOS labels, the decoder learns to trust low-frequency blobs and over-segments at test time, explaining the drop in precision on every split.

Hybrid, no-skip configuration trades precision for recall. Removing skips but keeping the ResNet stem (row 2) yields the best *recall* on both DOS-orig (0.776) and SAM2 (0.735) tests, but at a modest cost in precision. In scenarios where missing a crack is costlier than over-painting, this variant may be preferable.

Cross-domain behavior flips compared to SAM2-trained models. Recall that with *clean* SAM2 supervision, the ResNet + ViT + skip variant was best (§5.3.1). When supervision is *noisy*, the hierarchy reverses and the simpler Pure-DINO backbone excels. This underscores the interaction between architecture choice and label quality: richer decoders pay off only when the ground truth is reliable.

Interim summary. With *noisy* DOS supervision, the precision–oriented *Pure-DINO* variant delivers the strongest overall scores, while skip connections tend to amplify over-segmentation. Conversely, under *cleaner* SAM2 supervision (§5.3.1) the hybrid decoder with skips reclaims the lead, illustrating that the benefit of architectural capacity depends on label fidelity. The fact that DOS-trained models fare better on the hand-label test is also consistent with annotation style: both the DOS masks and the expert drawings use a *thick-brush* convention, whereas SAM2 labels contain many finer hair-line cracks and therefore induce a different decision boundary. Finally, this is also exacerbated by the introduced metric (Section 4.6) which uses a generous dilation of $r = 10$ pixels which conforms the predictions even more to the DOS automated annotation style.

5.3.3 Architecture Ablation on Crack500

To disentangle architectural effects from label noise and domain bias in our in-house data, we re-trained all model variants on the public Crack500 benchmark Yang et al. (2019) using its official train/val/test split. Table 5.5 reports precision (P), recall (R), F1, and IoU.

Table 5.5: CRACK500 ablation: isolating backbone/stem/skip choices. The *Pure DINOv2* TransUNet (no CNN stem) is best overall.

Variant	**P**	**R**	**F**1	**IoU**
Baseline TransUNet (TU)	0.851	0.809	0.810	0.712
TU + DINO + ResNet (*skip conn.*)	0.804	0.788	0.775	0.662
TU + DINO + ResNet (*no skip conn.*)	0.777	0.789	0.763	0.644
TU + *Pure* DINO backbone (TU decoder)	**0.856**	**0.867**	**0.851**	**0.762**

Key observations.

1. **Pure ViT wins.** Replacing the entire encoder with a pretrained DINOv2 ViT-B/14 and *omitting* the ResNet stem yields the best F1 (0.851) and IoU (0.762), outperforming the baseline TransUNet by +0.041 F1 / +0.050 IoU. This reinforces the benefit of strong self-supervised global features even on a relatively clean, RGB crack dataset.
2. **CNN stem can hurt.** Adding a ResNet stem in front of DINOv2 consistently lowers performance (F1 drops to 0.775/0.763). A plausible explanation is feature interference: the stem may downsample or distort cues that the ViT already captures at patch-level resolution (14×14), effectively bottlenecking the information passed to the transformer.
3. **Skips still help (a little).** Within the "DINOv2 + ResNet" setting, skip connections recover a small amount of precision (+0.027) and IoU (+0.018) over the no-skip variant, suggesting that low-level detail fusion remains useful when a CNN stem is present, albeit the overall ceiling is limited by the stem itself.
4. **Precision–recall trade.** Removing skips slightly increases recall (0.789) but lowers precision (0.777), hinting that the decoder hallucinated more positives without shallow features to "anchor" edges. The pure DINOv2 model achieves both the highest precision and recall, indicating a better global–local balance.

Implications & next steps. These results support the hypothesis that the ResNet stem may be *constraining* DINOv2 rather than helping it. Two concrete follow-ups are suggested:

- **Skip from ViT, not CNN.** Instead of CNN skips, expose intermediate ViT block features (multi-scale token maps or attention pyramids) directly to the decoder to recover fine detail without re-introducing a CNN bottleneck.
- **Ablate fusion adapters.** Insert lightweight adapters (e.g., 1×1 conv or MLP) to better align the statistics of any CNN stem with the ViT embeddings, testing whether the drop is due to feature mismatch rather than architectural redundancy.

Wrapping up ablation studies Finally, we caution that all numbers are single-seed runs; repeating with multiple random seeds would firm up confidence intervals. Nonetheless, the ranking is clear: *if labels are noisy or not,, a pure DINOv2 backbone paired with a lightweight decoder is the most effective configuration among those tested.*

5.4 NAIVE DATASET GENERATION (THRESHOLDING APPROACH)

Before investing in SAM2 we tested a trivial label-generation scheme: global gray-value thresholding followed by morphological opening (§4.4.1). Table 5.6 contrasts the resulting model against our stronger baselines on the hand-labelled benchmark.

Variant	**Training split**	**Prec.** ↑	**Recall** ↑	**F1** ↑	**IoU** ↑
TU + ResNet + DINOv2 (full) *threshold labels*	Thresholded	0.403	0.113	0.149	0.107
TU + ResNet + DINOv2 (full)	SAM2	0.366	0.277	0.283	0.221
TU + ResNet + DINOv2 (full)	DOS-orig	0.428	0.267	0.309	0.242
Baseline TU	DOS-orig	**0.454**	**0.300**	**0.342**	**0.278**

Table 5.6: Performance on the hand-labelled test set (tolerant metric, $r = 10\,\text{px}$). Best numbers are **bold**.

Conclusion The naïve threshold–opening pipeline falls well short of every learned alternative (-15–20 pp F1). Its extremely low recall confirms that fixed heuristics cannot capture the varied crack appearance found in practice, validating our choice to pursue SAM2-assisted relabelling instead.

