

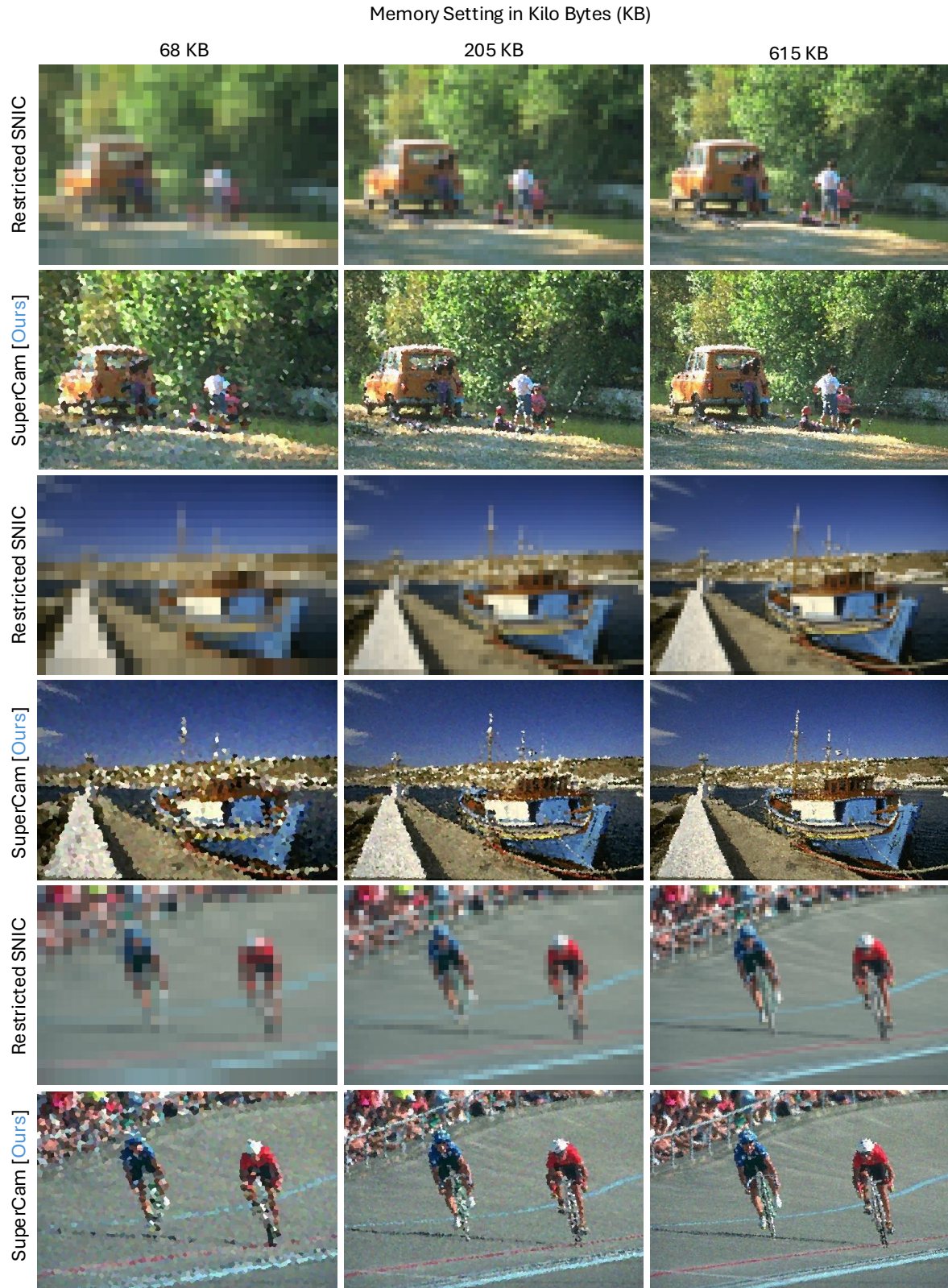
Supplementary Document for “Computer Vision with a Superpixelation Camera”

Sasidharan Mahalingam, Rachel Brown, Atul Ingle

Supplementary Note 1. Superpixel Segmentation Results

Suppl. Fig. [1](#) shows more qualitative visual comparisons generated using SuperCam and memory restricted SNIC [\[1\]](#) at varying memory budgets, shown here *without* Gaussian blur applied.

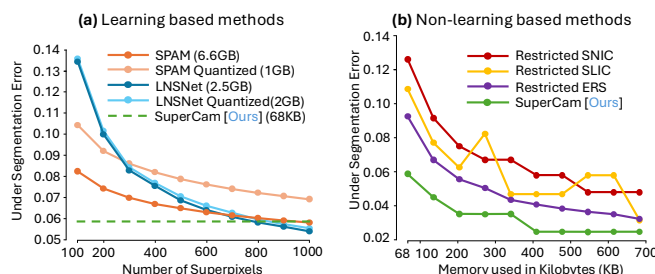
In the remaining supplement sections we provide additional qualitative and quantitative evaluations, including comparisons with additional superpixel algorithms and results for three computer vision tasks: image segmentation, object detection and monocular depth estimation. We also provide mathematical definitions for the quantitative error metrics that we used and additional detail regarding how we derived the optimal Gaussian blur kernel.



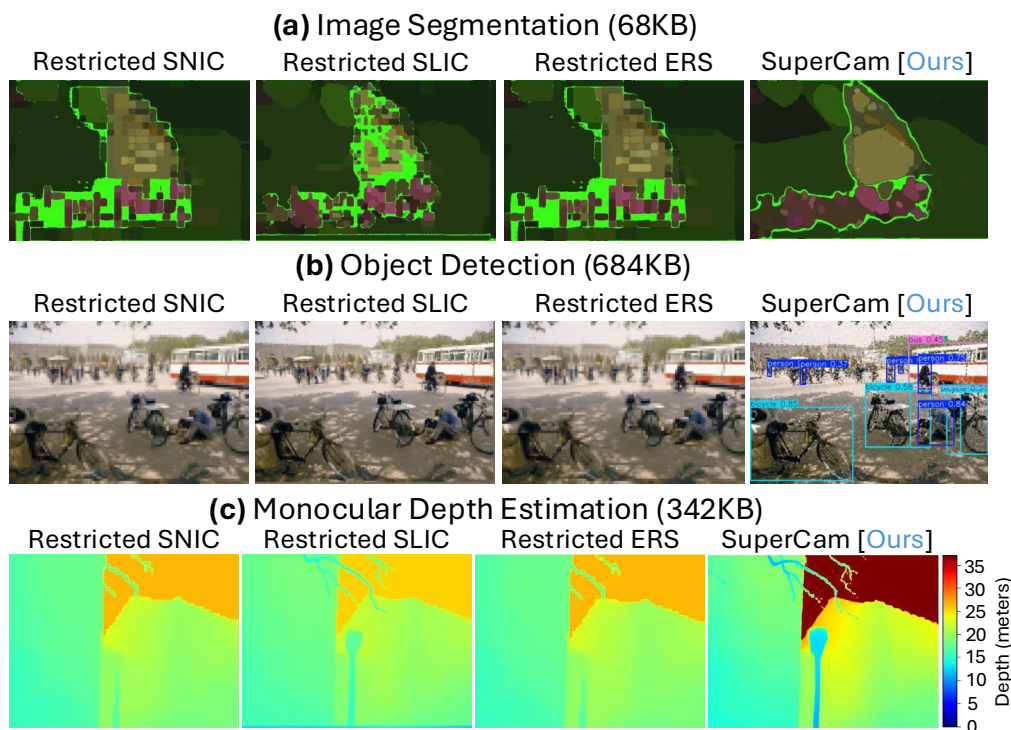
Supplementary Figure 1. **Comparison of memory-restricted SNIC and SuperCam superpixel images.** Images taken from the BSD500 dataset, shown at different memory settings in kilobytes (KB). These are raw output images *without* Gaussian blur applied. SuperCam results show better visual details and less aliasing.

