电子科技大学
格拉斯哥学院
Glasgow College, UESTC

Logbook

From: 29/09/2025 To: 02/11/2025

Month	**List the main activities** (only few words per activity)	**Interaction with the supervisor**			**Any other form of supervisory interaction** (second supervisor, industry, fellows etc.)
		Number of meetings	Mode of meeting (face-to-face, online e.g., Skype, WeChat etc.)	Number of emails exchanged	
09	1. Reading roadmap & baselines (crack detection) 2. Dataset review: SDNET2018, CRACK500, SUT-Crack	3	Online (WeChat ×2; Zoom ×1)	6	1. Weekly online check-in to align scope and reading plan; clarified deliverables. 2. Advice on dataset choice (SDNET2018/CRACK500/SUT-Crack), licensing and split strategy. 3. Peer fellow helped set up a literature matrix/Zotero, Q&A on screening rubric. 4. Library/IT assisted dataset access and compute environment setup.
10	1. Deep segmentation: DeepCrack, FPHBN 2. Lightweight models:	4	Face-to-face ×1; Online (WeChat ×2;	9	1. Guidance to reproduce baselines (DeepCrack/FPHBN) with checkpoints and logs. 2. Recommended lightweight options (FPCNet/U-Net)

	FPCNet, U-Net practices		Zoom ×1)		with latency & params targets. 3. Second supervisor suggested clean train/val/test protocol and ablations. 4. Labmate reviewed U-Net pipeline; fixed augmentation & logging issues.
11	1. Detectors vs. Transformers : YOLOv8/MS-YOLOv8/EMG-YOLO/YOLOv7-WMF vs. Swin/CSW-S/iSwin-UNet 2. E2E & data governance: RT-DETR/RT-DETRv3 + label-noise handling	1	Online (WeChat ×1)	2	1. Asked to justify Detectors vs. Transformers; fixed metrics (mIoU/F1, mAP). 2. Approved E2E plan (RT-DETR/RT-DETRv3); emphasized label-noise handling & docs. 3. Industry contacts shared annotation QC checklist; discussed class imbalance. 4. Reading group shared Swin-UNet refs; tips on mixed-precision training.

2025.9.29 ~ 2025.10.5

Weekly Goal:

Establish the reading trajectory from traditional methods => classical ML => datasets, and clarify core challenges and evaluation baselines for crack dection.

Weekly Reading List:

CrackTree (traditional curve-growth method)

CrackIT (crack-dection workflow & benchmark)

Random Structured Forests (RSF) (structed edges | frests)

SDNET 2018 (multi-material crack dataset)

CRACK500 (pixel-level crack annotations)

Key Takeways:

Crack Tree: Uses "growable curves + minimum-cost path." Pipeline: contrast / texture normalization => texture suppression or guided filtering => seed initialization => path growing guided by a cost combining intensity | gradient | geometric consistency, followed by thinning.

Pros: High recall for thin, continuous cracks; can bridge small gaps.

Limits: Sensitive to transverse joints, strong specular highlights, coarse aggregate textures; parameters (thresholds, structuring elements) don't tranfer well across scenes

CrackIT: Provides an end-to-end work flow and unified evaluation (P/R, connectivity, width error, etc.), Stressing that a fixed validation set and fixed thresholds / post-processing are needed for comparability.

RSF: Treats cracks are oriented fine structures via structured edge learning; better suppresses background-texture false edges than pure threshold / morphology, but out-of-domain generalization depends on data coverage.

Datasets:

SDNET2018: Keeps shadows / stains / diverse morphologies => good for robustness testing.

CRACK500: Pixel-level labels emphasize thin-crack

continuity & small-object recall; if later converting to detection, define a consistent mask => bbox protocol (e.g., min bounding rectangle + area threshold).

Issues & Risks

Traditional methods are brittle to illumination / texture shifts; SDNET2018 vs. CRACK500 distribution gaps imply augmentation must cover strong glare and transverse textures.

Next Week Plan

Move to deep segmentation representatives (DeepCrack, FPHBN); add SAM-Crack.

2025.10.6 ~ 2025.10.12

Weekly Goal

Systematize deep segmentation ideas for thin cracks / low contrast; complete the dataset dimensions.

Weekly Reading List

DeepCrack (multi-scale side outputs + hierarchical aggregation)

FPHBN (feature pyramid + hard-example re-weighting)

SUT-Crack (complex-condition crack dataset)

Key Takeaways

DeepCrack: Supervises features at multiple scales to fuse shallow texture with deep semantics; improves boundary integrity of slender cracks, but is sensitive to loss weights and pos/neg balance. Under strong textures it may spawn spurious fine edges → needs thining connected-component filtering.

FPHBN: Captures multi-scale cues and boosts

weak-contrast regions via hard-example re-weighting; more stable in complex backgrounds but adds compute, so deployment must balance latency.

