Efficient Building Segmentation from Aerial Imagery via Lightweight Ensembles of DeepLabV3+ Encoders

Nadeem Gauhar
Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur, Rajasthan, India

Dr. Manish Rai
Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur, Rajasthan, India

***Abstract*—Extracting building footprints from aerial imagery is an old remote-sensing problem with uses in urban planning, disaster response, and population mapping. Most published work chases the highest accuracy on a single benchmark and leaves a different question open: how much accuracy do you give up if you swap a heavy backbone for a light one, and how does inference cost actually move? This paper compares three DeepLabV3+ variants that differ only in their encoder. The encoders are MobileNetV2 (4.4M parameters), EfficientNet-B3 (11.7M), and ResNet50 (26.7M). All three are trained together on the Massachusetts Buildings and Inria Aerial Image Labeling datasets, evaluated on the Massachusetts test split with sliding-window inference and 4-way test-time augmentation, and combined into a validation-IoU-weighted ensemble. The ensemble reaches an IoU of 0.5501 ± 0.0815 and an F1 of 0.7066 ± 0.0672 at threshold 0.45, slightly above the strongest single model. Profiling on a Tesla T4 GPU shows that FLOPs are a bad proxy for latency. EfficientNet-B3 has the lowest FLOPs of any model tested but is more than twice as slow as ResNet50 at the same input size. MobileNetV2 sustains 158 frames per second with only a small accuracy loss. The ensemble inherits high recall from the lighter models and high precision from ResNet50, giving a more balanced precision-recall profile than any single backbone. We also report failure modes, fine-tuning behaviour, and a center-crop evaluation pitfall encountered during the work.**

***Index Terms*—semantic segmentation, building extraction, aerial imagery, DeepLabV3+, model ensemble, efficiency benchmark, remote sensing**

I. Introduction

Automatic extraction of building footprints from aerial and satellite imagery is useful for cadastral mapping, urban growth monitoring, post-disaster damage assessment, and population estimation. Manual digitisation of polygons is still common practice in many municipalities. It is slow, expensive, and hard to keep current. Encoder-decoder convolutional networks have therefore taken over the task in the last several years [1], [5], [6].

Much of the recent work in this area is organised around one goal: get the highest possible accuracy on a benchmark. That goal is fine, but it leaves several practical questions open. Which encoder should you choose when GPU time, deployment latency, or memory budget are tight? How much accuracy is given up by switching from a heavy ResNet to a lightweight mobile backbone? Do reported FLOP counts predict real wall-clock latency, or do other factors dominate? And when a small ensemble is used, which encoder combinations are worth the additional inference cost?

We address these questions empirically, not architecturally. We do not propose a new layer, loss, or attention mechanism. The segmentation framework is fixed to DeepLabV3+ [1], a well-established encoder-decoder with atrous separable convolutions, and only the encoder is varied. The three encoders compared are MobileNetV2 [3], EfficientNet-B3 [4], and ResNet50 [2]. They span more than a sixfold range of parameter counts and represent three different design philosophies: inverted residuals, compound-scaled blocks with squeeze-and-excitation, and standard residual blocks. Training data combines the Massachusetts Buildings dataset [7] with the much larger Inria Aerial Image Labeling Benchmark [8]. Evaluation is restricted to the Massachusetts test set, which exposes cross-domain behaviour. A validation-weighted ensemble is built at the prediction-averaging level.

The contributions are as follows.

- A controlled, single-architecture comparison of three encoders for building segmentation. Hyperparameters, augmentations, loss function, and evaluation protocol are identical across the three runs.
- A profiling study on a Tesla T4 GPU reporting parameters, multiply-accumulate FLOPs, peak memory, frames per second, and milliseconds per image. The numbers show that FLOPs do not predict latency for this family of models.
- Sliding-window evaluation with 4-way test-time augmentation, threshold sensitivity analysis on the validation set, and a validation-IoU-weighted ensemble whose principal benefit is a more balanced precision and recall profile rather than a large IoU lift.
- A transparent account of one fine-tuning experiment that helped only the smallest model, and a center-crop evaluation pitfall we encountered. Both are documented so that other practitioners can avoid the same mistake.

Section II surveys related work. Section III describes the

datasets, architecture, training, and evaluation protocol. Section IV reports profiling, single-model, fine-tuning, and ensemble results. Section V discusses what the numbers mean and where the limits are. Section VI concludes.

II. Related Work

A. Building extraction from aerial and satellite imagery

Building extraction has a long history in remote sensing, with early methods relying on hand-crafted spectral, geometric, and textural features. The shift to convolutional neural networks was accelerated by Mnih's release of the Massachusetts Buildings and Roads datasets [7], which provided the first large-scale aerial imagery benchmark with rasterised OpenStreetMap labels. Maggiori et al. later introduced the Inria Aerial Image Labeling benchmark [8], which deliberately splits train and test into different cities to expose generalisation failures.

Among recent encoder–decoder approaches, U-Net [5] and its many descendants are still a popular choice. Pan et al. showed that a U-Net with skip connections can segment dense urban village buildings in Guangzhou with IoU above 0.90 when trained and tested on co-located 8-band WorldView imagery at 0.5 m resolution [11]. Benchabana et al. report F1 scores above 0.97 on the WHU aerial imagery dataset using a superpixel-based pipeline with variational autoencoder feature extraction [12]. Irwansyah et al. apply a U-Net with a ResNet34 encoder to aerial photography of Pasar Minggu, Jakarta, and report a comparison table showing that published F1 scores in this domain span a wide range from about 0.54 to 0.89 depending on dataset density, urban morphology, and train–test split policy [13].

