Technical report · IF-TR-2026-07 · version 1.0

A from-scratch large-kernel ConvNet against ResNet-50 on Indian road scenes: BluConv-T on IDD, with TensorRT latency and INT8

Sherin Joseph Roy 18 November 2026 BluViT commit df0d01a (no release tag) P1 · Perception

Funded entirely by BluRabbit Mobility Private Limited.

Abstract

BluConv-T is a convolutional image encoder with no attention, added to the BluViT repository as a non-transformer candidate for a camera-only driving stack: four stages of BatchNorm blocks with 13x13 depthwise kernels, squeeze-excite, GRN and a camera gate. It was compared with ResNet-50, both with the same FPN segmentation head, both trained from scratch on the 14,027 training images of the public India Driving Dataset (IDD) and scored on its official val split, under two optimiser recipes fixed before any run. With AdamW, BluConv-T reached 65.89 mean IoU against ResNet-50's 63.57; with SGD it reached 55.68 against 57.97. The optimiser moved both networks by more than the networks differ, and the order flipped with it. In TensorRT on a workstation GPU, BluConv-T took 2.8 times ResNet-50's half-precision latency, and under INT8 quantisation with the setting fixed before the test it fell to 40.84 while ResNet-50 held 62.31. A rule written before the runs said that if BluConv-T did not clearly beat ResNet-50, the recommendation would be the ResNet-50 route; it did not, and that is the recommendation. The results also correct the earlier BluViT report: a better-trained ResNet-50 without pretraining comes within one point of the pretrained ViT and is ahead of it on person, rider and motorcycle. One seed per arm; nothing is measured on a vehicle or an embedded device.

1 Problem

Technical report IF-TR-2026-06 tested BluViT, a Vision Transformer trained from zero on Indian road imagery, and found that masked-autoencoder pretraining is what makes it work on the India Driving Dataset (IDD) [idd]. That study's only convolutional reference was a DeepLabV3 ResNet-50 trained with SGD, and the report said its number ranked a recipe, not an architecture.

The study then added a convolutional encoder with no attention, BluConv-T, to the same repository, and compared it with ResNet-50 [resnet] under one protocol. A rule was written before the runs: if BluConv-T does not clearly beat ResNet-50, the recommendation is the ResNet-50 route. The rule set no numeric threshold. The plan was to train both from scratch, measure their latency, and only then decide whether BluConv-T deserved masked pretraining of its own.

This report describes the two networks, the comparison, its speed and precision measurements, and the decision.

2 System design

BluConv-T

BluConv-T is a four-stage encoder built from BatchNorm convolution blocks, with the stage widths 80, 160, 320 and 640 and depths 3, 3, 12 and 3. A stem of two strided 3x3 convolutions brings the input to stride 4, and a strided 3x3 convolution sits between stages. Each block is:

- **a depthwise convolution** of 13x13 in stages 2 to 4 (stage 3 alternates 13x13 and 3x3 blocks) and 3x3 in stage 1, followed by BatchNorm;
- **squeeze-excite** [senet];
- **a pointwise expansion** by four, ReLU, **GRN** [convnextv2] and a pointwise projection with BatchNorm;
- a residual connection with stochastic depth.

The large kernels are trained with parallel dilated small-kernel branches, the reparameterisation of UniRepLKNet [unireplknet], following the large-kernel design of RepLKNet [replknet]. After training the branches are merged into one plain 13x13 depthwise convolution, so the exported graph contains only convolution, ReLU, multiplication, addition and global average pooling. A learned embedding per camera is mapped to a per-channel scale and shift at the entry of each stage, initialised to identity and unused in this study, as in BluViT.

ResNet-50 and the shared head

ResNet-50 is the standard network [resnet], here without pretrained weights. Both encoders feed the same segmentation head: a 160-channel feature pyramid [fpn] with a 128-channel merge in the style of Semantic FPN [panoptic-fpn], prediction at stride 4 and an auxiliary classifier on the stride-16 feature. As benchmarked, the BluConv-T network has `27.4 M` parameters and the ResNet-50 network `26.97 M`.

3 Method

Data and scoring

The data and scoring are those of IF-TR-2026-06. Both networks train for 40 epochs from random initialisation on the merged Part I and II segmentation train split of IDD (14,027 images), with batch 8, 512x896 crops, bf16 and EMA 0.999, and are scored on the official val split (2,036 images) at 1280x720 with the IDD protocol over the 26 level-3 classes. The reported value is the final epoch on EMA weights; no checkpoint is chosen on the val split. Each arm has one seed.

