The result in 10 seconds
MicroZoom ran and trained on one 24 GB RTX 3090 Ti. The trained AI beat untrained MicroZoom, but five seeds and five magnifications did not establish an overall advantage over ordinary Lanczos enlargement. The AI added plausible texture, not the real fibres in the reference photograph. Our practical result is a repeatable research path and a justified stop, not a production image tool.

Why we explored it

MicroZoom starts with a photograph of one physical object and several registered microscope close-ups. It learns the object's texture and can then synthesize an image that remains detailed at very large scale.

That promise matters to Instavar for two reasons. Creatively, it points toward product images that invite people to explore materials at unusual depth. Operationally, it tests whether research designed for expensive data-centre GPUs can be explored on hardware we already own.

The first principle is simple: a model has to fit before it can be useful, but fitting does not make it useful. We therefore separated the work into two questions:

1. Can we make the workflow execute within 24 GB of GPU memory?
2. If it executes, does training improve an image the model did not train on?
3. If training helps, does it beat ordinary image enlargement?

We first corrected our understanding of the product

MicroZoom is not a general image enhancer. It cannot take an arbitrary folder of Instavar images and reveal details that were never captured.

Each new object needs its own prepared dataset: a full photograph, aligned close-ups, masks, region maps, scale information, and written descriptions. The model then generates plausible texture. It does not recover unseen physical truth.

This changed the business question. We were no longer testing a universal upscaler. We were testing a specialist creative imaging method for objects that can be photographed under a controlled capture process.

Where this applies: planned product, material, art, or educational imagery where plausible synthesis is acceptable.

Where it does not apply: scientific inspection, defect detection, evidence, conservation records, or any use where a generated fibre could be mistaken for a fibre that was actually observed.

Making the experiment fit

The original workflow keeps several large models in GPU memory. We reduced the frozen parts to INT4, a compact numerical format, while leaving the small parts being trained in BF16, a higher-precision format.

In plain language, we compressed the parts that only needed to be read and kept the parts that needed to learn in a form suitable for training.

The final test used:

Part	Tested choice
GPU	One RTX 3090 Ti with 24 GB
Training crop	1024 by 1024 pixels
Images processed together	One
Trainable adapter	Rank 2 LoRA
Frozen model weights	INT4
Trainable weights	BF16

This was not a free speed-up. Loading and converting the models took most of the inference time. INT4 solved a memory problem in this experiment. It did not make startup fast.

We climbed an evidence ladder

Rather than spend hours on a long run whose first step might fail, we increased the cost only after each smaller test passed.

We first checked that gradients could reach the trainable adapters. Then we ran two 256 pixel steps and restored a checkpoint in a new process. Next came one monitored 1024 pixel step. Finally, we ran six 1024 pixel steps, saved a checkpoint after every step, sampled GPU memory every 200 milliseconds, and tracked the loss.

All six steps completed. The run crossed both of MicroZoom's training stages, and the saved adapter files changed only when their corresponding stage was active. This told us the intended training path executed, rather than merely finishing without an error.

Peak observed use was 23,194 MiB, leaving 919 MiB free. That is a fit, but not comfortable headroom. The workflow needs exclusive access to this GPU. Another model service or training job can make the same run fail.

What this establishes: short, checkpointed execution on the pinned tote fixture, software stack, and RTX 3090 Ti.

What it does not establish: long-run stability, safe GPU sharing, support for every 24 GB card, or useful image quality.

Then we tested whether training helped

A training loss tells us that the optimizer is changing weights. It does not tell us whether the resulting image is better.

We therefore held one microscope close-up out of training. From it, we made a fixed 1024 pixel centre crop and compared the pretrained model with later checkpoints. Every comparison used the same input, target, seed, and 28-step generation process.

The rank-2 checkpoint after 30 training steps improved all six tracked similarity measures over the pretrained control on that one crop. Increasing the adapter to rank 4 made every measure worse. Giving more steps to the second training stage also made five of the six measures worse than the original 15-step plus 15-step schedule.

Those negative results were valuable. They stopped us from spending more GPU time on changes that sounded larger or more powerful but performed worse in the case we could actually measure.

Visual inspection added the most important qualification: the best result was still smooth and did not reproduce individual fibres. The scores show closer agreement with one target crop. They do not turn generated texture into observed microscopic fact.

