Deep Learning HW2: Grad-CAM and Automated SAM Segmentation

Michał Paradowski

December 11, 2025

1 Grad-CAM Implementation

1.1 Methodology

Grad-CAM[1] uses the gradients of any target concept (e.g., 'circle') flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept.

We implemented a custom **`GradCAM`** class in PyTorch without relying on external interpretability libraries.

1.2 Visual Results

Figure 1 illustrates the Grad-CAM output for selected test samples. The heatmaps successfully localize the geometric shapes, indicating that the classifier is focusing on relevant object features rather than background noise. In some cases (such as the second image), Grad-CAM shows that the classifier focuses on the edge of the shape rather than the middle part.

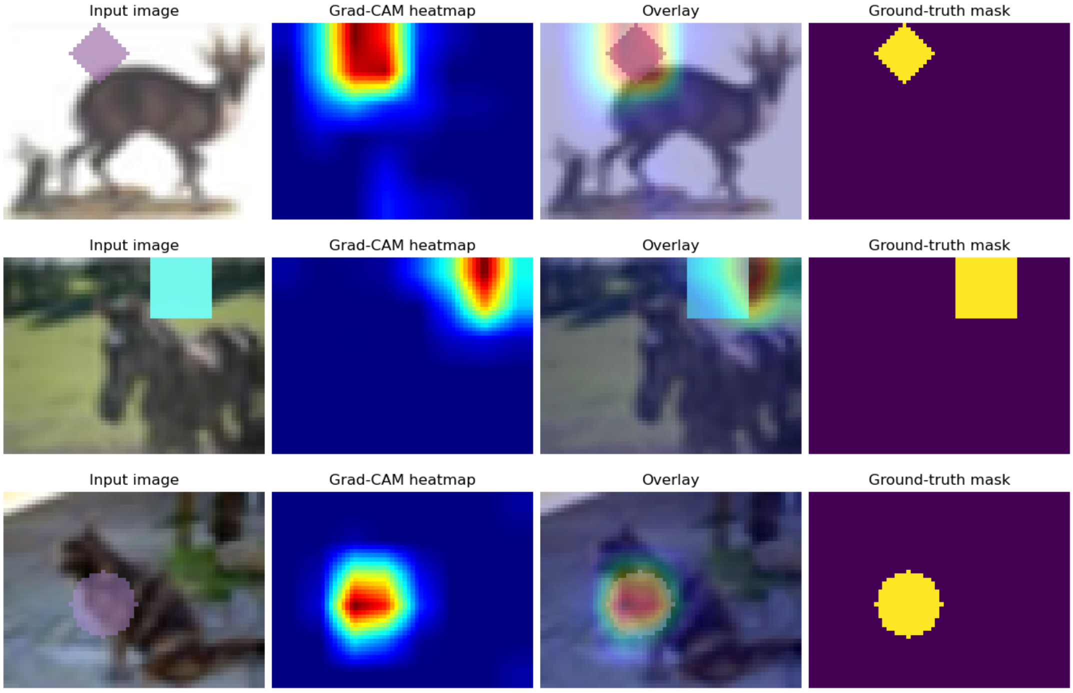

Figure 1: Grad-CAM visualization showing the Input Image, the generated Heatmap, an Overlay, and the Ground Truth Mask.

2 Automated SAM Segmentation

The Segment Anything Model (SAM) requires prompts (points, boxes) to segment objects. We automated this prompting process using the localization priors from our Grad-CAM implementation. We compared two approaches:

2.1 Pipeline 1: Foreground Points Only

(`SamForegroundPipeline`) relies on positive cues from Grad-CAM heatmaps:

- **Point Selection:** We identify the peak activation in the heatmap and calculate the center of mass for regions exceeding 80% of this maximum value.
- **Augmentation:** We generate a 3×3 grid of foreground points centered on this location (the center point plus its 8 neighbors with an offset of 1 pixel).

2.2 Pipeline 2: Foreground and Background Points

(`SamBackgroundPipeline`) adds negative constraints to refine the segmentation boundary.

- **Foreground:** Identical to Pipeline 1.
- **Background:** We identify regions with very low activation (heatmap values < 0.1). From these "cold" regions, we select 4 spatially distinct points (top-most, bottom-most, left-most, right-most) to explicitly signal background areas to SAM.

3 Results and Discussion

We evaluated both pipelines on a subset of the test dataset using three metrics:

- **Hit Rate:** The proportion of generated foreground/background points that fall within the correct area.
- **Distance:** The average Euclidean distance from the generated foreground points to the center of mass of the ground-truth mask.
- **IoU (Intersection over Union):** The overlap between the predicted segmentation mask and the ground truth.

3.1 Quantitative Results

The performance of the two pipelines is summarized in Table 1.

Table 1: Performance comparison of automated SAM Pipelines

Pipeline	**Fg Hit Rate**	**Bg Hit Rate**	**Distance (px)**	**mIoU**
Pipeline 1 (FG only)	0.815	-	3.848	0.860
Pipeline 2 (FG + BG)	0.815	1.000	3.848	0.831

3.2 Discussion

Both pipelines achieved excellent segmentation performance, significantly exceeding the target IoU of 65%.

- **Pipeline 1 (Foreground Only):** Achieved the highest mIoU of **0.860**. This suggests that for distinct geometric shapes on relatively clean backgrounds, identifying the core of the object is sufficient for SAM to infer the correct boundary. The "Hit Rate" of 0.815 indicates that our Grad-CAM derived points reliably fall within the object boundaries.
- **Pipeline 2 (FG + BG):** Achieved a slightly lower mIoU of **0.831**. While intuition suggests that background points should help constrain the mask, they may have introduced ambiguity if the low-activation regions were too close to the object boundary, or if SAM's internal biases for this specific dataset favored unconstrained expansion from a strong center seed.

Potential Improvements: The main issue was that no matter the background points, Pipeline 2 worked worse than Pipeline 1, even tho the only difference was the addition of background points. The proposed improvement would be to introduce background points, only if there were any leakages detected based on Grad-CAM after the initial segmentation with foreground points. The segmentation could then be recomputed. This way, background points would only be added when necessary, potentially improving the segmentation without introducing unnecessary noise.

References

[1] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization.* Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[2] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolge, C., ... & Girshick, R. (2023). *Segment Anything.* Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).