Arms

Two optimiser recipes were fixed before any run, and both are reported:

- **SGD**: lr 0.01, momentum 0.9, weight decay 5e-4, poly schedule to 1e-4, the recipe of the earlier CNN reference, so the comparison with it is like for like;
- **AdamW** [adamw]: lr 1e-3, weight decay 0.05, one warm-up epoch, poly schedule to 1e-5, the way this family of networks is usually published.

BluConv-T was run with both. ResNet-50 was first run with SGD only; when BluConv-T led only with AdamW, the same AdamW recipe was applied to ResNet-50 unchanged, before the decision was made, so that the comparison would not

set one network's better optimiser against the other's single run. Nothing was tuned after a result. Memory layout was chosen per network for speed only.

Speed and precision

Latency was measured with TensorRT 11.4 [tensorrt] on the same workstation GPU, at batch 1 with one fixed 3x720x1280 input, as the median of 300 runs after 50 warm-up runs, with input and output on the GPU. Half precision comes from a half-precision ONNX export. INT8 comes from post-training quantisation [int8] of the convolutions only, written as explicit quantise and dequantise nodes by ONNX Runtime's static quantiser [ort-quant], with per-channel symmetric weights, symmetric activations, 64 calibration images from the training split and every other layer in float32. Because the engines take one input size, their mean IoU is on the 1,980 val images that are 720x1280 after the loader's resize; PyTorch on all 2,036 reproduces the training logs.

The INT8 setting fixed before the test takes each activation range from the largest and smallest values seen over the calibration images. A second setting, which averages the per-image extremes, was added after the first failed. A sensitivity run then quantised one layer group of BluConv-T at a time, with every other layer in float32.

4 Validation

Accuracy

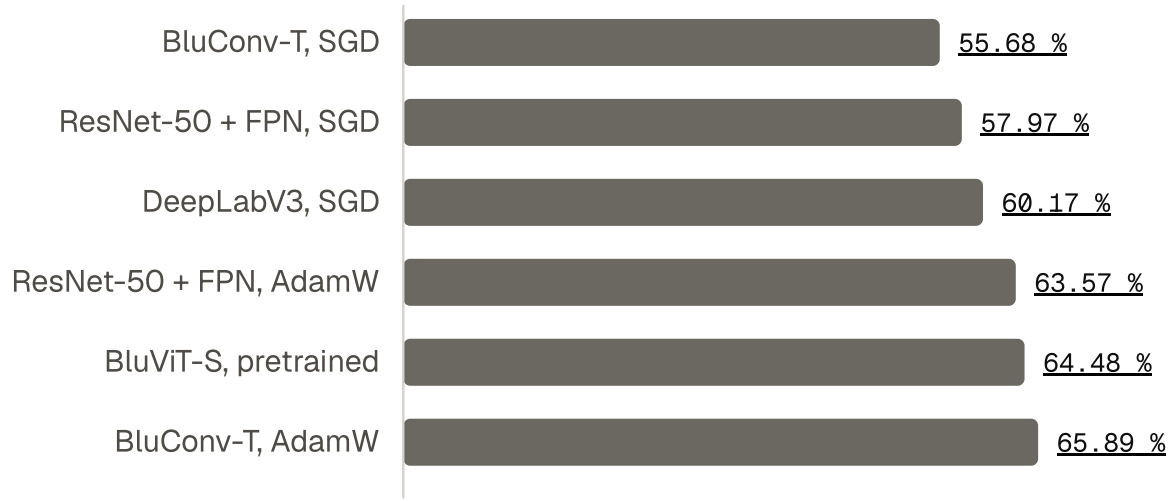


Figure 1. Mean IoU on the IDD val split (level 3, 26 classes), final epoch of 40, one seed per arm; every arm from scratch except BluViT-S, which had 400 epochs of masked pretraining on unlabelled IDD frames; DeepLabV3 (ResNet-50) and BluViT-S are from IF-TR-2026-06; the author's runs.

With AdamW, BluConv-T reached `65.89 %` and ResNet-50 `63.57 %`, a lead of `2.32 points` for BluConv-T. With SGD, ResNet-50 reached `57.97 %` and BluConv-T `55.68 %`, a difference of `-2.29 points`. The optimiser moved each network by more than the two networks differ, and the order flipped with it, so neither network is better on this evidence independently of its recipe.