The closest prior work to ours is Prathap and Afanasyev [9], who train an ensemble of three U-Net variants on SpaceNet imagery for Vegas, Paris, Shanghai, and Khartoum. Their three models differ in input modality (RGB, multispectral, multispectral plus OpenStreetMap layers) rather than in encoder backbone, and they report city-wise F1 scores between 0.58 and 0.88. Ayala et al. propose a U-Net with a ResNet34 encoder for sub-pixel building and road detection from fused Sentinel-1/Sentinel-2 imagery, using a Combo Loss that linearly combines binary cross-entropy and Dice and applying eight-way test-time augmentation [10]. Their best building IoU is 0.6027 at upsampled 2.5 m resolution. Liu and Cai apply a cascaded fully convolutional network to Gaofen-2 imagery of Changchun, China, for building roof change detection and report quality rates of 91% and 88% on industrial and residential buildings, respectively [14].

Outside segmentation, Xu et al. address building damage classification on Haiti, Mexico City, and Indonesia earthquake imagery with a twin-tower convolutional network. They document a substantial AUC drop, from 0.79 in-domain to 0.62 out-of-domain, when models are not retrained on the target region [15]. This cross-region degradation is directly relevant to our setting, in which most training data is from European and North American Inria cities while the test set is the Massachusetts holdout.

B. Lightweight backbones and the FLOPs–latency gap

Three encoder families are compared in the proposed method. ResNet [2] introduced residual connections so that very deep networks could be trained, and is still the standard backbone for many segmentation systems. MobileNetV2 [3] replaces standard convolutions with inverted residuals and linear bottlenecks, using depthwise separable convolutions to keep parameter counts very low. EfficientNet [4] applies compound scaling to width, depth, and input resolution, and uses squeeze-and-excitation blocks to reweight feature channels.

A repeated observation in the efficient-architecture literature is that FLOP count is an imperfect predictor of inference latency. Depthwise separable convolutions and squeeze-and-excitation modules have low FLOPs but limited arithmetic intensity. They leave GPU compute units idle while waiting for memory traffic. Our profiling results in Section IV-B reproduce this effect.

C. Ensemble methods for semantic segmentation

Output-level averaging of independently trained segmentation networks is a long-established technique for marginal accuracy gains [9], [10]. Ensemble weights can be uniform, validation-loss-weighted, or learned. We adopt validation-IoU weighting because it requires no additional training and because each model's contribution maps cleanly to its own holdout performance. Our ensemble contribution is not a methodological novelty. It is a careful empirical characterisation of when an ensemble helps and what changes in the precision and recall profile, particularly under cross-domain evaluation.

D. Comparative summary

Table I summarises the most relevant prior works alongside our setting. Reported IoU values vary by more than 0.35 across studies. The variation is driven mostly by resolution, spectral bands, dataset density, and whether train and test are drawn from the same distribution. Our work targets a deliberately constrained operational regime: 1 m RGB imagery, cross-domain training, a small in-domain training set, and a fixed segmentation framework. This lets the encoder choice be isolated as the experimental variable.

III. Methodology

Fig. 1 gives an overview of the proposed method. Each component is described in detail in the following subsections.

A. Datasets

Two publicly available aerial imagery benchmarks are used. The Massachusetts Buildings dataset [7] contains 151 RGB tiles of 1500×1500 pixels at about 1 m per pixel, drawn from the greater Boston area. The standard split is 137 training, 4 validation, and 10 test images, with rasterised OpenStreetMap polygons as ground truth. The dataset is small, but it uses a single coherent geographic distribution. That makes it well suited for in-domain test evaluation.

TABLE I

SUMMARY OF RELATED WORK ON DEEP-LEARNING-BASED BUILDING EXTRACTION. NUMBERS ARE TAKEN FROM THE CITED PAPERS AND ROUNDED TO THE PUBLISHED PRECISION. OUR ROW REPORTS THE MASSACHUSETTS TEST-SET ENSEMBLE RESULT.

Reference	Architecture	Dataset	Resolution	Reported metric
Prathap & Afanasyev [9]	U-Net ensemble (3 input variants)	SpaceNet (4 cities)	0.30 m, 8-band	F1 0.58–0.88
Ayala et al. [10]	U-Net + ResNet34, Combo Loss + TTA	Sentinel-1/2 + OSM, 44 Spanish cities	10 m $\rightarrow$ 2.5 m	IoU 0.60 (build.)
Pan et al. [11]	U-Net (modified)	WorldView-2, Guangzhou	0.50 m, 8-band	IoU $>$0.90
Benchabana et al. [12]	SLIC + VAE + CNN	WHU aerial + others	0.30 m	F1 0.97
Irwansyah et al. [13]	U-Net + ResNet34	Aerial photo, Jakarta	0.25 m	Test acc. 0.87
Liu & Cai [14]	Cascaded FCN	Gaofen-2, Changchun	1 m	Quality 0.88–0.91
Xu et al. [15]	Twin-tower CNN (damage class.)	Haiti / Mexico / Indonesia	0.30 m	AUC 0.62–0.86
Proposed method	**DeepLabV3+ ensemble (3 encoders)**	**Mass. Buildings + Inria**	**1 m, RGB**	**IoU 0.55, F1 0.71**

