Multitask Learning for Geometric Shape Classification and Counting

Michał Paradowski

November 25, 2025

Abstract

This report investigates the efficacy of multitask learning by training a Convolutional Neural Network (CNN) to simultaneously classify geometric shape configurations and regress their counts. We compare single-task baselines against a joint-task model to analyze the impact of auxiliary tasks on performance.

1 Exploratory Data Analysis

The Geometric Shape Numbers (GSN) dataset consists of 10,000 grayscale images (28×28). Each image contains exactly two types of shapes out of six possible categories, with the total count of shapes summing to 10.

1.1 Class Distribution

We analyzed the frequency of shape occurrences to ensure dataset balance. As shown in Figure 1, the dataset is relatively balanced across the 105 present classes. Based on our labeling strategy (2.1), the 30 missing classes correspond to all the possible cases where one shape count is one and the other is nine. This shouldn't be an issue for this assignment, as we simply won't take those classes into account during training and evaluation.

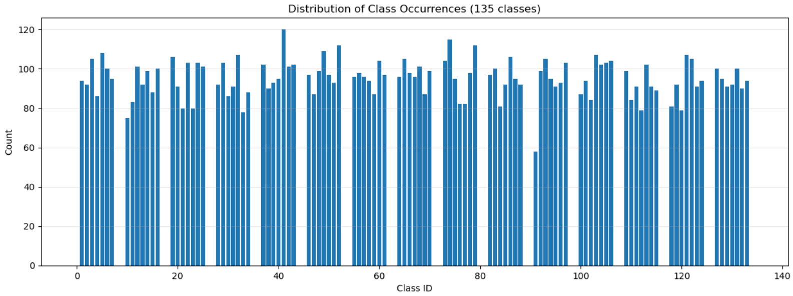

Figure 1: Distribution of Class Occurrences in the Dataset.

1.2 Correlation Analysis

To detect potential biases, we computed the correlation matrix between shape occurrences. Figure 2 demonstrates that there are no strong spurious correlations between specific shape types.

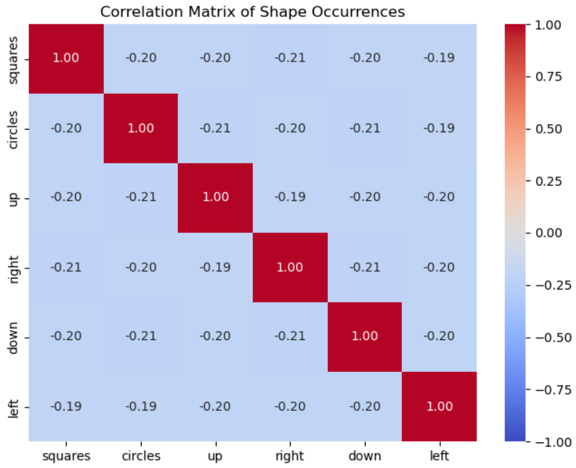

Figure 2: Correlation Matrix between Shape Counts.

We also counted all the shape occurrences and computed the mean of non-zero count labels for each shape type. The results are summarized in Table 1.

Table 1: Per-shape totals and mean of non-zero counts.

Shape	Total occurrences	Mean (non-zero) counts
squares	16574	5.055
circles	17149	5.029
up	16530	4.980
right	16770	5.000
down	16857	5.007
left	16120	4.928

We can conclude that the dataset is well-balanced, with no significant biases detected.

2 Methodology

2.1 Labeling Strategy

The classification task involves identifying the specific pair of shapes and their counts. Since there are $\binom{6}{2} = 15$ possible pairs and 9 possible count combinations (1 to 9) summing to 10, we map these to $15 \times 9 = 135$ unique classes. following this scheme:
For counting label

$$[a_0, a_1, a_2, a_3, a_4, a_5]$$

with the only non-zero elements a_i and a_j, where $i < j$ we will use a classification label equal to:

$$f(i, j) \cdot 9 + a_i - 1$$

Where for $g < h$, $f(g, h)$, is a simple indexing function:

$$f(g, h) = \binom{h}{2} + g$$

where $0 \leq g < h \leq 5$.

2.2 Model Architecture

We utilize a shared backbone with two task-specific heads. The backbone is a fixed sequential CNN provided in the assignment specifications, outputting a feature vector of size 256.