Quantity	Value	Conditions	Basis
Mean IoU on IDD val, BluConv-T + FPN head, AdamW, from scratch	65.89 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
Mean IoU on IDD val, ResNet-50 + FPN head, AdamW, from scratch	63.57 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
Mean IoU on IDD val, ResNet-50 + FPN head, SGD, from scratch	57.97 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
Mean IoU on IDD val, BluConv-T + FPN head, SGD, from scratch	55.68 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
Mean IoU difference, BluConv-T minus ResNet-50 (both + FPN head), AdamW	2.32 points	both arms: AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
Mean IoU difference, BluConv-T minus ResNet-50 (both + FPN head), SGD	−2.29 points	both arms: SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16

Table 1. Mean IoU per arm and the two differences, IDD val, final epoch, one seed per arm; run by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

The four separately reported classes

Every run reports person, rider, motorcycle and autorickshaw separately. With AdamW, BluConv-T is ahead of ResNet-50 on all four; with SGD, behind on all four.

Quantity	Value	Conditions	Basis
IoU, person, IDD val, BluConv-T + FPN head, AdamW, from scratch	69.15 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, rider, IDD val, BluConv-T + FPN head, AdamW, from scratch	73.9 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, motorcycle, IDD val, BluConv-T + FPN head, AdamW, from scratch	77.56 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, autorickshaw, IDD val, BluConv-T + FPN head, AdamW, from scratch	86.25 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, person, IDD val, ResNet-50 + FPN head, AdamW, from scratch	66.99 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, rider, IDD val, ResNet-50 + FPN head, AdamW, from scratch	72.05 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, motorcycle, IDD val, ResNet-50 + FPN head, AdamW, from scratch	75.59 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, autorickshaw, IDD val, ResNet-50 + FPN head, AdamW, from scratch	83.48 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, person, IDD val, ResNet-50 + FPN head, SGD, from scratch	60.52 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, rider, IDD val, ResNet-50 + FPN head, SGD, from scratch	64.91 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, motorcycle, IDD val, ResNet-50 + FPN head, SGD, from scratch	69.34 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, autorickshaw, IDD val, ResNet-50 + FPN head, SGD, from scratch	79.37 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16

Quantity	Value	Conditions	Basis
IoU, person, IDD val, BluConv-T + FPN head, SGD, from scratch	54.73 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, rider, IDD val, BluConv-T + FPN head, SGD, from scratch	60.23 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, motorcycle, IDD val, BluConv-T + FPN head, SGD, from scratch	66.65 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16
IoU, autorickshaw, IDD val, BluConv-T + FPN head, SGD, from scratch	78 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16

Table 2. Per-class IoU on the IDD val split, final epoch, one seed per arm; run by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

Against the pretrained Vision Transformer

IF-TR-2026-06 reported BluViT-S, pretrained as a masked autoencoder for 400 epochs, at 64.48 %, ahead of its from-scratch DeepLabV3 ResNet-50 at 60.17 % and ahead of it on all four classes. The from-scratch networks here change that picture. BluConv-T with AdamW is above the pretrained ViT with no pretraining at all, and ResNet-50 with AdamW comes within one point of it. On the four classes, ResNet-50 with AdamW is ahead of the pretrained ViT on person, rider and motorcycle, and behind it on autorickshaw.

Quantity	Value	Conditions	Basis
IoU, person, BluViT-S, MAE init (epoch 400)	64.15 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15
IoU, rider, BluViT-S, MAE init (epoch 400)	68.92 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15
IoU, motorcycle, BluViT-S, MAE init (epoch 400)	73.4 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15
IoU, autorickshaw, BluViT-S, MAE init (epoch 400)	84.92 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15
IoU, person, IDD val, ResNet-50 + FPN head, AdamW, from scratch	66.99 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, rider, IDD val, ResNet-50 + FPN head, AdamW, from scratch	72.05 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, motorcycle, IDD val, ResNet-50 + FPN head, AdamW, from scratch	75.59 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17
IoU, autorickshaw, IDD val, ResNet-50 + FPN head, AdamW, from scratch	83.48 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17

Table 3. Per-class IoU on the IDD val split, final epoch: BluViT-S from masked pretraining (IF-TR-2026-06) and ResNet-50 + FPN head from scratch with AdamW; run by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

The earlier report’s own caveat applies in both directions: each number ranks a recipe on 14,027 images in 40 epochs. The ViT’s advantage came from pretraining that the convolutional networks have not had yet, and the comparison says

nothing about how they rank once they have it.

