

Vision Transformer Ensembles for Panoramic Street Segmentation

Yunus Serhat Bıçakçı 

Department of Artificial Intelligence and Machine Learning
Faculty of Applied Sciences, Marmara University
Istanbul, Türkiye

Geospatial Data Science Group
School of Geographical and Earth Sciences
University of Glasgow, Glasgow, UK

5 October 2026

Abstract

Semantic segmentation of street panoramas can support detailed descriptions of urban environments, yet small datasets and unequal training costs make model selection difficult. This paper presents the system used for a first place submission to the PalmCity challenge in the leaderboard snapshot dated 5 October 2026. Nine pretrained segmentation systems are compared using approximately equal computation budgets. The candidates include DeepLabV3+, SegFormer, UPerNet, Mask2Former, DINOv3 with a linear decoder, and an Encoder only Mask Transformer using DINOv3. The two leading candidates are trained independently with three random seeds and longer budgets. Equal averaging of class probabilities from the three Encoder only Mask Transformer models, evaluated at three image scales with horizontal reflection, produces 60.95% mean intersection over union and 71.16% mean F1 on the 84 image public validation split. The submitted predictions receive 57.08% mean intersection over union and 67.96% mean F1 on the hidden test leaderboard. Producing all 249 test masks takes 251.49 seconds including model initialization and provenance checks on one NVIDIA RTX 5090. Peak allocated GPU memory is 2.70 GiB. The study reports all eligible models, all inference variants, class level errors, source conditions, and reproducibility checks, providing a documented challenge workflow with existing architectures.

1 Introduction

Urban environments are experienced from the street. A map may show that a park, a road, or a building exists, while an image taken at pedestrian height reveals how vegetation, traffic, footpaths, street furniture, and physical barriers share the same view. These details matter when describing the quality and accessibility of public space. Street imagery has therefore become a useful complement to remote sensing and conventional spatial data. The review by Biljecki and Ito describes its applications across urban planning, transport, environmental assessment, and related fields [1]. The growing variety of these applications creates a practical need for segmentation systems whose output can be inspected, reproduced, and interpreted at the class level.

The connection between visible surroundings and urban outcomes requires care. Helbich and colleagues examined associations between green and blue space measured from street imagery and depression symptoms among older adults in Beijing [2]. Their later Amsterdam study compared overhead and eye level greenness measures and found no statistically significant association with depression or anxiety scores [3]. The studies show that both the measurement perspective and the setting matter. A street facing camera and an overhead sensor do not necessarily represent the same exposure. At the same time, a segmentation mask is only one component of an environmental

measurement process. Sampling locations, image dates, geographical coverage, model errors, and the design of any subsequent health study also affect its conclusions.

Street imagery can support more specific analyses when its visual categories are linked to a clearly defined question. Ito and Biljecki used street imagery and computer vision to assess bikeability [4]. Recent work in the Fatih district of Istanbul used vision language descriptions and multimodal embeddings from 116696 Mapillary images to identify interpretable urban typologies without manual labels [5]. This provides another example of extracting spatially meaningful information from street observations. These applications motivate reliable visual measurements. PalmCity supplies semantic labels for image regions, and the present evaluation concerns the accuracy and cost of recovering those labels.

Much of the public evidence used to develop computer vision systems comes from a limited set of places and imaging conditions. A model that works well on one benchmark may encounter different streets, vehicles, surfaces, vegetation, or camera geometry elsewhere. PalmCity addresses part of this gap through panoramic street imagery captured in Mersin, Türkiye, and a benchmark motivated by underrepresented developing countries [6]. Its 32 categories cover both large regions such as road, building, sky, and vegetation, and small objects such as signs, traffic lights, benches, and infrastructure components. Panoramic imagery also places wide context and fine details within the same image. A useful challenge system must handle both.

The size of the labelled training split creates a second difficulty. There are only 497 training panoramas. Modern pretrained networks offer powerful visual features, but their performance after adaptation depends on the original training data, decoder, optimization settings, and the amount of computation available. Simply training every candidate for the same number of epochs gives slower systems a larger computation allowance. Conversely, a very short common schedule may fail to reveal the value of a large model. An evaluation that records both quality and cost is more informative than one that reports only the largest validation score.

Recent segmentation systems provide several plausible routes. DeepLabV3+ combines convolutional features with a decoder for spatial detail [7]. SegFormer uses a hierarchical transformer and a lightweight segmentation head [8]. UPerNet combines features at several resolutions [9], and can be paired with Swin Transformer [10] or ConvNeXt [11]. Mask2Former frames segmentation through learned mask queries [12]. DINOv3 supplies visual representations that can be adapted to dense prediction [13]. The Encoder only Mask Transformer, abbreviated EoMT, places mask queries within a vision transformer and provides an additional candidate [14]. Choosing among these alternatives is an empirical question under the resources available for the challenge.

Prediction fusion offers another practical choice after selecting a base model. Lakshminarayanan and colleagues describe deep ensembles that average predicted probabilities from independently trained neural networks [15]. Fusion takes different forms across vision tasks. In a 2026 preprint, Bıçakçı combines CNN and transformer detections in aerial imagery through Weighted Boxes Fusion [16]. Ozturk and colleagues average the softmax outputs of four CNN architectures for tea leaf disease classification [17]. In the selected PalmCity ensemble, members share one architecture and differ in their training seed. Their aligned class probabilities are averaged at each panorama pixel. The benefit is assessed through PalmCity experiments against individual models and alternative inference settings.

This paper follows the complete path from candidate selection to the submitted mask archive. A pilot stage compares nine systems using an estimated 600 seconds of training and validation computation per system. The two leading systems receive an estimated 2400 seconds each for three independent random seeds. A fixed set of eleven inference variants then compares individual checkpoints, probability ensembles, image scales, horizontal reflection, and overlapping windows. The final choice uses the official mean intersection over union, with class level performance, runtime, and memory providing additional evidence.

The resulting submission is based on three EoMT models with DINOv3 ViT L features. On the public validation split it reaches 60.95% mean intersection over union. The hidden test score

shown on Codabench is 57.08%, which placed the submission first in the leaderboard snapshot on 5 October 2026. The measured selection procedure is the central contribution. The published artifacts retain unsuccessful alternatives and classes on which the selected system remains weak. This emphasis on transparent and reusable procedures is consistent with recent recommendations for repeatable, reproducible, and expandable geospatial analysis [18].

2 Benchmark and evaluation

2.1 Data and split checks

The experiments use the official PalmCity split without modification. Table 1 gives its size. Every panorama has a width of 1024 pixels and a height of 512 pixels. Training and validation annotations are single channel images with direct integer class identifiers from 0 to 31. The labels are preserved throughout loading, training, evaluation, and archive construction. No label reduction inherited from an ADE20K configuration is applied.

