EE641 #HW1

#Q1:
Analysis of the Multi-Scale Detector

This report analyzes the key architectural features of the multi-scale, single-shot detector. We examine how the model uses a feature pyramid to handle objects of varying sizes, the critical role of anchor scales, and the nature of the learned visual features at each scale.

How Different Scales Specialize for Different Object Sizes
The model's architecture is explicitly designed to specialize its different feature maps for detecting objects of different sizes. By extracting features at three scales (56x56, 28x28, and 14x14), the network can efficiently handle the small circles, medium squares, and large triangles in the dataset.

This specialization occurs for two primary reasons:

1. Receptive Field Size: Deeper layers in the network produce lower-resolution feature maps (like 14x14). Neurons in these layers have a larger receptive field, meaning they "see" a wider area of the original input image. This makes them naturally suited for identifying large objects like triangles. Conversely, earlier, high-resolution layers (like 56x56) have smaller receptive fields, making them ideal for localizing small objects like circles.
2. Anchor Matching: During training, an object is assigned as a positive training example only to anchors that it has a high Intersection over Union (IoU) with. Since the small anchors ([16, 24, 32]) were specifically paired with the 56x56 feature map, only the small circles achieved high IoU with them. This process forces the detection head at this scale to learn features exclusively for detecting small circles. The same logic applies to the medium and large scales.

The plot generated by the evaluation script clearly validates this behavior. It shows that Scale 1 is almost exclusively responsible for detecting circles, Scale 2 for squares, and Scale 3 for triangles.

The Effect of Anchor Scales on Detection Performance
The configuration of anchor scales is one of the most critical factors for the detector's success. Anchors act as predefined "guesses" or priors that the model refines. The performance of the detector is highly sensitive to how well these priors match the actual objects in the dataset.

The key effects are:

- Ensuring High IoU: The chosen scales—[16, 24, 32] for small, [48, 64, 96] for medium, and [96, 128, 192] for large—were selected to closely match the sizes of the dataset's objects. This ensures that for every ground-truth object, there are anchors with a high initial IoU. A high IoU is required to create a positive sample for training the localization

and classification heads. If the anchors were poorly sized (e.g., all were small), the model would rarely find good matches for the large triangles, effectively starving it of the necessary training data.
- Simplifying the Regression Task: Object detection involves regressing the offsets from an anchor box to a ground-truth box (tx,ty,tw,th). When an anchor is already a good fit, these offsets are small and stable. If an anchor is a poor fit, the model must learn to predict very large offsets, which is a much harder task that can lead to unstable training and inaccurate localization. By providing well-fitting anchor priors, we make the learning task significantly easier for the network.

The anchor coverage visualization shows how the anchors from each scale are distributed across the image, with their sizes tailored for a specific object class.

Visualization of the Learned Features at Each Scale
While the evaluation script does not directly visualize the feature map activations, we can infer their properties from the model's specialized performance. To visualize them, one would use hooks in PyTorch to capture the output feature maps from each of the three scales for a given input image. By averaging the activations across the channel dimension, we can create a heatmap showing which regions of the image the network is "looking at."

Based on the model's behavior, we would expect to see the following:

- Scale 1 (56x56 Map): For an input image containing small circles, the activation heatmap for this scale would show high-intensity "hot spots" spatially aligned with the circles. The features learned by these early layers would likely correspond to simple, low-level patterns like curves and small, contained shapes.
- Scale 2 (28x28 Map): This heatmap would show strong activations in the locations of the medium squares. The features learned would be more complex, likely responding to patterns like right-angled corners and straight edges of a particular length.
- Scale 3 (14x14 Map): The heatmap for the deepest feature map would activate on the large triangles. These features are the most abstract, and their large receptive fields allow them to recognize the overall gestalt of a large object composed of multiple edges and corners.

In summary, these visualizations would confirm that the network learns a hierarchy of features that are not only progressively more complex but also specifically tuned to the scale of the objects they are designed to detect.

#Q2:

Comparative Analysis of Heatmap vs. Direct Regression for Keypoint Detection

1. Introduction

This report details the implementation and comparative analysis of two prominent deep learning methods for 2D keypoint detection: **Spatial Heatmap Regression** and **Direct Coordinate Regression**. The objective is to quantify the performance difference between these approaches on a controlled, synthetic "stick figure" dataset. Both models were trained for 30 epochs to identify the 5 keypoints of a figure in 128x128 grayscale images. Performance was evaluated using the Percentage of Correct Keypoints (PCK) metric.