Speed and precision

Quantity	Value	Conditions	Basis
Median latency, BluConv-T + FPN head, TensorRT half precision	5.818 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device	Reproduced 2026-11-17
Median latency, BluConv-T + FPN head, TensorRT INT8, ranges from extremes	5.543 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17
Median latency, BluConv-T + FPN head, TensorRT INT8, ranges averaged	5.596 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17
Median latency, ResNet-50 + FPN head, TensorRT half precision	2.078 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device	Reproduced 2026-11-17
Median latency, ResNet-50 + FPN head, TensorRT INT8, ranges from extremes	1.563 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17
Median latency, ResNet-50 + FPN head, TensorRT INT8, ranges averaged	1.594 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17

Table 4. Median latency at batch 1, 3x720x1280, on one NVIDIA GeForce RTX 5080 with TensorRT 11.4; AdamW weights of each network; measured by the author. A ranking on this GPU, not the latency of an in-car device. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

In half precision BluConv-T took `5.818 ms` and ResNet-50 `2.078 ms`, 2.8 times as long. The engines were not profiled, so this report does not say which layers account for the difference.

Quantity	Value	Conditions	Basis
Mean IoU, BluConv-T + FPN head, TensorRT half precision	65.29 %	1,980 IDD val images of 720x1280; AdamW weights	Reproduced 2026-11-17
Mean IoU, BluConv-T + FPN head, TensorRT INT8, ranges from extremes	40.84 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17
Mean IoU, BluConv-T + FPN head, TensorRT INT8, ranges averaged	64.02 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17
Mean IoU, ResNet-50 + FPN head, TensorRT half precision	62.91 %	1,980 IDD val images of 720x1280; AdamW weights	Reproduced 2026-11-17
Mean IoU, ResNet-50 + FPN head, TensorRT INT8, ranges from extremes	62.31 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17
Mean IoU, ResNet-50 + FPN head, TensorRT INT8, ranges averaged	62.74 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17

Table 5. Mean IoU of the TensorRT engines on the 1,980 IDD val images of 720x1280; measured by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

With the INT8 setting fixed before the test, BluConv-T fell to 40.84 % and ResNet-50 held 62.31 %. With averaged ranges, BluConv-T kept 64.02 % at 5.596 ms, against ResNet-50’s 62.74 % at 1.594 ms; ResNet-50 is 3.5 times as fast. INT8 buys BluConv-T almost nothing over half precision on this GPU.

1

1. The averaged-range setting was added after BluConv-T collapsed under the first, and the choice between the two was read on the val split. It is a selection between two calibration settings applied to both networks, small against a loss of this size, but it is a selection made with the result in hand.

Where BluConv-T loses accuracy under INT8

Quantity	Value	Conditions	Basis
mIoU lost when only the stem and downsampling convolutions are INT8, BluConv-T + FPN head	5.25 points	5 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17
mIoU lost when only the depthwise 13x13 and 3x3 convolutions are INT8, BluConv-T + FPN head	5.81 points	21 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17
mIoU lost when only the pointwise convolutions are INT8, BluConv-T + FPN head	3.44 points	42 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17
mIoU lost when only the squeeze-excite convolutions are INT8, BluConv-T + FPN head	0 points	42 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17
mIoU lost when only the FPN head convolutions are INT8, BluConv-T + FPN head	0.05 points	13 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17

Table 6. Mean IoU lost when one layer group of BluConv-T is quantised to INT8 with ranges from extremes and every other layer stays float32; measured by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

The loss is not in one place. The stem and downsampling convolutions, the depthwise kernels and the pointwise layers each cost several points alone, and the squeeze-excite layers and the head cost almost nothing. Averaging the ranges removes most of the loss, so the study reads the cause as rare large activations stretching the ranges, in layers with no ReLU after them, rather than the reparameterised 13x13 kernels in particular.

Decision

The study’s verdict is that **BluConv-T does not clearly beat ResNet-50, and the recommendation is the ResNet-50 route**. The bar used was to be ahead under both optimiser recipes and no slower in half precision; because the rule had no threshold, that bar was set with the numbers in hand. BluConv-T is the most accurate network trained from scratch in the study so far, but it leads under one of two optimisers, on one seed, at about three times the inference cost and less than half the training speed.

Cost

Quantity	Value	Conditions	Basis
Fine-tune wall clock, 40 epochs, BluConv-T + FPN head, AdamW	6.5 h	AdamW lr 1e-3, poly; 14,027 images, 512x896 crops, batch 8, bf16	Reproduced 2026-11-16
Fine-tune wall clock, 40 epochs, ResNet-50 + FPN head, AdamW	3.1 h	AdamW lr 1e-3, poly; 14,027 images, 512x896 crops, batch 8, bf16	Reproduced 2026-11-17
Training throughput, BluConv-T + FPN head, AdamW	26.2 img/s	median over epochs; 512x896 crops, batch 8, bf16	Reproduced 2026-11-16
Training throughput, ResNet-50 + FPN head, AdamW	59.9 img/s	median over epochs; 512x896 crops, batch 8, bf16	Reproduced 2026-11-17

