An MLOps Blueprint for Vision Models in Resource-Constrained Industry

The published MLOps playbook is written for teams that can scale out of any problem: add GPUs when memory is tight, add engineers when the schedule slips, add budget when the cloud bill climbs. Industrial computer-vision work rarely has those exits. We built VeerNet, our encoder-decoder segmentation network for digitising scanned raster well logs into LAS-grade curves, for an onshore Texas operator under three simultaneous hard ceilings: a GPU memory limit that forced the binary segmentation stage to a physical batch of one image per pass, a headcount cap, and a fixed price. This whitepaper is the blueprint we ran against, written from the inside. We treat the memory ceiling as the constraint that sets every other decision, and show how a custom collate function that pads each mini-batch to its widest member and masks the padding out of the loss recovered an effective batch of sixteen on the harder three-class multiclass stage. We cost training in the only unit that was actually scarce, wall-clock time, anchored on the 110 minutes the binary stage took for fifty epochs over two thousand images and the 550 minutes the multiclass stage took for fifty epochs over fifteen thousand. And we tie a deliberately compact architecture, a five-block encoder, a five-stage decoder, and two transformer attention layers on the bottleneck, to a delivery envelope of sixteen weeks, six full-time engineers, and 180,000 EUR, against a standard track of thirty-two weeks, four engineers, and 100,000 EUR. The argument throughout is that constraints are not obstacles to the blueprint; they are the blueprint. Three interactive instruments let a planner walk the memory ceiling, the training-time budget, and the delivery envelope directly.

Quamer Nasim
ML Research Engineer

Narendra Patwardhan
Research Collaborator

VEERNET / MLOPS / COMPUTER-VISION / RASTER-LOG-DIGITISATION / GPU-MEMORY / BATCH-SIZE

CONTENTS

> "
>
> The MLOps literature is written for teams that can scale out of any problem. We could not. Memory, people, and money were fixed before the first line of code, and the blueprint is what you do when the exits are closed.
>
> "

I

THE FRAME

When every axis of scale is already maxed out

A blueprint written from inside three ceilings

Most MLOps writing carries a silent assumption: that scale is a lever you can always pull. If memory is tight, rent a bigger GPU. If the schedule slips, add an engineer. If the cloud bill climbs, raise the budget. The discipline that follows from those assumptions is real and useful [1], but it is a discipline for teams whose constraints are soft. Industrial computer-vision work, the kind that runs inside an operating company rather than a hyperscaler, almost never has those exits open at the same time.

We built VeerNet, our encoder-decoder segmentation network for digitising scanned raster well logs into LAS-grade curves, for an onshore Texas operator whose three constraints were all hard at once. GPU memory was capped, so the binary segmentation stage was forced to a physical batch of a single image per pass. Headcount was capped, so the system had to be one a small team could build and the operator could later hold. And the price was fixed before we started, so every architectural and training decision had a budget consequence we could not paper over. This whitepaper is the blueprint we ran against, and the thesis is blunt: under these conditions the constraints are not obstacles to the blueprint, they are the blueprint. You design the memory, the time, and the money first, and the model second.

We want to be precise about what "hard" means, because the word is overused. A soft constraint is one you can buy your way out of inside the project: the GPU is a credit-card field away from being upgraded, the deadline has a week of slack hidden in it, the budget has a contingency line. A hard constraint is one that is fixed in the contract before the first commit and cannot move without renegotiating the engagement itself. All three of ours were the second kind. The price was a single number in an accepted proposal [1], not a range. The team was named individuals, not a hiring plan. And the card was the card we had, because the unit economics of the deliverable would not survive a fatter rental line.

When all three are hard simultaneously, the usual MLOps reflex of trading one resource for another stops working, because there is nothing soft left to trade against. That is the regime this document is written for, and it is more common in operating companies than the literature admits.

The raw material was a public archive. The Texas Railroad Commission, the state oil and gas regulator, publishes a vast set of scanned legacy logs, and the slice we worked against held roughly 136,771 TIFF images against only 7,781 already-digital LAS files. That ratio, better than seventeen scanned raster images for every one machine-readable curve set, is the whole commercial case: an enormous volume of subsurface measurement is locked in raster scans that no modern workflow can read, and freeing it is exactly the kind of high-value, low-glamour task an operator wants automated but cannot justify a research lab to solve. Every one of those scans is a decision a geoscientist once made and a measurement a tool once took, now stranded as pixels because the digital file was lost, never produced, or filed only on paper. The job was to ship a system that turns that pile back into curves, not to publish, and shipping under hard ceilings is a different engineering problem than the papers describe. The papers optimise the model; we had to optimise the model, the clock, and the cap on the people, all at once, against numbers we did not get to choose.

Memory ceiling

- The widest log, near 12,800 pixels, sets the physical batch, not the average
- Binary stage forced to a physical batch of 1 image per forward pass
- Multiclass effective batch of 16 bought back via padded collate plus accumulation
- GroupNorm keeps normalisation honest at a single-image physical batch

Headcount ceiling

- Six full-time engineers on the accelerated track, four on the standard

- A compact 5-encoder, 5-decoder, 2-attention architecture one team can hold
- No on-call rotation to staff: the model is a batch digitiser, not a live service
- Handover to the operator's own team is the acceptance test, not an afterthought

Budget ceiling

- Accelerated envelope: 16 weeks, 6 engineers, 180,000 EUR
- Standard envelope: 32 weeks, 4 engineers, 100,000 EUR
- Training costed in wall clock: 110 minutes binary, 550 minutes multiclass
- GPU rental sized to the job: 750 EUR per month high-end, 1,800 EUR advanced

Why the deliverable shape lets the ceilings bind so hard

A digitiser is a batch system, not a live service, and that single fact changes which ceilings matter. There is no request latency to defend, no autoscaling group, no on-call rotation. The model runs over an archive, emits curves, and a human reviews the output. So the serving side is cheap and the cost concentrates entirely in training and in the people who build it. That is why memory, training time, and the delivery envelope are the three axes this blueprint optimises, and why latency and uptime, the usual MLOps headline metrics, barely appear. The shape of the deliverable decides which constraints are load-bearing, and for a batch digitiser the load-bearing constraints are the three we were handed as fixed.