5.5 QUALITATIVE ANALYSIS

5.5.1 OBJECTIVE

We visually compare the predictions of the best architecture (TU + Pure DINOv2) when it is (1) trained on the *original DOS* labels and (2) trained on the *SAM2-improved* labels. The goal is to illuminate *how* SAM2's relabelling changes model behaviour, e.g. whether it removes omissions, introduces over-segmentation, or shifts the type of false positives, thereby complementing the headline scores of §5.2.

5.5.2 VISUAL PROTOCOL

All qualitative examples are drawn *exclusively* from the hand-labeled test set. We first rendered the analytic plots (coloured plots where the RGB channels represent, False Positive, True Positive and False Negative respectively) for **every** 224×1120 patch ($5 \times (224 \times 224)$ patch stitched together horizontally) and inspected them side–by–side for the two training regimes (DOS vs. SAM2 supervision). During inspection we tagged each patch with concise error descriptors (`FP{flood`, `hairline{miss`, `thick{mask`, `neutral`, `recall_↑/FP_↑`) and then clustered patches that shared the same dominant behavior.

Five consistent groups emerged:

1. **FP-flooding from SAM2 over-segmentation** (plots 3, 6): SAM2 training floods large background areas that DOS leaves untouched.
2. **Hairline-crack omissions** (plot 7): both models miss small 2-3 pixel–wide cracks, yielding high FN.
3. **Thick-label penalty cases** (plot 10): the expert mask is much wider than the physical crack, so both predictions are penalised.
4. **SAM2 adds no extra detail** (plots 19, 28): DOS already captures the crack; SAM2 neither helps nor hurts.
5. **Recall boost with FP trade-off** (plots 22, 40): SAM2 recovers additional crack fragments but introduces extra false positives.

From each cluster we selected *two* representative patches (listed in parentheses) to form the mosaic grids in Figure 5.1. This purposive sampling favors interpretability over statistical completeness;

nonetheless, the five patterns cover a majority of all tagged patches, so they characterize the dominant qualitative differences between the two training regimes.

5.5.3 VISUAL ANALYSIS

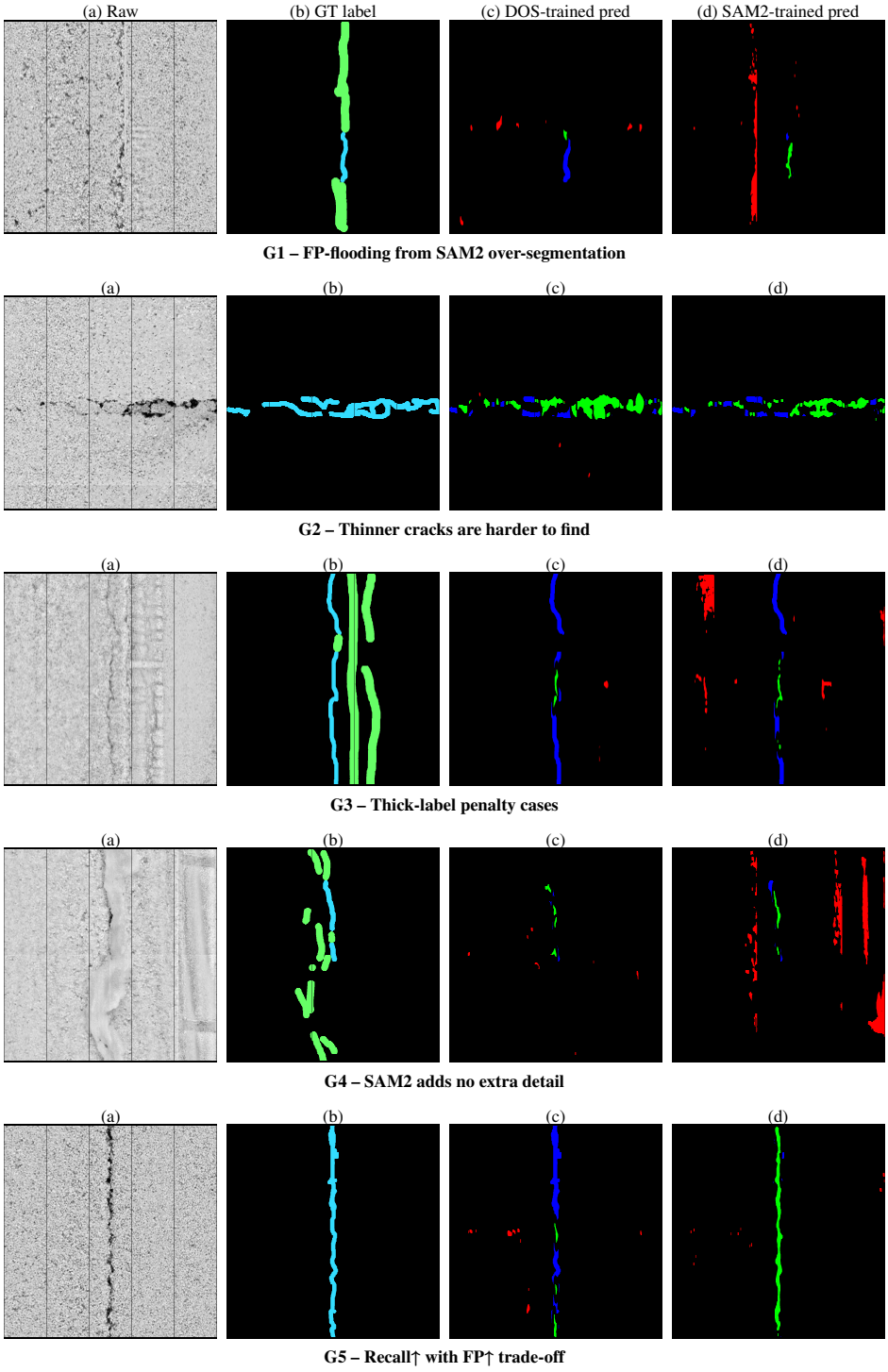


Figure 5.1: One representative patch per qualitative category (G1–G5). Green = TP, red = FP, blue = FN. See §5.5.2 for the selection protocol.