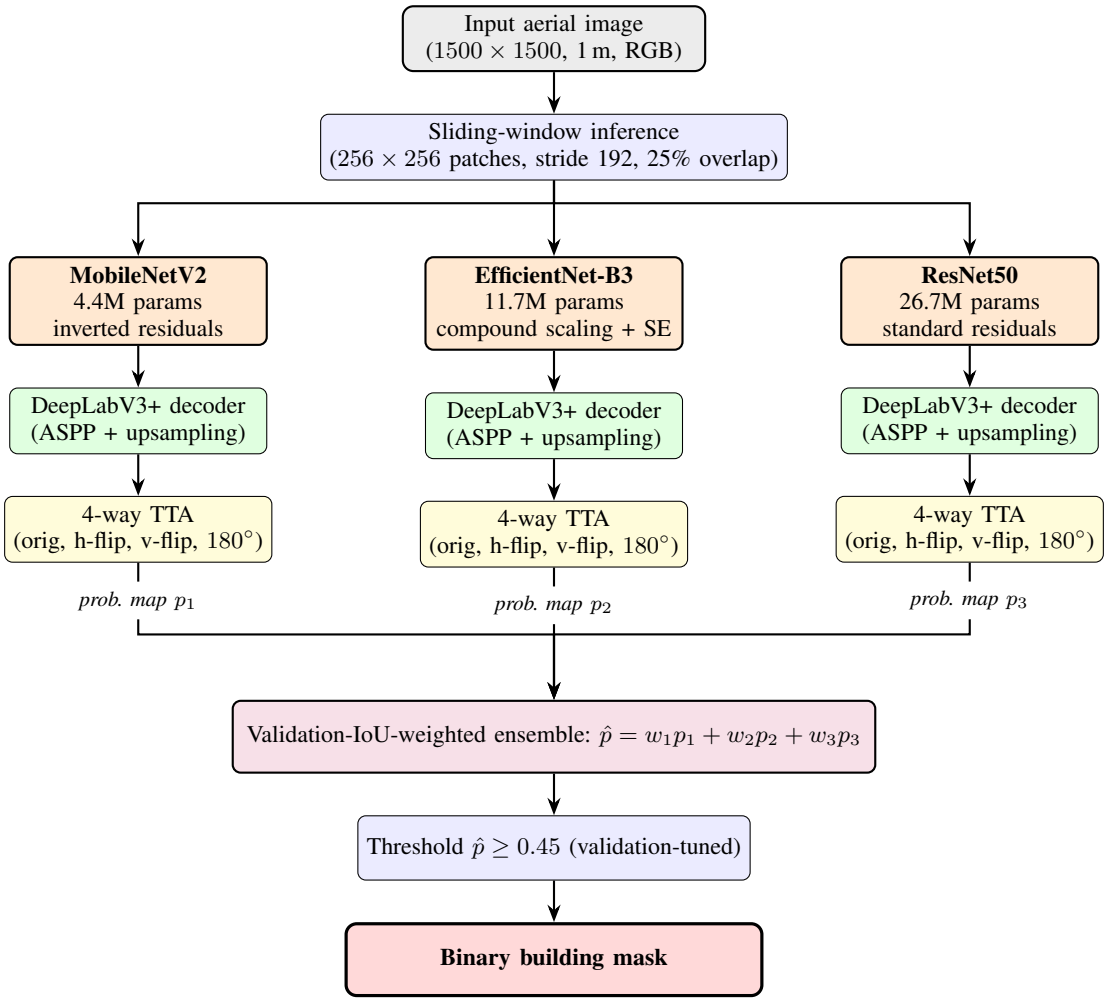

Fig. 1. Overview of the proposed method. Three DeepLabV3+ models with different encoder backbones (MobileNetV2, EfficientNet-B3, ResNet50) process every test image through sliding-window inference and 4-way test-time augmentation independently. The resulting probability maps are combined with weights proportional to each model's best validation IoU, and the weighted average is thresholded at the validation-selected value of 0.45 to produce the final binary building mask.

The Inria Aerial Image Labeling Benchmark [8] contains 360 RGB tiles of 5000×5000 pixels at 0.30 m per pixel, drawn from ten cities. The split is deliberately constructed so that train and test cities never overlap. We use a pre-tiled

redistribution of Inria at 1024×1024 pixels to reduce I/O cost.

After preprocessing, the unified training corpus contains 2,441 images (137 from Massachusetts and 2,304 from Inria). The validation set contains 580 images (4 from Massachusetts and 576 from Inria). The test set is the 10-image Massachusetts holdout. Inria images are downsampled to 512×512 before random cropping so that their spatial scale roughly matches that of Massachusetts. This prevents the much higher Inria building density at native resolution from biasing the learned receptive field.

B. Architecture

DeepLabV3+ [1] is used as the segmentation framework for all three models. DeepLabV3+ couples an encoder backbone with an atrous spatial pyramid pooling (ASPP) module that probes the encoder feature map at multiple dilation rates. A lightweight decoder then fuses high-level semantic features with low-level encoder features to recover spatial detail at object boundaries.

