ML HW1

zhenqun shao

September 2025

1 Multi-Scale Detector Analysis

1.1 Background

This project uses a synthetic object-detection dataset containing circle / square / triangle to evaluate how a multi-scale, anchor-based detector adapts to and specializes for objects of different sizes. Under a unified evaluation pipeline, I analyze how anchor scale choices, scale specialization, and qualitative visualizations together affect overall performance (mAP@0.5). The motivation is threefold: (1) real-world detection tasks typically contain objects of widely varying sizes, and a single scale struggles to balance recall and localization accuracy; (2) the size alignment between anchors and ground-truth objects directly determines the number of positive matches and the strength of training gradients; and (3) common industry approaches (e.g., FPN/YOLO) rely on multi-scale heads, but the exact division of labor among scales—and where performance bottlenecks arise—must be clarified through empirical analysis.

1.1.1 Model

I implemented a lightweight three-scale detector, **MultiScaleDetector()**. The input is fixed at 224×224, and the backbone consists of four convolutional blocks with stride=2 for progressive downsampling, producing three feature maps for detection at different scales.

- Block1: Conv(3→32, s=1)→BN→ReLU → Conv(32→64, s=2)→BN→ReLU → 112×112
- Block2: Conv(64→128, s=2)→BN→ReLU → 56×56(Scale-1)
- Block3: Conv(128→256, s=2)→BN→ReLU → 28×28(Scale-2)
- Block4: Conv(256→512, s=2)→BN→ReLU → 14×14(Scale-3)

At each scale, a detection head is attached: 3×3 Conv + BN + ReLU → 1×1 Conv, producing an output with $A * (5 + C)$ channels, where A=3 is the number of anchors per spatial location and C=3 is the number of classes. The term 5+C includes four regression offsets *(tx, ty, tw, th)*, one objectness score, and C class scores. Weights are initialized using Kaiming initialization, with BN weights set to 1 and biases set to 0. The model's forward pass returns three tensors:

Tensors	Size 2	Usage
P1	[B, A*(5+C), 56, 56]	S1, small objects, circle
P2	[B, A*(5+C), 28, 28]	S2, middle objects, square
P3	[B, A*(5+C), 14, 14]	S3, large objects, triangel

During inference, the decode_predictions function is used to transform the predicted offsets (tx, ty, tw, th) together with the pre-generated anchors into bounding boxes in corner format (x1, y1, x2, y2). The detection score is computed as 1:

$$score = \sigma(objectness) \times softmax(classscores), \tag{1}$$

where σ is the sigmoid activation. After this step, per-class Non-Maximum Suppression (NMS) is applied, with a default IoU threshold of 0.6 and a pre-filter confidence threshold of 0.5 (both adjustable). Finally, the remaining boxes are evaluated to compute Average Precision (AP). The Anchor configuration is :

Scale	Size
S1(56× 56)	[16, 24, 32]
S2(28× 28)	[48, 64, 96]
S3(14× 14)	[96, 128, 192]

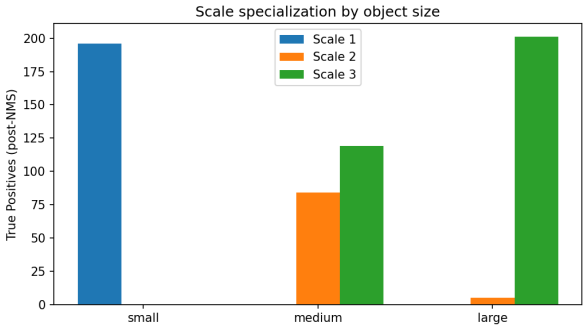

Figure 1: scale performance

1.1.2 evaluation

The evaluation script implements a VOC-style interpolated Average Precision (AP): predictions of the same class are sorted in descending order of confidence scores and matched to ground-truth boxes (a match is counted as a True Positive if IoU ≥ 0.5 and the GT has not already been assigned). From this, a precision–recall (P–R) curve is accumulated, and the AP is obtained by integrating the precision envelope.
In addition, the script outputs scale specialization statistics: for each True Positive after NMS, it records which detection head (S1/S2/S3) produced the prediction and which ground-truth size bucket (small/medium/large, approximated by $\sqrt{area}$) it belongs to. This provides objective evidence of "which scale is mainly responsible for which object size."
The script also saves detection visualizations and anchor coverage plots, which serve as qualitative tools to further support the analysis.