Where this result applies: one held-out tote crop, one seed, one 28-step generation setting, the pinned software environment, and the tested checkpoints.

Where it does not yet generalise: other crops, objects, seeds, scales, people's visual preferences, factual detail, or production outcomes.

Then the simple method beat the AI

Improving over the untrained model was not enough. A useful system should also beat the simplest reasonable alternative.

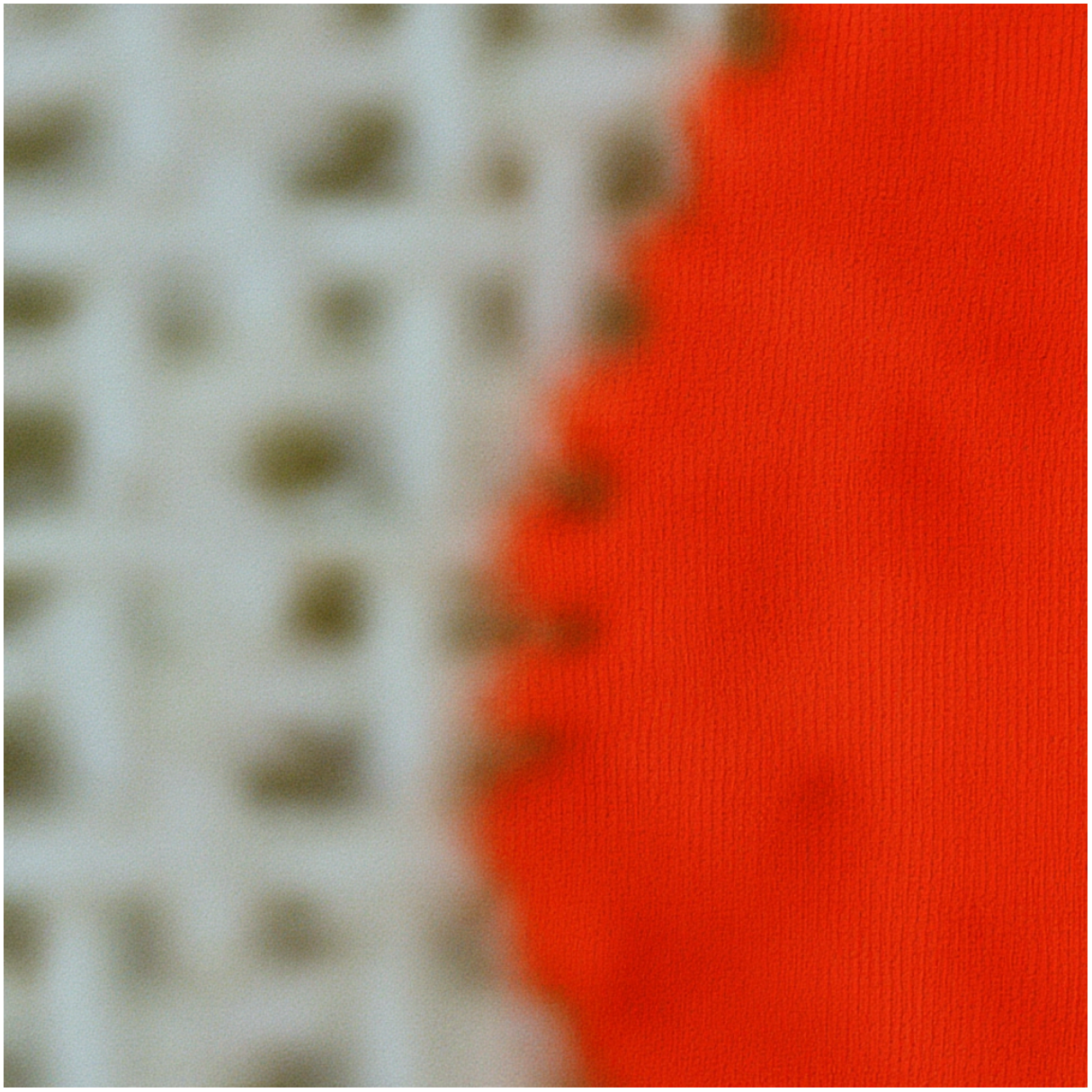
We therefore compared the trained MicroZoom output with Bicubic and Lanczos. Both are ordinary image-enlargement methods. They estimate new pixels with fixed mathematics and require no model, training, or GPU. Every method started from the same tiny 27 by 27 input and was compared with the same real 1024 by 1024 microscope crop.