It is worth dwelling on what falls away, because the canonical MLOps debt argument [1] enumerates a long list of hidden costs, and a batch digitiser simply does not pay most of them. There is no feedback loop where today's predictions become tomorrow's training data, so there is no entanglement to fear. There is no live feature pipeline to drift, because the input is a static scan and the features are pixels. There is no serving-skew between a training environment and a production one, because there is no production serving environment at all in the request-response sense: the same code that we ran in training runs over the archive in inference. The model can be re-run from scratch over any subset

of the archive at any time, and a wrong curve costs a human reviewer a correction, not a customer an outage. Removing the live-service surface removes whole categories of operational risk, and what is left is a much smaller, harder core: get the model right, get it trained inside the clock, and get it into hands that are not ours. Those are the three ceilings, and nothing in the deliverable lets us escape any of them.

II

THE FIRST CEILING

Memory decides the batch before the model does

The widest image, not the average, sets the physical batch

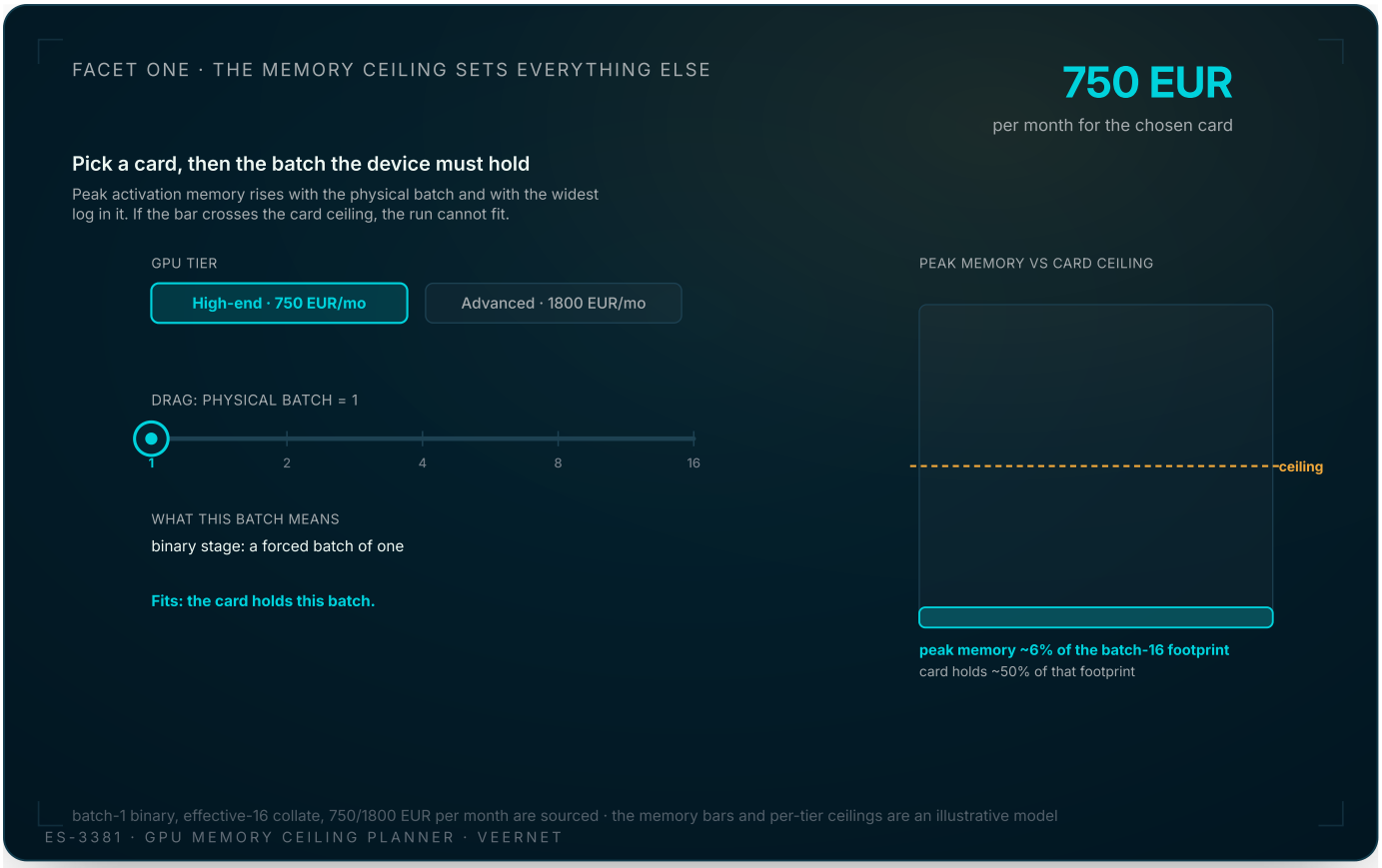
Scanned logs do not arrive at a tidy resolution. The slice we trained on spanned from roughly 3,200 to 12,800 pixels wide and 480 to 640 pixels tall, in no fixed ratio, because the physical logs were printed at different scales and scanned by different operators across decades. You cannot stack tensors of different shapes, and you cannot resize a one-pixel-wide curve trace without smearing the very signal the model is trained to find. So the dimensional spread is not a preprocessing nuisance; it is the thing that sets the memory ceiling.