Table 1: Official split used in the experiments. Hidden test annotations were not available to the training or selection procedure.

Split	RGB panoramas	Available masks
Training	497	497
Validation	84	84
Hidden test	249	0

The dataset audit checks file counts, image modes, dimensions, class ranges, and exact duplicates across splits. All 32 categories occur in both training and validation. No identical decoded RGB panorama occurs in more than one split. Some visually similar views nevertheless occur across the official training and validation splits. One inspected pair shows the same coastal walkway and nearby buildings from different viewpoints. This distinction between file separation and geographical separation is considered in the discussion.

Class support is highly unequal. The eight least frequent classes, selected from the training pixel histogram before the inference comparison, are Pruned Tree, Traffic Light, Driver, Bicycle, Traffic Sign, Sitting Bench, Railing, and Overpass. Their average IoU is reported as an additional diagnostic. It is not a substitute for the official metric. In validation, Overpass occupies only 40 labelled pixels in one image and Pruned Tree occupies 54 pixels in one image. Traffic Light is also present in only one validation image. These categories can have unstable scores even when the larger scene regions are predicted well.

2.2 Exact scoring convention

Evaluation accumulates a single confusion matrix over all pixels in the relevant split. Let C_{ij} be the number of pixels whose true class is i and predicted class is j . For class c , define $T_c = C_{cc}$, $P_c = \sum_i C_{ic}$, and $G_c = \sum_j C_{cj}$. The class scores are

$$\text{IoU}_c = \frac{T_c}{P_c + G_c - T_c}, \quad \text{F1}_c = \frac{2T_c}{P_c + G_c}. \quad (1)$$

A score with a zero denominator is set to zero. Mean IoU and mean F1 are the arithmetic means over all 32 classes. Class 31, named Void, remains part of both training and evaluation. A class absent from both the truth and the predictions contributes zero rather than being omitted. The metric is not an average of image specific mean IoU values.

This detail matters when comparing results with implementations that exclude Void or ignore absent categories. The local scorer was checked against the published PalmCity scoring program

on 103 synthetic confusion matrices, with exact agreement. The eleven validation output sets were also independently decoded and scored using dataset level confusion matrices. This establishes agreement with the published scorer. It does not claim that the local code is byte identical to the private Codabench evaluation package.

3 Candidate systems and transfer sources

The reference systems are DeepLabV3+ with ResNet50 and SegFormer with MiT B2. The larger candidates are SegFormer B5, UPerNet with ConvNeXt Large or Swin Large, Mask2Former with Swin Large, DINOv3 ViT B or ViT L with a linear segmentation head, and EoMT with DINOv3 ViT L. Table 3 lists the eligible results.

The initialization sources differ. The ResNet50 and MiT B2 reference encoders use ImageNet weights with newly initialized segmentation heads. SegFormer B5, UPerNet, Mask2Former, and EoMT start from complete semantic segmentation checkpoints trained on ADE20K [19]. The DINOv3 linear systems use pretrained DINOv3 encoders and a newly initialized segmentation head. Every PalmCity class prediction layer is adapted to 32 categories. All encoders are updated during PalmCity training.

The linear DINOv3 decoder provides a simple way to test the usefulness of the representation. The final normalized patch features are reshaped to the actual image grid and passed through dropout, batch normalization, and a 1×1 convolution. Predictions are interpolated to the input size. Class and register tokens are excluded from the patch grid.

EoMT is an especially relevant addition because it couples strong pretrained features with mask prediction in the encoder. The implementation uses the available DINOv3 ViT L ADE20K semantic checkpoint from the official model release [20]. This later release is distinguished from the original EoMT paper, which evaluates DINOv2 based systems. Its mask reshape grid is derived from the actual rectangular image dimensions on each forward pass to preserve the panorama layout. The training schedule records the decay of query attention masks. These masks are disabled during validation and inference, following the intended evaluation behavior.

The selected EoMT configuration has 24 transformer layers, a feature width of 1024, 16 attention heads, and patches of 16×16 pixels. A native panorama therefore supplies a 32×64 patch grid. One hundred learned mask queries are used in the final four transformer blocks. The query classification, mask, and Dice loss weights are 2, 5, and 5, with a no object weight of 0.1. Query attention mask probabilities decrease polynomially over four recorded update intervals, spanning fractions 0 to 0.2, 0.4 to 0.6, 0.6 to 0.8, and 0.8 to 1 of the training budget. The exponent is 0.9. These settings are included in the released configuration.

Native losses are retained when they are integral to a system. UPerNet uses cross entropy with its auxiliary supervision. Mask2Former and EoMT use their native mask and class objectives with query matching. DeepLabV3+, SegFormer, and the linear DINOv3 systems use cross entropy with a Dice term. For query systems, semantic targets contain one binary mask per class present in an image, including Void. Their additional no object query category is distinct from PalmCity class 31.

Pretrained sources and checkpoint versions are recorded in the public repository. Access conditions and checkpoint licences are documented separately from the code licence. The challenge workflow permits pretrained weights and external data. PalmCity remains the only dataset directly used for the adaptation runs, while the transferred weights retain information learned from their upstream datasets.

4 A bounded training protocol

4.1 Calibration and computation budgets

Every candidate is calibrated using real PalmCity data at the full 512×1024 input size. Calibration uses one warmup optimizer update and three measured updates, together with one warmup inference and three measured validation images. Update timings include decoding, augmentation, loading, device transfer, forward and backward computation, gradient clipping, and AdamW. Model initialization and checkpoint writing are recorded separately. A short calibration estimates steady state cost and is not treated as evidence of segmentation quality.

All models use BF16 mixed precision and an effective batch size of four. Microbatch sizes and gradient accumulation vary with the model. DeepLabV3+ and UPerNet use microbatches of two because their pooled batch normalization requires more than one sample. SegFormer B2 also uses microbatches of two. The remaining systems use microbatches of one with four accumulation steps. The linear DINOv3 systems use gradient checkpointing. Accumulation correctly weights the number of samples in the final partial group.

The pilot budget is an estimated 600 seconds of training plus validation computation for each candidate. Calibration translates this estimate into a fixed optimizer update limit, maximum epoch count, and warmup length. Ten validation evaluations are reserved in the budget and determine the checkpoint with the highest mean IoU. Fixed update limits and variation in measured update and validation speed cause the computation time to differ from the estimate. Eligible pilots use 603.15 to 628.02 seconds of measured training and validation computation. Initialization and checkpoint writing add to the whole process duration and are excluded from these figures. The comparison therefore uses approximately equal computation.

The top two pilot systems are then trained from their original pretrained sources, rather than by continuing the pilot weights. Each receives an estimated 2400 seconds for seeds 42, 123, and 2026. EoMT has a limit of 6351 updates and 51 epochs, while DINOv3 ViT L with a linear head has 7940 updates and 64 epochs. Each run again has ten validation evaluations. All six runs finish within 1.81% to 4.55% above the computation target. EoMT and DINOv3 confirmations run on separate 32 GB RTX 5090 cards. No model combines the memory of the two cards.