Supplementary Note 2. Comparisons with other superpixel algorithms

In the main paper we compared SuperCam results with memory restricted SNIC. Here we compare SuperCam with two recent learning based superpixel algorithms, SPAM [59], LNSNet [75], and also two memory restricted non-learning based algorithms, SLIC [2] and ERS [36]. Suppl. Fig. 2 shows quantitative results for the Under Segmentation Error for the two learning based superpixel algorithms and all four non-learning based methods at a range of memory levels. In Suppl. Fig. 3 we also show a snapshot of the qualitative results for the downstream computer vision tasks of Image Segmentation with Segment Anything Model v2 [44], Object Detection with YOLOv12 [53], and Monocular Depth estimation with Depth Anything v2 [69] using images from the BSD500 [40], COCO [35] and DIODE [57] datasets.



Supplementary Figure 2. **Comparisons of SuperCam with other superpixel algorithms.** Quantitative comparisons of SuperCam with other learning based and non-learning based superpixel algorithms. **(a) Comparisons of SuperCam with learning based superpixel algorithms:** We show comparisons of SuperCam with two recent learning based algorithms SPAM and LNS-Net. **(b) Comparisons of SuperCam with non-learning based superpixel algorithms:** Here we show comparisons of SuperCam with memory restricted SNIC, SLIC and ERS on the BSD500 dataset. We can see that the overall trend is similar for all the memory restricted algorithms and SuperCam performs better than all algorithms.



Supplementary Figure 3. **Results of downstream computer vision applications for other non-learning based superpixel algorithms.** We show a snapshot of qualitative comparisons for SuperCam against memory restricted SNIC, ERS and SLIC. SuperCam performs better than all the other algorithms on all computer vision tasks.

Supplementary Note 3. Superpixel Evaluation

In this section we provide all the formulae for various metrics used to evaluate the superpixel segmentation algorithms.

The Under Segmentation Error is defined as:

$$UE_{NP}(G, S) = \frac{1}{N} \sum_{G_i} \sum_{S_j \cap G_i \neq \emptyset} \min\{|S_j \cap G_i|, |S_j - G_i|\} \quad (S1)$$

where S_j where $1 \leq j \leq K$ and G_i where $1 \leq i \leq K$ are the corresponding partitions of the same image and N is the total number of pixels in the image. We use the implementation of the Under Segmentation Error given in [49].

We also show superpixel algorithm performance using a precision vs recall plot. Precision is defined as:

$$Pre(S, G) = \frac{TP}{TP + FP} \quad (S2)$$

where TP is the number of true positives and FP is the number of false positives. TP and FP are calculated as:

$$TP(S, G) = \sum_{i=1}^N \mathbb{1}_{j \in \mathcal{N}(i, \epsilon)}(S_j, G_i) \quad (S3)$$

$$FP(S, G) = \sum_{i=1}^N [1 - \mathbb{1}_{j \in \mathcal{N}(i, \epsilon)}(S_j, G_i)] \quad (S4)$$

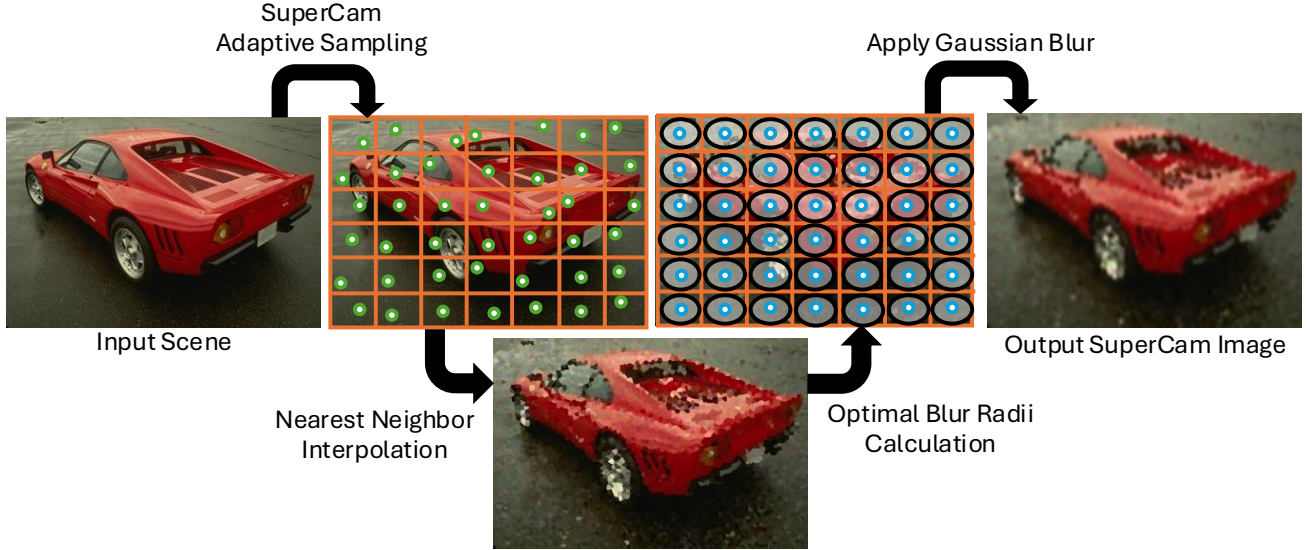
where \mathcal{N} represents a $\epsilon \times \epsilon$ boundary around the pixel position i . S_j where $1 \leq j \leq K$ and G_i where $1 \leq i \leq K$ are the corresponding partitions of the same image and N is the total number of pixels in the image. The function $\mathbb{1}$ returns 1 if a superpixel boundary overlaps with the ground truth boundary pixel within the neighborhood \mathcal{N} and 0 otherwise. ϵ is defined as $(2r + 1)$, where r is 0.0025 times the diagonal of the image rounded to the next integer. Recall can be calculated as:

$$Rec(S, G) = \frac{TP}{TP + FN} \quad (S5)$$

where FN can be expressed as:

$$FN(S, G) = \left(\sum_{i=1}^N G_i \right) - TP \quad (S6)$$

We found that applying a post-processing Gaussian blur to both the SuperCam and the SNIC superpixel images, improved performance for all the computer vision models we tested (Segment Anything Model 2 [44], YOLOv12 [53] and DepthAnythingV2 [69]). In order to calculate the optimal blur that has to be applied for the SuperCam images, we did both a theoretical and an empirical analysis. In the next section we show the explanation of the experiments we carried out.



