Image Segmentation Review: DeepLabV3 and U-Net

Experiments on Cityscapes, Kvasir-SEG and Supervisely Person

Rami Aridi

November 18, 2025

Abstract

This review evaluates and compares two widely-used segmentation architectures, **U-Net** and **DeepLabV3**, on several benchmark datasets for semantic segmentation: **Cityscapes, Kvasir-SEG** and **Supervisely Person**. We provide a concise overview of each method, outline the experimental setup including targeted ablations and cross-dataset evaluation, and present quantitative results with metrics, visualizations, and analyses. The goal is to highlight the strengths and limitations of each architecture in different segmentation scenarios and provide insights for future small-scale segmentation studies.

1 Introduction

Semantic segmentation, the task of assigning a class label to every pixel in an image, is a fundamental problem in computer vision with applications ranging from autonomous driving and medical imaging to scene understanding. Early approaches relied on hand-crafted features and classical machine learning techniques, which were limited in accuracy and scalability. The introduction of Fully Convolutional Networks (FCNs) [6] marked a turning point, enabling end-to-end training and dense predictions directly from raw images.

Architectures such as U-Net [8] introduced encoder-decoder structures with skip connections to capture fine spatial details, while the DeepLab family [1, 2] leveraged atrous (dilated) convolutions and spatial pyramid pooling to capture multi-scale context without reducing resolution.

This review focuses on two representative families:

- **DeepLabV3** (encoder + ASPP) which captures multi-scale context via atrous spatial pyramid pooling [1].
- **U-Net** (symmetric encoder–decoder with skip connections) that is particularly effective for precise boundary recovery and small-object segmentation [8].

We evaluate these methods on standard datasets, including Cityscapes [3], Kvasir-SEG [5], and Supervisely Person [9]. Our study includes an *ablation analysis*, examining architectural and component changes such as backbone depth and encoder initialization, a *hyperparameter study*, investigating tunable training and configuration settings such as output stride and input resolution, and cross-dataset generalization to assess domain shift. Performance is measured using standard segmentation metrics, including mIoU, per-class IoU, mDice, per-class Dice, and pixel accuracy [6, 7].

2 Methods

2.1 Architectures

DeepLabV3 DeepLabV3 employs atrous (dilated) convolutions together with an Atrous Spatial Pyramid Pooling (ASPP) module to capture multi-scale contextual information while controlling feature map resolution. Typical implementations pair the ASPP head with a ResNet

backbone (e.g., ResNet-50 or ResNet-101) and allow different output strides (OS) that trade spatial detail for computational cost [1, 2]. In practice, these design choices affect the model's ability to recover fine boundaries and to model long-range context.

U-Net The U-Net family uses a contracting path (encoder) to aggregate context and a symmetric expanding path (decoder) to recover spatial detail, with skip-connections that directly pass high-resolution encoder features to the decoder for improved localization. Originally developed for biomedical segmentation, U-Net and its variants remain popular for tasks that require precise boundary recovery and for handling small objects [8].

2.2 Datasets

We evaluate and analyze the models on three commonly used segmentation datasets to cover both multi-class urban scenes and binary object segmentation tasks:

- **Cityscapes**: An urban-scene semantic segmentation dataset used in DeeplabV3's paper [1] widely used for autonomous-driving research; it contains 20 fine-grained classes for road-scene understanding and is a standard benchmark for multi-class segmentation [3]. In this work we operate on a subset of 3116 images to keep experiments reproducible and fast.
- **Kvasir-SEG**: A medical imaging dataset of 1000 images for polyp segmentation (binary mask: polyp vs background), used to evaluate small-object sensitivity and boundary accuracy in biomedical-like settings [5].
- **Supervisely Person**: A person segmentation collection (binary mask: person vs background) useful for cross-dataset generalization experiments across natural images. In this work we use a subset of 2667 images. [9].

Kvasir-SEG and Supervisely Person are binary segmentation datasets (object vs background); we use cross-dataset evaluation (train on Kvasir-SEG $\rightarrow$ test on Supervisely Person) to probe domain shift and model robustness to appearance and annotation differences.

For the exact dataset used in this study, please refer to Appendix 5.

2.3 Ablation / Hyperparameters Factors

Ablation study :

Backbone depth (DeepLab: ResNet-50 vs ResNet-101 (capacity vs generalization).

Encoder initialization (U-Net): ImageNet pretrained weights vs random initialization.

Hyperparameter study :

Output stride (DeepLab): OS = 8 vs OS = 16 (feature map resolution).

Input resolution (U-Net): 256 vs 512 pixels (sensitivity to small objects and boundary fidelity).

Note on terminology and methodology. For clarity: in this report an **ablation** denotes a deliberate change to the model architecture or a core component (for example replacing ResNet-50 with ResNet-101, or removing a module). A **hyperparameter** denotes a tunable training/configuration choice that does not change the model's structural components (for example output stride, input resolution, learning rate, or batch size). Each ablation/hyperparameter experiment is executed by changing a single item relative to its corresponding baseline so as to isolate its effect.

2.4 Metrics

The primary metric is **mean Intersection-over-Union (mIoU)** [6], we also report the **Dice coefficient (mDice)**[7]. Supporting diagnostics include per-class IoU, per-class Dice, and **pixel accuracy (mPA)**; note that pixel accuracy can be misleading on highly imbalanced datasets, so the emphasis of analysis is on mIoU and Dice for fairness across classes.