Peak training memory is, to a first approximation, the activation footprint of the network evaluated at the largest image in the mini-batch, multiplied by the physical batch size, plus the fixed cost of parameters, gradients, optimiser state, and the backpropagation workspace. Every term except the first is constant. The first term is governed by your widest log, and a batch sampler can pull a 12,800-pixel-wide image at any moment. So you size the physical batch to survive that worst case, which means the physical batch is small. On the binary segmentation stage it was forced all the way down to one. A single image per forward pass, because a single one of the widest logs already commits a serious slice of the device.

It helps to see why the spread is so punishing in memory terms. A convolutional encoder holds, for the backward pass, the activations at every spatial resolution it produces, and for a dense-prediction encoder-decoder the spatial dimensions dominate the channel dimensions in the early and late stages. The widest log at 12,800 pixels is four times the width of the narrowest at 3,200, and the tallest at 640 pixels is a third again the shortest at 480. Because activation memory scales with the product of height and width at each resolution, the worst case is not a little worse than the average, it is several times worse, and it can arrive in any single sample. A sampler that happened to draw one 12,800-by-640

log into a batch alongside three small ones would size the whole batch's memory to that one giant, because the collate has to pad the small ones up to the giant's footprint. This is the asymmetry that pins the physical batch: you cannot average your way to safety, because one outlier in the draw sets the peak. VeerNet itself is lean by design, a single grayscale input channel and a 128-wide feature embedding at depth, but leanness in the channel dimension buys you little when a single image is twelve thousand pixels across.

That is the constraint the first instrument makes concrete. Pick the GPU tier the budget allows, the high-end card at 750 EUR per month or the advanced card at 1,800 EUR per month, then drag the physical batch the device must hold and watch the peak memory climb against the card's ceiling. The cheaper card runs the batch-of-one binary regime comfortably and cannot hold the larger batch; the more expensive card is what admits the effective batch of sixteen that the multiclass stage needs. The instrument is built to make a specific planning trap visible: the temptation to provision for the average image and discover, three epochs into a run, that an out-of-memory crash on the first wide log has thrown away an afternoon. Provisioning for the worst case is not conservatism here, it is the only setting that completes a run, and the meter shows exactly where the worst-case batch crosses each card's ceiling so the choice is made before the run, not during it.



A GPU-memory ceiling planner. The memory ceiling is the constraint everything else is built on: the binary stage was forced to a physical batch of one variable-size log per pass, and the multiclass stage reached an effective batch of sixteen only because a custom collate function pads each mini-batch to its widest member and masks the padding out of the loss. Pick the high-end card at 750 EUR per month or the advanced card at 1,800 EUR per month, then drag the physical batch the device must hold. The teal column shows the peak activation memory the batch requires; if it crosses the card ceiling the run cannot fit and the overage turns orange. The batch-of-one binary regime, the effective-sixteen multiclass collate regime, and the 750 and 1,800 EUR per-month costs are the engagement's own figures; the relative memory bars and the per-tier ceilings are an illustrative linear model and are flagged as such.

GroupNorm is not a preference here, it is a requirement

A physical batch of one quietly breaks a normalisation scheme that depends on batch statistics. Estimating a mean and variance from a single example is meaningless, and a training loop that does it produces noise where it should produce stable normalisation. Because the memory ceiling forces the physical batch small, we cannot treat the normalisation choice as a tuning detail. We use GroupNorm, whose statistics are computed across channel groups within a single example and therefore do not depend on the batch at all [2]. It behaves identically at a physical batch of one as at sixteen, which is exactly the invariance a memory-constrained loop demands. The ceiling chose the normalisation for us; we just had to recognise that it had.

The detail that matters in implementation is how the groups are sized, and here we learned to be defensive rather than fixed. A GroupNorm layer must divide its channels into groups, and a hard-coded group count breaks the moment a layer has fewer channels than the count, or a count that does not divide the channel dimension evenly. Our layers therefore size the group count adaptively, taking the smaller of half the channels or sixteen, so a layer with thirty-two channels gets sixteen groups and a shallow layer with eight channels gets four, never an illegal configuration that throws at construction time. It is a small piece of plumbing, but it is the kind of small piece that decides whether a five-stage encoder and five-stage decoder instantiate cleanly or fail on the third block of a fresh run. We mention it because the move from BatchNorm to GroupNorm is not free: the legacy first cut of the network carried BatchNorm with a 0.01 momentum and a LeakyReLU slope of 0.2, tuned for a regime that the memory ceiling then made unreachable. Recognising that the ceiling had invalidated the old normalisation, and rebuilding the normalisation and its group-sizing logic to be batch-independent, was one of the earliest forced decisions in the project, and everything in the training loop sits on top of it.

Buying back a real batch without buying more memory

The binary stage tolerated a batch of one because the task was simpler and the data path stayed trivial. The multiclass stage did not. Expanding to three classes, background plus two curves, on a synthetic dataset of fifteen thousand instances built up from an initial run of twenty thousand generated logs, the sparse-foreground imbalance grew worse and a noisy single-example gradient was no longer good enough. The arithmetic of the imbalance is brutal: a curve trace is roughly a pixel wide and runs the height of a log that is thousands of pixels in the other dimension, so the foreground classes occupy a tiny fraction of a percent of the pixels and the background swamps everything. A gradient computed from one image, on a task that sparse, is dominated by background and barely informed by the two curves we actually care about. We needed a real batch, and we needed it without a memory budget that would allow holding sixteen of the widest logs at once.

The answer was two pieces of engineering that work together. First, a custom collate function: it inspects each sampled mini-batch, finds the maximum height and width across its members, pads every image and mask up to that per-batch maximum, and carries a per-pixel validity mask that marks real pixels from padding. The padded regions are fiction and are multiplied out of the loss, so every padded pixel contributes exactly zero gradient. Pad

to make the stack legal; mask to keep the gradient honest. Second, gradient accumulation: forward a small physical batch the device can hold, call backward to pile gradient into the buffers without stepping, repeat until sixteen images' worth of gradient has accumulated, then take one optimiser step. The optimiser updates on the averaged gradient of an effective batch of sixteen while the GPU never holds more than the small physical batch the memory ceiling allows. You trade wall-clock for an effective batch the memory could never hold directly, and the trade is worth it because the foreground sparsity needs the averaging to stay stable.