Table 7. Fine-tune wall clock and training throughput with AdamW, 512x896 crops, batch 8, bf16, on one workstation GPU; measured by the author. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

5 Limitations

- **One seed per arm.** The differences between the networks are about two points, of the size a second seed could move, and the optimiser moves each network by more.
- **The bar was set after the numbers.** The rule fixed before the runs had no threshold; the bar used to read it was chosen with the results known.
- **One arm was added after a result.** ResNet-50 with AdamW was run after BluConv-T’s AdamW lead was known, with the recipe unchanged. Without it the comparison would have been unfair to ResNet-50.
- **No pretraining.** Every convolutional arm starts from random weights, while the ViT it is compared with had 400 epochs of masked pretraining. Masked pretraining of ResNet-50 in the style of SparK [spark] started on 2026-11-17 and is not in this report. BluConv-T’s has not been started.
- **One GPU, not the device.** Latency is a ranking on a workstation GPU. Nothing is measured on the Jetson AGX Orin or any in-car device, and the networks were not compared at equal latency, for instance against a larger ResNet.
- **Engine accuracy on a subset.** The engines were scored on the 1,980 val images of one size, so their mean IoU sits slightly below the training logs on all 2,036.
- **The repository is not public.** The commands and commit below can be checked by the author but not by readers.

6 Reproducibility

The BluConv-T encoder is part of the BluViT repository at commit df0d01a on its main branch; the repository is local, with no remote and no release tag. Every training run records the SHA-256 of its training script, the CNN baseline script and the encoder file, and refuses to resume across a change. The commands come from the study’s scripts, which sit beside the repository:

- Training: `python scripts/train_conv_seg.py --arch bluconv_t --opt adamw`, and the same with `--arch r50_fpn` and `--opt sgd`.
- Report tables: `python scripts/conv_report.py`.

- Latency and engine accuracy: `python scripts/bench_latency.py`, in a separate environment with TensorRT 11.4.0.106.
- INT8 sensitivity: `python scripts/int8_sensitivity.py`.

Hardware and software: one NVIDIA GeForce RTX 5080 (15.5 GB usable), an AMD Ryzen 9 9900X with about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 for training. The IDD releases are available from IIIT Hyderabad under their own terms; no IDD image is reproduced in this report.

References

1. G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. WACV 2019. doi:10.1109/WACV.2019.00190
2. K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. CVPR 2016. doi:10.1109/CVPR.2016.90
3. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie. Feature pyramid networks for object detection. CVPR 2017. arxiv.org/abs/1612.03144
4. A. Kirillov, R. Girshick, K. He, P. Dollár. Panoptic feature pyramid networks. CVPR 2019. arxiv.org/abs/1901.02446
5. X. Ding, X. Zhang, J. Han, G. Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs. CVPR 2022. arxiv.org/abs/2203.06717
6. X. Ding et al. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. CVPR 2024. arxiv.org/abs/2311.15599
7. J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks. CVPR 2018. arxiv.org/abs/1709.01507
8. S. Woo et al. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. CVPR 2023. arxiv.org/abs/2301.00808
9. I. Loshchilov, F. Hutter. Decoupled weight decay regularization. ICLR 2019. arxiv.org/abs/1711.05101
10. B. Jacob et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. CVPR 2018. arxiv.org/abs/1712.05877
11. NVIDIA. TensorRT documentation. Software documentation. docs.nvidia.com/deeplearning/tensorrt/
12. ONNX Runtime. Quantize ONNX models. Software documentation. onnxruntime.ai/docs/performance/model-optimizations/quantization.html
13. K. Tian et al. Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. ICLR 2023. arxiv.org/abs/2301.03580

Evidence

Quantity	Value	Conditions	Basis	Source and reproduction
Mean IoU on IDD val, BluConv-T + FPN head, AdamW, from scratch	65.89 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), accuracy table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)`
Mean IoU on IDD val, ResNet-50 + FPN head, AdamW, from scratch	63.57 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17	CONV_ENCODER.md (study report), accuracy table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`
Mean IoU on IDD val, ResNet-50 + FPN head, SGD, from scratch	57.97 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), accuracy table; runs/seg_r50_fpn_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt sgd (study scripts, alongside bluvit)`
Mean IoU on IDD val, BluConv-T + FPN head, SGD, from scratch	55.68 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of	Reproduced 2026-11-16	CONV_ENCODER.md (study report), accuracy table; runs/seg_bluconv_t_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16