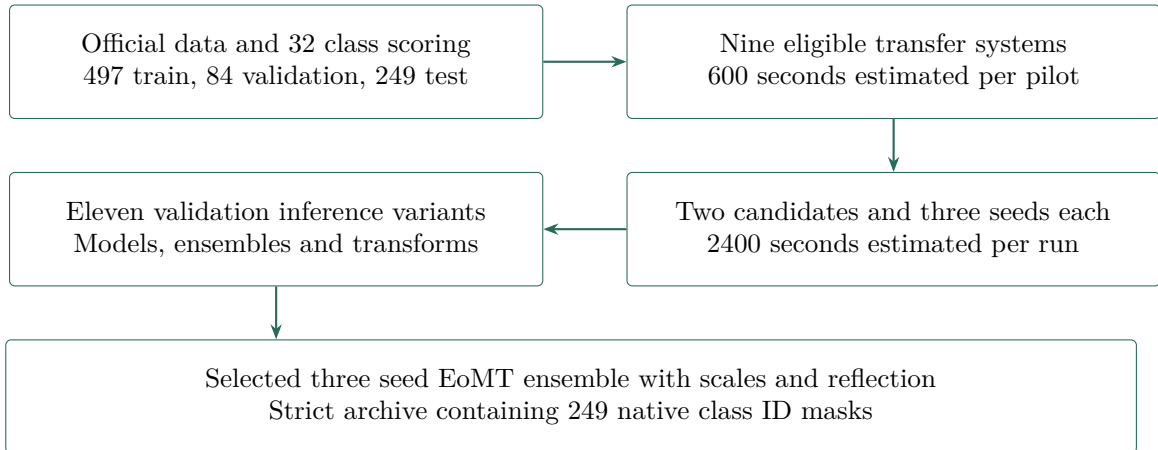


Figure 1: The selection procedure uses public validation labels throughout. Hidden test labels are never used for training, checkpoint selection, or inference variant selection.

4.2 Optimization and recorded recipe corrections

All runs use AdamW [21]. Bias and normalization parameters are excluded from weight decay. Learning rates use a polynomial schedule with power 0.9, a minimum value of 10^{-6} , and warmup

occupying 5% of the update budget. The data augmentation is a horizontal reflection with probability 0.5. There are no random crops, panorama rolls, or changes to the 2 : 1 aspect ratio. Training remains at the native spatial resolution.

Table 2 gives the eligible recipes. These are adaptations for a short PalmCity fine tuning run, rather than reproductions of the full source training schedules. EoMT uses a common learning rate of 6×10^{-5} for the encoder and prediction layers. Its original ADE20K schedule uses a larger base learning rate and a longer training procedure. ConvNeXt uses one scalar encoder multiplier as an approximation to the source stage dependent values. These choices limit implementation and search complexity, and their effect remains part of the evaluated configuration.

Table 2: Eligible optimizer settings. The encoder learning rate equals the base rate multiplied by the encoder factor. Gradient clipping uses the recorded norm threshold.

System	Base learning rate	Encoder factor	Decay	Clip
DeepLabV3+ ResNet50	10^{-4}	1	0.01	1
SegFormer B2 and B5	6×10^{-4}	0.1	0.01	1
DINOv3 linear B and L	6×10^{-4}	0.1	0.01	1
EoMT DINOv3 L	6×10^{-5}	1	0.01	1
UPerNet ConvNeXt L	10^{-4}	0.3	0.05	1
UPerNet Swin L	6×10^{-5}	1	0.01	1
Mask2Former Swin L	10^{-4}	0.1	0.05	0.01

An initial set of nine pilots used a more uniform adapted recipe. Review of the source settings revealed four changes worth applying before the primary comparison. The changes concern the SegFormer encoder and head learning rate ratio, the Mask2Former encoder rate, decay and clipping, and the ConvNeXt encoder rate and decay. These four configurations were rerun under the same pilot budget. The primary cohort uses the specified corrected recipes and the five unchanged original runs. It does not select the larger score from each old and corrected pair. In particular, the corrected Mask2Former score is slightly lower than its original score and remains the eligible result. Both versions are retained in the experiment record. There are therefore 13 pilot executions and six confirmation executions, giving 19 training runs in total.

The complete recorded training and validation computation is 6.3443 GPU hours, including the four additional diagnostic pilot runs and excluding initialization and checkpoint input and output. The inference sweep takes another 392.63 seconds of elapsed time and finishes within its fixed 1800 second allowance.

4.3 Reproducibility controls

Before training, source code and dependencies are frozen. Each run records the resolved model configuration, pretrained source, dataset inputs, random seed, update limits, validation schedule, and measured resources. Checkpoints include model weights, optimizer and scheduler state, mixed precision state, and random number generator state. Resume operates at an epoch boundary and checks that the saved experiment settings match before proceeding.

The environment uses Python 3.12.12, PyTorch 2.9.1 with CUDA 12.8, torchvision 0.24.1, Transformers 5.5.2, segmentation models pytorch 0.5.0, timm 1.0.30, and SciPy 1.17.1. The dependency lock is retained. Random seeds are recorded, and strict CUDA determinism is disabled. Hardware and numerical differences can therefore affect a rerun.

5 Probability ensembles and inference variants

5.1 Comparable semantic probabilities

The inference pipeline converts every model output into a distribution over the same 32 semantic classes. For a dense predictor this is the softmax of its class logits. Query predictors require a different conversion. If a_{qc} is the softmax probability that query q has semantic class c , and $m_q(u)$ is the sigmoid mask value at pixel u , the unnormalized semantic score is

$$z_c(u) = \sum_q a_{qc} m_q(u), \quad p_c(u) = \frac{z_c(u)}{\sum_{k=0}^{31} z_k(u)}. \quad (2)$$

Only the separate no object query category is discarded. Void is retained. Small positive clamping prevents numerical failure before normalization. Returning the logarithm of the normalized probabilities allows the shared dense inference interface to recover the intended distribution through a softmax. Without this step, applying a softmax directly to positive mask scores would change the ensemble meaning.

For the selected system, let $p_{s,t,c}(u)$ be the probability from training seed s and inference transform t , aligned back to the native image coordinates. The ensemble is

$$\bar{p}_c(u) = \frac{1}{3 \cdot 6} \sum_{s \in \{42, 123, 2026\}} \sum_{t \in \mathcal{T}} p_{s,t,c}(u), \quad \hat{y}(u) = \arg \max_{c \in \{0, \dots, 31\}} \bar{p}_c(u). \quad (3)$$

The transform set $\mathcal{T}$ contains scales 0.75, 1, and 1.25, each with the original and horizontally reflected image. Reflection is reversed before averaging. Scale inputs preserve the panorama aspect ratio and correspond to heights 384, 512, and 640 pixels. Outputs are aligned to the original 512×1024 grid using bilinear interpolation. The pipeline averages probabilities, never integer class masks.