The three encoders compared are MobileNetV2 [3], EfficientNet-B3 [4], and ResNet50 [2]. All encoders are initialised with ImageNet-pretrained weights. The decoder, ASPP module, and classification head are kept identical across all three models. Any difference in segmentation performance is therefore attributable to the encoder. All models are implemented using the segmentation_models.pytorch library [16].

C. Training procedure

All three models are trained with identical hyperparameters. Inputs are randomly cropped to 256×256 pixels. We use a batch size of 8, the Adam optimiser [17] with a learning rate of 10^{-4} and weight decay 10^{-5}, and cosine annealing with warm restarts [18] with $T_0 = 5$ and $T_{\text{mult}} = 2$. Each model is trained for 15 epochs. Mixed-precision training is enabled.

The loss function is a weighted sum of soft Dice loss [19] and binary cross-entropy applied to logits:

$$\mathcal{L} = 0.6 \cdot \mathcal{L}_{\text{Dice}} + 0.4 \cdot \mathcal{L}_{\text{BCE}}. \tag{1}$$

This is conceptually similar to the Combo Loss used by Ayala et al. [10], but with a slight bias towards Dice. The bias addresses the foreground–background class imbalance characteristic of building segmentation, where buildings typically occupy 10 to 20% of pixels.

Augmentations are kept deliberately light: random horizontal and vertical flips, random 90-degree rotations, and modest brightness, contrast, and HSV shifts, all implemented in Albumentations [20]. We evaluated heavier augmentations such as elastic deformation, grid distortion, and optical distortion in pilot experiments. They increased training time substantially without measurable improvement on the validation set, which is consistent with the rectilinear nature of building footprints.

For reproducibility, we enable cudnn.deterministic, seed worker processes, and use seeded random generators. Each model checkpoints its best validation-IoU state during training. Per-epoch training and validation losses, IoU, and pixel accuracy are logged to disk. No reported number relies on memory or post-hoc reconstruction.

D. Evaluation protocol

The Massachusetts test images are 1500×1500 pixels, much larger than the 256×256 training crops. A naive solution is to resize the test image to 256×256 and predict in one pass. We initially tried this and got anomalously low IoU values (below 0.20 for all three models) despite visually plausible probability maps. That prompted a closer look at the evaluation pipeline. Resizing destroys most of the building detail. A center-crop alternative discards 75% of every test image and biases evaluation towards a single region.

We therefore use a sliding-window inference scheme. The full 1500×1500 image is padded to a multiple of the patch size using border reflection, then traversed with a sliding 256×256 window at stride 192 (25% overlap). Predictions in overlapping regions are averaged. Reflective padding avoids boundary artefacts at the image edges. The overlap suppresses tiling seams that would otherwise be visible at patch boundaries.

To reduce single-orientation bias, we additionally apply 4-way test-time augmentation. Predictions are averaged across the original image, its horizontal flip, its vertical flip, and its 180-degree rotation. Each of these forward passes uses sliding-window inference as above.

The decision threshold is selected on the validation set rather than fixed at the conventional 0.50 value. The optimal operating point depends on building density and on the relative cost of false positives and false negatives in the deployment setting. We sweep thresholds from 0.30 to 0.60 in steps of 0.05 and report the value that maximises validation IoU.

We report five metrics: IoU (Jaccard index), pixel accuracy, precision, recall, and F1 score. Each is computed per test image and reported as mean $\pm$ standard deviation across the 10-image test set. The standard deviation matters. In a small test set, average accuracy without dispersion can hide substantial per-image variance.

E. Ensemble construction

The ensemble is constructed at the prediction-averaging level. Each of the three trained models produces a probability map for the test image (post sigmoid, post sliding-window TTA averaging). The three probability maps are combined as a weighted average,

$$\hat{p} = w_1 p_1 + w_2 p_2 + w_3 p_3, \tag{2}$$

where the weights w_i are proportional to each model's best validation IoU and normalised to sum to one. The weighted probability map is then thresholded with the validation-tuned threshold to produce the final binary mask.

This procedure has three properties that we consider desirable for a reproducibility-focused study. First, the weights require no additional training and depend only on validation performance that is computed during model selection anyway. Second, no model is excluded a priori; the weights down-weight rather than discard underperforming members. Third, the procedure is symmetric across encoders, so the ensemble result is not advantaged by ordering or by an asymmetric voting rule.

F. Profiling protocol

For each individual model and the ensemble, we measure the following on a Tesla T4 GPU at 256×256 input size and batch size 1: parameter count (using PyTorch summary), multiply–accumulate FLOPs (using thop), peak GPU memory during forward pass, frames per second over 100 forward passes after a 20-pass warm-up, and milliseconds per image. For the ensemble, parameters and FLOPs are summed across members. FPS and milliseconds reflect a sequential pass through all three encoders.

We report thop FLOP counts but flag a known limitation: thop does not always track the cost of squeeze-and-excitation modules accurately. This causes EfficientNet-B3 to appear lower in FLOPs than the published value would suggest. The caveat is preserved in Section IV-B for the reader's interpretation.

IV. Experiments and Results

A. Training behaviour

All three models converge cleanly within 15 epochs. Fig. 2 shows training and validation loss, IoU, and pixel accuracy curves. Validation IoU rises with model capacity at the end of training: MobileNetV2 reaches 0.6399, EfficientNet-B3 reaches 0.6574, and ResNet50 reaches 0.6759. Validation curves track training curves closely, indicating no significant overfitting in any of the three models. The cosine annealing schedule with warm restarts produces visible loss spikes at epoch boundaries that the optimiser then re-flattens.