3 Experiments

We evaluate DeepLabV3 and U-Net using a consistent, reproducible experimental protocol. Experiments include baseline reproductions, a focused ablation / hyperparameter study, and cross-dataset generalization tests.

3.1 Baseline experiments

The baselines establish a common reference point for all comparisons. Baseline configurations were chosen to reflect common practice for small-to-medium scale segmentation experiments and to be reproducible across datasets:

- **U-Net (baseline):** ResNet-50 encoder (ImageNet pretrained), input size 256.
- **DeepLabV3 (baseline):** ResNet-50 encoder (ImageNet pretrained), output stride = 8, input size 256.

All baseline experiments were trained under a unified configuration to ensure fair comparisons. Unless otherwise specified, models are trained at an input resolution of 256×256 with a batch size of 16, using the Adam optimizer (learning rate 1×10^{-4}, weight decay 1×10^{-4}) for 50 epochs on ImageNet-pretrained ResNet-50 backbones. Normalization follows standard ImageNet statistics (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]). The number of output classes is dataset-dependent (2 for Supervisely/Kvasir, 20 for Cityscapes), and all experiments use the same augmentation policy and an ignore index of 255 for invalid labels.

Baseline results (baseline training). The table below [1] reports the baseline runs trained with the default, reproducible training recipe described previously (ResNet-50 encoder/backbone, ImageNet pretrained weights, input size = 256; DeepLabV3 uses output stride = 8).

Model	Dataset	mIoU	mDice	mPA
DeepLabV3	Cityscapes	0.5171	0.6459	0.9041
DeepLabV3	Kvasir	0.8830	0.9361	0.9664
U-Net	Cityscapes	0.4831	0.6004	0.9119
U-Net	Kvasir	0.8820	0.9354	0.9659

Table 1: Baseline results (baseline training)

3.2 Experiment naming convention

To ensure clear traceability across logs, plots and exported CSV files, each experiment is associated with a unique model name which are listed in Table 2. These names encode the key configuration choices (architecture, dataset, backbone, output stride or input resolution). To get the models used in this study, please refer to Appendix 5.

Model Name	Model	Dataset	Backbone	Pre-trained	OS	Input	Notes
deeplabv3-cityscapes	DeepLabV3	Cityscapes	resnet50	imagenet	8	256	Baseline
deeplabv3-101-cityscapes	DeepLabV3	Cityscapes	resnet101	imagenet	8	256	Ablation (backbone)
deeplabv3-os16-cityscapes	DeepLabV3	Cityscapes	resnet50	imagenet	16	256	Hyperparameter (OS)
deeplabv3-kvasir	DeepLabV3	Kvasir	resnet50	imagenet	8	256	Baseline
unet-cityscapes	U-Net	Cityscapes	resnet50	imagenet	N/A	256	Baseline
unet-noimagenet-cityscapes	U-Net	Cityscapes	resnet50	None	N/A	256	Ablation (init)
unet-kvasir	U-Net	Kvasir	resnet50	imagenet	N/A	256	Baseline
unet-512-kvasir	U-Net	Kvasir	resnet50	imagenet	N/A	512	Hyperparameter (input size)

Table 2: Summary of model configurations and experiment labels. Training and evaluation were both performed on the dataset listed in the "Dataset" column.

3.3 Ablation design and rationale

A compact, targeted ablation / hyperparameter study was selected to probe the most influential design choices while keeping the total number of runs manageable. For each experiment, only a single variable is changed relative to the corresponding training baseline [3.1].

Backbone depth (ResNet-50 vs ResNet-101). Increasing backbone depth increases representational capacity and the effective receptive field, which can improve performance on complex or texture-rich classes. We compare ResNet-50 and ResNet-101 backbones for DeepLabV3 to quantify gains in mIoU. Results are reported in Table 3.

Output stride (OS = 8 vs OS = 16), DeepLab only. Smaller output stride (OS=8) provides higher-resolution feature maps and typically improves boundary localization at the cost of increased computation. We evaluate how boundary accuracy differs between these two output stride settings. Quantitative results and boundary-focused appear in Table 3.

Encoder initialization (ImageNet pretrained vs random) ImageNet is a large-scale image classification dataset containing millions of labeled images across a wide range of object categories. Pretraining an encoder on ImageNet helps the model learn rich, generic visual features, such as edges, textures, shapes, and object parts, that transfer well to downstream tasks like segmentation. To assess the importance of this prior knowledge, we compare a U-net model whose encoders are initialized with ImageNet-pretrained weights against a U-net model initialized randomly. This comparison reveals how much performance degrades without pretrained features and highlights the value of starting from ImageNet. Quantitative results are summarized in Table 4.

Input resolution (256 vs 512) Higher input resolutions provide more spatial detail, which helps the model better capture small objects and fine boundaries. However, increasing resolution

also raises computational cost and memory usage. We compare models trained with 256×256 and 512×512 inputs to quantify the effect of resolution on segmentation accuracy. Results are reported in Table 4.