There is a small implementation detail in the spatial alignment. Query mask logits are interpolated before sigmoid aggregation. The common inference interface then interpolates log probabilities when mapping the scaled result back to the native grid and normalizes them with softmax. This is the recorded implementation used for the reported scores. It should be retained when reproducing the output rather than silently replacing it with a different interpolation order.

5.2 Fixed comparisons and memory control

The inference plan evaluates eleven variants on the same 84 validation images. It includes the best EoMT checkpoint, the best linear DINOv3 checkpoint, three weights for their combination, an equal ensemble of the three EoMT seeds, reflection alone, three scales with reflection, and windows of 256×512 pixels with 0.5 overlap. The candidate list is fixed before the sweep. It is not expanded in response to individual scores.

Only one panorama is processed at a time. Probability sums remain on the CPU, and the ensemble models are moved to the GPU sequentially for each image. This permits three checkpoints to be used without keeping all model weights in GPU memory together. It also adds model transfer cost. Reported prediction time includes image decoding, tensor preparation, transfers, forward inference, transformation reversal, class selection, and PNG writing. Checkpoint loading and source checks are included in the separate total runtime. Allocated memory is measured through PyTorch and is not the complete memory usage reported by the graphics driver.

6 Results

6.1 Eligible pilot comparison

Table 3 reports the nine eligible pilot results. EoMT has the highest mean IoU at 57.565%. The linear DINOv3 ViT L system reaches 52.151% and Mask2Former reaches 51.926%. The difference between the second and third candidate is only 0.226 percentage points. Selecting the second candidate for confirmation is therefore a bounded search decision, not strong evidence that Mask2Former cannot perform better with a different schedule.

Table 3: Eligible pilot results at approximately 600 seconds of training and validation computation. Scores are percentages. Rare IoU is the mean over the eight categories fixed from training pixel frequency. Memory is peak allocated training memory in GiB.

System	mIoU	mF1	Rare IoU	Seconds	GiB
EoMT DINOv3 L	57.565	68.515	26.843	624.20	10.58
DINOv3 L linear	52.151	63.566	24.841	628.02	5.71
Mask2Former Swin L	51.926	62.885	15.591	609.14	9.36
DINOv3 B linear	49.110	60.300	21.979	603.15	1.99
UPerNet ConvNeXt L	48.784	59.043	13.314	609.52	9.59
UPerNet Swin L	46.876	56.323	5.802	605.99	13.05
SegFormer B2	46.221	56.780	12.502	608.80	4.09
SegFormer B5	45.742	55.962	7.524	618.96	6.02
DeepLabV3+ ResNet50	42.306	51.539	5.057	619.87	2.43

The pilot ranking does not simply follow model size. SegFormer B5 trails B2 under the eligible recipes and budget. DINOv3 ViT B with a linear decoder exceeds both UPerNet variants, while using less training memory. EoMT combines the highest score with the highest rare category mean in the pilot. These observations support the decision to continue EoMT and the linear DINOv3 ViT L system. They also show why a broad initial comparison is useful before investing in longer runs.

The recipe corrections have a visible effect. SegFormer B5 changes from 34.637% to 45.742% mean IoU, B2 changes from 44.323% to 46.221%, and ConvNeXt UPerNet changes from 47.747% to 48.784%. Mask2Former changes from 52.015% to 51.926%. The magnitude of the B5 change illustrates that a weak short run may reflect an unsuitable adaptation recipe. It does not justify selecting every source setting after seeing the result, which is why the correction cohort is explicitly retained.

6.2 Longer training with three seeds

The confirmation runs in Table 4 strengthen the selection evidence. EoMT achieves a mean IoU of 58.791% across the three seeds, compared with 54.488% for the linear DINOv3 ViT L system. Every EoMT run exceeds every linear DINOv3 run. The mean difference is 4.303 percentage points. EoMT uses about 10.58 GiB of allocated training memory, compared with 5.71 GiB for the linear system.

The population standard deviation of mean IoU across seeds is 0.695 percentage points for EoMT and 0.276 for the linear DINOv3 system. These values describe observed training variation. The best individual EoMT checkpoint is seed 2026 at epoch 31. Its score of 59.670% is used as the individual reference in the inference comparison. The best linear DINOv3 checkpoint is seed 123 at epoch 52, with 54.706%.

Table 4: Independent confirmation runs with approximately 2400 seconds of training and validation computation each. The checkpoint is selected from ten scheduled validation evaluations per run.

System	Seed	mIoU	mF1	Rare IoU	Best epoch
EoMT DINOv3 L	42	57.970	68.576	31.336	26
EoMT DINOv3 L	123	58.731	69.353	30.846	36
EoMT DINOv3 L	2026	59.670	69.718	31.103	31
DINOv3 L linear	42	54.660	66.021	28.406	45
DINOv3 L linear	123	54.706	65.862	29.475	52
DINOv3 L linear	2026	54.098	65.523	27.448	58
EoMT mean		58.791	69.216	31.095	
DINOv3 linear mean		54.488	65.802	28.443	

6.3 Inference selection and its cost

The full inference comparison appears in Table 5. Averaging the three EoMT seeds at the native image scale gives 59.852% mean IoU, a modest improvement of 0.181 percentage points over the strongest single checkpoint. Using three scales with reflection on the single checkpoint gives a larger improvement, reaching 60.671%. Combining the three seeds with the same transforms gives the highest validation score, 60.953% mean IoU and 71.165% mean F1.

Table 5: All eleven inference variants on 84 validation panoramas. E denotes the best EoMT checkpoint and D the best linear DINOv3 checkpoint. Ensemble weights apply to normalized semantic probabilities. Seconds measure the complete prediction loop, including transfers and PNG output.

Inference variant	mIoU	mF1	Seconds	GiB
Three EoMT seeds, scales and reflection	60.953	71.165	81.72	2.70
E, scales and reflection	60.671	70.828	16.77	2.70
Three EoMT seeds, native	59.852	70.093	51.82	2.32
E, native	59.670	69.718	7.85	2.32
$0.75E + 0.25D$, native	59.652	69.624	34.72	2.32
Three EoMT seeds, reflection	59.510	69.595	56.95	2.39
E, overlapping windows	59.070	69.809	13.68	1.91
E, reflection	59.025	68.930	8.34	2.39
$0.5E + 0.5D$, native	58.364	68.725	34.78	2.32
$0.25E + 0.75D$, native	55.863	66.842	34.69	2.32
D, native	54.706	65.862	6.96	1.79

The selected ensemble improves mean IoU by 1.283 percentage points over the native single checkpoint and by 0.282 points over the transformed single checkpoint. Prediction time increases from 16.77 to 81.72 seconds for 84 images, a factor of 4.87. Figure 2 shows the tradeoff. Peak allocated inference memory remains about 2.70 GiB because models are processed sequentially. The highest scoring variant is selected because its runtime remains feasible for the challenge. The transformed single checkpoint offers a faster option at 60.671% mean IoU.