Quantity	Value	Conditions	Basis	Source and reproduction
		40, EMA; one seed		Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt sgd (study scripts, alongside bluvit)
Mean IoU difference, BluConv-T minus ResNet-50 (both + FPN head), AdamW	2.32 points	both arms: AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17	CONV_ENCODER.md (study report), Go/no-go commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/conv_report.py (study scripts) after the four training runs
Mean IoU difference, BluConv-T minus ResNet-50 (both + FPN head), SGD	-2.29 points	both arms: SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), Go/no-go commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/conv_report.py (study scripts) after the four training runs
IoU, person, IDD val, BluConv-T + FPN head, AdamW, from scratch	69.15 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)
IoU, rider, IDD val, BluConv-T + FPN head, AdamW, from scratch	73.9 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16

Quantity	Value	Conditions	Basis	Source and reproduction
		40, EMA; one seed		Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)
IoU, motorcycle, IDD val, BluConv-T + FPN head, AdamW, from scratch	77.56 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)
IoU, autorickshaw, IDD val, BluConv-T + FPN head, AdamW, from scratch	86.25 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)
IoU, person, IDD val, ResNet-50 + FPN head, AdamW, from scratch	66.99 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)
IoU, rider, IDD val, ResNet-50 + FPN head, AdamW, from scratch	72.05 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol);	Reproduced 2026-11-17	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X,

Quantity	Value	Conditions	Basis	Source and reproduction
		final epoch of 40, EMA; one seed		about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`
IoU, motorcycle, IDD val, ResNet-50 + FPN head, AdamW, from scratch	75.59 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`
IoU, autorickshaw, IDD val, ResNet-50 + FPN head, AdamW, from scratch	83.48 %	AdamW lr 1e-3, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-17	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`
IoU, person, IDD val, ResNet-50 + FPN head, SGD, from scratch	60.52 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt sgd (study scripts, alongside bluvit)`
IoU, rider, IDD val, ResNet-50 + FPN head, SGD, from scratch	64.91 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce

Quantity	Value	Conditions	Basis	Source and reproduction
		classes, IDD protocol); final epoch of 40, EMA; one seed		RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt sgd (study scripts, alongside bluvit)`
IoU, motorcycle, IDD val, ResNet-50 + FPN head, SGD, from scratch	69.34 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt sgd (study scripts, alongside bluvit)`
IoU, autorickshaw, IDD val, ResNet-50 + FPN head, SGD, from scratch	79.37 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_r50_fpn_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt sgd (study scripts, alongside bluvit)`
IoU, person, IDD val, BluConv-T + FPN head, SGD, from scratch	54.73 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt sgd (study scripts, alongside bluvit)`
IoU, rider, IDD val, BluConv-T + FPN head, SGD, from scratch	60.23 %	SGD lr 0.01, poly; from scratch; IDD val (2,036	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_sgd/metrics.jsonl

Quantity	Value	Conditions	Basis	Source and reproduction
		images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed		commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt sgd (study scripts, alongside bluvit)`
IoU, motorcycle, IDD val, BluConv-T + FPN head, SGD, from scratch	66.65 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt sgd (study scripts, alongside bluvit)`
IoU, autorickshaw, IDD val, BluConv-T + FPN head, SGD, from scratch	78 %	SGD lr 0.01, poly; from scratch; IDD val (2,036 images, 26 classes, IDD protocol); final epoch of 40, EMA; one seed	Reproduced 2026-11-16	CONV_ENCODER.md (study report), per-class table; runs/seg_bluconv_t_sgd/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt sgd (study scripts, alongside bluvit)`
Mean IoU on IDD val, BluViT-S, MAE init (epoch 400)	64.48 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at	Reproduced 2026-11-15	BASELINE.md (study report), arm table; runs/seg_mae_init_ep400/metrics.jsonl commit 59b8054 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python -m bluvit.train.finetune_seg --config configs/seg_vit_s_idd.yaml train.epochs=40 seg.class_weights=configs/idd_class_weights.json data.workers=8 train.init_from=runs/mae_vit_s/mae_last.pt train.seed=1337`

Quantity	Value	Conditions	Basis	Source and reproduction
		1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed		
Mean IoU on IDD val, DeepLabV3 ResNet-50, from scratch (reference)	60.17 %	reference CNN: torchvision DeepLabV3 ResNet-50, no pretrained weights, SGD with a poly schedule; IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-10	BASELINE.md (study report), arm table; runs/seg_cnn_deeplabv3_r50/metrics.jsonl commit 4261989 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_cnn_baseline.py (study scripts, alongside bluvit)`
IoU, person, BluViT-S, MAE init (epoch 400)	64.15 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036	Reproduced 2026-11-15	BASELINE.md (study report), per-class table; runs/seg_mae_init_ep400/metrics.jsonl commit 59b8054 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python -m bluvit.train.finetune_seg --config configs/seg_vit_s_idd.yaml`