2.2.1 Classification Head

The classification head takes the 256-dimensional feature vector from the backbone. It consists of a linear layer maintaining the 256 dimensions, followed by a ReLU activation and Dropout ($p = 0.2$) for regularization. The final linear layer maps these features to the 135 class logits.

$$\text{Head}_{cls} : \mathbb{R}^{256} \xrightarrow{Linear} \mathbb{R}^{256} \xrightarrow{ReLU,Dropout} \mathbb{R}^{135}$$

2.2.2 Regression Head

The regression head branches from the same backbone output. It reduces the dimensionality to 128 via a linear layer, followed by ReLU and Dropout ($p = 0.2$). The final layer outputs 6 continuous values representing the counts for each shape type.

$$\text{Head}_{cnt} : \mathbb{R}^{256} \xrightarrow{Linear} \mathbb{R}^{128} \xrightarrow{ReLU,Dropout} \mathbb{R}^{6}$$

Hyperparameters such as layer sizes and dropout rates were chosen based on preliminary experiments to balance model capacity and overfitting risk.

2.3 Data Augmentation

To improve generalization, the following augmentations were applied during training:

- **Horizontal/Vertical Flip:** Increases diversity. Note that labels for directional triangles (up, right, down, left) are switched accordingly to maintain label correctness.
- **Rotation (±90°):** Also increases diversity. Labels for directional triangles (up, right, down, left) are also permuted accordingly.
- **Gaussian Noise:** Improves robustness to pixel-level variations.
- **Random Erasing:** We did not use this augmentation as it sometimes removes critical parts of the shapes, potentially confusing the model.

3 Experimental Setup

We conducted three experiments to evaluate the multitask learning approach. The model was trained using the Adam optimizer ($lr = 1e - 3$) with early stopping (patience=15).

The total loss function is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{CLS} + \lambda_{cnt} \cdot \mathcal{L}_{CNT} \tag{1}$$

where $\mathcal{L}_{CLS}$ is the Negative Log Likelihood Loss and $\mathcal{L}_{CNT}$ is the Smooth L1 Loss.

- **Exp 1 (Classification Only):** $\lambda_{cnt} = 0$. The regression head is ignored.
- **Exp 2 (Regression Only):** Classification loss is ignored. Only regression loss is used.
- **Exp 3 (Multitask):** $\lambda_{cnt} = 1$. Both tasks are learned simultaneously.

3.1 Experiment 1

3.1.1 Training

In this experiment, we trained the model solely for the classification task. The regression head was not utilized, and the total loss was equivalent to the classification loss. Early stopping was employed based on validation classification accuracy. The training concluded after 37 epochs. Training curves are presented in Figure 3.