In Figure 5.1 the **5 distinct** groups that are observed are shown. This subsection will discuss key-points from each group separately and conclusions are drawn along the way. The full analysis can be found where the results from this qualitative analysis are combined with those from the quantitative analysis in either Chapter 6 or in § 5.6.

G1: FP–flooding caused by SAM2 over-segmentation. Figure 5.1 (top row) contrasts the same road patch under three supervision regimes. Panel **(b)** shows the hand-labelled mask, where **blue** pixels denote *confirmed* cracks and **green** pixels mark *uncertain* regions[1].

With **DOS**-trained weights (panel **(c)**) the network behaves conservatively: it recovers part of the true crack (**TP**) while introducing only a few small false–positive speckles (**FP**).

Training instead on the **SAM2** masks (panel **(d)**) inverts this pattern. The model now fires almost continuously along the vertical stripe, a behavior inherited from SAM2's frequent *over-segmentation* of faint sensor streaks. Recall on the true crack increases (more (**TP**) but at the expense of a flood of **FP**, which overwhelms precision.

The visual evidence therefore explains the quantitative drop in precision observed for this error category: SAM2 supervision biases the model toward over-predicting artifacts that resemble cracks, whereas DOS supervision remains more restrained due to the more reserved labels present in the DOS data.

G2: finer cracks remain elusive. Panel **(b)** depicts the hand-labeled mask for a nearly small pixel count–wide transverse crack. Both supervision regimes struggle to recover its full extent:

- **DOS-trained model (c).** Roughly half of the crack is retrieved (**TP**) but extensive **FN** gaps indicate that the network fails whenever the signal narrows to a single pixel or is partially occluded by sensor noise. A few isolated **FP** speckles appear where loose aggregates in the raw image resemble tiny cavities.
- **SAM2-trained model (d).** The recall pattern is almost identical: the same thin segments are missed. On the positive side, FP speckles disappear, suggesting that SAM2 supervision teaches the model to be more conservative on small blob-like artifacts. Yet this comes at no clear recall benefit, the long, hair-line portions are still **unsegmented**. Take note, that our introduced metrics also takes part in the role of effecting the results due to morphological opening operations performed to clean up small noise blobs which could be these smaller 1-2px fine cracks.

The failure mode highlights a fundamental limitation shared by both training sets: neither provides enough examples of small-count-pixel cracks with accurately thinned masks. Consequently the decoder learns a bias towards two–to three-pixel-wide strokes and overlooks truly slender fissures unless they are reinforced by strong local contrast. Future work could mitigate this via *label thinning* or explicit *super-resolution* decoding stages that preserve crack continuity below the native patch resolution.

G3: Thick–label penalty cases. In this patch the hand annotation (**b**) was brushed with a ≈ 10-px diameter, producing a *corridor* of ground truth that far exceeds the physical crack width visible in the raw scan (**a**). Under the tolerant-disk metric ($r = 10$ px) this has two side-effects:

- **Inflated FN count.** Both models faithfully trace the true, finer crack, yet the FN band dominates because every pixel inside the thick label that is not predicted as crack is counted as a miss. This is particularly evident for the DOS-trained network (**c**), where a single-pixel ridge is correctly segmented but still scored as largely false negative.
- **FP artifacts for SAM2 training.** The SAM2-supervised model (**d**) inherits the tendency, showcased in G1, to *over-complete* narrow structures. It fills gaps around the crack core (more **TP**), but also spills red **FP** paint wherever the thick annotation encroaches on textured, non-crack pavement. This underscores how SAM2 masks can bias the network towards aggressive, context-agnostic filling.

[1] Uncertain (`unknown`) pixels are ignored during metric computation; they neither contribute to TP, FP, TN, nor FN.

What appears as severe model failure in the confusion map is largely a *label–format mismatch*: a crack that is one–to–two pixels wide is judged against a ten-pixel template. Future evaluations should thin hand-annotations (or reduce r) to avoid penalizing geometrically accurate predictions, or alternatively adopt boundary-aware metrics that down-weight interior pixels of bloated masks.

G4: SAM2 adds no extra detail The raw patch (**a**) shows a moderately wide longitudinal crack whose edges are somewhat ragged. The hand mask (**b**) marks the main crack body with a mix of *certain* (blue) and *uncertain* (green) pixels but is otherwise free of spurious annotations.

- **DOS-trained model (c).** The network predicts only a few connected pixels. Most fall squarely on the true crack (**TP**), leaving very few false positives. Precision is therefore high, albeit at the cost of limited recall.
- **SAM2-trained model (d).** Contrary to expectations, the SAM2 variant recovers virtually no additional crack pixels beyond those already detected by the DOS model. Instead, it introduces numerous vertical streaks of false positives that follow irrelevant texture seams. These artifacts mirror over-segmentation patterns occasionally present in the SAM2 training masks.

When the DOS supervision already captures the essential crack region, SAM2 offers little benefit and can even degrade precision by encouraging context-blind fill-in behavior. The added value of SAM2 therefore appears case-dependent: helpful for bridging gaps (cf. G1, G5), but potentially harmful where the original labels are already adequate or atleast, comparatively.

G5: Recall↑ with FP↑ trade-off The patch in **(a)** contains a well-defined longitudinal crack that extends almost the full image height. The hand mask **(b)** marks the entire fissure as *certain crack* (blue) without ambiguous regions.