1.2 Result

Load results/best_model.pth to evaluate :
AP@0.5

- circle : 0.6611
- square : 0.3449
- triangle : 035627

mAP@0.5 : 0.5229

Scale	small	medium	large
Scale-1	196	0	0
Scale-2	0	84	5
Scale-3	0	119	201

1.3 Result Analysis

1.3.1 Scale Specialize

All 196 True Positives (TPs) for **small** objects come exclusively from **Scale-1**, while Scales-2 and -3 contribute none. This indicates that the small-scale detection head is highly aligned with small objects, which directly explains the strong performance of the circle class (AP = 0.66). The anchors of Scale-1 [16,24,32] are relatively well matched to the actual size of small circles in the dataset, providing sufficient positive samples. As a result, both objectness and regression receive stable gradients, leading to the high AP for circles.
For **medium objects**, there are 203 TPs in total, with **Scale-3** covering 58.6% (119 TPs) and Scale-2 covering 41.4% (84 TPs). The larger receptive field of Scale-3 enables stronger coverage of medium-sized objects, while Scale-2 plays a supportive role.
For **large objects**, among 206 TPs, **Scale-3** accounts for 97.6% (201), while Scale-2 contributes only 5

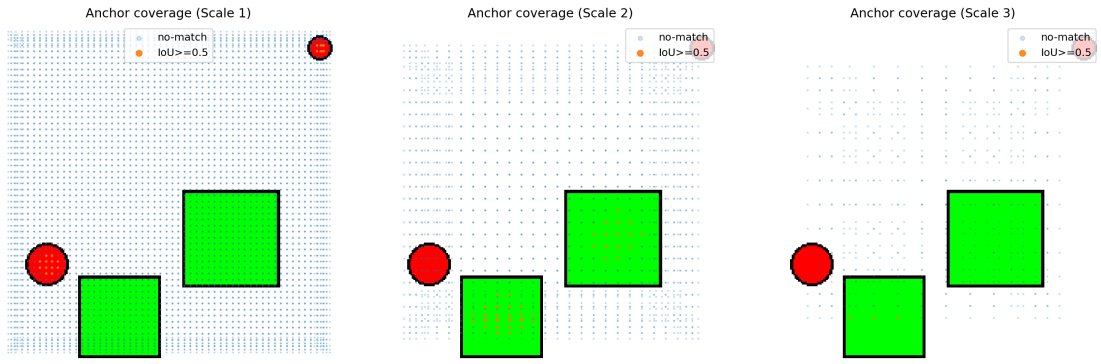

Figure 2: Anchor Scale

and Scale-1 none. Scale-2 and Scale-3 have overlapping anchor ranges, which allows Scale-3 to capture part of the medium objects. Consequently, Scale-2 ends up with fewer and lower-quality positive samples, weakening its classification and localization ability and resulting in the low AP for the square class. Scale-3, on the other hand, almost monopolizes large objects, consistent with its low-resolution, large receptive field characteristics.

Overall, the model shows an ideal division of labor: S1 $\rightarrow$ small, S2 $\rightarrow$ medium, S3 $\rightarrow$ large. However, medium objects are still disproportionately captured by Scale-3, leaving Scale-2 less impactful, which correlates with the relatively low AP of the square class (0.345).

Visualizations further confirm this [2]: Scale-1 produces dense, high-scoring candidates around small circles; Scale-3 forms smooth, broad responses on large triangles; while Scale-2 shows peaks at the edges and centers of squares, but in a more sparse and less stable manner.