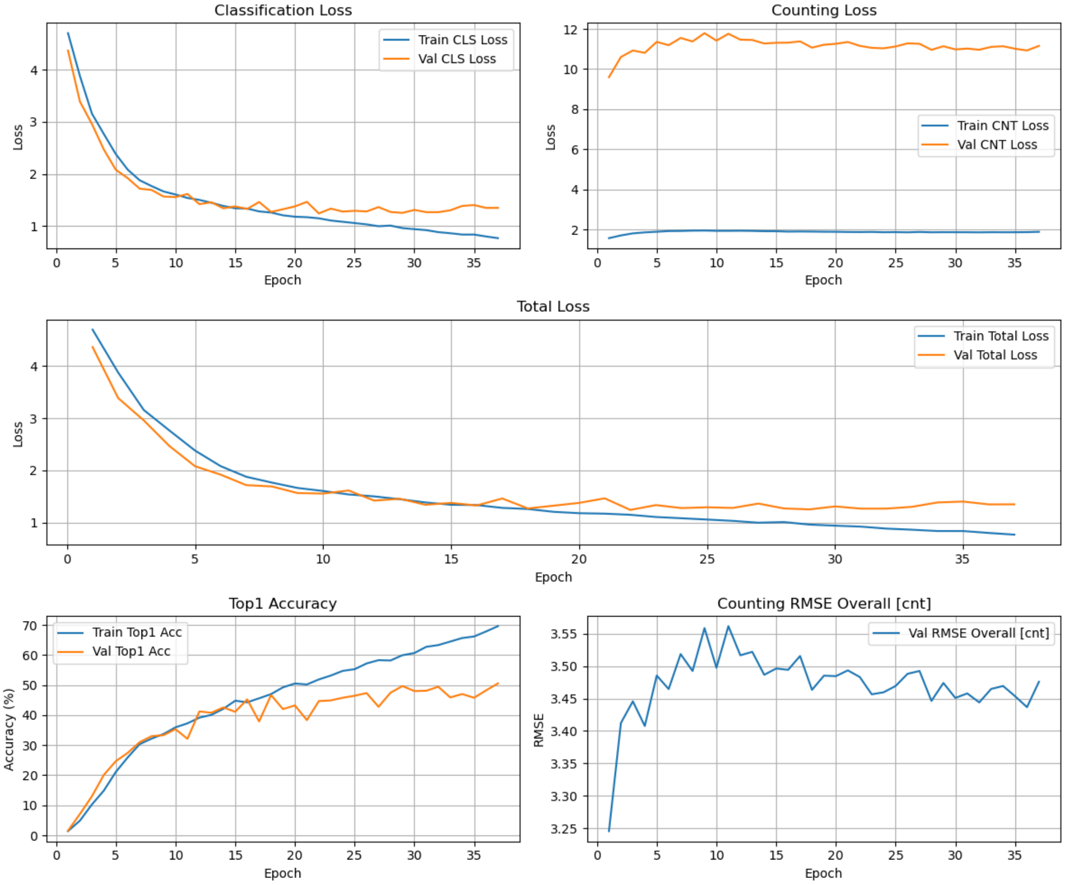

Figure 3: Training and Validation metrics for Experiment 1 (Classification Only).

As expected, the classification accuracy improved steadily, reaching a plateau towards the end of training while the counitng RMSE and counting loss changes are just random noise, as they are not optimized in this experiment.

3.1.2 Evaluation

The final model was evaluated on the validation set, yielding a Top-1 accuracy of 48.10% and a per-pair accuracy of 94.40%. The accuracy per pair is very high, indicating that most errors are due to miscounting rather than misclassifying the shape types. Detailed per-class accuracy and a confusion matrix are presented on the bar chart in the notebook.

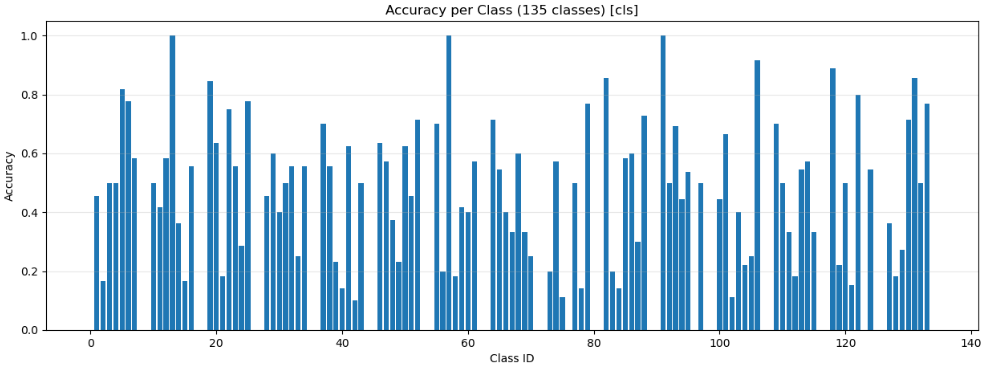

Figure 4: Per-class Accuracy for Experiment 1 (Classification Only).

Due to the dataset not containing examples where one shape count is 1 and the other is 9, the bar chart is clearly seperated into 15 groups. We can observe that some of those groups are classified with lower accuracy than others, indicating that certain shape pairs are inherently harder to classify (like group 9/15 corresponding to {up, down} pair).
We also present the scatter plots of predicted vs true labels for classifaction proglem and counts for each shape in Figure 5.

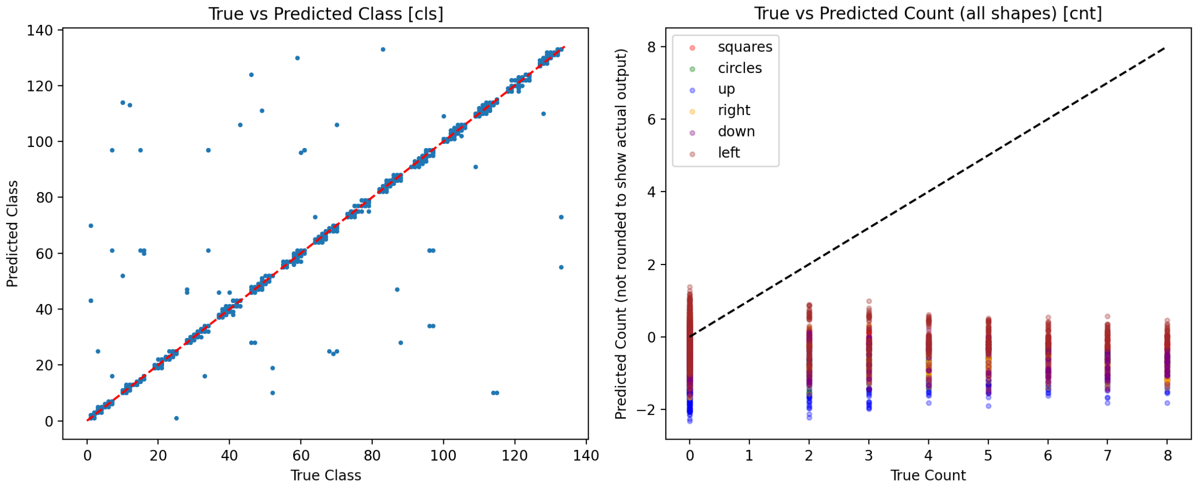

Figure 5: Scatter Plots of Predicted vs True Labels and Counts (Experiment 1).