Combining EoMT and the linear DINOv3 system does not improve on the best native EoMT checkpoint for any tested weight. Reflection alone and the tested overlapping window approach also reduce mean IoU. These results caution against treating ensemble diversity or additional transformations as automatic improvements. The observed benefit belongs to the specific three seed, three scale, reflection configuration evaluated here.

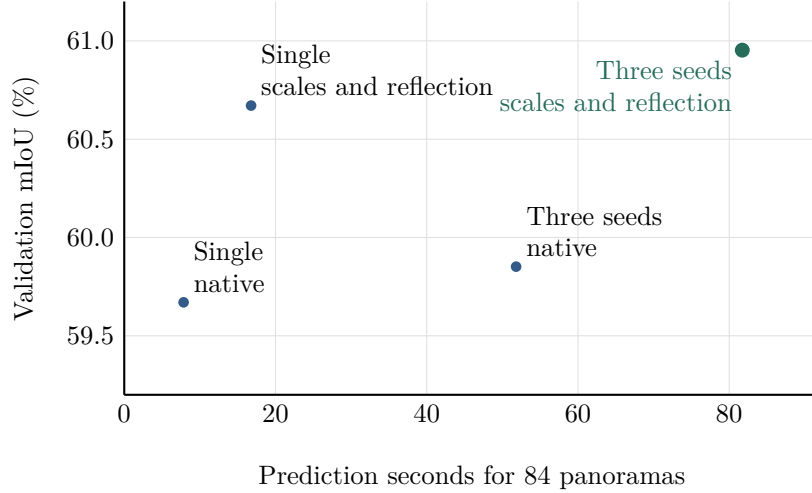


Figure 2: Measured accuracy and prediction cost of four EoMT inference configurations. The selected ensemble yields the highest validation score, while the transformed single model is much faster. The graph begins at 59.2% on the vertical axis to show the small differences clearly.

6.4 Class level behavior

The selected prediction performs strongly on large scene regions. Sky reaches 98.88% IoU, Road 95.07%, Building 94.38%, Void 92.65%, Car 91.69%, and Water Surface 91.24%. Table 7 in the appendix reports every category. The average over the eight rare categories is 32.044%, compared with 31.103% for the best native single checkpoint.

Some gains are concentrated in difficult categories. Parking Lot increases from 28.41% to 39.73% and Railing from 0.38% to 11.23%. The ensemble also improves Sidewalk, Pedestrian, Truck, Infrastructure Box, and Parking Barrier. Other categories lose accuracy. Motorcycle decreases by 7.83 percentage points, Bus by 4.91, and Driver by 2.84. Bicycle remains weak at 15.86% and Stairs at 13.92%. These tradeoffs are visible even though the overall metric improves.

Overpass and Pruned Tree have zero IoU in the final prediction. Their extremely small validation support limits what can be inferred from this result, but they remain part of the official mean. Success on broad scene structure coexists with unresolved small and rare category errors.

6.5 Qualitative results

Figure 3 compares real panoramas, reference annotations, and the saved predictions of the selected ensemble. The examples are selected by ranking all 84 validation images by the fraction of correctly labelled pixels and taking the nearest lower quartile, median, and upper quartile. This diagnostic gives an explicit selection rule for the illustrations. It differs from the official mean IoU because large regions contribute more heavily to pixel agreement. The examples retain much of the broad scene structure, while the disagreement maps reveal boundary errors and missed small objects.

Figure 4 shows the validation image with the lowest pixel agreement, 65.33%. A shaded walkway is largely labelled as Road instead of Sidewalk, and the stairs are largely assigned to Operator and Shadow. The enlarged detail is the 384×192 window with the largest number of incorrect pixels whose reference category is not Void. The displayed reference masks and predictions are unchanged. This example makes the remaining ambiguity between street surfaces and small structures visible alongside the stronger results.

The illustrated images and reference annotations are credited to PalmCity [6] and are shown for academic illustration under its published noncommercial research and academic use conditions.