Ablation and Hyperparameter Results :

Model Name	Dataset	Variant	mIoU	mDice	mPA
deeplabv3-cityscapes	Cityscapes	Baseline	0.5170	0.6458	0.9040
deeplabv3-101-cityscapes	Cityscapes	backbone=resnet101	0.5127	0.6411	0.9067
deeplabv3-os16-cityscapes	Cityscapes	OS=16	0.4647	0.5895	0.8908

Table 3: Ablation and hyperparameter results for DeepLabV3. All models were trained and evaluated on the dataset listed in the "Dataset" column. Metrics: mIoU = mean Intersection over Union, mDice = Dice coefficient, mPA = mean Pixel Accuracy.

Model Name	Dataset	Variant	mIoU	mDice	mPA
unet-cityscapes	Cityscapes	Baseline	0.4831	0.6003	0.9118
unet-noimagenet-cityscapes	Cityscapes	Random init encoder	0.3211	0.3984	0.8653
unet-kvasir	Kvasir	Baseline	0.8819	0.9354	0.9658
unet-512-kvasir	Kvasir	input_size=512	0.8764	0.9321	0.9647

Table 4: Ablation and hyperparameter results for U-Net. Layout identical to Table 3

3.4 Cross-dataset evaluation (domain shift)

To assess generalization to dataset shift, we perform cross-dataset evaluation: models trained on one dataset are evaluated on another without further fine-tuning. The cross-evaluation in this work is **Kvasir-SEG → Supervisely Person** (train on Kvasir-SEG, evaluate on Supervisely Person). This choice is motivated by two practical considerations: (1) the Kvasir-SEG baseline achieves high performance under our training protocol (Table [1]) (it is easier to train to good quality than the Cityscapes subset), and (2) Kvasir-SEG and Supervisely Person are both binary segmentation datasets (object vs background), making direct comparison meaningful. By contrast, cross-evaluating binary datasets against Cityscapes (a 20-class multi-class dataset) is not directly comparable and is therefore not the focus of our cross-evaluation experiments.

The cross-evaluation protocol is as follows: train the model on the source dataset using the baseline training recipe [3.1], evaluate in-domain on the source test set and cross-domain on the target dataset without any fine-tuning, and the drops for each metric.

Cross-evaluation results are summarized in Table 5 for DeepLabV3 and in Table 6 for U-net.

Train → Test	mIoU	Drop mIoU	mDice	Drop mDice	mPA	Drop mPA
Kvasir-SEG → Kvasir-SEG	0.8830	-	0.9361	-	0.9664	-
Kvasir-SEG → Supervisely	0.3959	0.4872	0.4953	0.4407	0.7096	0.2567

Table 5: Cross-evaluation for DeepLabV3 (experiment key example: `deeplabv3-kvasir`).

3.5 Practical considerations

All experiments, including training, inference, and evaluation, were conducted on Kaggle notebooks under Kaggle's resource constraints: 57 GB of disk space, 30 GB of RAM, and a single

Train → Test	mIoU	Drop mIoU	mDice	Drop mDice	mPA	Drop mPA
Kvasir-SEG → Kvasir-SEG	0.8820	-	0.9361	-	0.9664	-
Kvasir-SEG → Supervisely	0.3736	0.5084	0.4610	0.4745	0.6979	0.2680

Table 6: Cross-evaluation for U-Net (experiment key example: `unet-kvasir`).

GPU with 15 GB memory.

We complement accuracy-focused metrics with practical measurements to aid deployment decisions:

- **Compute cost:** Training time per epoch and peak GPU memory usage are reported for the heaviest configurations (ResNet-101 backbone, input size = 512, output stride = 8).
- **Equal training budget:** To ensure fair comparisons, the optimizer, total number of training iterations, and learning rate scheduler are kept identical across runs unless an ablation explicitly studies a training hyperparameter.

4 Analysis

4.1 Overview

This Analysis section interprets the experimental results reported earlier (baseline reproduction, ablations, cross-dataset evaluation), per-class statistics and qualitative visualizations.

4.2 Baseline results and comparison with the original papers

Table 1 reports our reproducible baseline runs (ResNet-50 encoder, ImageNet pretraining, input size 256×256, 50 epochs). Below we interpret these results, compare models across datasets, and relate our numbers to values reported in the literature.

Observations.

- **Cityscapes.** DeepLabV3 achieves mIoU = **0.5171**, mDice = 0.6459 and mPA = 0.9041, while U-Net achieves mIoU = **0.4831**, mDice = 0.6004 and mPA = 0.9119. DeepLabV3 therefore outperforms U-Net on Cityscapes by a clear margin in mIoU (≈ 3.4 percentage points).
- **Kvasir (polyp segmentation).** Both architectures perform very well on the Kvasir dataset (binary segmentation): DeepLabV3 mIoU = 0.8830, mDice = 0.9361, mPA = 0.9664; U-Net mIoU = 0.8820, mDice = 0.9354, mPA = 0.9659. Performance is nearly identical, with DeepLabV3 showing a very small advantage in mIoU and mDice.
- **mPA vs. mIoU differences.** On Cityscapes U-Net shows a slightly higher mean pixel accuracy (mPA) despite lower mIoU. This suggests U-Net is predicting many pixels correctly for dominant/large classes but is worse on per-class segmentation balance (which mIoU penalizes more).

Why Cityscapes mIoU is lower than typical literature numbers. There are multiple, co-occurring reasons why our Cityscapes mIoU ($\approx$0.52 for DeepLabV3) is substantially lower than the best-reported Cityscapes numbers in the literature (DeepLab variants and strong encoder/backbones report $>$ 0.75–0.82 mIoU on Cityscapes dataset; see [1, 2]):

1. **We trained on a subset.** Our experiments used a reduced subset (not the full training protocol / coarse+fine or full-resolution training) which reduces effective sample size and hurts generalization.