The averaged batch fixes the variance of the gradient, but it does not fix what the gradient points at, and that is where the loss function does the work the batch cannot. We evaluated five losses against the multiclass objective: Dice, Focal, Lovasz, soft cross-entropy, and Tversky. The comparison was not academic, it decided the deliverable. Dice, the natural first choice for segmentation, left the curve classes weak: with Dice loss the multiclass intersection-over-union landed at 0.94 on the background mask but only 0.26 and 0.21 on the two curve masks, with per-curve F1 scores of 0.37 and 0.32 against 0.97 on background. Those background-versus-foreground gaps are the signature of a loss that is satisfied by getting the easy, abundant class right. Tversky, which exposes separate weights on false positives and false negatives, let us tilt the objective toward recall on the sparse foreground, and that tilt is what carried the final model [5]. Read on the deliverable rather than the mask, the difference is stark: Tversky reached a peak coefficient of determination of 0.9891 on a recovered curve where Dice's mean absolute error on the same curves sat at 0.0367 and 0.0774, against Tversky's 0.0277 and 0.1241, and Tversky's mean squared error fell to 0.0021 on the first curve. The lesson is that under sparse foreground the loss is not a hyperparameter you sweep at the end; it is a structural choice that the batch engineering makes possible but cannot substitute for.

III

THE SECOND CEILING

The bill is wall-clock, so cost the clock

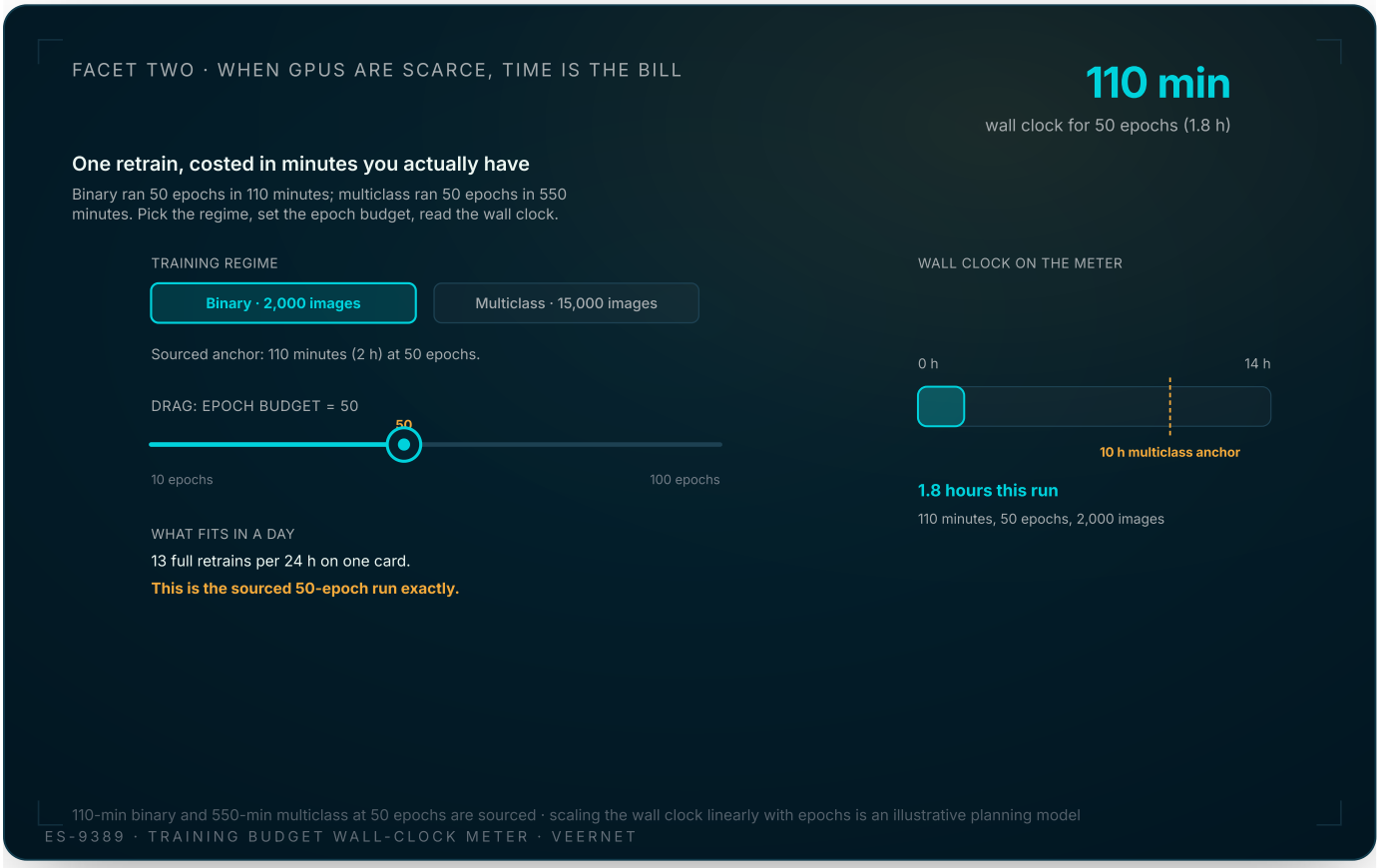
Training time is the only training cost that is actually scarce

When GPUs are rented and headcount is fixed, the marginal cost of an experiment is not money in any direct sense; it is time. A retrain you cannot afford to run twice in a working day is a retrain that slows every downstream decision, and the planning question that governs the project is simple to state and easy to get wrong: how many full retrains fit in the time we have. We answer it by costing training in wall-clock minutes against two firm anchors. The binary stage ran fifty epochs over its two thousand images in 110 minutes, roughly two hours. The multiclass stage ran fifty epochs over its fifteen thousand images in 550 minutes, roughly ten hours. Those two numbers are the calibration for every training-budget conversation we had.