- **DOS-trained model (c).** The prediction only overlaps the ground truth in a few short segments (**TP**, green) and misses the majority of the crack (**FN**, blue). A handful of isolated specks away from the crack appear as false positives (red). Overall, the model is conservative: high precision but very low recall.
- **SAM2-trained model (d).** Here the network traces the crack almost continuously, converting most former FN pixels into TP (green) and thus **greatly boosting recall**. The improvement comes at a cost: the prediction also hallucinates additional vertical streaks and small blobs in the background, inflating the false–positive count. Precision therefore drops, but the F1 score still rises because the recall gain outweighs the extra FP under the tolerant evaluation metric.

SAM2 supervision can *recover near-complete crack extent* on well-delineated fissures, validating its utility for recall-critical applications. However, the accompanying FP increase underscores the need for post-processing or a stricter evaluation regime when precision is paramount.

5.6 Chapter Summary

- **Cleaner labels first, architecture second.** SAM2 relabelling delivers the largest single boost when the evaluation domain matches the training domain; architectural upgrades (ResNet stem, DINOv2 backbone, skip connections) add a further ≈2 pp F1/IoU once label noise is reduced.
- **Choose the backbone to match label quality.** A lightweight, *Pure-DINO* encoder is most robust under noisy DOS supervision, while the full ResNet + DINOv2 + skip hybrid excels on cleaner SAM2 data.
- **Architecture isolation on Crack500.** Re-training all variants on the clean Crack500 benchmark (§5.3.3) showed that a *pure* DINOv2 backbone with a TU decoder achieves the best F1/IoU (0.851/0.762), the ResNet stem *hurts* performance, and skip connections give only marginal gains, confirming that strong self-supervised ViT features are sufficient when label noise is minimal to none-exisitent.
- **Residual challenges.** Hair-line cracks remain poorly recalled and thick hand-drawn masks can penalize otherwise accurate predictions, signaling the need for finer-grained annotation and boundary-aware metrics.
- **Practitioner takeaway.** Invest in semi-automatic relabeling (e.g. SAM2) before increasing model complexity; select a recall-oriented hybrid decoder when labels are reliable, and a precision-oriented ViT-only model when label noise is high.

6 Discussion & Conclusion

This chapter reflects on the work presented in the preceding chapters, answers the research questions posed in §1.2, evaluates the strength of the evidence, and candidly describes the limitations that accompany the results. We close with directions for future work.

6.1 What This Thesis Did

In response to the opaque, low-fidelity crack masks produced by the Pavemetrics laser-line system with some post-processing done by the DOS pipeline. (Chapter 3), this thesis:

1. Curated a large corpus of laser-derived strip imagery and noisy auto-generated (*DOS*) crack masks; designed a robust patch extraction, de-duplication, and deterministic split scheme that prevents spatial leakage (§3.5).
2. Produced *refined* training targets at scale by prompting Segment Anything 2 (SAM2) with low-resolution DOS masks (§4.4.2); yielding a paired dataset of identical RGB patches with two label variants (DOS-orig vs. SAM2-refined; §3.7).
3. Collected an *expert hand-labeled benchmark* with three classes (crack/unknown/background) and a group-wise evaluation protocol to aggregate sub-patch predictions into mother-patch metrics (§3.6, §4.6).
4. Integrated and fine-tuned a self-supervised DINOv2 ViT-B/14 backbone into a TransUNet-style encoder–decoder where we trained the decoder from scratch (§4.5), exploiting pretrained global context while retaining shallow convolutions for fine structure.
5. Developed tolerant, uncertainty-aware evaluation metrics (disk-kernel counting; unknown-class masking) suitable for noisy / incomplete ground truth (§4.6).
6. Benchmarked multiple training regimes (DOS vs. SAM2 supervision) and architectures (baseline TransUNet vs. DINOv2 variants) across three evaluation splits, including the expert benchmark (§5.3).

6.2 Answering the Research Questions

This thesis consisted of one encompassing research question which was subdivided into two research question that were tackled during this thesis. The main research question was formulated as such:

How can a semantic segmentation model be designed and implemented to reliably detect and segment fine-grained structures in heightmap data, given the practical constraints of imperfect groundtruth and large-scale data?

The following subsections will tackle answering the subquestions outlined in the introduction and encompassed within the main research question.

RQ1: How can DINOv2 be effectively integrated into existing segmentation backbones to enhance fine-grained crack detection?

Integration strategy. We replaced the standard Transformer encoder in TransUNet with a pretrained DINOv2 ViT-B/14 feature extractor and added shallow convolutional stem layers plus skip fusions to reinject high-resolution spatial detail (as done in the original TransUNet) (§4.5.3). A lightweight decoder (UNet-style upsampling + fusion) converts token features to pixel logits. Integrating a DINOv2 ViT-B/14 Transformer was hypothesized to have better global context extraction with more fine-grained crack masks due to the higher patch count per image versus a ViT-B/16.

Effect on performance. When trained on the same supervision, DINOv2-backboned models consistently matched or exceeded the baseline TransUNet across automatic test splits and, critically, achieved the *best* scores on the expert hand-labeled benchmark (Table 5.1): $\mathrm{F1} = 0.376$, $\mathrm{IoU} = 0.315$ vs. baseline $0.342/0.278$ under identical training data (*Orig. train/val*). Gains persisted, though were

smaller, on the SAM2 refinement split. These results indicate that rich self-supervised ViT features can improve generalization to out-of-domain labels (expert masks) even when absolute label noise remains high.

Architecture isolation on CRACK500. To decouple architecture from label noise, we re-trained all variants on the clean CRACK500 benchmark. A *pure* DINOv2 backbone with the TU decoder achieved the strongest scores (F1 $= 0.851$, IoU$= 0.762$), surpassing the baseline TransUNet (0.810/0.712). Re-introducing a ResNet stem *reduced* performance (0.775/0.662 with skips; 0.763/0.644 without), suggesting the stem can bottleneck the ViT features when labels are clean. Skip connections helped slightly but did not close the gap to the pure-DINO variant. These findings support the hypothesis that, once supervision quality is high, DINOv2's pretrained representation is sufficient and additional CNN stems may be unnecessary or even detrimental.