2. **Many classes and class imbalance.** Cityscapes has 20 classes and many small object classes. With limited data and a compact training recipe (input size 256×256, 50 epochs), the models struggle on small/rare classes and mIoU drops.

3. **Lower input resolution and training budget.** Standard Cityscapes benchmark runs use larger crop sizes (e.g. 512–769) and often longer training, stronger augmentations, multi-scale training and test-time augmentation. Our more modest recipe (256, 50 epochs) is intentionally reproducible and fast, but it trades off peak performance.

Cityscapes requires more data, larger inputs and longer training to reach literature-level mIoU."

Why Kvasir results are much higher. Kvasir (polyp segmentation) is a binary segmentation task with a much smaller label space and (in our runs) we trained on the full dataset (no "subset" restriction). This explains why both models achieve high performance (mIoU $\approx$ 0.882–0.883, mDice $\approx$ 0.935–0.936). Binary tasks are typically easier to learn and generalize faster; models converge to high Dice/IoU quickly.

The high scores are consistent with published Kvasir results and, depending on the exact variant/setup used in the cited papers, our baselines may be competitive or even exceed some previously reported numbers for vanilla DeepLab/U-Net baselines (see dataset and polyp papers [5, 10, 4]).

Model-to-model comparison.

- **Cityscapes:** DeepLabV3 > U-Net (mIoU: 0.5171 vs 0.4831). This is expected because DeepLab-style decoders (ASPP + dilated backbone) better capture multi-scale context which is important for urban scenes with varied object scales [1, 2].

- **Kvasir:** DeepLabV3 $\approx$ U-Net (mIoU: 0.8830 vs 0.8820). For binary polyp segmentation, both architectures (with the same ResNet-50 encoder and same training recipe) perform equivalently; any small differences are likely within experimental noise.

Comparison to literature (paper results).

- DeepLab variants with stronger backbones and larger inputs report substantially higher Cityscapes mIoU (e.g. DeepLab/DeepLabV3+ reports in the high 70s–low 80s mIoU under full benchmark settings) [1, 2]. The gap between our Cityscapes numbers and literature values is therefore largely explained by differences in *setup* (backbone size, input resolution, number of training iterations, use of coarse labels, multi-scale / multi-crop evaluation).

- For Kvasir-SEG, published works report a range of DeepLab / U-Net based scores depending on added attention modules, task-specific losses, and post-processing [5, 4]. Our baselines produce very competitive mIoU/mDice relative to many vanilla baselines (U-Net's mIoU on Kvasir-SEG is **0.7472** in [10] vs **0.882** (ours))

4.3 Cross-dataset results (Kvasir $\rightarrow$ Supervisely)

Tables 5 and 6 summarize the cross-dataset evaluation protocol (train on Kvasir-SEG, test in-domain on Kvasir-SEG and cross-domain on Supervisely Person). Below we interpret these results quantitatively, diagnose the dominant causes of degradation, and list concrete follow-ups to mitigate the observed domain gap.

Quantitative summary. DeepLabV3 and U-Net both experience substantial performance degradation under cross-dataset evaluation (when evaluated zero-shot on Supervisely Person after training on Kvasir-SEG). For DeepLabV3, mIoU drops from 0.8830 in-domain to 0.3959 cross-domain, an **absolute** drop of 0.4871 and a **relative** reduction of approximately 55.2%. U-Net exhibits a similar trend: mIoU decreases from 0.8820 in-domain to 0.3736 cross-domain, corresponding to an **absolute** drop of 0.5084 and a **relative** reduction of about 57.6%. Similar patterns hold for mDice and mPA: both metrics degrade substantially for both models, with DeepLabV3 slightly outperforming U-Net in absolute cross-domain performance (mIoU 0.3959 vs 0.3736) and slightly smaller relative drops, while the mean pixel accuracy (mPA) remains higher than the foreground-sensitive metrics, indicating that background pixels are still largely predicted correctly.

Interpretation and causes of the large domain gap. Although both datasets are treated as binary segmentation tasks in our protocol, the large cross-domain failure is expected and can be traced to several non-exclusive causes:

1. **Semantic / object mismatch:** Kvasir-SEG contains endoscopic images of polyps (medical images) whereas Supervisely Person contains natural images of people. Even when both are formulated as "foreground vs background", the *semantics* of the foreground class (appearance, typical shape, texture) differ drastically. A model that learns polyp appearance will not generalize to person appearance.

2. **Appearance and imaging modality shift:** endoscopy vs RGB natural photos differ in color distribution, illumination, texture, camera viewpoint and noise characteristics. These low-level shifts strongly degrade convolutional models trained on one modality.

3. **Scale and context differences:** objects in Kvasir are often small, bright and centrally located; person silhouettes in Supervisely vary in scale and context. Models trained on Kvasir learn different scale priors.

4. **Annotation policy / mask shape differences:** medical masks and person masks may follow different contour/labeling practices (tightness, inclusion of border pixels), which penalizes pixel-wise metrics when transferred.