The real held-out photograph

This image was excluded from training. It is the reference that all three methods tried to match.

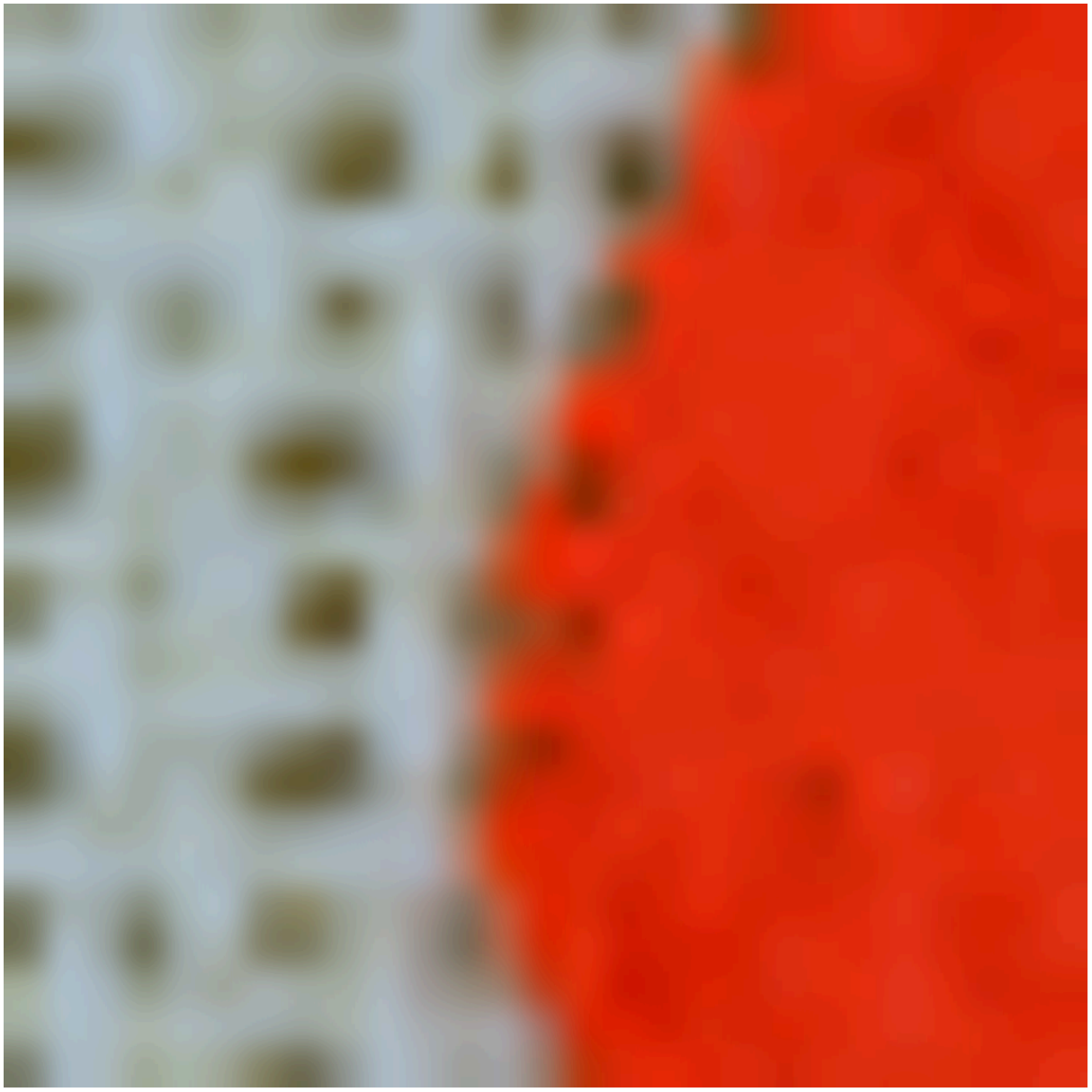
Rank-2 LoRA: the trained MicroZoom result

The AI generates visible texture, especially across the orange region. But it does not reconstruct the interwoven fibres in the real photograph. It is a plausible surface, not the observed microscopic structure.