B. Profiling results

Table II reports profiling results on a Tesla T4 at 256×256.

FLOPs are not a reliable predictor of latency on this hardware. EfficientNet-B3 has the lowest FLOPs of any model in the table, yet it runs more than twice as slowly per image as ResNet50, which has roughly fourteen times more FLOPs. The reason is that EfficientNet-B3's depthwise separable convolutions and squeeze-and-excitation modules have low arithmetic intensity. They perform little compute per byte of memory access and so fail to saturate the T4's compute units. ResNet50's 3×3 standard convolutions, by contrast, are exactly the operation that cuDNN is most aggressively optimised for.

MobileNetV2 dominates the throughput axis. At 158 FPS it is more than two and a half times faster than ResNet50 and nearly three times faster than EfficientNet-B3, while using only one-sixth the parameters of ResNet50 and one-third the parameters of EfficientNet-B3.

Ensembling carries a sequential cost. The ensemble's effective throughput is bounded by the sum of its members' latencies and lands at roughly 30 FPS, an order of magnitude slower than MobileNetV2 alone.

Fig. 3 visualises the accuracy and speed trade-off. The ensemble is positioned in the upper-left corner (highest IoU, lowest FPS), MobileNetV2 in the lower-right (highest FPS, lowest IoU). EfficientNet-B3 and ResNet50 occupy intermediate positions.

C. Single-model and ensemble accuracy

Table III reports test-set metrics for each individual model and for the validation-IoU-weighted ensemble.

ResNet50 is the strongest single model on IoU and pixel accuracy, but it has the lowest recall of any model (0.665). It misses more buildings than its lighter counterparts. MobileNetV2 and EfficientNet-B3 both have higher recall (above 0.72) but lower precision. The ensemble inherits the high recall of the lighter models (0.752, marginally higher than any single member) while keeping precision in the same range as EfficientNet-B3 (0.671). The net effect is a modest IoU gain over the best single model (0.550 vs. 0.543), a slightly higher F1 (0.707 vs. 0.701), and visibly more balanced precision and recall.

This is the principal practical benefit of the ensemble. The IoU improvement is small and arguably within noise. The change in operating profile is more substantial. A practitioner who needs to maximise recall (do not miss buildings) gains about 9 percentage points by using the ensemble instead of ResNet50 alone. The cost is roughly four times the inference budget.

Fig. 4 plots this trade-off across all five metrics with error bars.

D. Threshold sensitivity

Decision thresholds were swept on the validation set in steps of 0.05 between 0.30 and 0.60. The validation-optimal threshold for the ensemble was 0.45. Threshold 0.50 produced an IoU of 0.5478 and threshold 0.40 produced 0.5433, so the ensemble result is not very sensitive to threshold choice. The curve is shallow around the maximum. We retain 0.45 because that is what validation selected.

E. Fine-tuning experiment

To explore whether limited in-domain Massachusetts data could improve cross-domain performance, we fine-tuned each pretrained model for 5 additional epochs on the Massachusetts training set alone (137 images). The learning rate was reduced to 2×10^{-5} to preserve learned features. Table IV summarises the outcome.

Fine-tuning helped only the smallest model. MobileNetV2 gained almost 5 IoU points. EfficientNet-B3 lost about 3 points. ResNet50 was essentially unchanged. We attribute this asymmetry to overfitting capacity: with only 137 in-domain images, larger networks have more parameters than the in-domain signal can support and drift away from useful features learned on the much larger Inria distribution. We therefore use the fine-tuned MobileNetV2 in the final ensemble but the original (Inria + Massachusetts) checkpoints for EfficientNet-B3 and ResNet50.

F. Qualitative results

Fig. 5 shows best-case, median-case, and worst-case test predictions for the ensemble. The model captures dense suburban building grids well. Individual house outlines are reproduced reasonably faithfully in residential areas. Failure modes

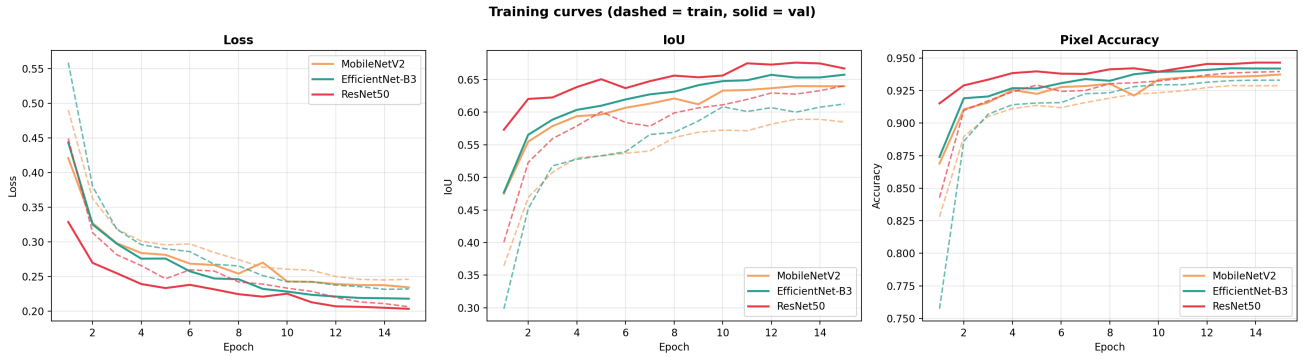

Fig. 2. Per-epoch training (dashed) and validation (solid) curves for the three models. Left: combined Dice + BCE loss. Centre: validation IoU. Right: pixel accuracy. All three models converge by epoch 12–15. The cosine annealing schedule with warm restarts produces the visible perturbations at epochs 5 and 10.