The two anchors also tell you something the headline figures hide, which is that wall-clock does not scale linearly with the dataset. The multiclass set is seven and a half times the size of the binary set, two thousand images against fifteen thousand, yet the run is only five times longer, 110 minutes against 550. Part of that is fixed per-epoch overhead that amortises better over the larger set, and part is that the gradient-accumulation schedule changes the ratio of compute to optimiser steps between the two stages. The practical consequence is that you cannot extrapolate a training budget from images alone, you have to anchor on a measured run of the actual loop with the actual collate and accumulation in place. We measured both, which is why the meter has two calibration points rather than one curve fit through the origin. A planner who assumed linear scaling from the binary anchor would over-budget the multiclass run by the better part of two hours, and over-budgeting time is itself a cost when the schedule is one of the three ceilings.

The second instrument turns those anchors into a planner. Pick the regime, drag the epoch budget, and read the wall clock back in minutes and hours, with the count of full retrains that fit in a twenty-four-hour day on a single card. The fifty-epoch tick is marked, because

that is the figure that is actually sourced; everything either side of it is a straight-line estimate from the anchor, and the instrument says so. The point of putting it in front of a planner is to make the time cost of an idea visible before the idea is run. A proposal to push the multiclass stage from fifty to a hundred epochs is not an abstract quality bet; it is a decision to spend the better part of an extra working day of GPU time, and the meter shows that the moment you reach for it. The same meter quietly answers a question that comes up in every fixed-price engagement: how many experiments can the team actually afford. At ten hours per multiclass run, a single rented card gives you two full retrains in a day with margin for nothing else, which means a five-loss comparison like the one that chose Tversky over Dice is not an afternoon, it is the better part of a working week of GPU time. Seeing that on the meter changes how a team rations its experiments, and rationing experiments well is most of what staying inside the clock means.



A training-budget meter in wall-clock minutes. With headcount and money fixed, the real cost of an experiment is time, and the two known regimes set the scale: the binary stage ran 50 epochs over 2,000 images in 110 minutes (about 2 hours), and the multiclass stage ran 50 epochs over 15,000 images in 550 minutes (about 10 hours). Pick the regime, then drag the epoch budget; the meter reads the wall clock in minutes and hours and tells you how many full retrains fit in a day on one card. The 110-minute binary and 550-minute multiclass figures at 50 epochs are the engagement's own; scaling the wall clock linearly with epoch count is an illustrative planning model, exact at 50 epochs by construction and a straight-line estimate elsewhere, and is flagged as such.

Why ten hours is a feature, not a failure

It would be easy to read a ten-hour multiclass run as evidence of an under-resourced setup. It is the opposite. Ten hours for fifty epochs over fifteen thousand variable-dimension instances, on hardware that cannot hold sixteen of the widest logs at once, is what correctness costs when you refuse to resize the data and refuse to drop the foreground sparsity on the floor. The cheaper paths, downsampling every log to a fixed small box, or running a noisy batch of one through the harder three-class objective, would have finished faster and produced a worse model. Consider what downsampling would actually do: a curve trace that is one pixel wide on a 12,800-pixel log does not survive being squeezed into a 1,024-pixel box; it dissolves into the background or smears across neighbours, and the very signal the model exists to extract is destroyed before the first epoch. The ten hours is the cost of keeping that signal intact through the whole pipeline, padding instead of resizing, masking instead of cropping, accumulating instead of shrinking the batch. The wall clock is the honest price of a training loop that respects the data, and a blueprint that hides that price behind a faster, wronger loop is not a blueprint worth shipping. We would rather hand an operator a model that took ten hours and reads their narrowest curve than one that trained in two and learned to ignore it.

> **“Every shortcut that would have cut the ten hours also cut the curve quality. The clock was telling the truth about the work.”**
>
> — FROM OUR OWN TRAINING LOGS