Supplementary Figure 4. **SuperCam Algorithm.** We show a diagram of the proposed SuperCam algorithm. The pixels are first adaptively sampled to create an initial superpixel image that has holes. The holes are filled using nearest neighbor interpolation and a Gaussian Blur is applied to it to generate the final SuperCam image. The circles with the green border are the pixels that are exposed adaptively. The ellipses are the calculated optimal 2D Gaussian blur kernels for the pixel locations marked with blue borders.

Supplementary Note 4. Deriving the Optimal Blur Kernel

We apply a post-processing Gaussian blur on the raw superpixel image produced by SuperCam. This improves the performance of all applications (image segmentation, object detection and monocular depth estimation) for both the SuperCam and SNIC images. We provide the theoretical and empirical derivation of the optimal blur for SuperCam. It is not possible to derive an optimal blur kernel for SNIC due to the absence of either a lower or upper bound on the superpixel size. However, empirically we found that the same size blur kernel derived for SuperCam also improves performance for SNIC superpixel images.

Supplementary Note 4.1. Theoretical Derivation of the Optimal Blur Kernel for SuperCam

Suppl. Fig. 4 shows a pictorial representation of the SuperCam algorithm. As the proposed SuperCam divides the image into rectangles of equal size and seeds each superpixel by a randomly chosen location within the rectangle, we know that the seeded location cannot be outside its corresponding grid. The intuition behind applying the Gaussian blur kernel is that doing so smooths the transition across superpixels. As each superpixel in a SegCam image will be within the initial grid of uniformly shaped rectangles, we apply a 2D Gaussian blur kernel with blur radii in each direction (width and height), equal to half of the width and height of the initial superpixel grid. The conversion from the blur radius to the standard deviation (σ) is done as follows:

A 1D Gaussian blur kernel used can be represented as:

$$g(x) = 255e^{-\frac{1}{2\sigma^2}x^2} \quad (S7)$$

where σ is the standard deviation of the blur kernel applied and x is the distance from origin. For a digital 1D signal the lowest value that is non-zero is 1, and we want this to happen exactly when we reach a distance of r , which is the known blur radius for that kernel. As the radius has to be non-inclusive, we substitute $r + 1$ instead of x in eqn S7. Doing this we get:

$$1 = 255e^{-\frac{1}{2\sigma^2}(r+1)^2} \quad (S8)$$

Now rearranging and taking log on both sides, we end up with the relation:

$$r = \sigma\sqrt{2\log_e 255} - 1 \quad (S9)$$

This is the relation between the blur radius r and the standard deviation of the blur kernel σ for a 1D Gaussian blur operation. We use the fact that the 2D Gaussian blur kernel is separable to calculate the optimal blur kernel sigma for both the x and y dimensions of the image and apply two 1D Gaussian blur kernels.

Supplementary Note 4.2. Empirical Derivation of the Optimal Blur Kernel for SuperCam

In order to confirm our theoretical calculation for the optimal blur radii, we conducted experiments testing image segmentation using several different multiples of our theoretical value. The resulting error metrics confirmed that the theoretically derived optimal blur kernel produced the lowest error.

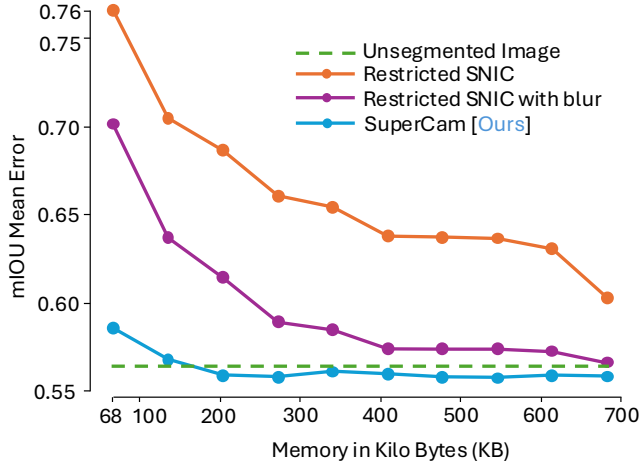
Supplementary Note 5. Image Segmentation

For validating the image segmentation results, we run Segment Anything Model V2 [44] on the BSD500 [40], NYUV2 [47] and SBD [21] datasets and compare them using the mIOU Error metric. In this section we go through the definitions of the error metric used and also provide supplementary results to those show in the paper. The mIOU Error is defined as:

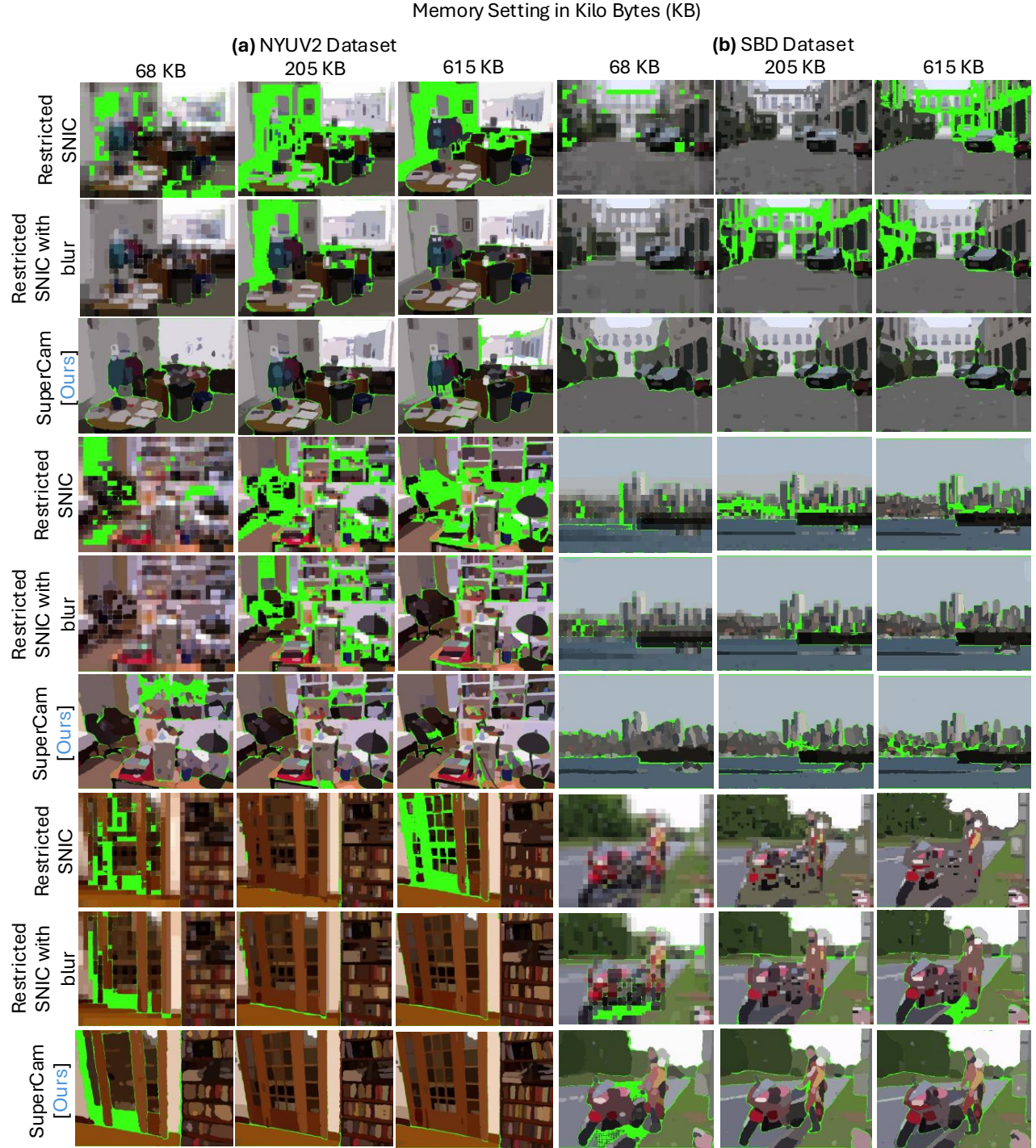
$$mIOUError = \frac{1}{N} \sum_{i=1}^N 1 - \frac{|S_i \cap G(S_i)|}{|S_i \cup G(S_i)|} \quad (S10)$$