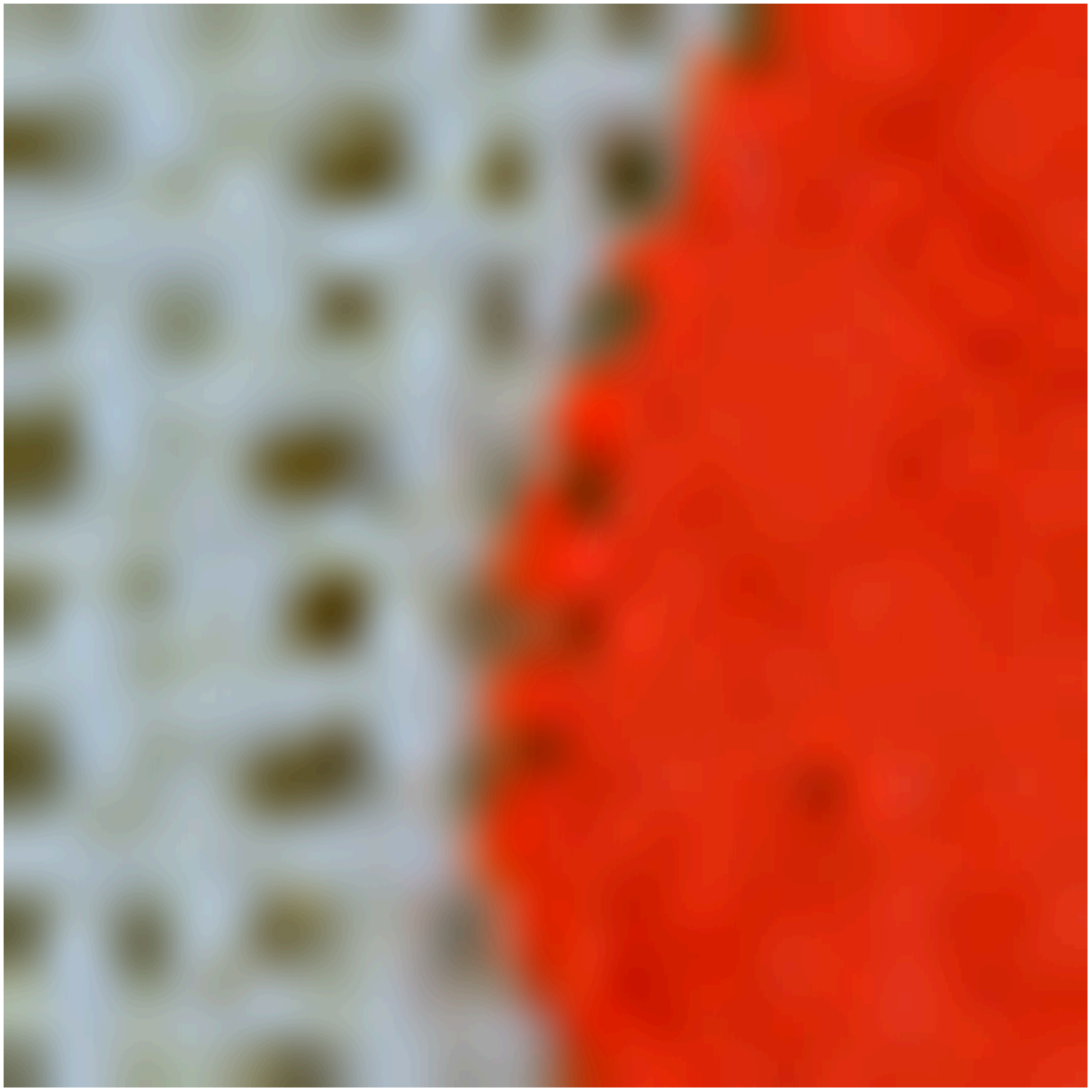
Bicubic: the familiar smooth enlargement

Bicubic estimates new pixels from nearby pixels. It stays blurry because the 27 by 27 input contains too little information to reveal real fibres.



Lanczos: the strongest simple baseline

Lanczos is another fixed enlargement method. Its result looks similar to Bicubic, with a slightly different edge treatment. It beat the trained AI on PSNR, SSIM, DISTS, LPIPS, and boundary error. The AI won only the frequency measure, which rewards its added texture without proving that the texture is correct.