Take-away. Fine-tuning a large self-supervised ViT and adapting its multi-scale tokens through a thin decoder is a practical path to leveraging global context for slender, low-contrast structures such as cracks, provided that high-frequency detail is reintroduced via skips. *On clean labels, however, a pure DINOv2 encoder already excels, and extra CNN stems can be counterproductive, use them only when noisy supervision demands additional low-level guidance.*

RQ2: WHAT TRAINING AND EVALUATION METHODOLOGIES IMPROVE PERFORMANCE WHEN LABELS UNDER-REPRESENT CRACK PIXELS?

Label refinement at scale. Prompting SAM2 with low-resolution DOS masks produced thinner, better-aligned pseudo-labels without manual interaction (§4.4.2). Training on these refined labels yielded improved scores *when evaluated on SAM2-style ground truth* (Table 5.1, rows 2 vs. 4), confirming that the refinement pipeline increases usable signal. SAM2 refined masks are *not* the best because, most likely, the train-test domains do not align. The DOS data more closely resemble the expert hand-labeled dataset, biasing the results in favor of the raw DOS dataset even when in general the DOS dataset under segments cracks. This points us towards further refining the SAM2 data refinement pipeline to remove over segmentation and in general refining its output.

Unknown-aware metrics. For hand labels we introduced an "Unknown" class that is ignored during metric computation (`ignore_index`), preventing ambiguous regions from biasing scores (§3.6). Most of the time the expert labeler was not sure if a region of pavement was either crack or something else, it could be either or, and this should not influence the metric calculation. A disk-kernel tolerance (r=10px) reduces sensitivity to pixel-level misalignment that arises from broad annotation brushes (§4.6). Since the exact pixel perfect prediction does not necessarily matter compared to the crack being detected in the general vicinity, this method was developed to more fairly judge crack prediction. Although, this method does punish models that can/may predict finer grained cracks (models trained on SAM2 data) since the morphological opening operations being conducted on the predictions may remove fine-grained crack pixels, which can results in an even more biased result towards the thicker more coarse DOS labels.

Group-wise aggregation. Sub-patch predictions are summed in confusion-space before computing precision/recall/F1/IoU, ensuring that metrics reflect the entire crack extent within a mother patch rather than arbitrary tile boundaries (§3.6.2).

Take-away. A combination of (i) scalable label refinement, (ii) uncertainty-aware evaluation (unknown masking + distance tolerance), and (iii) hierarchical metric aggregation provides a workable recipe for training and fairly assessing models under incomplete / noisy crack annotations. Although, as stated, these methods currently hold some inherent bias towards the raw DOS data because of the immaturity of the methods development. Further research needs to be done on SAM2 refinement and more fine-grained expert labels need to be annotated to represent the better crack masks that SAM2 produces.

6.3 Interpreting the Results

Three high-level patterns emerged (cf. Table 5.1):

1. **Label quality dominates within-domain testing.** Models trained and tested on the same label type perform best on that domain (DOS→DOS, SAM2→SAM2, DOS→Expert labels). Cleaner supervision yields immediate gains without architectural change. This is due to the data characteristics of the SAM2 and DOS dataset being so different due to the current implementation of the SAM2 refinement pipeline producing higly over-segmented crack masks. In further research this can be mitigated with a less naive, more novel mask generation/selection strategy. This should also close the gap cross dataset domains.
2. **Architecture helps cross noisy/clean label domains.** The DINOv2 variant shows the largest relative improvement on the expert benchmark, suggesting that stronger pretrained features regularize against label noise and domain shift. This is most likely due to DINOv2 ViT-B/14s higher resolution with patch size of 14×14 pixels versus ViT-B/16s 16×16 pixels patches. This high resolution achieves better/finer crack delineation. Together with DINOv2's self-supervised training regime translating to better domain agnostic segmentation performance. This should explain the higher relative performance increase.
3. **Manual benchmark remains hard.** All models drop sharply on the expert set, underscoring the gap between auto-generated supervision and human judgment. This, most likely is happening because of the abrupt domain switch between train-test splits, the hand labeled test set is simply to coarse and to far removed from either the DOS or SAM2-refined dataset. Even so, the DINOv2 model closes part of the gap and produces qualitatively crisper, less over-dilated masks (§5.5).

6.4 LIMITATIONS

The study has several important caveats that should temper interpretation:

1. **Noisy foundation labels.** DOS masks are coarse, incomplete, and generated by an unrevealed proprietary pipeline; residual errors persist even after SAM2 refinement. Having no insight in the most fundamental part of the data generation process is a crutch this research has not found a definitive resolution for. Fine-tuning the SAM2-refinement method is for now hypothesized as the best bet for dealing with this foundational issue.
2. **Small expert test set.** Only 127 mother patches were annotated; In our DOS dataset split the test set consisted of 15% of the dataset, this was 387 images. We now have only 127 which is not even half of what we previously have. Also, the cracks for the test set need to span a broad range of crack scenarios which, for now, was not the focus during hand annotation.
3. **Thick hand annotations.** The expert used a 10–20+ px brush (possibly wider for the *unknown* class), inflating crack width; tolerant metrics reduce but do not eliminate this bias. This wide crack annotation biases the expert hand labeled set even more towards the original DOS data because these auto-generated labels were also over dilating cracks while not being precise in the annotation of fine-grained crack pixels, just like the hand labeled set.
4. **Single sensor domain.** All imagery stems from one laser-line platform and one pre-processing stack (depth flattening, clipping). Cross-sensor generalization is untested.
5. **Patch-centric training.** Models see cropped, crack-centric tiles; In the pre-processing pipeline outlined in the thesis, and also in the DOS pipeline, cracks are mostly centered in the extracted patches. But this presents a bias towards center pixels during inference time. The model has been trained on data where cracks tend to skew towards the center region of patches. Whilst cracks can occur in various different parts of a patch. This is something needed to be taken into account when reviewing results and thinking about viability.
6. **Tolerance parameter.** Scores depend on the disk radius r; smaller radii penalize coarse masks more strongly and rewards models which predict finer-grained masks. Only one radius is reported in headline tables. Because of a singular radius being used we cannot observe the impact of varying radii on the final scores and qualitatively analyze these predictions. This is something that is interesting to tackle later.
7. **Limited literature coverage.** The related-work survey (Chapter 2) was not conducted as structured as needed. The selection of DINOv2 ViT-B/14 as a backbone and the use of SAM2 as a label refinement foundation were well researched. On the other hand, the baseline selection was done to quick. A solid model was found outside of the pavement-crack detection field because of previous models tested not providing reproducible results. Because of time constraints we moved outside of the field. But looking back when writing the related work section a lot more well established models were found for crack detection which performed better than TransUNet, like CrackFormer (Liu et al., 2021) for example. (see §6.5).
8. **Dense or sparse predicitons** Dense semantic segmentation expects pixel-accurate supervision; our DOS and SAM2 masks are coarse, incomplete, and unevenly dilated. Given the abundance of strip-level detections but scarcity of reliable fine masks, a bounding-box or centreline detection formulation may be better aligned with available supervision and downstream maintenance needs. The fact that a tolerant distance-based disk metric was needed already hints we are grading models more like localization than exact segmentation, supporting a future shift to detection.

6.5 On the Literature Review

The literature study in Chapter 2 should be viewed as a targeted, practice-driven survey rather than a comprehensive systematic review. Time constraints, inconsistent terminology across civil and computer-vision venues, reproducibilty and other issues made it a complicated matter to complete a comprehensive literature review that spanned the full scope of state-of-the-art crack detection models Readers seeking a broader overview should consult recent surveys of pavement distress detection and looking at well established models; incorporating such sources would strengthen a future version of this thesis.

6.6 Concluding Remarks

This thesis set out to improve crack segmentation on laser-derived road imagery in the face of noisy, proprietary auto-labels. By pairing scalable label refinement (SAM2 mask prompting) with a DINOv2-augmented TransUNet, and by evaluating with uncertainty-aware, group-wise metrics against a new expert benchmark, we demonstrated measurable, if modest, improvements in both in-domain and out-of-domain performance. The absolute numbers remain far from human reliability, yet the pipeline establishes an open, reproducible baseline and a pathway for incremental improvement: clean the labels, leverage large pretrained features, measure fairly.

The road from opaque black box to transparent, extensible crack mapping is long; this work covers the first few meters. We hope the data handling, evaluation protocol, and integration lessons documented here make the next steps faster for others.

7 Future Work & Practical Recommendations

This chapter looks beyond the present study. The first section distils *immediate, low-effort actions* that TNO can adopt today using the insights and tooling developed in this thesis. The second section outlines *research directions* that would materially strengthen crack segmentation from laser-derived pavement imagery, address the limitations cataloged in §6.4, and extend the pipeline toward operational deployment at scale.

7.1 Immediate Operational Recommendations

Even without additional research, several practical steps can improve the quality and utility of crack data products derived from laser line systems:

1. **Bulk label upgrading via promptable segmentation.** The workflow in §4.4.2 shows that noisy, over-dilated auto-labels (DOS) can be upgraded in batch by low-resolution mask prompting with SAM2, yielding thinner and more spatially aligned supervision. Asset owners should prioritize running existing archives through such a refinement pass before anything else.
2. **Leverage large ViT features when labels are scarce.** As demonstrated in §6.2 (RQ1) and Table 5.1, a pretrained DINOv2 ViT-B/14 backbone, lightly adapted with a convolutional stem and UNet decoder, provides competitive performance even under imperfect supervision. This is a pragmatic way to bootstrap new projects that lack dense, high-quality crack annotations.
3. **Deterministic data partitioning and provenance tracking.** The hash-based split in §3.5 prevents spatial leakage across train/val/test and ensures reproducibility. Store and version the mapping from original strip IDs to patch-level files; doing so guards against overly optimistic metrics and simplifies cross-study comparisons.
4. **Include an "Unknown" label in manual QA.** The three-class expert protocol (§3.6) avoids penalizing models for ambiguous pavement artifacts by masking those pixels at evaluation time. The DOS software pipeline could integrate this *unknown* class generation into their pipeline such that during training and testing the model can learn what pixels are actually usefull for predictions and which ones should be diregarded entirely.

Taken together, these four actions can materially raise data quality and the credibility of reported performance metrics with minimal engineering effort.

7.2 Research Directions

The work reported in this thesis opens several avenues for deeper study. The subsections are ranked on their immediate importance except for the last three sections. (§7.2.6, - §7.2.8)

7.2.1 Detection-First or Weakly Supervised Formulations

Dense semantic segmentation presumes pixel-level truth, yet our supervision (§3.4, §4.4.2) remains noisy and coarse. Recasting crack mapping as *object detection* (bounding boxes around crack spans) or *centreline/segment detection* could better exploit abundant but low-fidelity annotations. Candidate workflow: (i) derive boxes from DOS polylines with width margin; (ii) train lightweight detectors (, YOLO-family, RT-DETR) to localize crack instances; (iii) optionally refine detected regions to masks via promptable segmentation (SAM2) or lightweight region decoders. Such a two-stage pipeline may deliver more stable maintenance-grade outputs under weak supervision than end-to-end dense segmentation. Firstly, it should be clear what is needed in the prediction task, a pixel level mask that exactly pinpoint a crack, or a general idea if cracks exist in a localized area. From here the implementation/research can differ broadly, so care should be taken into what direction is truly best suited.