where i is the id of the segment in the segmented image, N refers to the total number of segments in the segmented image, S_i represents the segment with the id i and $G(S_i)$ returns the ground truth segment that has maximum overlap with the segment S_i .

We show the mIOU Error plots for both the BSD500 [40] and the [47] datasets in the paper. Suppl. Fig. 5 shows the error plot on the SBD [21] dataset. The trends are similar to what we see in the other two datasets shown in the paper. SuperCam performs consistently better than SNIC and converges to the unsegmented image error as we increase the memory used. We also show more comparisons of SuperCam and SNIC in Suppl. Fig. 6.



Supplementary Figure 5. **Quantitative evaluation of image segmentation.** We compare the quality of image segmentation results produced by the SegmentAnythingV2 model using the mIOU metric on publicly available SBD dataset. Observe that our proposed SuperCam method achieves a lower mIOU mean error than SNIC. The error approaches that of an unsegmented image as the number of superpixels is increased.



Supplementary Figure 6. **Comparison of qualitative Image Segmentation results for memory-restricted SNIC and SuperCam.** Image segmentation results produced using the SAMV2 model on superpixel images for SuperCam, raw SNIC, and SNIC with the optimal blur kernel applied. Images are drawn from the (a) NYUV2 and (b) SBD datasets for different memory settings in kilobytes (KB). SuperCam results look visually better and converge to the ground truth as we increase the memory used.

Supplementary Note 6. Object Detection

We evaluate the object detection performance of YOLOv12 [53] on the COCO [35] dataset with SuperCam and SNIC using the $mAP(50 - 95)$ score. This section provides background on the metric definition and additional qualitative results for object detection. The $mAP(50 - 95)$ score refers to the average of the mean average precision across a range of Intersection over Union (IOU) thresholds in increments of 0.05, starting from 0.50 (which denotes at least a 50% overlap in the detected bounding boxes) to 95% (which denotes at least a 95% overlap in the detected bounding boxes).

The Intersection Over Union (IOU) is defined as:

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} \quad (\text{S11})$$

Precision is defined as:

$$Pre(S, G) = \frac{TP}{TP + FP} \quad (\text{S12})$$

where TP is the number of true positives and FP is the number of false positives. Now $mAP50$ can be denoted as:

$$mAP50 = \frac{1}{NC} \sum_{i=1}^N \sum_{j=1}^C Precision(D_{ij} * \mathbb{1}(IOU(D_{ij}))) \quad (\text{S13})$$

where N is the number of images in the dataset, C is the number of categories in the dataset. D_{ij} refers to the detection of category j in the image i and $\mathbb{1}$ denotes a function that returns 1 if $IOU(D_{ij}) \geq 0.5$, 0 otherwise. Suppl. Fig. 7 shows additional object detection results for the validation partition of the COCO [35] dataset using the YOLOv12 [53] model.



Supplementary Figure 7. **Comparison of memory-constrained SNIC and SuperCam superpixel results for Object Detection.** Results on the YOLOv12 model applied to superpixel images for SuperCam, raw SNIC, and SNIC with the optimal blur kernel applied. Images are drawn from the COCO dataset, and each column shows a different memory setting in kilobytes (KB). SuperCam results look visually better and converge to the ground truth as we increase the memory used.

Supplementary Note 7. Monocular Depth Estimation

We evaluate the performance of the DepthAnythingV2 [69] model on the KITTI [18], DIODE [57], NYUV2 [47] and the Sintel [8] dataset images generated by SuperCam and SNIC. Here we present the error metrics used to evaluate the estimated depth, show additional visual results, and discuss a few failure cases.

We use Absolute Relative Error and Threshold Accuracy to compare the results. Absolute Relative Error (AbsRel) measures how much the predict depth deviates from the ground truth in terms of percentages. Threshold Accuracy (δ_1) measures the percentage of pixels that differ by no more than 25%.

The predicted depth is compared with the ground truth depth available in the KITTI [18], DIODE [57], NYUV2 [47] and the Sintel [8] datasets. The error metrics are defined as follows:

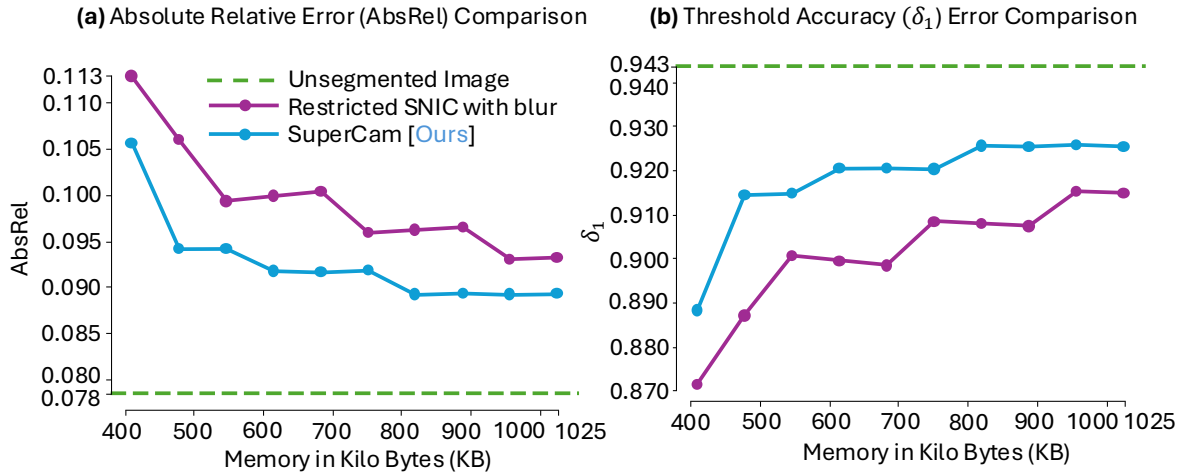
$$AbsRel = \frac{1}{M} \sum_{j=1}^M \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \frac{|d_{ij} - \hat{d}_{ij}|}{d_{ij}} \right) \quad (S14)$$

$$\delta_1 = \frac{1}{M} \sum_{j=1}^M \left(\frac{1}{N_j} \sum_{i=1}^{N_j} \mathbb{1} \left(\max \left(\frac{d_{ij}}{\hat{d}_{ij}}, \frac{\hat{d}_{ij}}{d_{ij}} \right) < 1.25 \right) \right) \quad (S15)$$