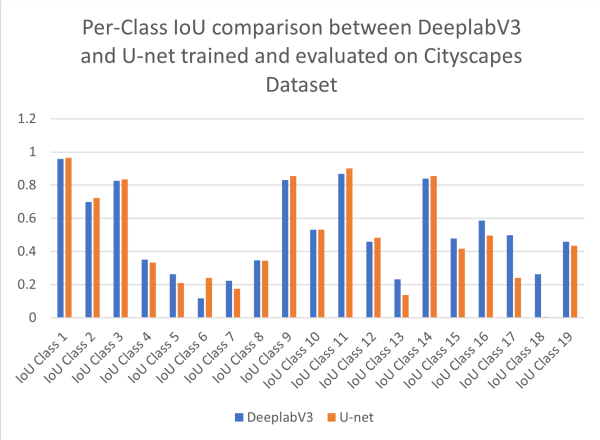
Metric sensitivity to cross-dataset evaluation The large **mIoU** and **mDice** drops (50–58% relative) indicate both gross segmentation failure and poor foreground localization on the target domain.

The smaller relative drop in **mPA** (≈26-28%) compared to mIoU suggests that many background pixels remain correctly classified (background is often dominant), while foreground localization (which affects mIoU much more) collapses. This pattern is typical when a model retains conservative background predictions but fails to find foreground instances it has never seen.

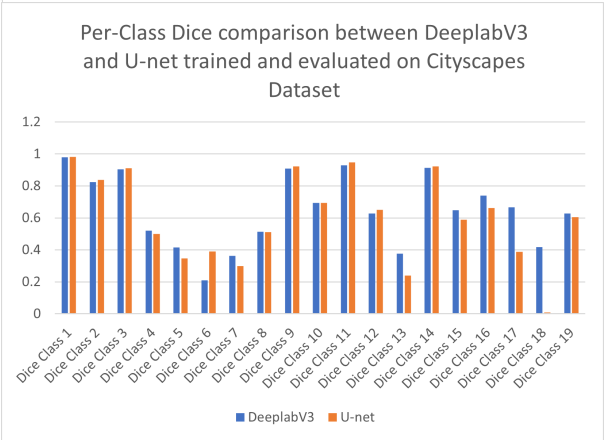
DeepLabV3's slightly smaller relative drops imply it is marginally more robust to this type of shift (likely because ASPP / larger receptive field offers some context robustness), but the improvement is small compared to the overall degradation.

4.4 Per-class histograms comparison

Figure 1 displays per-class IoU and per-class Dice distributions for DeepLabV3 and U-Net on Cityscapes. These histograms reveal which classes drive the overall mIoU / mDice differences and whether gains (or losses) are uniform or concentrated on a few classes.



(a) Per-class IoU Histogram Comparison



(b) Per-class Dice Histogram Comparison

Figure 1: Per-class visualizations,

High-level summary. DeepLabV3 has a per-class IoU mean of ≈ 0.517 (std ≈ 0.249) and mean Dice ≈ 0.646 (std ≈ 0.220). U-Net's per-class IoU mean (from these same per-class values) is ≈ 0.525 (std ≈ 0.280) and mean Dice ≈ 0.600 (std ≈ 0.270). Put differently: Dice favors DeepLabV3 on average, while per-class IoU is mixed and shows a comparable overall spread i.e., one model is not uniformly better class-by-class.

Notable per-class differences. Most classes are handled well by both models (several classes have IoU > 0.80 for both networks), but a small subset of classes produces the largest differences and therefore drives the mIoU gap. The table below (7) lists the classes with the largest absolute IoU differences ($\|\Delta\text{IoU}\| > 0.05$), with values rounded to three decimals ($\Delta = DeepLabV3 - UNet$).

Class	DL IoU	UNet IoU	ΔIoU	DL Dice	UNet Dice	ΔDice
5	0.262	0.210	+0.051	0.415	0.348	+0.067
6	0.117	0.241	-0.125	0.209	0.389	-0.180
13	0.232	0.137	+0.096	0.377	0.240	+0.137
16	0.585	0.137	+0.449	0.739	0.662	+0.076
17	0.498	0.855	-0.357	0.665	0.389	+0.277
18	0.263	0.418	-0.155	0.417	0.008	+0.409

Table 7: Classes with the largest per-class IoU differences ($\|\Delta\text{IoU}\| > 0.05$). Positive Δ indicates DeepLabV3 advantage; negative Δ indicates U-Net advantage.

Interpretation of the table and histograms :

Most large gains and losses are concentrated in a handful of classes (Table 7), which means overall mIoU changes are driven by a few "difficult" or "rare" classes rather than uniform shifts across all classes.

DeepLabV3 shows clear advantages for class 16 (+0.449 IoU) and class 13 (+0.096 IoU). These are large, class-specific wins that explain much of DeepLabV3's better performance on some metrics (notably mean Dice).

U-Net substantially outperforms DeepLabV3 on class 17 (0.357 IoU) and class 6 (0.125 IoU). In particular, U-Net's class-17 IoU is very high (≈ 0.855) while DeepLabV3 is middling ($\approx$0.498), which pulls U-Net's per-class IoU average up.

There are classes where Dice and IoU tell slightly different stories (e.g., class 18: IoU advantage for U-Net, but Dice advantage for DeepLabV3 is large in absolute value). This suggests differences in contour prediction vs. bulk overlap. Dice is more forgiving to region overlap while IoU penalizes per-class false positives/negatives differently.