IV

THE THIRD CEILING

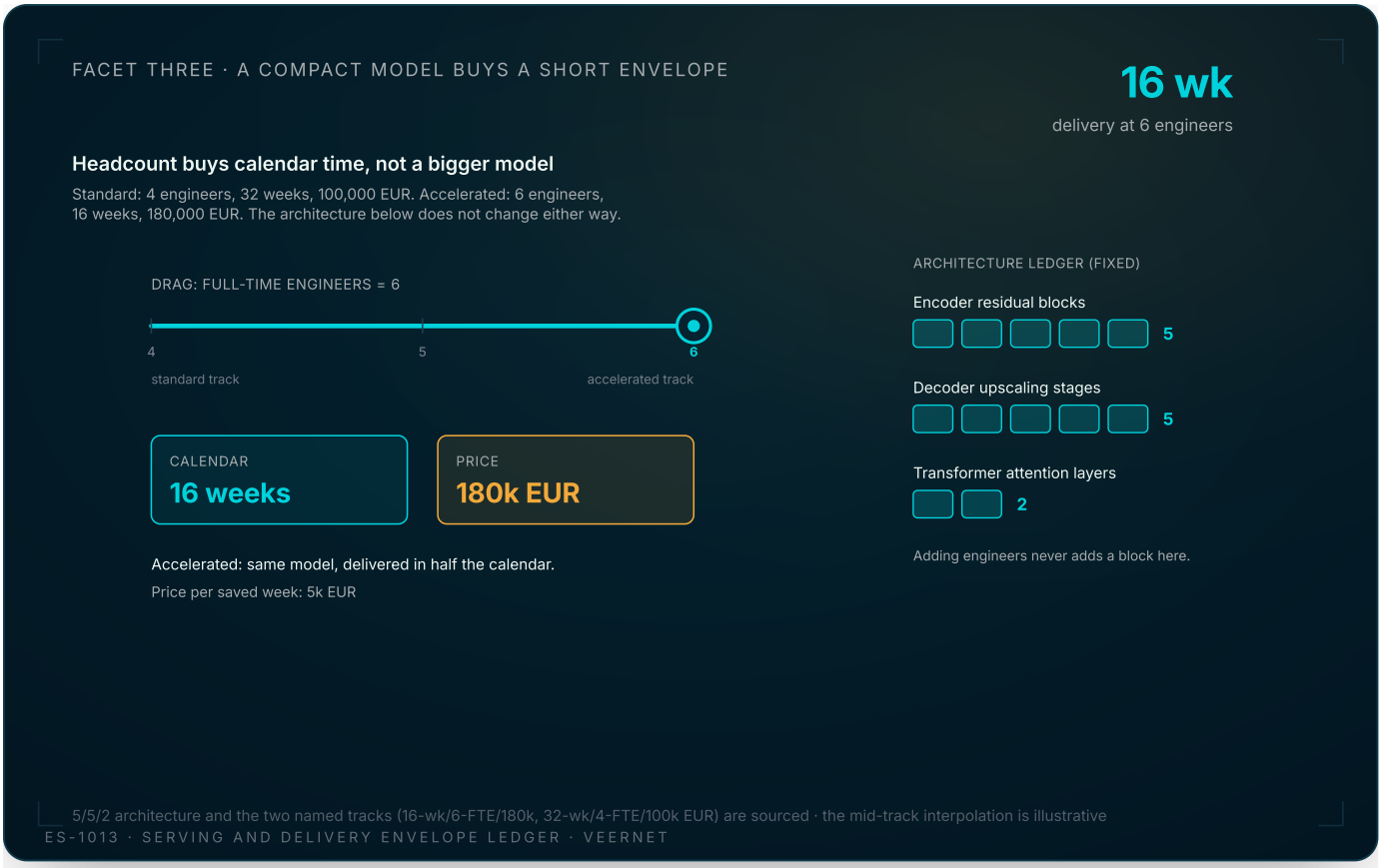
A compact model is what makes the envelope feasible

Headcount and price are bought with architecture restraint

The third ceiling is the combined one: people and money, expressed as a delivery envelope. The accelerated track was sixteen weeks, six full-time engineers, and 180,000 EUR. The standard track was thirty-two weeks, four engineers, and 100,000 EUR. Neither envelope leaves room for a sprawling system that takes a quarter just to stand up, and that is the deeper reason VeerNet is deliberately compact: a five-block residual encoder, a five-stage decoder, and only two transformer attention layers on the bottleneck [3][4]. The architecture is not small because the problem is easy. It is small because a small architecture is one a six-person team can build, debug, train inside the wall-clock budget, and hand to the operator's own people without a fleet of specialists to keep it alive.

The two attention layers deserve a word, because they are the one place the compact design spends complexity rather than saving it. A pure convolutional encoder-decoder sees only local neighbourhoods at each layer, and a well-log curve is a long, thin, globally coherent object: a stretch of trace at the top of the log constrains what is plausible at the bottom. Placing two self-attention layers at the bottleneck, where the spatial resolution is smallest and attention is therefore affordable, lets the network reason about the curve as a whole without paying the quadratic cost of attention at full resolution [4]. It is a deliberate, bounded investment, two layers and no more, in exactly the capability a five-by-five convolutional stack lacks. That restraint is the pattern of the whole architecture: spend where the data demands it, save everywhere else, and never add a block whose only justification is that a larger model might score marginally higher. The compactness is what keeps the model trainable inside the ten-hour multiclass budget on a single card, and a model that did not fit that budget would have broken the time ceiling to chase the envelope, which is no win at all.

The third instrument draws the relationship directly. The architecture block ledger on the right is fixed: five encoder blocks, five decoder stages, two attention layers, regardless of how the project is staffed. Drag the staffing lever between the standard and accelerated tracks and the envelope trades calendar time for headcount and price, but the model never changes. That is the load-bearing point. More engineers do not buy a bigger network; they buy the same network sooner. Moving from the standard four-engineer track to the accelerated six-engineer one halves the calendar from thirty-two weeks to sixteen, at the cost of moving from 100,000 to 180,000 EUR. The instrument lets a sponsor see exactly what each saved week costs, which is the conversation a fixed-price engagement actually turns on. The arithmetic is worth saying plainly, because it is not the arithmetic of pure linear scaling: halving the calendar costs eighty percent more money, not the fifty percent a naive doubling of throughput would predict, because the extra two engineers are coordination overhead as well as capacity, and because the accelerated track front-loads work that the standard track could spread. The ledger surfaces that premium so a sponsor decides on the real number. Speed is buyable here, but it is not cheap, and the right answer depends on what the operator is paying for the digitised curves to unlock downstream, not on a reflex that faster is always better.



A serving-and-delivery cost ledger. The model is small by design: a 5-block encoder, a 5-stage decoder, and 2 transformer attention layers on the bottleneck, and that compactness is what makes a short delivery envelope feasible. Drag the staffing lever between the standard track (4 engineers, 32 weeks, 100,000 EUR) and the accelerated track (6 engineers, 16 weeks, 180,000 EUR) and watch the envelope trade calendar time for headcount and price. The architecture ledger on the right never changes: more people buy the same model sooner, not a bigger one. The 5-encoder, 5-decoder, 2-attention architecture and the two named delivery envelopes are the engagement's own figures; interpolating weeks and price between the two tracks as the lever moves is an illustrative model, with only the two endpoints sourced, and is flagged as such.