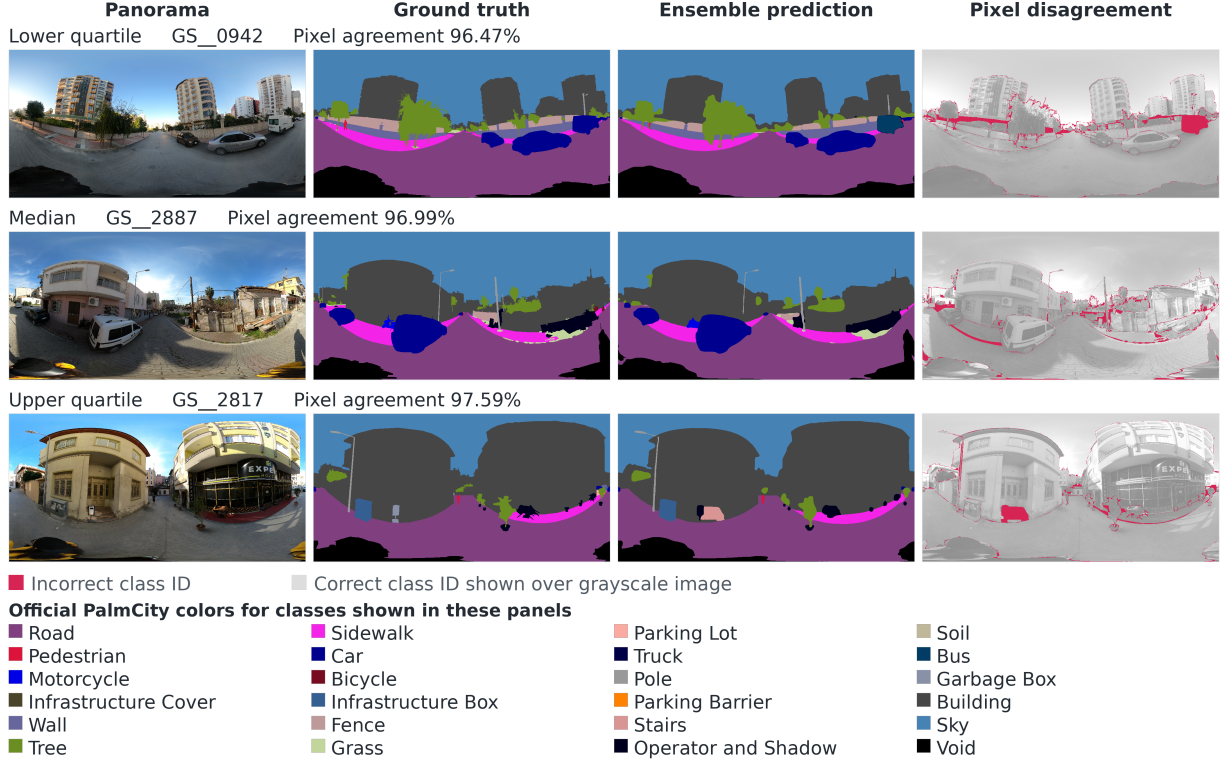


Figure 3: Real validation examples nearest the lower quartile, median, and upper quartile of pixel agreement across all 84 images. This selection uses pixel agreement rather than class balanced IoU. Predictions are the saved outputs of the selected three seed ensemble with scales and reflection. Masks use the official PalmCity colors. Pink marks incorrect class identifiers, including Void.

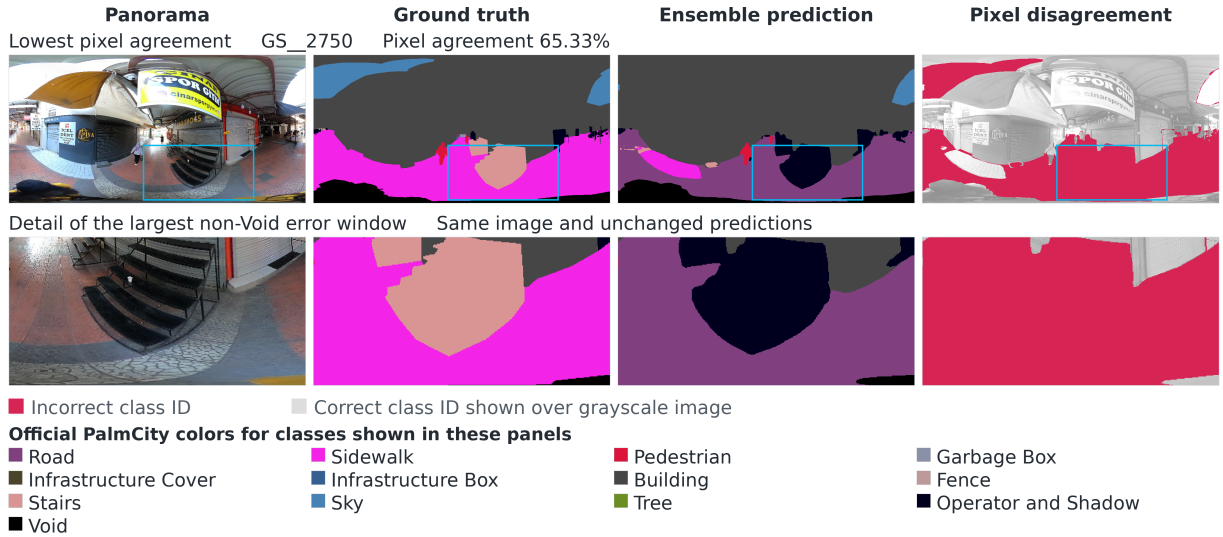


Figure 4: The validation image with the lowest pixel agreement, GS__2750. The blue box marks the detail shown below, selected as the 384×192 window with the most incorrect pixels outside reference Void regions. The shaded walkway is largely predicted as Road instead of Sidewalk, while the stairs are largely assigned to Operator and Shadow. Pink marks all incorrect class identifiers.

6.6 Hidden test submission

The final archive contains exactly 249 PNG files with the official test basenames. Every file is a single channel image at 1024×512 resolution, and every pixel contains an integer class identifier from 0 to 31. An independent audit checks these conditions and confirms that the archive contains the outputs of the three selected checkpoints. The archive is 1.82 MiB.

Test prediction takes 242.19 seconds for the 249 images. Total runtime is 251.49 seconds after including checkpoint loading and provenance checks. Peak allocated memory is 2.7015 GiB and peak reserved memory is 3.6133 GiB on one RTX 5090. This is a measured whole split result rather than a forward pass throughput estimate.

Table 6 records the leaderboard snapshot associated with the submitted archive, independently retrieved from the public Codabench API [22]. Submission 961441 by `yunusserhat`, dated 5 October 2026 at 09.15 Istanbul time, has 57.08% mean IoU and 67.96% mean F1. It leads the next listed entry by 11.05 percentage points in mean IoU and 10.98 points in mean F1 when calculated from the full precision API values. The ranking refers to this dated snapshot. It does not compare every possible method or future submission.

Table 6: PalmCity Test Task leaderboard snapshot dated 5 October 2026. Values are percentages. The table identifies the submitted system without inferring the training procedure of other participants.

Rank	Participant	Submission	mIoU	mF1
1	yunusserhat	961441	57.08	67.96
2	onurbayrak	945126	46.02	56.99
3	MaxwellEQ	950834	33.54	40.93

The hidden test mean IoU is 3.88 percentage points below the selected public validation score. The mean F1 difference is 3.20 points. The leaderboard score was obtained only after the final inference setting and archive were fixed.

7 Discussion

7.1 What the comparison supports

The evidence favors EoMT with the recorded DINOv3 ViT L checkpoint and PalmCity recipe among the systems tested. It wins the eligible pilot and remains ahead of the other confirmed candidate across three independent seeds. The final probability ensemble with scale and reflection transforms increases validation accuracy further. These results make it a defensible choice for this challenge and computation budget.

The experiment also supports a more general procedural lesson. Correct class semantics, suitable source initialization, budget calibration, and output verification matter alongside the architecture name. Query probability normalization and rectangular mask alignment affect the predicted pixels, while CPU and GPU model transfers affect the measured runtime. These implementation choices accompany the reported score so that another researcher can recover the same procedure.

The systems have different pretrained data, decoders, and loss functions. EoMT begins from an ADE20K segmentation checkpoint, while the linear DINOv3 candidate begins from a representation checkpoint with a new decoder. Larger or more suitable decoders for DINOv3 may narrow the difference. The pilot budget gives a practical answer about transfer configurations under limited search. Isolating architectural contributions would require matched pretraining and additional controlled comparisons.

Only two candidates receive longer training. Mask2Former is close to the second candidate in the pilot and could merit confirmation in a larger study. The attention mask schedule and optimizer adaptations may interact with the limited number of PalmCity images. Longer schedules, broader augmentation, additional decoder comparisons, and rare object sampling remain useful questions for subsequent experiments.