We can observe that the classification scatter plot shows a clear diagonal trend and clearly separated clusters corresponding to different classes. The counting scatter plot is just random noise, as the counting head was not trained in this experiment.
Fianally, we present some numerical metrics in Tables 2 and 3.

Table 2: Classification Metrics (Exp 1)

Metric	Value
Top-1 Accuracy	48.10%
Macro F1-score	0.3507
Per-pair Accuracy	94.40%

Table 3: Regression Metrics (Exp 1)

Class	RMSE	MAE
squares	3.2210	1.8476
circles	3.5655	2.4965
up	3.8386	2.7638
right	3.6806	2.3880
down	3.4350	2.1683
left	3.0722	1.9072
Overall	**3.4787**	**2.2619**

We can concude the experiment as sucesful.

3.2 Experiment 2

3.2.1 Training

In this experiment, we trained the model solely for the regression task. The classification loss was not utilized, and the total loss was equivalent to the regression loss. Early stopping was employed based on validation loss for counting. The training concluded after 91 epochs. Training curves are presented in Figure 6.

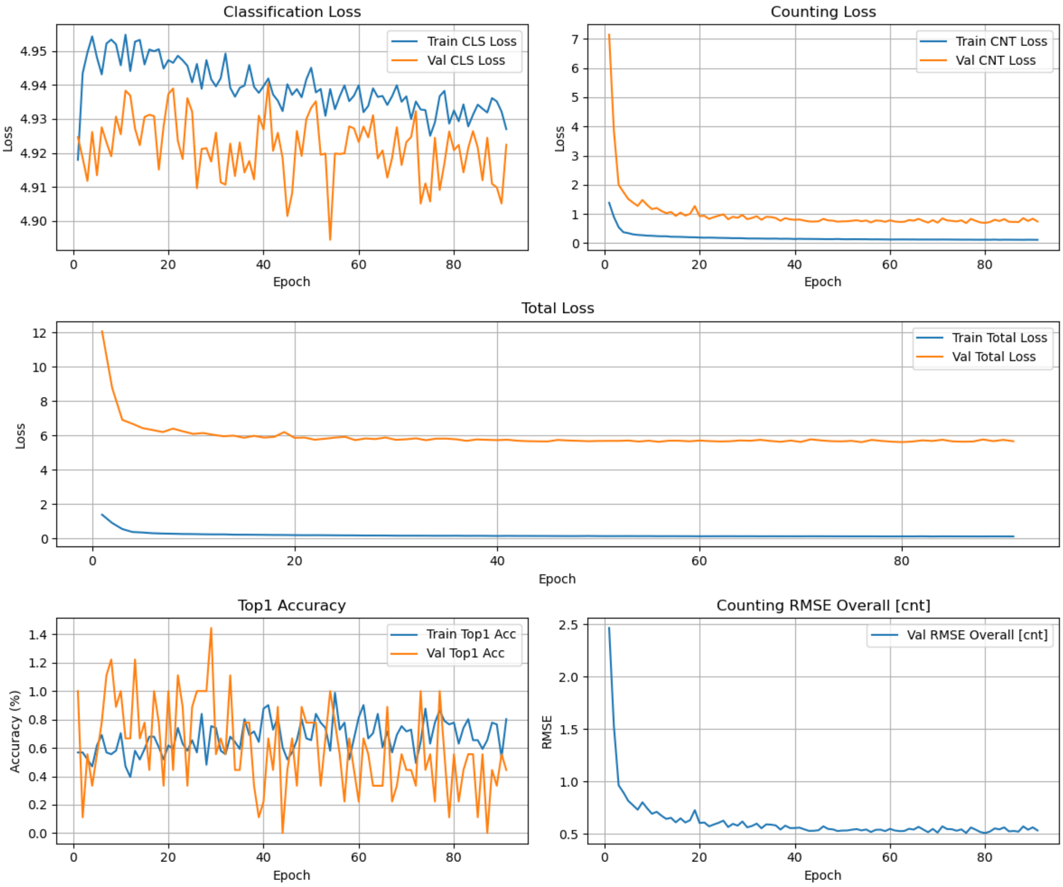

Figure 6: Training and Validation metrics for Experiment 2 (Regression Only).