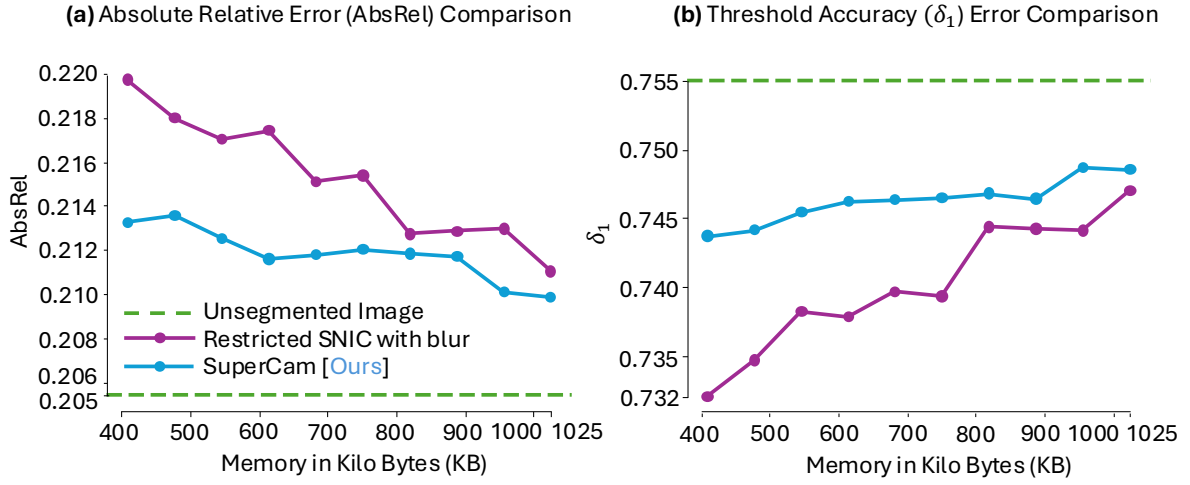
where M denotes the number of images in the dataset, N_j denotes the number of pixels in the image j , d_{ij} denotes the ground truth depth at pixel i and image j and $\mathbb{1}$ denotes the indicator function.

Suppl. Figs. 8, 9, 10 show the $AbsRel$ and δ_1 error metrics for the monocular depth estimates for the DepthAnythingV2 [69] model on the KITTI [18], DIODE [57] and Sintel [8] datasets. SuperCam images give better $AbsRel$ and δ_1 errors than SNIC for the KITTI [18], DIODE [57] and NYUV2 [47] datasets. For the Sintel [8] dataset, SuperCam gives better δ_1 errors and similar $AbsRel$ errors. The mean $AbsRel$ error values plotted for SuperCam are high due to outlier error values produced by a few images. Some SuperCam images have a few pixels that have very high or very low estimated depth values when compared to the ground truth. Fig. 11 shows the median error plots for $AbsRel$ and δ_1 for the Sintel [8] dataset. It can be seen that apart from two data points in the $AbsRel$ plot (for memory settings 205KB and 273KB) all other data points show better results for SuperCam.

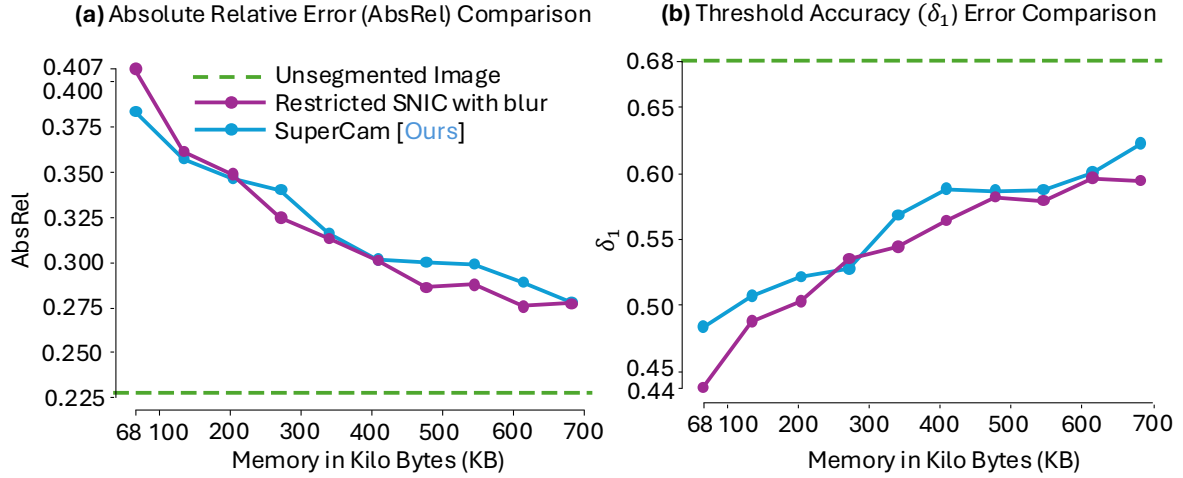
Suppl. Figs. 12, 13 and 14 show depth estimates given by the DepthAnythingV2 [69] model on the DIODE [57], KITTI [18] and Sintel [8] datasets.



Supplementary Figure 8. **Quantitative evaluation of monocular depth estimation on the KITTI dataset.** We compare the (a) Absolute Relative Error (AbsRel) and (b) Threshold Accuracy (δ_1) metrics for depth estimates produced by the DepthAnythingV2 model on the KITTI dataset. Observe that SuperCam error metrics are better when compared to restricted SNIC across all memory settings.

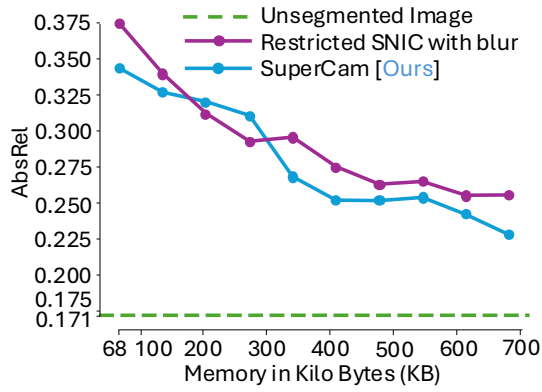


Supplementary Figure 9. **Quantitative evaluation of monocular depth estimation on the DIODE dataset.** We compare the (a) Absolute Relative Error (AbsRel) and (b) Threshold Accuracy (δ_1) error metrics for depth estimates produced by the DepthAnythingV2 model on the DIODE dataset. Observe that SuperCam error metrics are better when compared to restricted SNIC using the same amount of memory.