Root causes for per-class behavior :

Class frequency / imbalance. Rare classes (small objects / infrequent labels) typically show much lower IoU and higher variance across models. If a few classes are rare, they will dominate mIoU drops.

Object size and boundary complexity. Small, thin or high-frequency boundary classes suffer more from downsampling (input size = 256) and from decoder receptive field trade-offs. DeepLab's ASPP can help for context but may lose thin structures; U-Net's skip connections can help recover edges, this aligns with some of the class-wise wins/losses observed.

4.5 Analysis of Ablation / Hyperparameter Results

The ablation / hyperparameter tables (3 and 4) summarize how single-design changes affect final segmentation performance. Below we discuss the main observations, possible explanations, and practical recommendations.

DeepLabV3: backbone depth and output stride Table 3 reports a baseline DeepLabV3 (ResNet-50, OS=8) mIoU of **0.5170**. Replacing the backbone with ResNet-101 yields a slightly lower mIoU of **0.5127** (Δ = -0.0043), while switching the output stride to OS=16 produces a substantially lower mIoU of **0.4647** (Δ = -0.0523).

These results suggest two conclusions. First, increasing backbone depth from ResNet-50 to ResNet-101 did not improve mIoU on this Cityscapes subset; the small decrease implies that the baseline dataset size / distribution or the training budget did not allow the larger model to realize its capacity advantage, and may indicate mild overfitting or optimization sensitivity for the deeper backbone. Second, the large drop for OS=16 confirms the expected role of output stride: a coarser feature map (OS=16) significantly harms boundary and fine-structure recovery, degrading overall mIoU and Dice. The relative sensitivity to output stride indicates that, for tasks and datasets where boundary fidelity matters, OS=8 is preferable despite higher compute.

Importantly, the output stride trade-off we observe is consistent with DeepLab analyses [1]: reducing output stride to 8 preserves fine spatial detail and boosts mIoU at the cost of memory and compute, whereas OS=16 speeds training but produces coarser feature maps and worse boundary fidelity.

Practical implication: For similar small-to-medium urban segmentation subsets, prefer ResNet-50 + OS=8 for a balanced trade-off between accuracy and compute. If resources permit and the dataset is substantially larger, re-evaluate ResNet-101 with extended training (or stronger regularization) as the larger backbone may pay off with more data.

U-Net: encoder initialization and input resolution Table 4 shows a strong dependence on encoder initialization. For the Cityscapes subset, the pretrained U-Net baseline attains mIoU = **0.4831**, whereas the randomly initialized encoder run yields mIoU = **0.3211** (= -0.1620). This large gap indicates that ImageNet pretraining substantially improves optimization and final performance on this dataset; random initialization struggles to reach competitive results within the same training budget.

For Kvasir (binary polyp segmentation), the U-Net baseline is high (mIoU = **0.8819**); increasing the input size to 512 produces a small decrease (mIoU = **0.8764**, Δ = -0.0055). The small negative change suggests that, for this dataset and model/training setup, increasing input resolution does not bring clear gains and may slightly hurt (likely due to optimization or the

need for adjusted learning rates / regularization at higher resolution). Given the high baseline performance on Kvasir, the model already captures the polyp structures well at 256 input size.

Practical implication: Always use ImageNet-pretrained encoders when training U-Net or Deeplab on limited data: the ablation shows large benefits in both convergence and final metrics. Increase input resolution only after validating optimization settings (LR, batch size) and when per-class analysis shows consistent small-object gains, otherwise the compute cost may not justify the marginal or negative improvements.