Quantity	Value	Conditions	Basis	Source and reproduction
		images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed		`train.epochs=40 seg.class_weights=configs/idd_class_weights.json data.workers=8 train.init_from=runs/mae_vit_s/mae_last.pt train.seed=1337`
IoU, rider, BluViT-S, MAE init (epoch 400)	68.92 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15	BASELINE.md (study report), per-class table; runs/seg_mae_init_ep400/metrics.jsonl commit 59b8054 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python -m bluvit.train.finetune_seg --config configs/seg_vit_s_idd.yaml train.epochs=40 seg.class_weights=configs/idd_class_weights.json data.workers=8 train.init_from=runs/mae_vit_s/mae_last.pt train.seed=1337`
IoU, motorcycle, BluViT-S, MAE init (epoch 400)	73.4 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes),	Reproduced 2026-11-15	BASELINE.md (study report), per-class table; runs/seg_mae_init_ep400/metrics.jsonl commit 59b8054 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16

Quantity	Value	Conditions	Basis	Source and reproduction
		official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed		`Local repository, not public. python -m bluvit.train.finetune_seg --config configs/seg_vit_s_idd.yaml train.epochs=40 seg.class_weights=configs/idd_class_weights.json data.workers=8 train.init_from=runs/mae_vit_s/mae_last.pt train.seed=1337`
IoU, autorickshaw, BluViT-S, MAE init (epoch 400)	84.92 %	encoder from the MAE checkpoint at epoch 400 (unlabelled IDD frames only); IDD level 3 (26 classes), official val split (2,036 images), final epoch of 40, EMA weights, scored at 1280x720 (IDD protocol); trained on 14,027 IDD train images, 512x896 crops, batch 8, bf16; one seed	Reproduced 2026-11-15	BASELINE.md (study report), per-class table; runs/seg_mae_init_ep400/metrics.jsonl commit 59b8054 (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python -m bluvit.train.finetune_seg --config configs/seg_vit_s_idd.yaml train.epochs=40 seg.class_weights=configs/idd_class_weights.json data.workers=8 train.init_from=runs/mae_vit_s/mae_last.pt train.seed=1337`
Parameters, BluConv-T + FPN head (encoder and head)	27.4 M	the network as benchmarked, with the FPN segmentation head	Reproduced 2026-11-17	latency.md (study report); latency.json commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/bench_latency.py (study`

Quantity	Value	Conditions	Basis	Source and reproduction
				scripts, in a separate TensorRT environment)
Parameters, ResNet-50 + FPN head (encoder and head)	26.97 M	the network as benchmarked, with the FPN segmentation head	Reproduced 2026-11-17	latency.md (study report); latency.json commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Median latency, BluConv-T + FPN head, TensorRT half precision	5.818 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Median latency, BluConv-T + FPN head, TensorRT INT8, ranges from extremes	5.543 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Median latency, BluConv-T + FPN head, TensorRT INT8, ranges averaged	5.596 ms	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges averaged	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)

Quantity	Value	Conditions	Basis	Source and reproduction
		(added afterwards)		
Median latency, ResNet-50 + FPN head, TensorRT half precision	`2.078 ms`	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)`
Median latency, ResNet-50 + FPN head, TensorRT INT8, ranges from extremes	`1.563 ms`	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)`
Median latency, ResNet-50 + FPN head, TensorRT INT8, ranges averaged	`1.594 ms`	batch 1, 3x720x1280, median of 300 runs; AdamW weights; workstation GPU, not a device; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)`
Mean IoU, BluConv-T + FPN head, TensorRT half precision	`65.29 %`	1,980 IDD val images of 720x1280; AdamW weights	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/bench_latency.py (study`

Quantity	Value	Conditions	Basis	Source and reproduction
				scripts, in a separate TensorRT environment)
Mean IoU, BluConv-T + FPN head, TensorRT INT8, ranges from extremes	40.84 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Mean IoU, BluConv-T + FPN head, TensorRT INT8, ranges averaged	64.02 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges averaged (added afterwards)	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)
Mean IoU, ResNet-50 + FPN head, TensorRT half precision	62.91 %	1,980 IDD val images of 720x1280; AdamW weights	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Mean IoU, ResNet-50 + FPN head, TensorRT INT8, ranges from extremes	62.31 %	1,980 IDD val images of 720x1280; AdamW weights; INT8 convolutions, ranges from extremes (fixed in advance)	Reproduced 2026-11-17	latency.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/bench_latency.py (study scripts, in a separate TensorRT environment)
Mean IoU, ResNet-50 + FPN head, TensorRT INT8, ranges averaged	62.74 %	1,980 IDD val images of 720x1280; AdamW weights; INT8	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0