7.2.2 Open sourcing data

During the thesis this idea was thrown around, but at the end it was concretized more. The idea of open sourcing a large chunk of the raw data or processed data ran through the DOS software pipeline. During the literature study it became clear that there were not many or any annotated large scale laser-line datasets that match ours. For the entire field of pavement distress detection we think it can be of great value to release the dataset in a conference setting such that the broader field can tackle some problems for us. Or even more naively, that they will properly label a good chunk of data we can use again for evaluation or even training later down the line. We believe the broader field will benefit from the release of such a dataset with the caveat that any method developed with this data needs to be completely open source to the benefit of the scientific community.

7.2.3 Larger & Finer Expert Dataset

The current hand-labeled benchmark (§3.6) spans only 127 mother patches and employs a broad 10–20+ px brush. Using a more intelligent annotation method that could potentially help the annotator suggest regions of finer cracks. Or using an actual finer brush would tremendously boost annotation quality and would give deeper, more trustworthy insights into the quantitative and qualitative analysis done. Taking the time to label a moderate amount of labels to a fine degree is paramount for proper assessment of any developed method

7.2.4 SAM2-Refinement; Further research on the raw output

At the moment, as described in §4.4.2 the second highest IoU prediction out of the three predictions is taken in bulk across the entire dataset. This is a naive method of mask selection. A more novel way of selecting a mask can be to take the raw logits SAM2 also outputs and develop a method ontop of these raw predictions. First extract statistics across all raw predictions to try and detect patterns in the data and act on these patterns. This would be a logical first step to continue on with the SAM2 refinement pipeline. We are still adamant that SAM2 can provide major improvement across the entire dataset. Further experimentation was planned, but due to unforeseen circumstances this research had to be cut short.

7.2.5 Metric Sensitivity Studies

Headline scores used a single disk-tolerance radius ($r{=}10$ px; §4.6). Sweeping r exposes how models trade boundary precision against coverage, critical when comparing coarse DOS-style labels, SAM2 refinements, and future fine annotations. Similarly, reporting curves over probability thresholds (ODS/OIS; see §5.3) can standardize evaluation across studies. This experiment was planned, but due to lack of time it was not conducted.

7.2.6 Iterative Human-in-the-Loop Refinement

SAM2 refinement (§4.4.2) is a one-shot process. Active-learning loops could flag low-confidence or high-disagreement regions for targeted expert correction; refined masks would feed successive training rounds, progressively improving both model and pseudo-labels. This deviates a lot from an automated process and this step, if applicable time-wise, should be thought out carefully before implementing to avoid downstream problems.

7.2.7 Uncertainty Modeling & Confidence Propagation

Current outputs are binary masks. Calibrated per-pixel uncertainty, via ensembles, Monte-Carlo dropout, or temperature scaling, would enable risk-aware crack prioritization (e.g., maintenance triage) and allow metrics that weight confident errors more heavily than uncertain ones (cf. §4.6).

7.2.8 MULTI-CHANNEL/SENSOR FUSION

All experiments used single-channel laser height imagery (§3.2). Pavemetrics also capture intensity, and co-registered RGB imagery. Fusing complementary modalities may disambiguate shallow texture from true cracks and improve robustness to lighting, contamination, or wear patterns. Domain adaptation across hardware vendors (Pavemetrics vs. future TNO builds) is a related challenge (§6.4). Take note, there was an intensity channel available for the available data, but it was decided not to use it early on because it did not fit the format of popular model inputs.

7.3 CLOSING NOTE

Improving crack segmentation accuracy is only one link in a broader infrastructure chain. By (i) upgrading noisy legacy labels at scale, (ii) harnessing transferable self-supervised vision backbones, and (iii) adopting evaluation protocols that acknowledge ground-truth uncertainty, practitioners and researchers alike can shorten the path from raw survey data to defensible maintenance insight. The directions above are intended as a roadmap for that journey.

Bibliography

Willem van Aalst, Giljam Derksen, Peter-Paul Schackmann, Petra Paffen, Frank Bouman, and Wim van Ooijen. Automated ravelling inspection and maintenance planning on porous asphalt in the netherlands. In *International Symposium Non-Destructive Testing in Civil Engineering (NDTCE 2015). Berlin*, 2015.

Lev Ayzenberg, Raja Giryes, and Hayit Greenspan. Dinov2 based self supervised learning for few shot medical image segmentation. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2024.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Christian Benz and Volker Rodehorst. Omnicrack30k: A benchmark for crack segmentation and the reasonable effectiveness of transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3876–3886, 2024.

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pp. 205–218. Springer, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Dongjie Cheng, Ziyuan Qin, Zekun Jiang, Shaoting Zhang, Qicheng Lao, and Kang Li. Sam on medical images: A comprehensive study on three prompt modes. *arXiv preprint arXiv:2305.00035*, 2023.

Chu Chu, Linbing Wang, and Haocheng Xiong. A review on pavement distress and structural defects detection and quantification technologies using imaging approaches. *Journal of Traffic and Transportation Engineering (English Edition)*, 9(2):135–150, April 2022. ISSN 2095-7564. doi: 10.1016/j.jtte.2021.04.007. URL `https://www.sciencedirect.com/science/article/pii/S2095756422000010`.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. (arXiv:1604.01685), April 2016. doi: 10.48550/arXiv.1604.01685. URL `http://arxiv.org/abs/1604.01685`. arXiv:1604.01685 [cs].

Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Shuyang Gu, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Clip itself is a strong fine-tuner: Achieving 85.7 (arXiv:2212.06138), December 2022. doi: 10.48550/arXiv.2212.06138. URL `http://arxiv.org/abs/2212.06138`. arXiv:2212.06138 [cs].

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10(15):2733, 2022.