TABLE II

PROFILING ON TESLA T4 GPU AT 256 × 256 INPUT, BATCH SIZE 1. FLOPS REPORTED VIA THOP; SEE SECTION IV-B FOR CAVEAT REGARDING SQUEEZE-AND-EXCITATION ACCOUNTING.

Model	Params (M)	FLOPs (G)	FPS	Peak Mem (MB)	ms/img
MobileNetV2	4.38	1.54	158.5	40.7	6.31
EfficientNet-B3	11.68	0.65	54.5	81.5	18.35
ResNet50	26.68	9.23	125.3	145.5	7.98
Ensemble	42.74	11.42	30.6	160.1	32.64

TABLE III

TEST-SET METRICS ON MASSACHUSETTS HOLDOUT (10 IMAGES), REPORTED AS MEAN ± STANDARD DEVIATION. ALL SINGLE-MODEL RESULTS USE SLIDING-WINDOW INFERENCE WITH 4-WAY TTA; THE ENSEMBLE ADDITIONALLY USES VALIDATION-IOU WEIGHTING AND A THRESHOLD OF 0.45.

Model	IoU	Acc.	Prec.	Recall	F1
MobileNetV2	0.488 ± 0.082	0.860 ± 0.050	0.594 ± 0.099	0.725 ± 0.069	0.650 ± 0.075
EfficientNet-B3	0.539 ± 0.084	0.889 ± 0.040	0.673 ± 0.066	0.726 ± 0.090	0.697 ± 0.067
ResNet50	0.543 ± 0.083	0.901 ± 0.038	0.751 ± 0.058	0.665 ± 0.108	0.701 ± 0.071
Ensemble	$\mathbf{0.550 \pm 0.082}$	$\mathbf{0.891 \pm 0.039}$	0.671 ± 0.074	$\mathbf{0.752 \pm 0.087}$	$\mathbf{0.707 \pm 0.067}$

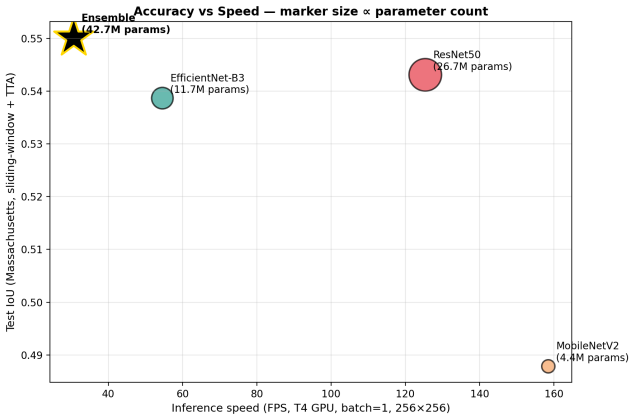

Fig. 3. Test-set IoU plotted against inference speed (FPS) on a Tesla T4 at 256 × 256 input. Marker area is proportional to parameter count. The ensemble (star) gains less than 0.01 IoU over the best single model (ResNet50) at roughly one-quarter the throughput.

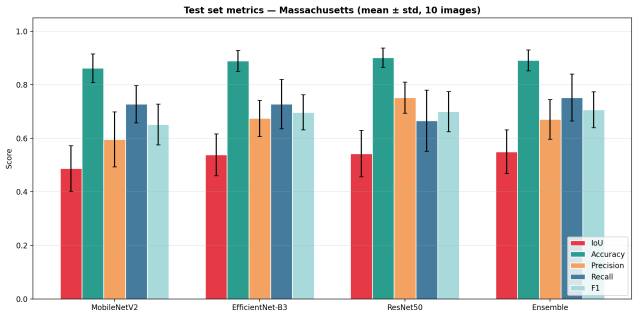

Fig. 4. Per-image mean and standard deviation of the five metrics across the 10-image Massachusetts test set, for the three single models and the ensemble. The ensemble's most distinctive feature is high recall combined with mid-range precision.

cluster around three categories: very small or very narrow buildings that are partially submerged in the receptive field; commercial structures with unusually reflective or coloured roofs that do not appear in the training distribution; and

TABLE IV
EFFECT OF 5-EPOCH FINE-TUNING ON MASSACHUSETTS TRAINING DATA ALONE. IOU VALUES ARE COMPUTED VIA SLIDING-WINDOW INFERENCE ON THE TEST SET, WITHOUT TTA, TO ISOLATE THE EFFECT OF FINE-TUNING FROM OTHER EVALUATION-TIME CHOICES.

Model	Pretrained IoU	Fine-tuned IoU	Delta
MobileNetV2	0.4260	0.4755	$+0.0495$
EfficientNet-B3	0.5194	0.4911	-0.0283
ResNet50	0.5283	0.5267	-0.0016

densely packed adjacent buildings that the model merges into a single connected region.