Quantity	Value	Conditions	Basis	Source and reproduction
		convolutions, ranges averaged (added afterwards)		Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)
mIoU lost when only the stem and downsampling convolutions are INT8, BluConv-T + FPN head	5.25 points	5 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)
mIoU lost when only the depthwise 13x13 and 3x3 convolutions are INT8, BluConv-T + FPN head	5.81 points	21 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)
mIoU lost when only the pointwise convolutions are INT8, BluConv-T + FPN head	3.44 points	42 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)
mIoU lost when only the squeeze-excite	0 points	42 convolutions	Reproduced 2026-11-17	int8_sensitivity.md (study report)

Quantity	Value	Conditions	Basis	Source and reproduction
convolutions are INT8, BluConv-T + FPN head		INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280		commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)`
mIoU lost when only the FPN head convolutions are INT8, BluConv-T + FPN head	`0.05 points`	13 convolutions INT8 (ranges from extremes), rest float32; against float32 TensorRT (65.3); 1,980 IDD val images of 720x1280	Reproduced 2026-11-17	int8_sensitivity.md (study report) commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080, TensorRT 11.4.0.106, PyTorch 2.14.1 with CUDA 13.0 `Local repository, not public. python scripts/int8_sensitivity.py (study scripts, in a separate TensorRT environment)`
Fine-tune wall clock, 40 epochs, BluConv-T + FPN head, AdamW	`6.5 h`	AdamW lr 1e-3, poly; 14,027 images, 512x896 crops, batch 8, bf16	Reproduced 2026-11-16	CONV_ENCODER.md (study report), accuracy table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)`

Quantity	Value	Conditions	Basis	Source and reproduction
Fine-tune wall clock, 40 epochs, ResNet-50 + FPN head, AdamW	3.1 h	AdamW lr 1e-3, poly; 14,027 images, 512x896 crops, batch 8, bf16	Reproduced 2026-11-17	CONV_ENCODER.md (study report), accuracy table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`
Training throughput, BluConv-T + FPN head, AdamW	26.2 img/s	median over epochs; 512x896 crops, batch 8, bf16	Reproduced 2026-11-16	CONV_ENCODER.md (study report), accuracy table; runs/seg_bluconv_t_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch bluconv_t --opt adamw (study scripts, alongside bluvit)`
Training throughput, ResNet-50 + FPN head, AdamW	59.9 img/s	median over epochs; 512x896 crops, batch 8, bf16	Reproduced 2026-11-17	CONV_ENCODER.md (study report), accuracy table; runs/seg_r50_fpn_adamw/metrics.jsonl commit df0d01a (bluvit main; local repository, no release tag) · NVIDIA GeForce RTX 5080 (15.5 GB), AMD Ryzen 9 9900X, about 30 GB RAM, Linux 7.0.0-42 (Ubuntu), PyTorch 2.14.1 with CUDA 13.0, bf16 `Local repository, not public. python scripts/train_conv_seg.py --arch r50_fpn --opt adamw (study scripts, alongside bluvit)`

Table 8. Every measured quantity in this report, with its source, commit, hardware and the command that reproduces it. “Reproduced” rows were run by the author; “Reported” rows are quoted from the repository at the commit shown.

Cite this report

APA

Roy, S. J. (2026). A from-scratch large-kernel ConvNet against ResNet-50 on Indian road scenes: BluConv-T on IDD, with TensorRT latency and INT8 (Technical Report IF-TR-2026-07, Version 1.0). IsoFrames. https://isoframes.io/reports/if-tr-2026-07/

BibTeX

```
@techreport{isoframes_if_tr_2026_07,
  author      = {Roy, Sherin Joseph},
  title       = {A from-scratch large-kernel ConvNet against ResNet-50 on Indian road scenes:
BluConv-T on IDD, with TensorRT latency and INT8},
  institution = {IsoFrames, BluRabbit Mobility},
  number      = {IF-TR-2026-07},
  type        = {Technical Report},
  year        = {2026},
  month       = nov,
  note        = {Version 1.0},
  url         = {https://isoframes.io/reports/if-tr-2026-07/}
}
```