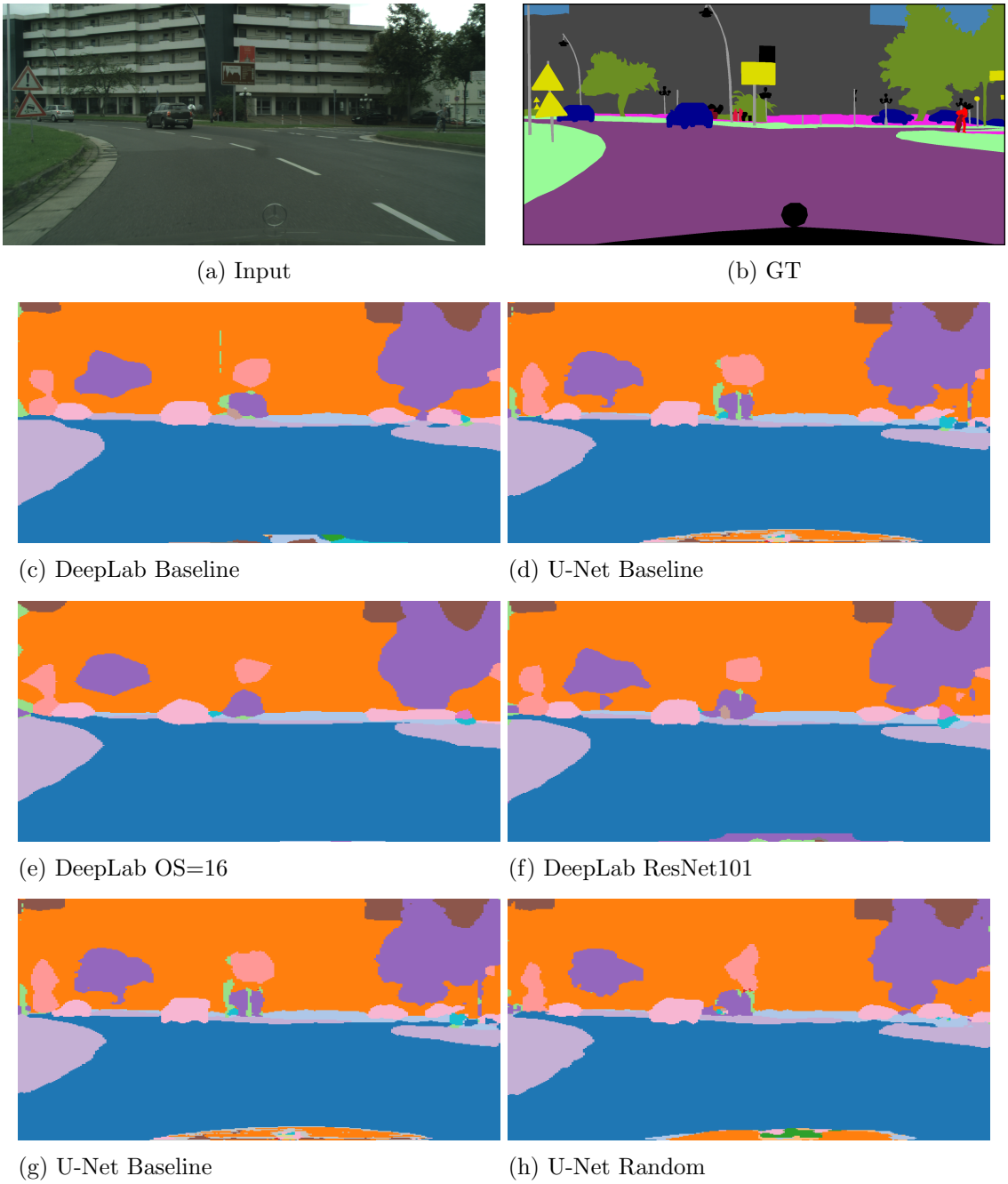
Reliability and statistical considerations The presented numbers appear as single-run results (unless otherwise noted). For more robust conclusions, critical ablations (backbone depth, encoder initialization) should be repeated with multiple random seeds and reported as mean $\pm$ std.

4.6 Visualization of inference (Cityscapes)

Qualitative interpretation. Figure 2 provides an illustrative comparison of how architectural and training choices affect predicted label shapes and boundaries on Cityscapes. Several observations emerge (with metric values corresponding to the ablation / hyperparameter tables reported earlier Tables 3, 4), and these quantitative differences manifest clearly in the visual examples.

- **Gap to ground truth:** Both DeepLab and U-Net (baselines) remain visibly far from the ground-truth segmentation. Large regions (roads, sidewalks, buildings) are generally recognized, but object boundaries, fine structures, and small instances are often misaligned or incomplete. This qualitative gap is consistent with the overall quantitative performance (mIoU ≈ 0.5 on the Cityscapes subset), confirming that the models have not yet learned the full structural complexity of the dataset under the limited training setup.
- **Boundary fidelity and thin structures:** DeepLab Baseline (OS=8) better preserves thin and elongated structures such as trees and traffic signs, compared with OS=16. The OS=8 predictions show more coherent and well-shaped object contours, whereas OS=16 frequently erodes, fragments, or completely misses narrow structures. This visual degradation directly corresponds to the substantial mIoU drop observed for OS=16 in Table 3.
- **Large objects and coarse regions:** U-Net and DeepLab tend to agree on large, homogeneous regions (road, buildings, sky). This is consistent with the relatively high mean pixel accuracy (mPA) seen in the quantitative tables despite lower mIoU.
- **Effect of initialization:** the U-Net random-init panel shows visibly noisier predictions, matching the large mIoU gap vs the pretrained U-Net.
- **Depth (ResNet-101):** the ResNet-101 DeepLab panel shows only modest visual improvement over ResNet-50, which suggests that, under the current dataset subset and training budget, it does not translate into substantially better per-image masks.

Important note on colors and visual comparison. The particular colors used to render predicted classes in these visualizations are **arbitrary label-to-color mappings** and therefore **do not carry semantic meaning**. What matters visually are the **class identities, shapes, spatial extent, boundary precision, and topological errors** (missed holes, merged instances, spurious islands), not the palette.



(a) Input (b) GT

(c) DeepLab Baseline (d) U-Net Baseline

(e) DeepLab OS=16 (f) DeepLab ResNet101

(g) U-Net Baseline (h) U-Net Random

Figure 2: Visual comparison of input, ground-truth segmentations, and model predictions on Cityscapes. Baselines (DeepLab and U-Net) are shown in the second row for comparison, while other variants (DeepLab OS=16, DeepLab ResNet101, U-Net random init) are shown in the subsequent rows.

5 Conclusion

This review compared two representative segmentation families, DeepLabV3 and U-Net, under a single, reproducible training recipe (ResNet-50 backbone, ImageNet pretraining, 256×256 input, 50 epochs) across Cityscapes, Kvasir-SEG and Supervisely Person. On the multi-class Cityscapes subset DeepLabV3 consistently outperformed U-Net in terms of mIoU, reflecting the benefit of atrous convolutions and ASPP for multi-scale contextual reasoning. For the binary polyp task (Kvasir-SEG) both architectures delivered nearly identical, high performance, which highlights that for simple foreground/background problems the choice between these two canonical architectures is less critical when using the same encoder and training recipe.