V. DISCUSSION

A. What the ensemble actually buys

The ensemble's principal benefit is not a large IoU lift. It is a more balanced precision and recall profile. Practitioners should choose the ensemble when missing buildings is more costly than over-segmenting, and choose ResNet50 alone when the opposite is true. For latency-constrained settings, MobileNetV2 alone gives up about 0.06 IoU but runs at roughly five times the throughput of the ensemble.

B. Why FLOPs do not predict latency on this hardware

EfficientNet-B3 has the lowest FLOPs in our profiling table but is the slowest single model. There are two reasons. Depthwise separable convolutions perform fewer operations per byte of memory traffic than dense 3×3 convolutions, so they do not saturate GPU compute units. Squeeze-and-excitation modules also introduce small fully-connected operations interleaved with convolutional layers; these are difficult to fuse and add launch overhead disproportionate to their nominal compute cost. ResNet50, in contrast, consists almost entirely of dense 3×3 convolutions that cuDNN heavily optimises. The takeaway is that papers reporting FLOPs as an efficiency proxy may significantly mis-rank backbones on real GPUs. Wall-clock latency on the target deployment hardware should be measured directly.

C. Cross-domain limitations

Our training distribution is approximately 95% Inria (predominantly European cities) and 5% Massachusetts. The test set is exclusively Massachusetts. The 0.55 IoU we report is in the same range as cross-city SpaceNet results from Prathap and Afanasyev [9] and the building IoU reported by Ayala et al. [10], but well below the 0.90+ achieved by Pan et al. [11] on co-located WorldView-2 imagery, and below the 0.97 F1 reported by Benchabana et al. [12] on WHU. The dominant explanation is not architecture. It is the combination of resolution (1 m vs. 0.30 to 0.50 m), spectral bands (RGB only vs. 8-band), and train–test domain match. A practitioner deploying our pipeline in a new city should expect to fine-tune on a small amount of in-domain data, a regime that our fine-tuning experiment suggests is non-trivial: only the smallest model benefited.

D. The center-crop pitfall

Our initial test evaluation used a center crop of 256×256, which produced IoU values below 0.20 across all three models despite visually plausible probability heatmaps. The cause was simple but not obvious: a center crop discards 75% of every 1500×1500 test image. The diagnostic was to overlay the per-pixel probability map onto the input image and observe that the model was correctly identifying buildings outside the crop region. When reported IoU is anomalously low for a model whose qualitative outputs look reasonable, the evaluation pipeline should be inspected before the model is.

E. Limitations and threats to validity

The Massachusetts test set contains 10 images, which is too small to detect IoU differences below about ± 0.02 with statistical confidence. Our 0.007 ensemble lift over the best single model is therefore best read as suggestive rather than significant. Inria's pre-tiled redistribution at 1024×1024 has been resampled and may differ in fine detail from the original 5000×5000 tiles. We trained on Kaggle T4 GPUs, which limited our hyperparameter search to a single configuration. Our ensemble weights are derived from validation IoU on a combined validation set rather than Massachusetts-only validation; results may differ if weights are tuned on the deployment domain.

VI. CONCLUSION AND FUTURE WORK

This paper compared three encoders for building segmentation under a fixed DeepLabV3+ framework. Three findings stand out from the experiments. First, reported FLOPs are a poor proxy for latency in this family of models on a Tesla T4 GPU. EfficientNet-B3 is more than twice as slow as ResNet50 despite having lower FLOPs. Second, a validation-IoU-weighted ensemble produces a small IoU gain (about 0.007) over the strongest single model but a meaningfully more balanced precision and recall profile. Third, limited in-domain fine-tuning helps only the smallest model, which suggests that overfitting capacity rather than architecture choice dominates in low-data regimes. The full pipeline, including training logs and per-image metrics, is reproducible from a single Kaggle notebook.

This study began as the first author's undergraduate major project. The most useful lesson from it was the value of inspecting evaluation pipelines before doubting models. The center-crop pitfall described in Section V cost roughly a day of debugging and would not have been caught without overlaying probability maps on the input imagery. We hope its description here saves a similar day for someone else.

Future work could go in several directions. The encoder set could be extended to recent transformer-based backbones such as SegFormer or Swin-UNet under the same controlled-comparison methodology. The pipeline could be evaluated on additional cross-domain settings, including South Asian urban morphologies that are closer to the deployment region of interest for our research group. The architecture could also be adapted to the change-detection setting suggested by Liu