1.4 Conclusion

The model structure is simple yet effective, with the three detection heads clearly exhibiting scale specialization: S1 focuses on small objects, S3 handles large ones, and S2 assists with medium objects. The current overall performance is mAP@0.5 = 0.5229. The strength lies in small-object detection (circle 0.66), while the weakness is in medium objects (square 0.345), and large-object performance falls in between (triangle 0.563). The main performance bottleneck arises from the unclear boundary between the responsibilities of S2 and S3, which leads to insufficient positive samples and weak feature learning for S2, while S3 carries too heavy a burden. Future improvements can be achieved by narrowing S2's anchors to be more "purely medium" (e.g., [48,56,72] or [56,64,80]), and slightly shifting S3 upward (e.g., [112,144,192/208]) to reduce overlap between the two. In addition, moderately increasing S3's confidence threshold or slightly lowering the NMS IoU threshold could help reduce residual detections of medium objects by S3. On the classification side, applying Focal Loss with a light alpha weighting for the square class may further improve balance.

Overall, this experiment confirms the effectiveness of the multi-scale detector in size specialization and highlights medium-sized objects as the main area for improvement. With more refined anchor partitioning and improved positive-sample assignment strategies, the AP for squares can be significantly boosted, thereby further improving the overall mAP.

2 Heatmap vs Direct Regression for Keypoint Detection

2.1 Background

Keypoint detection is a fundamental task in computer vision with applications. The goal is to localize a predefined set of landmarks within an image as accurately as possible. In this problem, I consider a simplified setting using synthetic "stick figure" images, each annotated with five keypoints: head, left hand, right hand, left foot, and right foot.

Two major approaches exist for keypoint localization. First, Spatial heatmap regression. Each keypoint is represented by a Gaussian distribution over a 2D spatial grid. The model predicts per-keypoint heatmaps, and the keypoint coordinates are extracted as the location of maximum activation. Second, Direct coordinate regression. Instead of producing heatmaps, the network directly outputs normalized

(x,y) coordinates for each keypoint. This approach is more compact and avoids the need for heatmap generation, but is generally harder to optimize since the network must learn precise regression without intermediate spatial supervision. Each method has its own pros and cons. Heatmap regression provides spatially interpretable outputs but is more computationally expensive, whereas direct regression is lightweight but less interpretable. Heatmaps allow dense supervision signals across pixels, which may yield higher accuracy, while regression only provides sparse point-level supervision. Understanding when heatmaps significantly outperform regression can guide practical system design, especially in resource-constrained or real-time applications. To systematically evaluate these approaches, I adopt a shared-encoder, two-head design. The backbone extracts multi-scale features from the 128×128 grayscale input, and then branches into: (i) HeatmapNet, an encoder–decoder head with skip connections that predicts per-keypoint heatmaps at 64×64 resolution; and (ii) RegressionNet, a lightweight head that applies global average pooling followed by fully connected layers to regress normalized (x,y) coordinates directly. Training and data are strictly controlled and shared between both models, ensuring a fair comparison of accuracy, robustness, and efficiency under identical conditions.

2.2 Result

2.2.1 Evaluation protocol (PCK) and tooling

I evaluate with PCK (Percentage of Correct Keypoints)[1] at thresholds [0.05,0.10,0.15,0.20], normalizing distances by the bounding-box diagonal (default) or the torso length as an alternative. The full evaluation is implemented in evaluate.py and consists of three steps: Keypoint extraction from heatmaps: For each predicted heatmap [B,K,H,W], I flatten to [H×W] and take the argmax to obtain the peak location. The $(\hat{x}, \hat{y})$ indices are then converted back to image coordinates, scaling from heatmap size (W,H) to the input image size (128×128); PCK computation: Let $d_{i,j} = ||\hat{P}_{i,j} - P_{i,j}||$ be the Euclidean error for sample i and keypoint j. I normalize by either the bbox diagonal or the torso proxy; the function computes both robustly (with fallbacks if a torso is degenerate). For each threshold τ, report

$$PCK(\tau) = \frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K} (\frac{d_{i,j}}{norm_i} \leq \tau) \tag{2}$$

The function returns a dict {threshold: accuracy} so the same driver can compare multiple models; Visualization & curves: draws Heatmap vs Regression PCK curves on the same axes 3. Function visualize_predictions() overlays GT (green circles) and predictions (red crosses) on the grayscale image, also drawing dashed lines between them to indicate the per-keypoint error vector. Visualized 10 sample predictions for each method, Figure 4 shows heatmap(left) and regression(right) prediction.