Hongren Gong, Liming Liu, Haimei Liang, Yuhui Zhou, and Lin Cong. A state-of-the-art survey of deep learning models for automated pavement crack segmentation. *International Journal of Transportation Science and Technology*, 13:44–57, March 2024. ISSN 2046-0430. doi: 10.1016/j.ijtst.2023.11.005. URL https://www.sciencedirect.com/science/article/pii/S2046043023001028.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Ju Huyan, Wei Li, Susan Tighe, Zhengchao Xu, and Junzhi Zhai. Cracku-net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Structural Control and Health Monitoring*, 27(8):e2551, 2020.

Xiao Jiang, Shanjun Mao, Mei Li, Hui Liu, Haoyuan Zhang, Shuwei Fang, Mingze Yuan, and Chi Zhang. Mfpa-net: An efficient deep learning network for automatic ground fissures extraction in uav images of the coal mining area. *International Journal of Applied Earth Observation and Geoinformation*, 114:103039, 2022.

Joint Departments of the Army and the Air Force, USA. Procedures for u.s. army and u.s. air force airfield pavement condition surveys. Technical Report TM5-826-6 / AFR93-5, U.S. Department of the Army and U.S. Department of the Air Force, Washington, DC, July 1989. Public domain; July 1989.

Narges Kheradmandi and Vida Mehranfar. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, 321:126162, February 2022. ISSN 0950-0618. doi: 10.1016/j.conbuildmat.2021.126162. URL https://www.sciencedirect.com/science/article/pii/S0950061821038940.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3041–3050, 2023.

Qingquan Li, Dejin Zhang, Qin Zou, and Hong Lin. 3d laser imaging and sparse points grouping for pavement crack detection. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2036–2040. IEEE, 2017.

Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3783–3792, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*, 2023.

OpenCV Team. Morphological transformations. https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html, 2025. Accessed: 2025-06-04.

OpenCV Team (moukthika). Resizing and rescaling images with opencv. https://opencv.org/blog/resizing-and-rescaling-images-with-opencv/, March 2025. Accessed: 2025-07-07.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Pavemetrics Systems Inc. Laser Crack Measurement System 4M (LCMS 4M). https://www.pavemetrics.com/applications/road-inspection/laser-crack-measurement-system-lcms-4m/, 2024a. Accessed: 2025-07-10; includes technical overview of features like 3D triangulation, PASER/PSCI scoring, 4m lane coverage, and automated distress analysis.

Pavemetrics Systems Inc. *LCMS Road Surface Condition Measurement System: User Processing Manual*, 2024b. URL https://www.pavemetrics.com. Version 5.x.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

Antonella Ragnoli, Maria Rosaria De Blasiis, and Alessandro Di Benedetto. Pavement distress detection methods: A review. *Infrastructures*, 3(4):58, 2018.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Dhvanil Shah. Improving remote sensing-based semantic segmentation by adapting pre-trained vision transformers for multispectral data. Master's thesis, The Cooper Union for the Advancement of Science and Art, 2024.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1415–1428. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0a9fdbb17feb6ccb7ec405cfb85222c4-Paper.pdf.

Huaqi Tao, Bingxi Liu, Jinqiang Cui, and Hong Zhang. A convolutional-transformer network for crack segmentation with boundary awareness. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 86–90, 2023. doi: 10.1109/ICIP49359.2023.10222276.

Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. (arXiv:2203.09795), March 2022. doi: 10.48550/arXiv.2203.09795. URL http://arxiv.org/abs/2203.09795. arXiv:2203.09795 [cs].

Willem van Aalst. The current dutch pavement monitoring system. In *Proceedings of the European Road Profiler User Group Symposium (ERPUG 2021)*, Vienna, Austria, 2021. Presented at ERPUG 2021, session "The current Dutch pavement monitoring system" on 11 November 2021 :contentReference[oaicite:1]index=1.

Hanguang Xiao, Li Li, Qiyuan Liu, Xiuhong Zhu, and Qihang Zhang. Transformers in medical image segmentation: A review. *Biomedical Signal Processing and Control*, 84:104791, 2023.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.

Haoyuan Zhang, Ning Chen, Mei Li, and Shanjun Mao. The crack diffusion model: An innovative diffusion-based method for pavement crack detection. *Remote Sensing*, 16(6):986, 2024.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10145–10155, 2021.

Tong Zhao, Yichen Xie, Mingyu Ding, Lei Yang, Masayoshi Tomizuka, and Yintao Wei. A road surface reconstruction dataset for autonomous driving. *Scientific data*, 11(1):459, 2024.

Lele Zheng, Jingjing Xiao, Yinghui Wang, Wangjie Wu, Zhirong Chen, Dongdong Yuan, and Wei Jiang. Deep learning-based intelligent detection of pavement distress. *Automation in Construction*, 168:105772, December 2024. ISSN 0926-5805. doi: 10.1016/j.autcon.2024.105772. URL https://www.sciencedirect.com/science/article/pii/S0926580524005089.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. (arXiv:1608.05442), October 2018. doi: 10.48550/arXiv.1608.05442. URL http://arxiv.org/abs/1608.05442. arXiv:1608.05442 [cs].

Guijie Zhu, Jiacheng Liu, Zhun Fan, Duan Yuan, Peili Ma, Meihua Wang, Weihua Sheng, and Kelvin C. P. Wang. A lightweight encoder–decoder network for automatic pavement crack detection. *Computer-Aided Civil and Infrastructure Engineering*, 39(12):1743–1765, 2024a. ISSN 1467-8667. doi: 10.1111/mice.13103. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.13103.

Guijie Zhu, Jiacheng Liu, Zhun Fan, Duan Yuan, Peili Ma, Meihua Wang, Weihua Sheng, and Kelvin CP Wang. A lightweight encoder–decoder network for automatic pavement crack detection. *Computer-Aided Civil and Infrastructure Engineering*, 39(12):1743–1765, 2024b.

Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE transactions on image processing*, 28(3):1498–1512, 2018.