7.2 Implications for urban image analysis

The high scores for road, sky, building, tree, and water suggest that the system can produce useful broad scene descriptions on imagery similar to the benchmark. However, a high mean IoU does not certify the reliability of a derived environmental indicator. For example, a measure concerned with bicycle infrastructure, access barriers, or small traffic controls may be sensitive to the classes that remain weak. Error inspection must be tied to the intended measurement rather than to a single overall ranking.

Visible greenness and environmental quality have been studied through different sensors and sampling strategies [1, 3]. PalmCity segmentation could be one input to a subsequent urban analysis. Linking it to health, urban redesign, or policy would require a separate study of acquisition conditions, geographical sampling, temporal coverage, and task specific error. A benchmark trained model should be evaluated again before use in a new city or comparison between groups of residents.

Decoder design and adaptation also deserve attention across image domains. The author’s earlier ATTransUNet work combines attention and transformer components for building segmentation from aerial imagery and laser data [23]. Panoramic street parsing presents a complementary problem with different image geometry, annotation needs, and transfer conditions.

Temporal consistency provides another potential extension when connected panoramas or street video become available. Recent VidEoMT work extends encoder focused mask prediction to video [24]. This direction could be evaluated on future data with temporal connections, while the present system processes each panorama independently.

7.3 Limits of the evidence

The 84 validation images select checkpoints, candidates for confirmation, individual seeds, and the final inference variant. This reuse creates selection dependence. The 0.282 percentage point advantage of the selected ensemble over the faster transformed single model may change on an independent sample. Three seed population deviations describe training variation, not confidence intervals over new streets. Visually similar views across the official split further limit claims of geographical independence. Route, GPS, and acquisition time metadata would be needed for a spatial evaluation. The validation and hidden test difference may reflect split composition, sampling variation, or repeated validation selection, but its causes cannot be isolated from aggregate scores.

The result establishes the highest score among the three public entries in the dated leaderboard snapshot. Architecture wide superiority, final competition victory, and generalization to other cities would require additional evidence. The reported 19 training runs constitute a bounded search with adapted source recipes, rather than exhaustive optimization of each candidate.

7.4 Public release

The public code release lets another researcher prepare data, obtain permitted pretrained sources, train the selected systems, and generate a challenge archive in a separate workspace. It includes the dependency lock, configurations, result summaries, source identities, and commands for the experiment stages. The project code is released under the GNU General Public License version 3 only, identified as `GPL-3.0-only`. PalmCity images, dependencies, and third party pretrained

weights retain their original conditions. Some evaluated sources have research or noncommercial restrictions, and DINOv3 has its own access conditions.

The published recipe allows repetition of the training procedure, while CUDA nondeterminism and hardware behavior can change a rerun. The training and inference commands document how the submitted predictions were produced.

8 Conclusion

This study documents a first place PalmCity challenge submission in the leaderboard snapshot of 5 October 2026. Approximately equal computation pilots, three seed confirmation of the leading candidates, and a complete inference comparison lead to a three seed EoMT DINOv3 ViT L probability ensemble with three scales and horizontal reflection. The system reaches 60.95% validation mean IoU and 57.08% hidden test mean IoU, while producing 249 test masks in 251.49 seconds on one RTX 5090. The complete record exposes the cost of the ensemble, the limits of the small validation split, and persistent rare category errors. It provides a reproducible challenge solution and a basis for further street image segmentation research.

Data and code availability

PalmCity is available through the dataset authors' [official repository](#). The training and inference code is available in the [PalmCity challenge repository](#). The three trained seed models are provided as native Transformers safetensors checkpoints in the [Hugging Face model release](#), with instructions for direct inference without retraining. The adapted weights retain their applicable source licences, including the DINOv3 terms supplied with the release. The manuscript and its source are maintained separately from the code repository.

Acknowledgements

The author thanks the Geospatial Data Science Group at the University of Glasgow. The author also thanks the PalmCity dataset creators and challenge organizers for providing the benchmark and evaluation platform, Meta AI for DINOv3, and the developers of EoMT and the other models and software used in the experiments.

A Complete class results

Table 7: Validation IoU for every official category. Single refers to EoMT seed 2026 at the native scale. Selected is the three seed ensemble with scales and reflection. Differences are percentage points calculated before rounding.

ID	Class	Single	Selected	Difference
0	Road	94.81	95.07	+0.26
1	Sidewalk	77.36	78.45	+1.10
2	Parking Lot	28.41	39.73	+11.31
3	Soil	76.21	78.94	+2.73
4	Pedestrian	65.88	69.22	+3.34
5	Driver	73.91	71.08	-2.84
6	Car	91.26	91.69	+0.44
7	Truck	69.58	72.47	+2.89
8	Bus	80.41	75.51	-4.91
9	Motorcycle	71.53	63.70	-7.83
10	Bicycle	17.63	15.86	-1.77
11	Traffic Light	44.34	44.85	+0.51
12	Traffic Sign	43.49	43.09	-0.40
13	Pole	59.52	61.56	+2.04
14	Garbage Box	53.78	57.48	+3.70
15	Sitting Bench	69.07	70.25	+1.18
16	Infrastructure Cover	74.19	75.63	+1.44
17	Infrastructure Box	62.14	65.28	+3.14
18	Parking Barrier	51.90	54.83	+2.93
19	Building	93.86	94.38	+0.52
20	Wall	68.36	70.85	+2.50
21	Fence	63.05	63.57	+0.52
22	Stairs	12.54	13.92	+1.37
23	Railing	0.38	11.23	+10.84
24	Overpass	0.00	0.00	0.00
25	Water Surface	88.91	91.24	+2.33
26	Sky	98.76	98.88	+0.12
27	Tree	81.11	81.73	+0.62
28	Grass	66.97	67.13	+0.16
29	Pruned Tree	0.00	0.00	0.00
30	Operator and Shadow	38.09	40.23	+2.15
31	Void	92.03	92.65	+0.63

B Independent checks

Independent training checks cover 19 runs, 190 validation evaluations, and 1162 training and validation input files. Independent inference checks directly decode 924 PNG files from the eleven validation variants and reconstruct their confusion matrices. The archive check covers all 249 members, PNG modes, native dimensions, class ranges, archive integrity, source images, and selected checkpoints. These checks confirm that the reported scores and the submitted masks correspond to the recorded experiments.

C Reproduction outline

C.1 Direct inference with the released models

The public repository separates project code from the data and output workspace. After cloning it, a reader prepares a project environment using the supplied setup script and its dependency lock. The released three seed models can then be applied to a local panorama or a directory of panoramas without downloading PalmCity or running training. Input panoramas must be twice as wide as they are high. The following command downloads the trained checkpoints and applies the selected ensemble with scales and reflection.

```
export PALMCITY_WORKSPACE=/absolute/user-owned/workspace
export PALMCITY_PYTHON=python3.12
bash scripts/setup.sh
bash scripts/verify.sh
HF_HUB_OFFLINE=0 bash scripts/run.sh python -m palmcity.hub \
  --download --input /absolute/path/panoramas \
  --output-dir outputs/challenge --mode challenge --colorize
```

The command writes class identifier PNGs and optional color visualizations into the selected workspace. The model release documents the source licences and the project helper preserves the rectangular mask geometry, Void category, and probability normalization used in the reported experiments. The faster single model options are also documented. Once the required weights are cached, inference can run offline.