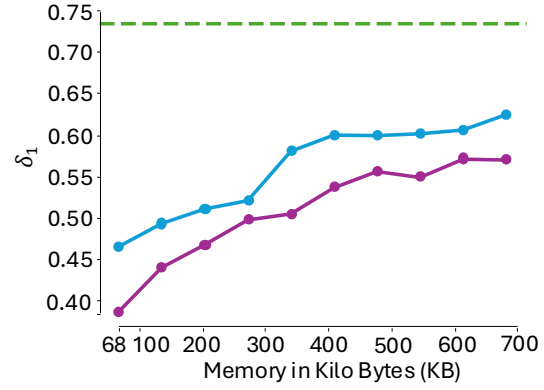


Supplementary Figure 10. **Quantitative evaluation of mean absolute error for monocular depth estimation on the Sintel dataset.** We compare the (a) mean Absolute Relative Error (AbsRel) and (b) Threshold Accuracy (δ_1) error metrics for depth estimates produced by the DepthAnythingV2 model on the Sintel dataset. AbsRel and Threshold Accuracy values are comparable to Restricted SNIC for all memory settings.

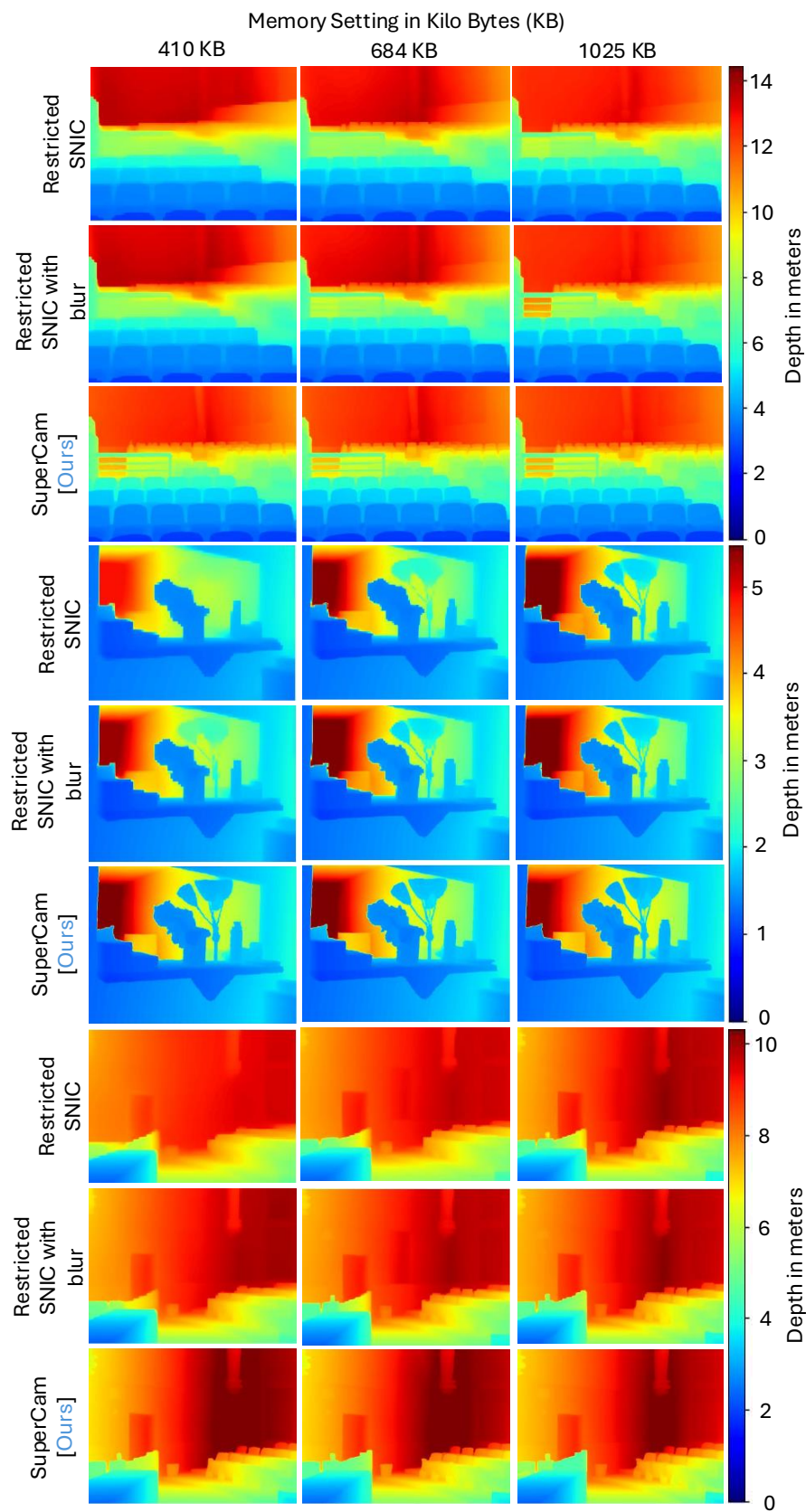
(a) Median Absolute Relative Error (AbsRel) Comparison