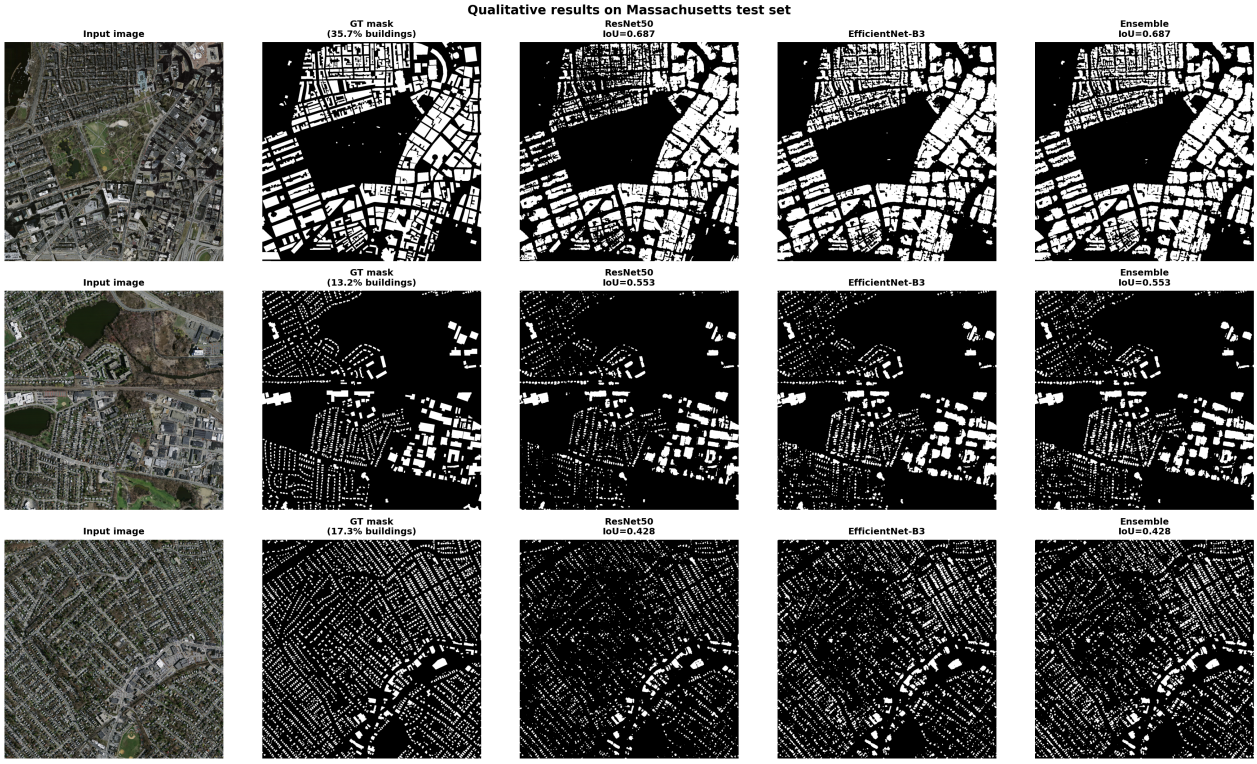

Fig. 5. Best-case (top), median-case (middle), and worst-case (bottom) test predictions. Columns from left to right: input image, ground-truth mask, ResNet50 prediction, EfficientNet-B3 prediction, ensemble prediction. The IoU value of each prediction is given above its panel.

and Cai [14], in which the same model is applied to multi-temporal imagery to track building footprint evolution over time.

Acknowledgements

The authors thank Dr. Manish Rai of the Department of Computer Science and Engineering at Manipal University Jaipur for his supervision and guidance throughout this project. The training and profiling experiments described in this paper were conducted on the free Kaggle Notebooks platform, whose Tesla T4 GPU allocation made an undergraduate-budget reproduction of these results possible. The Massachusetts Buildings dataset and the Inria Aerial Image Labeling benchmark were obtained through their respective public distributions. The authors acknowledge the use of Anthropic's Claude as a writing assistant during manuscript preparation; all technical work, training runs, results, and conclusions are the authors' own.

References

[1] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th International Conference on Machine Learning (ICML)*, PMLR, vol. 97, 2019, pp. 6105–6114.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[7] V. Mnih, *Machine Learning for Aerial Image Labeling*. Ph.D. dissertation, University of Toronto, 2013.

[8] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The Inria aerial image labeling benchmark," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 3226–3229.

[9] G. Prathap and I. Afanasyev, "Deep learning approach for building detection in satellite multispectral imagery," in *2018 International Conference on Intelligent Systems (IS)*, IEEE, 2018, pp. 461–465.

[10] C. Ayala, R. Sesma, C. Aranda, and M. Galar, "A deep learning approach to an enhanced building footprint and road detection in high-resolution satellite imagery," *Remote Sensing*, vol. 13, no. 16, p. 3135, 2021.

[11] Z. Pan, J. Xu, Y. Guo, Y. Hu, and G. Wang, "Deep learning segmentation and classification for urban village using a Worldview satellite image based on U-Net," *Remote Sensing*, vol. 12, no. 10, p. 1574, 2020.

[12] A. Benchabana, M.-K. Kholladi, R. Bensaci, and B. Khaldi, "Building detection in high-resolution remote sensing images by enhancing superpixel segmentation and classification using deep learning approaches," *Buildings*, vol. 13, no. 7, p. 1649, 2023.

[13] E. Irwansyah, Y. Heryadi, and A. A. S. Gunawan, "Semantic image segmentation for building detection in urban area with aerial photograph image using U-Net models," in *2020 IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology (AGERS)*, 2020, pp. 48–51.

[14] H. Liu and G. Cai, "CNN based detection of building roofs from high resolution satellite images," *The International Archives of the*

Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-3/W10, pp. 187–192, 2020.

[15] J. Z. Xu, W. Lu, Z. Li, P. Khaitan, and V. Zaytseva, "Building damage detection in satellite imagery using convolutional neural networks," *NeurIPS Workshop on AI for Humanitarian Assistance and Disaster Response*, 2019, arXiv:1910.06444.

[16] P. Iakubovskii, "Segmentation models PyTorch," GitHub repository, 2019. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[18] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, 2017.

[19] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[20] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.