C.2 Repeating the training

A reader downloads PalmCity from the official source, creates and audits the split manifest, and obtains the pretrained checkpoint at its recorded repository revision. Access gated sources require the reader’s own authorized account. The supplied selected configuration trains EoMT independently for seeds 42, 123, and 2026 with the recorded update and validation limits. After the environment setup above, the main training commands are

```
bash scripts/reproduce.sh data
HF_HUB_OFFLINE=0 bash scripts/reproduce.sh prepare --download
bash scripts/reproduce.sh train
bash scripts/reproduce.sh val
bash scripts/reproduce.sh test
```

After training, the inference command uses the three best checkpoints, equal weights, scales 0.75, 1, and 1.25, and horizontal reflection. The submission command packages the test masks and rejects an incomplete archive. The essential inference settings are

```
--weights 1 1 1
--scales 0.75 1 1.25
--hflip-tta
--split test
```

The repository documents the complete commands, expected split counts, checkpoint access conditions, memory checks, and output locations. Full comparison commands are also supplied for calibration, eligible pilots, confirmation, inference evaluation, and archive construction. A reader may reproduce the selected system alone or repeat the broader study. The original experiment records remain available alongside the portable training workflow.

References

- [1] Filip Biljecki and Koichi Ito. Street view imagery in urban analytics and GIS: A review. *Landscape and Urban Planning*, 215:104217, 2021. [10.1016/j.landurbplan.2021.104217](https://doi.org/10.1016/j.landurbplan.2021.104217).
- [2] Marco Helbich, Yao Yao, Ye Liu, Jinbao Zhang, Penghua Liu, and Ruoyu Wang. Using deep learning to examine street view green and blue spaces and their associations with geriatric depression in Beijing, China. *Environment International*, 126:107–117, 2019. [10.1016/j.envint.2019.02.013](https://doi.org/10.1016/j.envint.2019.02.013).
- [3] Marco Helbich, Ronald Poppe, Daniel Oberski, Maarten Zeylmans van Emmichoven, and Raoul Schram. Can’t see the wood for the trees? an assessment of street view- and satellite-derived greenness measures in relation to mental health. *Landscape and Urban Planning*, 214:104181, 2021. [10.1016/j.landurbplan.2021.104181](https://doi.org/10.1016/j.landurbplan.2021.104181).
- [4] Koichi Ito and Filip Biljecki. Assessing bikeability with street view imagery and computer vision. *Transportation Research Part C: Emerging Technologies*, 132:103371, 2021. [10.1016/j.trc.2021.103371](https://doi.org/10.1016/j.trc.2021.103371).
- [5] Yunus Serhat Bıçakçı, Joseph Shingleton, Yu Wang, and Anahid Basiri. Mapping the semantics of the street: A VLM-driven geospatial analysis in Fatih, Istanbul. In *Proceedings of the 1st International Conference on Geospatial Artificial Intelligence*, Ghent, Belgium, 2026. [10.5281/zenodo.19390648](https://doi.org/10.5281/zenodo.19390648). URL [Full text](#).
- [6] Muzaffer Can Iban, Onur Can Bayrak, Serkan Kartal, Dogu Ilmak, and Dursun Zafer Seker. PalmCity: An emerging benchmark dataset for semantic segmentation of panoramic street view images in under-represented developing countries. In *Computational Science and Its Applications – ICCSA 2025 Workshops*, volume 15899 of *Lecture Notes in Computer Science*, pages 165–175, Cham, 2026. Springer. [10.1007/978-3-031-97663-6_14](https://doi.org/10.1007/978-3-031-97663-6_14). URL [Full text](#).
- [7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision – ECCV 2018*, pages 833–851. Springer, 2018. [10.1007/978-3-030-01234-2_49](https://doi.org/10.1007/978-3-030-01234-2_49).
- [8] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, pages 12077–12090, 2021. URL [Full text](#).
- [9] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Computer Vision – ECCV 2018*, pages 432–448. Springer, 2018. [10.1007/978-3-030-01228-1_26](https://doi.org/10.1007/978-3-030-01228-1_26).
- [10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. URL [Full text](#).
- [11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. URL [Full text](#).
- [12] 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*, pages 1290–1299, 2022. URL [Full text](#).
- [13] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL [Full text](#).
- [14] Tommie Kerssies, Niccolò Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan de Geus. Your ViT is secretly an image segmentation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25303–25313, 2025. URL [Full text](#).
- [15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, pages 6402–6413, 2017. URL [Full text](#).
- [16] Yunus Serhat Bıçakçı. Hybrid CNN-transformer ensemble for enhanced tank detection in aerial imagery. Research Square preprint, February 2026. URL [Full text](#). Version 1, posted 5 February 2026.

- [17] Ozan Ozturk, Beytullah Sarica, and Dursun Zafer Seker. Interpretable and robust ensemble deep learning framework for tea leaf disease classification. *Horticulturae*, 11(4):437, 2025. [10.3390/horticulturae11040437](#). URL [Full text](#).
- [18] Siqin Wang, Xiao Huang, Filip Biljecki, Francisco Rowe, and Veruska Muccione. Advancing sustainable geospatial analytics and geoinformatics through repeatable, reproducible, and expandable (RRE) framework and design. *International Journal of Applied Earth Observation and Geoinformation*, 149: 105239, 2026. [10.1016/j.jag.2026.105239](#). Editorial.
- [19] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. [10.1007/s11263-018-1140-0](#).
- [20] EoMT authors. EoMT Model Zoo – DINOv3. Official model release and implementation, n.d. URL [Full text](#). Official software resource. Accessed 5 October 2026.
- [21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL [Full text](#).
- [22] PalmCity challenge organizers. PalmCity Test Task public leaderboard. Codabench leaderboard 20330, 2026. URL [Full text](#). Snapshot accessed 5 October 2026. Submission 961441 by yunusserhat.
- [23] Yunus Serhat Bıçakçı and Beytullah Sarica. ATTransUNet: Semantic segmentation model for building segmentation from aerial image and laser data. *Nordic Machine Intelligence*, 2(3), 2023. [10.5617/nmi.10039](#).
- [24] Narges Norouzi, Idil Esen Zulfikar, Niccolò Cavagnero, Tommie Kerssies, Bastian Leibe, Gijs Dubbelman, and Daan de Geus. VidEoMT: Your ViT is secretly also a video segmentation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35177–35186, 2026. URL [Full text](#).