SUT-Crack: Broad coverage of scenes / illumination → good source for a robustness validation set; before merging with CRACK500 / SDNET2018, unify color space / exposure to reduce domain bias.

Segmentation → Detection conversion: Connected components + min bounding rectangles give quick bboxes but lose width / orientation; keep a skeleton / width-estimation hook in the GUI to compensate later.

Issues & Risks

Multi-scale side-output performance is very sensitive to loss weighting; weak-contrast zones may over/underfit without explicit augmentation average.

Next Week Plan

Read FPCNet and U-Net practices in crack

segmentation; compile a lightweight × pixel-accuracy trade-off list.

2025.10.13 ~ 2025.10.19

Weekly Goal

Evaluate lightweight segmentation routes that retain pixel accuracy while controlling compute.

Weekly Reading List

FPCNet (dilated convolutions + lightweight attention)

U-Net family practices for crack segmentation (transfer learning | strong augmentation | lr schedules, etc.)

Key Takeaways

FPCNet: Expands receptive fields via dilated conv and strengthens fine-texture responses with lightweight attention; under offline constraints, can keep boundary detail with fewer params, but ultra-thin | fragmented cracks still benefit from morphological post-processing

U-Net practices:

Transfer learning (pretrained backbones,

markedly stabilizes convergence on small data.
Strong augmentation.
(brightness | contrast | blur | noise | random crops) helps, but must preserve morphological consistency to avoid breaking crack topology.
Reproducibility: fix random seeds; save full yaml | logs | PR curves; consistent records are essential.

Issues & Risks
Over-aggressive augmentation may distort connectivity | skeleton, hurting downstream dection and width estimation.

Next Week Plan
Shift to dection | Transformer: YOLOv8 variants, YOLOv7-WMF, and Swin | CSW-S |; Swin-UNet; contrast real-time vs. global representation.

2025.10.20 ~ 2025.10.26

Weekly Goal

Contrast real-time detection (YOLO family) vs. global representation (Transformer family) and identify fit points.

Weekly Reading List

Improved YOLOv8 (small-object / texture-oriented variants)
MS-YOLOv8 (multi-scale feature enhancement,
EMG-YOLO (multi-granularity fusion)
YOLOv7-WMF (mask-guided fusion; width-friendly)
Swin-Transformer / CSW-s / iSwin-UNet (global attention / hierarchical Transformers in crack tasks)

Key Takeaways

YOLOv8 variants: Multi-scale / multi-granularity paths boost small-object recall; light weight attention helps separate texture noise vs. true cracks. However, results are highly sensitive to NMS / confidence thresholds; under transverse textures or

glare, FPs rise → require connected-component + morphology as secondary filtering.

YOLOv7-WMF: Mask-guided fusion enhances geometric consistency and naturally supports width/area stats in the GUI; training is more complex and demands label consistency.

Transformers (Swin / CSW-S / Swin-UNet): Global attention reduces false detections and bridges broken segments, but with higher cost/inference cost and sensitivity to data.

Issues & Risks

YOLO pipelines hinge on threshold choices; Transformers have higher training barriers and overfitting risk.

Next Week Plan

Read RT-DETR / RT-DETRv3 (end-to-end real time) and noise-label governance papers to form an "E2E real-time + data-quality" strategy set.

2025.10.27 ~ 2025.11.2

Weekly Goal

Focus on end-to-end real-time (reducing unstable post-processing, and label-noise governance to prepare for November implementation.

Weekly Reading List

RT-DETR (end-to-end real-time detection)

RT-DETRv3 (hybrid encoder | dense supervision improvements)

Noise-label | weakly-supervised reweighting (distribution-aware, confidence decay, etc.)

Crack detection | segmentation surveys (method taxonomy + datasets | evaluation protocols)

Key Takeaways

RT-DETR family: Moves toward no-NMS end-to-end pipelines, simplifying matching | inference and reducing threshold fragility-friendly for offline batch GUI. But it's more sensitive to data organization and

training strategy; safer to proceed after the environment is stable.

Noise-label governance: Crack data often suffer mislabelling | omissions | boundary jitter; prioritize cleaning | resampling | sample reweighting | consistency regularization - these can outperform pure model tweaks

Strategy: Provide a consensus method taxonomy and data set | evaluation map that can be ported to mid-term slides and experimental records.

Issues & Risks

E2E migration needs stable operator support and export paths (ONNX / TensorRT). Noise strategies must match the data set's actual distortion / mislabel distribution - avoid copy-pasting.

Next Week Plan

Standardize CUDA / driver / PyTorch; run a dataloader dry-run; no training, and full GW I/O check; refine

the post-processing thresholds and morphological filtering plan based on this mask's readings.