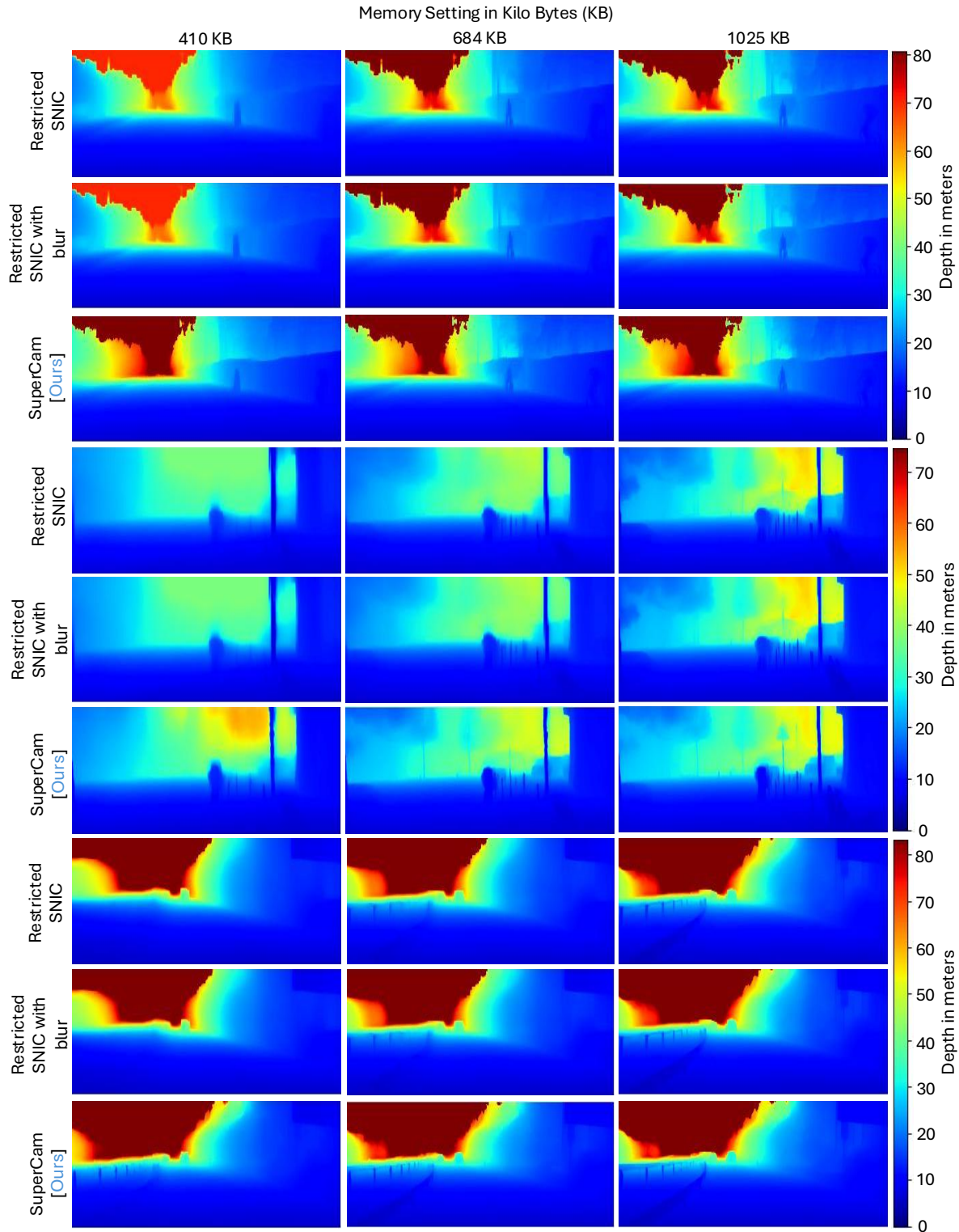
(b) Median Threshold Accuracy (δ_1) Error Comparison



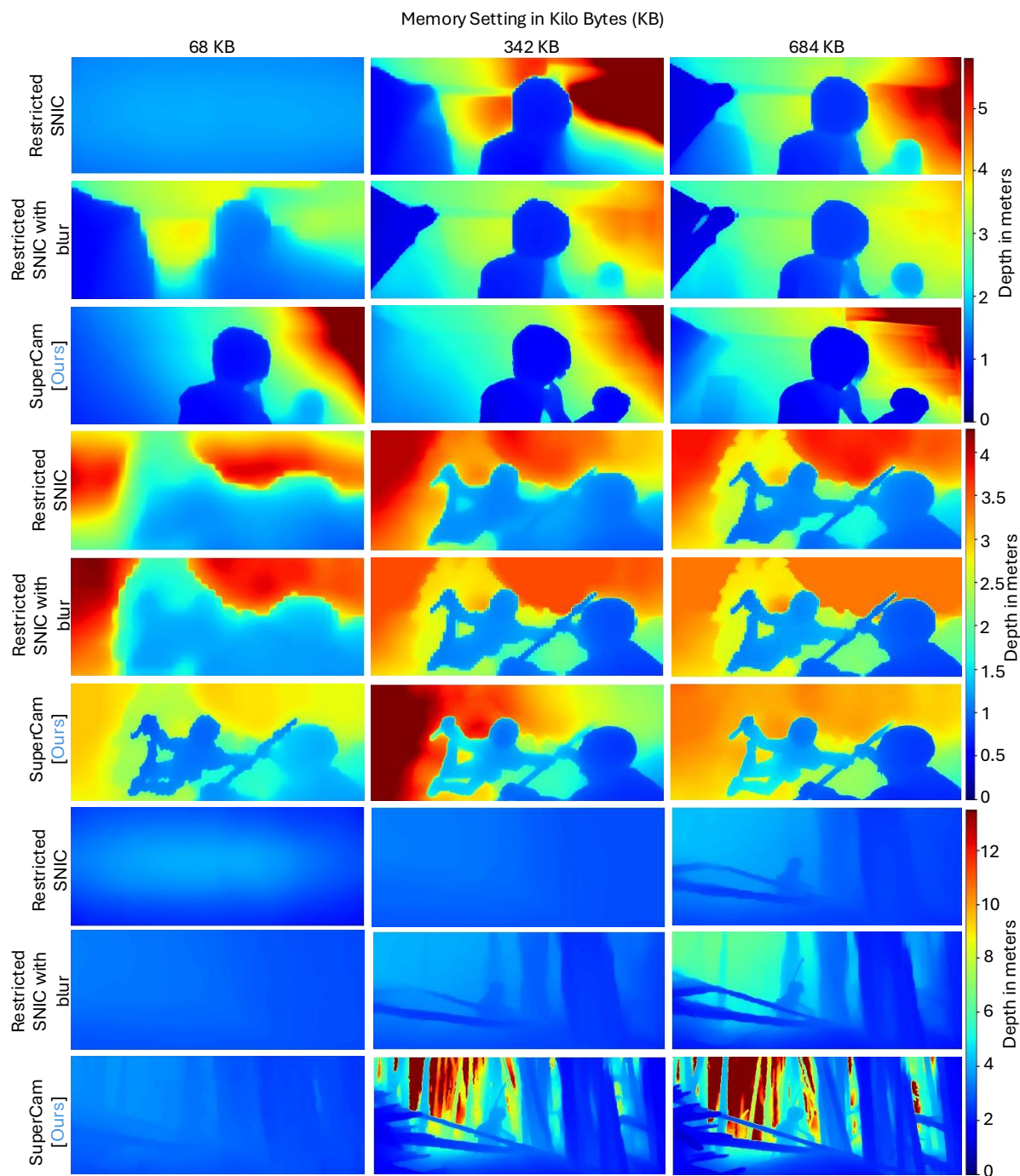
Supplementary Figure 11. **Quantitative evaluation of median absolute error for monocular depth estimation on the Sintel dataset.** We compare the median (a) Absolute Relative Error (AbsRel) and (b) Threshold Accuracy (δ_1) error metrics for depth estimates produced by the DepthAnythingV2 model on the Sintel dataset. AbsRel are comparable to or better than Restricted SNIC for all memory settings. Average Threshold Accuracy values are higher than Restricted SNIC across memory settings, although subsequent figures show that the underlying error distributions are comparable.



Supplementary Figure 12. **Comparison of Monocular Depth Estimation results for the DIODE dataset.** Object Detection results on the DepthAnythingV2 model applied to the superpixel images for SuperCam, Restricted SNIC, and SNIC with the optimal blur kernel applied. Images are drawn from the DIODE dataset and columns show different memory settings in kilobytes (KB). SuperCam results look visually better, particularly for lower memory settings.