The ablations and hyperparameter study clarify which design choices matter in practice. Output stride strongly affects boundary-sensitive classes: OS=8 preserves fine spatial details and raises mIoU at the cost of memory and latency, while OS=16 degrades thin-object and boundary performance. ImageNet pretraining is essential for quick convergence and high final accuracy on limited data, as shown by the large performance drop for randomly initialized U-Net encoders. Increasing backbone capacity (ResNet-101) produced marginal gains under the current budget and dataset subset, suggesting that deeper models require larger datasets or longer training to realize their potential.

Cross-dataset experiments expose severe domain sensitivity: zero-shot transfer from Kvasir to Supervisely yielded dramatic drops in mIoU and Dice for both models. The root causes are semantic and modality mismatch (endoscopic polyps vs natural person images), differences in scale and scene context, and annotation-policy differences. In this setting DeepLabV3 showed a small edge in absolute cross-domain numbers, but the difference is minor beside the overall failure to generalize without adaptation. Practically, this indicates that zero-shot cross-domain use is more reliable between closely aligned datasets.

Finally, per-class analyses revealed that a handful of difficult or rare classes drive most of the mIoU variance. Global metrics alone can therefore mask class-specific weaknesses; reporting per-class IoU and Dice alongside qualitative overlays is necessary to understand failure modes and to guide remedies (e.g., higher input resolution, class reweighting, boundary losses, or task-specific modules). For practitioners, our recommendation is to adopt ResNet-50 + ImageNet pretraining and OS=8 as a robust starting point, then invest compute into dataset-specific adjustments (higher resolution, longer schedules, or semi-/self-supervised domain adaptation) when targeting benchmark performance or cross-domain robustness.

Before final submission, we recommend completing multi-seed repeats for the critical ablations, adding per-class delta tables and representative qualitative examples, and reporting compute costs (training time and peak memory) for the heaviest configurations to be able to weigh accuracy against practical constraints.

Appendix: Data Availability and Reproducibility

All code, experiment logs, result CSVs, pretrained checkpoints, and inference visualizations used in this review are publicly available. Exact file names, download links, and checksums are listed below to enable exact reproduction of the reported results.

Code and Notebook

The main notebook used to run and reproduce the experiments is available at:

`https://github.com/RamiAridi03/S9-CE6190-ImageSegmentationReview-Unet-DeepLabV3.git`
Notebook file: `ce6190-segmentation-review-unet-deeplabv3.ipynb`.

And at: https://www.kaggle.com/code/ramiaridi/ce6190-segmentation-review-unet-deeplabv3

Pretrained Models, CSV Results, and Visualizations

A single archive containing pretrained models, evaluation CSVs, and inference visualizations used to produce figures is available at:
https://www.kaggle.com/datasets/ramiaridi/ce6190-segmentation-review-unet-deeplabv3-dataset.

Datasets (Mirrors and Download Dates)

- Cityscapes (subset): https://www.kaggle.com/datasets/xiaose/cityscapes; downloaded on <2025-10-15>.
- Kvasir-SEG: https://www.kaggle.com/datasets/abdallahwagih/kvasir-dataset-for-classification-and-segmentation; downloaded on <2025-10-15>.
- Supervisely Person: https://www.kaggle.com/datasets/tapakah68/supervisely-filtered-segmentation-person-dataset; downloaded on <2025-10-15>.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. URL https://arxiv.org/abs/1706.05587. Documentation: https://smp.readthedocs.io/en/latest/models.html#id44.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation (deeplabv3+). In *ECCV*, 2018. URL https://arxiv.org/abs/1802.02611. Documentation: https://smp.readthedocs.io/en/latest/models.html#id44.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. URL https://www.cityscapes-dataset.com.

[4] Feixiang Du, Zhongliang Wang, Joel C. M. Than, Hadi Nabipour Afrouzi, and Nianxia Qian. Doubleaanet: Auxiliary attention and area adaptive loss for robust polyp segmentation. *Journal of Computational Design and Engineering*, 12(8):1–13, 2025. doi: 10.1093/jcde/qwaf061. URL https://doi.org/10.1093/jcde/qwaf061.

[5] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. de Lange, D. Johansen, and H. D. Johansen. Kvasir-seg: A segmented polyp dataset. *arXiv:1911.07069*, 2019. URL https://arxiv.org/abs/1911.07069.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. doi: 10.1109/CVPR.2015.7298965. URL https://doi.org/10.1109/CVPR.2015.7298965.

[7] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. URL https://arxiv.org/abs/1505.04597. Documentation: https://smp.readthedocs.io/en/latest/models.html#id44.

[9] Supervisely. Supervisely persons dataset. https://ecosystem.supervisely.com/projects/persons, 2024. Accessed 2024.

[10] Nikhil Kumar Tomar, Debesh Jha, Ulas Bagci, and Sharib Ali. Tganet: Text-guided attention for improved polyp segmentation. In *MICCAI (Lecture Notes in Computer Science)*, 2022. URL https://arxiv.org/abs/2205.04280. Provides Kvasir-SEG baselines and compares to U-Net, DeepLab variants.