The handover is inside the budget, not after it

A digitiser that only its builders can run has not been delivered, it has been demonstrated. The headcount ceiling cuts both ways: it limits the team that builds the system and it describes the team that must keep it. So the compact architecture is doubly justified, because the same restraint that lets six engineers build it in sixteen weeks lets the operator's own staff operate it afterward without hiring a research group. The persistent shortage of in-house ML capability that operators report [6] is precisely why a model you can hand over matters more than a model that scores a fraction of a point higher on a benchmark and needs its authors on call forever. We budget the handover, the runbooks,

and the operator-side training inside the envelope, not as a closing courtesy bolted on at the end, because a system nobody on the customer's side can rebuild a year later was never really delivered against the price they paid.

This is also where the batch-system shape pays off a second time. Because there is no live service to keep up, the operating burden after handover is genuinely small: there is no on-call rotation to staff, no latency budget to defend, no autoscaling policy to tune. The operator's team needs to be able to run the loop over a new tranche of scans, read the wall-clock cost off the same two anchors we used, and review the curves the model emits. That is a skill set an existing subsurface data team can absorb, not a new function they have to hire. The compact architecture, the documented collate and accumulation logic, the batch-independent normalisation that does not need re-tuning when the data shifts: each of these was chosen partly so that the handover is a teachable afternoon rather than a months-long apprenticeship. A blueprint that respects the headcount ceiling on the build side and forgets it on the operate side has only solved half the problem, and the half it skipped is the half the operator lives with.

V

THE NUMBERS

What the three ceilings produced

The result, stated against the constraints

A blueprint is only worth the artefact it produces, so here is the artefact in the terms the ceilings set. Under the memory ceiling, the binary stage trained correctly at a physical batch of one and the multiclass stage reached an effective batch of sixteen through the padded collate function and gradient accumulation, on a card the budget could afford, with GroupNorm keeping the single-image normalisation valid. Under the time ceiling, the binary stage completed fifty epochs in 110 minutes and the multiclass stage in 550 minutes, both reproducible figures a planner can build a schedule on. Under the envelope, the compact five-five-two architecture was built and handed over inside the sixteen-week, six-engineer, 180,000 EUR accelerated track.

THE BLUEPRINT, IN THE UNITS THAT WERE SCARCE

1

Physical batch on the binary stage, set by the memory ceiling

16

accumulation

Effective batch on multiclass, bought via padded collate

550 min

Wall clock, multiclass, 50 epochs over 15,000 images

180,000 EUR

vs 100,000

Accelerated envelope: 16 weeks, 6 engineers

The model itself, graded on its deliverable rather than its mask, reached a peak coefficient of determination of 0.9891 against native LAS curves with a lowest mean absolute error of 0.0132 and a lowest mean squared error of 0.0004, which the companion evaluation whitepaper covers in full. It is worth stating plainly why we report the deliverable metric and not the mask metric, because the gap between them is the whole point of the design. On the mask the peak intersection-over-union was only 0.51 and the peak F1 was 0.55, numbers that look mediocre next to a segmentation leaderboard. But a one-pixel-wide curve trace punishes mask overlap metrics structurally: miss the trace by a single pixel and the intersection collapses even though the recovered curve, read as a function of depth, is almost exactly right. The peak recall of 0.97 tells the truer story, the model finds the trace, and once the trace is found the curve it produces tracks the native LAS measurement to a

coefficient of determination of 0.9891. Grading a thin-object digitiser on mask IoU is grading it on the wrong axis; grading it on the recovered curve is grading it on the deliverable the operator actually pays for.

What matters for this document is that those quality numbers were reached without breaching a single one of the three ceilings. The binary stage trained at a physical batch of one. The multiclass stage trained at an effective batch of sixteen on a card the budget allowed. The full five-loss comparison that selected Tversky, two firm wall-clock anchors, and the compact architecture all fit inside the sixteen-week accelerated envelope. The constraints did not degrade the result; designing around them produced it. A team that had treated the memory ceiling as a problem to be funded away, rather than the constraint that sets the batch, would have spent the budget on a bigger card and still had to solve the variable-dimension and sparse-foreground problems that the collate, the accumulation, and the loss choice actually solved.

VI

THE METHOD

The moves, in the order a planner needs them

The blueprint as a sequence of decisions

Read end to end, the blueprint is an ordered set of decisions, each one forced by a ceiling and each one constraining the next.

- **Start from the widest input.** The memory ceiling is set by the largest plausible image, near 12,800 pixels, not the average. Size the physical batch to survive that worst case, and accept that it will be small. Everything downstream follows from this single sizing.
- **Pick normalisation that survives a batch of one.** Because the physical batch is forced small, a batch-statistic normalisation would silently poison the run. GroupNorm's batch independence [2] is mandatory in this regime, not optional.
- **Decouple effective batch from physical batch.** When the optimiser needs more averaging than the memory can hold, a custom collate function that pads and masks plus gradient accumulation buys the effective batch in software [5]. Pad to enable the stack, mask to protect the gradient, accumulate to recover the batch.
- **Choose the loss to match the sparsity, then measure it.** The batch engineering fixes gradient variance but not gradient direction. Against three classes with a foreground that occupies a fraction of a percent of the pixels, we compared five losses and let Tversky's tunable false-positive and false-negative weights carry recall on the curves [5], where Dice left the curve masks at intersection-over-union of 0.26 and 0.21. Select the loss on the recovered-curve metric, not the mask.
- **Cost every experiment in wall-clock minutes.** With GPUs rented and headcount fixed, time is the scarce input. Anchor on the 110-minute binary and 550-minute multiclass figures at fifty epochs, remember the scaling is not linear in dataset size, and plan retrains against the day, not against an abstract compute budget.
- **Keep the architecture small enough to hold and hand over.** The five-five-two design is what lets six engineers build it in sixteen weeks and the operator's own team run it afterward. Restraint in the model is what makes the envelope feasible and the handover real.