In my experimental results, the Heatmap method consistently outperforms Regression across all PCK thresholds (see Figure 3). For example, at the strictest threshold $\tau = 0.05$, Heatmap already achieves approximately 0.84 PCK, while Regression lags far behind at only 0.08, indicating that direct coordinate regression is highly unstable for pixel-level precision. As the threshold relaxes to $\tau = 0.20$, the Heatmap curve further rises to ≈ 0.92, maintaining robust accuracy, whereas Regression improves but plateaus around 0.53, leaving a substantial gap.

This discrepancy can be explained by the nature of the two approaches: HeatmapNet preserves spatial structure through convolutional feature maps and predicts dense distributions aligned with the image grid, while RegressionNet compresses features into a global vector before directly regressing coordinates, lacking spatial constraints and therefore more prone to drift. The qualitative visualizations in Figure4 corroborate this: Heatmap predictions (left, red crosses) cluster tightly around the green GT circles with only minor offsets, whereas Regression (right) often exhibits systematic shifts, especially at extremities like hands and feet.

Overall, these results confirm that the heatmap-based approach provides superior localization robustness and accuracy, while regression—despite its computational simplicity—fails to generalize reliably under complex poses, becoming the primary bottleneck.

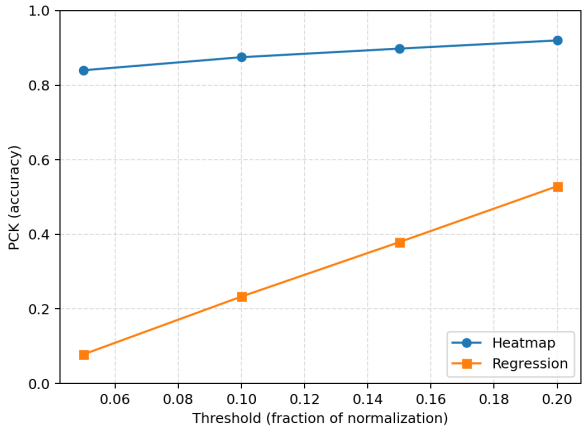

Figure 3: PCK Comparison

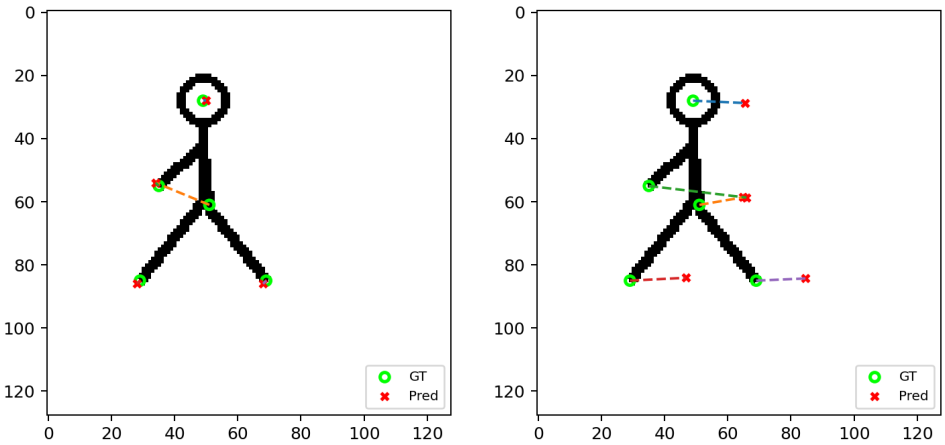

Figure 4: Prediction Comparison

2.2.2 Ablation

My ablation suite in baseline.py isolates the key design factors of the heatmap approach and quantifies their impact using validation MSE on predicted heatmaps. Varying the heatmap resolution (32, 64, 128) shows that while the 64×64 case unexpectedly performs worse than 32×32, increasing to 128×128 yields a sharp drop in error (Figure 5), confirming that higher spatial resolution can substantially improve localization at the cost of greater compute and memory. Interestingly, the 64×64 heatmap resolution performs worse than the 32×32 case, which at first glance seems counter-intuitive since higher resolution should provide finer spatial supervision. I hypothesize several factors behind this anomaly. First, intermediate resolutions may suffer from an unfavorable trade-off: the supervision is not as coarse and stable as 32×32, yet not fine-grained enough to fully exploit the added spatial detail, leading to unstable convergence. Second, the stochasticity of training (initialization, data sampling) can amplify this instability at mid-scale resolutions, causing the network to overfit noise while failing to capture consistent keypoint structures. Third, lower-resolution targets implicitly act as a form of label smoothing—making optimization easier and gradients more stable—whereas mid-resolution targets may lose this regularization effect without gaining the full precision benefit of 128×128. Together, these factors suggest that performance is not a monotonic function of resolution; instead, it reflects a complex interaction between signal sharpness, optimization stability, and model capacity.

The Gaussian sigma sweep ($\sigma \in 1.0, 2.0, 3.0, 4.0$) reveals that very small $\sigma \approx 1.0$ achieves the lowest error, whereas moderate values around 2–3 actually increase MSE, and larger $\sigma = 4.0$ partially recovers (Figure 6 left). This indicates that supervision sharpness has a non-monotonic effect, with overly broad Gaussians degrading precision. Finally, removing skip connections in the decoder leads to consistently

higher error compared to the standard U-Net-style architecture (Figure 6 right), confirming that low-level spatial detail is critical for accurate keypoint heatmaps. All ablations share the same lightweight training loop with Adam ($lr = 10^{-3}$) and fixed epochs, ensuring the observed differences stem from the tested factor rather than confounds.