This is the practical finding: more visible detail is not necessarily more true detail. For this one crop, the cheaper method stayed closer to the photograph.

Where this result applies: one registered tote crop and the exact degradation used in our experiment.

Where it does not generalise: other objects, capture systems, scales, or a revised MicroZoom model. It does justify stopping larger experiments until the AI can first beat this simple baseline.

We challenged the negative result

One seed and one extreme scale could have produced a misleading result. We therefore kept the checkpoint, target, and 28-step generation process fixed and

tested five diffusion seeds at 37.372x.

Seed choice changed the image and the balance between measures. Four seeds beat Lanczos on DISTS, LPIPS, and the frequency measure, while every seed lost on PSNR, SSIM, and boundary error. No seed won a majority of the six measures. Seed 42 was unusually weak on two learned perceptual scores, but replacing it did not turn MicroZoom into the stronger overall method.

We then gave MicroZoom a favorable test. Seed 123 had the best PSNR and frequency score among the five, so we used it at five magnifications:

Magnification	Measures MicroZoom won against Lanczos
2x	0 of 6
4x	0 of 6
8x	2 of 6
16x	3 of 6
37.372x	3 of 6

Our promotion rule required at least four wins at one scale, followed by a visual check for invented structure. MicroZoom reached neither condition. At 2x and 4x, Lanczos won every measure. At larger scales, MicroZoom restored more visible texture and improved some perceptual scores, but it generated clean, regular fabric patterns that were not the irregular fibres in the reference.

This strengthens the decision without making it universal. It covers one tote crop, one checkpoint, five seeds at the extreme scale, and one favorable seed across five scales. It does not tell us what a revised model or a separately captured object would do.

The failed attempts made the release better

Several failures had nothing to do with the research idea. A sample path did not match the code, one scale file used a format the parser rejected, optional Deep Zoom code blocked ordinary runs, dependencies were missing, and CUDA selected a compiler version it could not use.

We fixed or documented each issue and repeated bounded checks from a clean release checkout. This matters because a research result is only useful to another team if they can tell exactly what ran, what failed, and what changed.

The broader operational lesson is reusable: distinguish model failure from environment failure, and distinguish a warm file cache from an actual software speed improvement.

What Instavar gained

The immediate value is not a product launch. It is a lower-cost research capability and a better decision process.

- We can test this class of large image model on hardware we already own.
- We now have a checkpointed, monitored path instead of a one-off demo.
- We have a held-out evaluation loop that can reject unhelpful training changes.
- We now require every trained image model to beat a simple non-AI baseline before it earns a larger experiment.
- We know that the baseline decision survived both seed variation and a favorable-seed magnification curve in the tested case.
- We know the capture work required before trying a real Instavar object.
- We know which claims are safe to make and which would overstate the evidence.

This pattern generalises beyond MicroZoom: reduce a new research claim to one answerable question, run the cheapest test that could disprove it, preserve the evidence, and increase cost only when the result earns the next step.

What we released

The source-only compatibility port is public at instavar/microzoom-int4. It contains the code, pinned requirements, test scripts, machine-readable result manifests, the experiment record, and the autonomous evaluation protocol.

The source inherited from MicroZoom is MIT licensed. The complete model stack is not. FLUX.1-dev and the Jasper ControlNet use the FLUX.1-dev Non-Commercial License. We therefore do not distribute their weights, our INT4 exports, or our LoRA checkpoints. Users must obtain gated model components from their owners and accept their terms.

The honest conclusion

We proved that a carefully bounded MicroZoom inference and LoRA training path can run on our 24 GB RTX 3090 Ti. The rank-2 checkpoint scored better than the pretrained control on one unseen tote crop. Across five seeds and five tested magnifications, however, it never met our threshold for an overall win against simple Lanczos enlargement.

We did not prove that MicroZoom reconstructs true microscopic detail, that the improvement transfers to another object, or that the workflow is ready for a customer production pipeline.

The next experiment should not be longer training or a more expensive new object capture. MicroZoom first needs a revised configuration that beats the registered classical baseline on the existing held-out crop. Only then would a new, independently captured object be worth the extra capture and GPU cost.

References and evidence

- MicroZoom paper
- MicroZoom project page
- MicroZoom source repository
- Instavar MicroZoom INT4 release
- FLUX.1-dev license
- Jasper ControlNet model card