Each decision narrows the next. The widest input sets the physical batch; the small physical batch forces batch-independent normalisation; the small batch on a sparse objective forces the collate-and-accumulation recovery and the loss choice; the wall-clock those choices produce sets the experiment budget; and the experiment budget plus the headcount cap forces the compact architecture that both fits the clock and survives the handover. There is no step you can take in isolation, which is exactly why we present the blueprint as a chain rather than a menu. A team that reaches for a bigger card to escape the first link finds the rest of the chain unchanged: the data is still variable-dimension, the foreground is still sparse, the loss still has to be chosen, and the operator still has to be able to run the thing. The memory ceiling was never the real difficulty; it was the forcing function that exposed the difficulties that were there all along.

VII

THE IMPLICATIONS

What a constraints-first blueprint changes downstream

Where this leaves the next engagement

The reason this blueprint is worth writing down is that the conditions that produced it are not unusual; they are the normal conditions of machine-learning work inside an operating company rather than a lab. The dataset ratio that opened this document, better than seventeen scanned images for every machine-readable one, repeats across every operator with a paper archive, and the in-house ML capability needed to attack it is exactly the capability that survey after survey finds scarce in the sector [6]. The next engagement that looks like this one will again have a memory ceiling set by an outlier input, a clock that is the true currency, and a delivery envelope that is fixed before the work starts. What changes if a team carries this blueprint in is that none of those three is treated as a surprise. The batch is sized from the worst-case input on day one. The training budget is anchored on a measured run before the schedule is promised. The architecture is held compact deliberately, so the handover is a teachable afternoon and the operator is not left dependent on the people who built it.

There is a roadmap implication too. Because the system is a batch digitiser with no live-service surface, scaling it from the slice we trained on toward the full 136,771-image archive is a throughput problem, not a re-architecture: the same loop runs over more scans, the wall-clock anchors predict the cost, and the compact model trains on whatever fresh tranche the operator labels. The expensive, irreversible decisions, the normalisation, the collate, the loss, the architecture, were all made under the ceilings and do not have to be remade to grow. That is the quiet dividend of designing the constraints first. A blueprint built to survive the tightest version of the problem scales up gracefully, where a blueprint built for the comfortable case has to be torn down the moment the comfort runs out.

WHAT TO CARRY OUT OF THIS

1. Under hard ceilings the constraints are the blueprint. We designed the **memory**, the **time**, and the **money** first and the model second.
2. The widest log, near 12,800 pixels, sets the physical batch. The binary stage was forced to **1**; the multiclass stage reached an effective batch of **16** through a padded collate function and gradient accumulation.
3. Training cost is wall-clock: **110 minutes** for the binary stage and **550 minutes** for the multiclass stage, both at fifty epochs. Cost the clock, not an abstract compute budget.
4. A compact **5-encoder, 5-decoder, 2-attention** architecture is what makes the **16-week, 6-engineer, 180,000 EUR** envelope feasible and the handover real.
5. GroupNorm is not a preference at a physical batch of one; it is the only normalisation that stays valid in the regime the memory ceiling imposes.

GLOSSARY

Collate function — The component a data loader calls to turn a list of sampled examples into the batched tensors a training step consumes. Our custom version pads each mini-batch to its widest member and carries a validity mask so padding never contributes to the loss.

Delivery envelope — The fixed combination of calendar time, headcount, and price that bounds the engagement. The accelerated track is sixteen weeks, six engineers, and 180,000 EUR; the standard track is thirty-two weeks, four engineers, and 100,000 EUR.

Effective batch — The number of images whose gradients are averaged before a single optimiser step. A software variable, decoupled from the physical batch by gradient accumulation. We reached sixteen on the multiclass stage.

Gradient accumulation — Running several small forward and backward passes, piling gradient into the parameter buffers, then taking one optimiser step on the accumulated total. Buys a large effective batch without the memory cost of holding it all at once.

GroupNorm — Group Normalization. A normalisation scheme whose statistics do not depend on the batch, so it behaves identically at a physical batch of one, the regime the memory ceiling pushes us into.

LAS — Log ASCII Standard, the canonical text format for digital well-log curves. The deliverable a digitised raster log is exported to and graded against.

Physical batch — The number of images the GPU actually holds in memory during one forward and backward pass. Pinned to the

memory ceiling that the widest log sets. On the binary stage it was forced to one.

References

1. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., Dennison, D. (2015). *Hidden Technical Debt in Machine Learning Systems*. NeurIPS. https://papers.nips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
2. Wu, Y., He, K. (2018). *Group Normalization*. ECCV. https://arxiv.org/abs/1803.08494
3. Ronneberger, O., Fischer, P., Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. MICCAI. https://arxiv.org/abs/1505.04597
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). *Attention Is All You Need*. NeurIPS. https://arxiv.org/abs/1706.03762
5. Salehi, S. S. M., Erdogmus, D., Gholipour, A. (2017). *Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks*. MLMI. https://link.springer.com/chapter/10.1007/978-3-319-67389-9_44
6. Koroteev, D., Tekic, Z. (2021). *Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future*. Energy and AI. https://www.sciencedirect.com/science/article/pii/S2666546820300033

Get the full whitepaper

This page is the long-form summary. The complete whitepaper adds the per-stage memory profile at the 12,800-pixel worst case, the reference implementation of the padding-and-masking collate function, the full epoch-by-epoch wall-clock breakdown for both stages, and the staffing-plan derivation behind the sixteen-week accelerated envelope.

EARTHSCAN

An MLOps Blueprint for Vision Models in Resource-Constrained Industry

Published November 2023. Generated 2026-07-22.
https://www.earthscan.io/whitepapers/mlops-blueprint-resource-constrained-vision