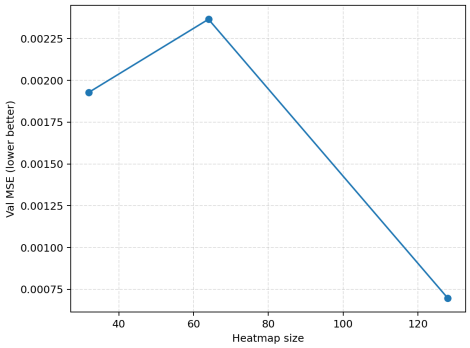

Figure 5: ablation heatmap resolution

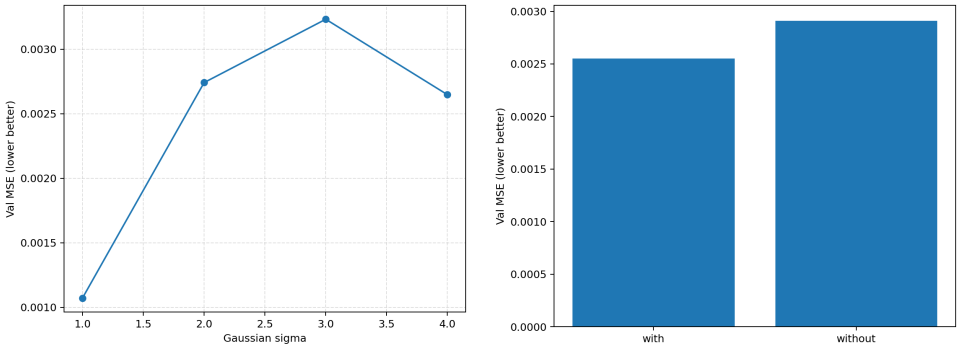

Figure 6: ablation sigma and skip

2.2.3 Failure analysis

In the analysis of failure cases, I divide predictions into three categories: type A denotes instances where the heatmap method succeeds but regression fails, type B where regression succeeds but the heatmap fails, and type C where both methods fail. In practice, there are only type A Figure7 and type C Figure8 cases, with type B absent. This outcome is consistent with the overall results, where the heatmap approach consistently outperforms regression across all thresholds, leaving little room for regression to succeed uniquely. A closer inspection reveals that both type A and type C failures predominantly occur in poses where the two hands are close together or pressed against the torso, creating regions of high similarity and ambiguity that are inherently difficult to resolve. In such situations, the heatmap method still demonstrates strong robustness: because it produces dense spatial distributions rather than single coordinates, it can form multiple candidate peaks on the image grid and leverage local competition as well as contextual body-structure cues to select the most plausible joint location. As a result, it often delivers correct predictions, while regression, lacking spatial constraints, tends to drift or confuse left versus right hands or hands versus torso, thereby producing type A cases where only the heatmap succeeds. However, when the two hands completely overlap or are heavily occluded by the torso, even the heatmap may generate nearly equal competing peaks or suffer from resolution limits and annotation noise, leading both methods to fail and thus creating type C cases. Taken together, the distribution of failure cases further confirms the superiority of the heatmap approach: not only does it significantly outperform regression in standard scenarios, but it also maintains much greater reliability in complex and ambiguous poses.