Supplementary Figure 13. **Comparison of Monocular Depth Estimation results for the KITTI dataset.** Object Detection results on the DepthAnythingV2 model applied to superpixel images for SuperCam, Restricted SNIC, and SNIC with the optimal blur kernel applied. Images are drawn from the KITTI dataset, and columns show different memory settings in kilobytes (KB). SuperCam results look slightly better visually. The lack of significant difference visually between SuperCam and SNIC can be explained by the fact that the KITTI dataset is very sparse and this affects the conversion of relative depth to absolute depth as there are only a few valid data points to do the conversion.



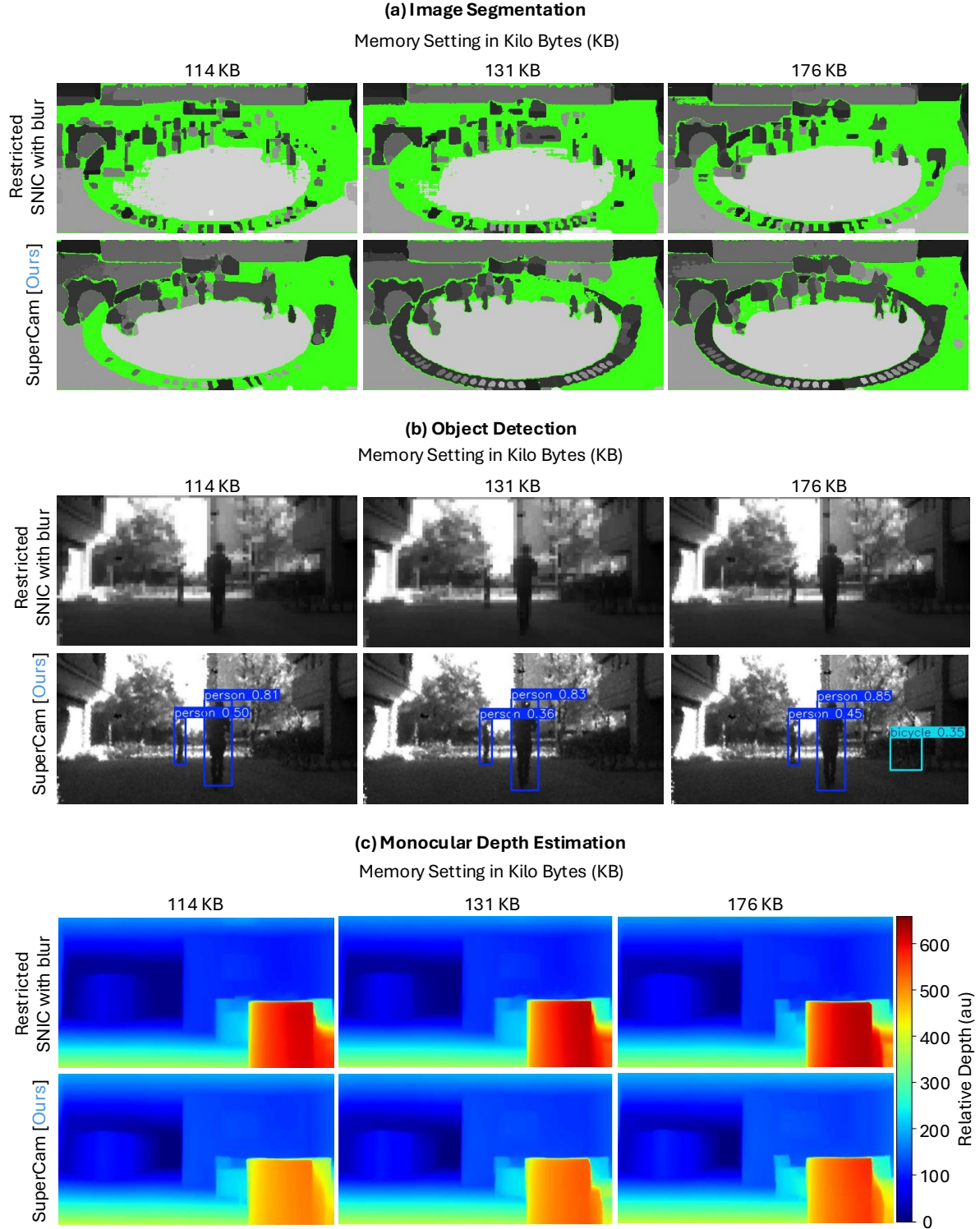
Supplementary Figure 14. **Comparison of Monocular Depth Estimation results for the Sintel dataset.** Object Detection results on the DepthAnythingV2 model on the superpixel images for SuperCam, Restricted SNIC, and SNIC with the optimal blur kernel applied. Images are drawn from the Sintel dataset and columns show different memory settings in kilobytes (KB). SuperCam results look visually better.

Supplementary Note 8. Real Hardware Results

So far we have synthetically simulated the images that SuperCam would capture from a normal RGB image. We simulate the photon streams corresponding to every pixel and generate the SuperCam image from the simulated data. Next we show how a SuperCam would perform when real-world captures are used.

Lastly, we provide results obtained on a publicly available real-world dataset, [39] captured using a SwissSPAD sensor [54]. We compare SuperCam results with memory restricted SNIC on the captured sequences. Suppl. Fig. 15 shows the results obtained using the captured sequences. It can be observed that SuperCam results are visually better when compared to SNIC for image segmentation and object detection. For monocular depth estimation, we show only relative depth as we do not have ground truth for the captured scenes. The depthmaps for SNIC are pixelated and blocky when compared to the SuperCam results.

To summarize all the real hardware results and to show that they have temporal consistency, we also provide a short video sequence (named “supercam_demo_video.mp4”) of all three computer vision tasks; image segmentation, object detection and monocular depth estimation on a captured sequence that is part of the publicly available real-world dataset [39]. The video sequence is captured in high dynamic range lighting conditions, has motion-blur due to movement of the camera, subjects and objects in the scene and are noisy. Nevertheless, SuperCam performance is comparable to the unsegmented image and is visually better than the memory-restricted SNIC results. For image segmentation, the area of frame that is unsegmented is less for SuperCam when compared to SNIC. The segmented regions are also more consistent and similar to the unsegmented image results. For object detection, SuperCam gets more detections than SNIC and misses only the relatively small objects among the objects detected in the unsegmented video. Compared to this the SNIC video performs poorly. For monocular depth estimation, we show the relative depth for the captured scene. The SNIC results are pixelated, especially along the boundaries of objects, when compared to SuperCam. Note that there is some flickering in the image segmentation results for all methods including the unsegmented video. This is caused by the photon noise present in the reconstructed image from the photon cube.



Supplementary Figure 15. **Qualitative comparison of experimental results.** We show experimental results of SuperCam on real world SPAD data from the SWISS SPAD sensor used in the Burst-Vision work. **(a)** Image Segmentation results using SAM2 on experimental SPAD data collected from the SWISS SPAD sensor. The florescent green color refers to regions of the image that were not given any mask by SAM2 model. **(b)** Object Detection results using the YOLOv12 model on experimental SPAD data collected from the SWISS SPAD sensor. **(c)** Monocular Depth Estimation results using the DepthAnythingV2 model on experimental SPAD data collected from the SWISS SPAD sensor. Observe that results the SuperCam results are visually better than the SNIC result that uses the same amount of memory. The results improve with increase in the amount of memory used.