2. Methodology

2.1 Model Architectures

Two models were implemented, sharing a common convolutional encoder:

1. **HeatmapNet**: An encoder-decoder architecture with skip connections (U-Net style). It processes an image and outputs 5 distinct 64x64 heatmaps, where each map represents the probability distribution for a single keypoint. The final coordinate is derived from the location of the maximum value in each heatmap.
2. **RegressionNet**: This model uses the same encoder but replaces the decoder with a regression head. The head consists of a Global Average Pooling layer followed by fully-connected layers, which directly output a vector of 10 normalized (x, y) coordinates.

2.2 Training

Both models were trained for 30 epochs using the Adam optimizer (lr=0.001) and Mean Squared Error (MSE) loss. The training logs show consistent convergence for both models, although HeatmapNet achieved a significantly lower validation loss (0.000143) compared to RegressionNet (0.007310), indicating a much better fit to the data.

2.3 Evaluation Metric

The primary metric for comparison is the **Percentage of Correct Keypoints (PCK)**. A keypoint is deemed "correct" if the Euclidean distance between the predicted and ground-truth coordinate is within a specified threshold. This threshold is normalized by the diagonal of the ground-truth keypoints' bounding box to ensure scale invariance.

3. Results and Discussion

The experimental results demonstrate a decisive performance advantage for the heatmap regression method.

3.1 Quantitative Analysis (PCK)

The PCK scores, calculated on the unseen test set, reveal a stark contrast in performance.

Threshold	HeatmapNet Accuracy	RegressionNet Accuracy
0.05	**98.9%**	3.9%
0.10	**100.0%**	14.6%
0.15	**100.0%**	29.1%
0.20	**100.0%**	43.4%

The HeatmapNet achieves near-perfect localization, with 98.9% of keypoints being accurate even at the strictest threshold. In contrast, RegressionNet struggles significantly, with its accuracy failing to surpass 44% even at the most lenient threshold. The PCK curve below visually captures this performance gap.

3.2 Analysis: Why the Heatmap Approach is Superior

The dramatic difference in performance can be attributed to several architectural and conceptual advantages of the heatmap approach:

1. **Preservation of Spatial Information**: HeatmapNet's encoder-decoder structure maintains a 2D spatial representation throughout the network. Convolutions are naturally suited to this, allowing the model to learn *where* features are. In contrast, RegressionNet collapses all spatial information into a 1D vector after the encoder, forcing the final layers to learn a difficult and less intuitive mapping from abstract features to precise coordinates.
2. **A More Effective Learning Target**: The MSE loss on a heatmap provides a rich, spatially-aware gradient for every pixel in the output map. This gives the model direct feedback on how to adjust its weights to move the heatmap peak. For direct regression, the loss provides a sparse gradient that is less informative about the spatial relationship between the image features and the output coordinates.
3. **Implicit Uncertainty Modeling**: A heatmap can naturally represent uncertainty. A broad, dim peak can signify low confidence, while a sharp, bright peak shows high confidence. Direct regression outputs a single coordinate with no such built-in confidence measure.

3.3 Visualization of Learned Heatmaps

The training progression of HeatmapNet shows the model learning to refine its predictions from diffuse blobs into sharp, accurate Gaussian peaks, demonstrating its effective localization capability.

3.4 Failure Case Analysis

Qualitative analysis confirms the quantitative results. The baseline.py script easily identified numerous cases where HeatmapNet was successful while RegressionNet failed. A common failure mode for RegressionNet is predicting a biologically implausible "average pose" or collapsing keypoints, whereas HeatmapNet remains robust.

4. Ablation Study Results (Hypothetical)

While full ablation experiments were not run, we can hypothesize the following outcomes based on common practice:

- **Effect of Heatmap Resolution**: Increasing the heatmap resolution (e.g., to 128x128) would likely provide a marginal increase in accuracy by reducing quantization error, but at a higher computational cost. Decreasing it (e.g., to 32x32) would likely degrade performance.
- **Effect of Gaussian Sigma**: The sigma used to generate ground-truth heatmaps is crucial. A very small sigma makes the target too sparse and difficult for the model to learn. A very large sigma makes the localization imprecise. The value of sigma=2.0 used in this experiment appears effective, likely falling within an optimal range.

5. Conclusion

This investigation decisively concludes that for keypoint detection tasks, the **spatial heatmap regression approach is fundamentally superior to direct coordinate regression**. The preservation of spatial information, combined with a more effective learning target, allowed HeatmapNet to achieve near-perfect results. In contrast, RegressionNet's architectural bottleneck of collapsing spatial features led to poor performance. This study underscores the importance of choosing a network architecture that aligns with the spatial nature of the problem.