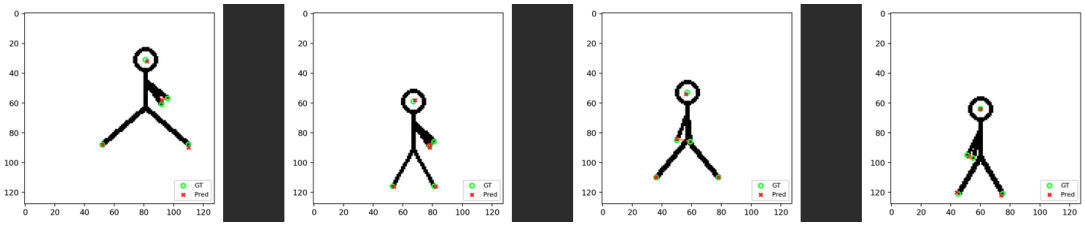

Figure 7: typeA

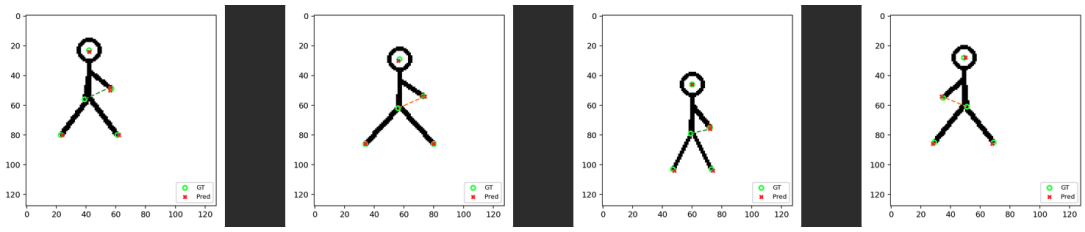

Figure 8: typeC

2.3 Conclusion

Through controlled experiments on synthetic stick-figure data, we compared heatmap-based and direct regression approaches for keypoint detection under identical training and evaluation settings. Our results consistently demonstrate that heatmap regression provides substantially higher localization accuracy and robustness, achieving up to 0.84 PCK at the strictest threshold while direct coordinate regression lags far behind. Ablation studies further reveal that spatial resolution, Gaussian sharpness, and skip connections are critical factors for effective heatmap supervision, with higher-resolution and sharper targets enabling more precise predictions. Failure case analysis shows that the most challenging poses—such as hands close together or pressed against the torso—still expose limitations for both methods, yet heatmaps remain significantly more reliable by leveraging dense spatial distributions and contextual cues. Overall, these findings confirm that heatmap-based formulations, despite their greater computational cost, deliver markedly superior accuracy and robustness, and should be favored in scenarios where precise localization is critical, while direct regression may only be viable in resource-constrained settings where efficiency outweighs accuracy.

References

[1] L. Pishchulin, E. Insafutdinov. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. *arXiv preprint arXiv:1511.06645*, 2016.