As expected, the counting RMSE and counting loss improved steadily, reaching a plateau towards the end of training while the classification accuracy changes are just random noise, as they are not optimized in this experiment.

3.2.2 Evaluation

The exact same metrics and plots as in Experiment 1 are presented here for Experiment 2.

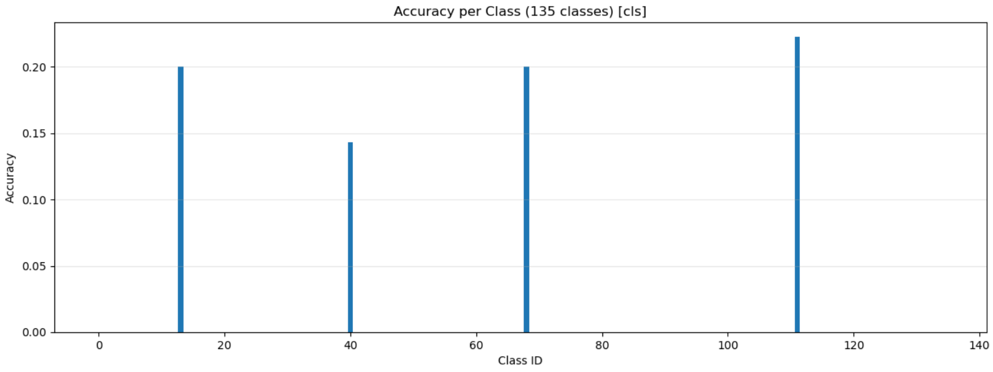

Figure 7: Per-class Accuracy for Experiment 2 (Classification Only).

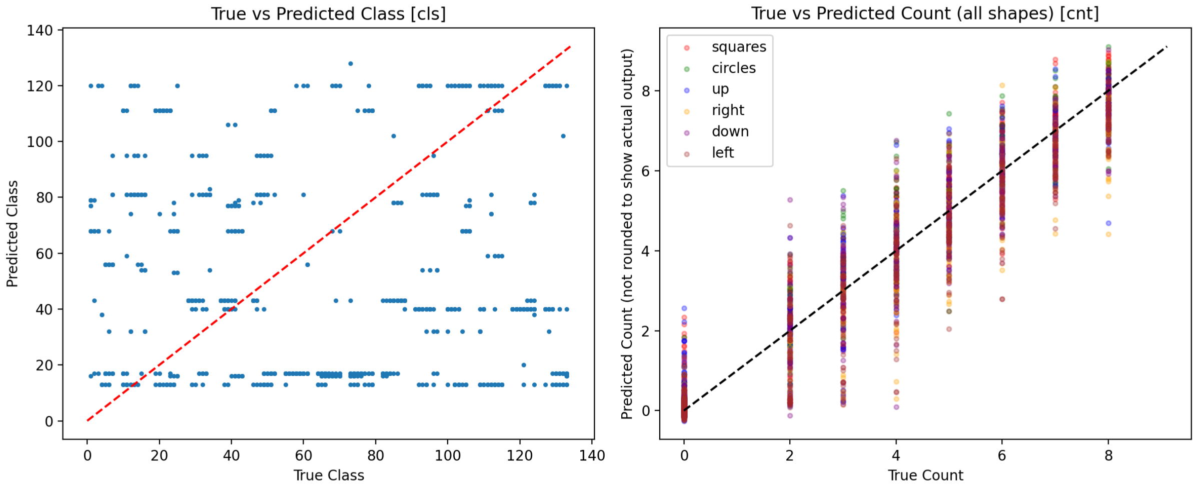

Figure 8: Scatter Plots of Predicted vs True Labels and Counts (Experiment 2).

Table 4: Classification Metrics (Exp 2)

Metric	Value
Top-1 Accuracy	0.60%
Macro F1-score	0.0010
Per-pair Accuracy	6.00%

Table 5: Regression Metrics (Exp 2)

Class	RMSE	MAE
squares	0.5035	0.2963
circles	0.4870	0.2780
up	0.5575	0.3073
right	0.6178	0.3234
down	0.5468	0.2928
left	0.6751	0.3588
Overall	**0.5683**	**0.3094**

As expected, the classification metrics are very poor, as the model was not trained for this task. The regression metrics are quite good, with overall RMSE of 0.5683 and MAE of 0.3094. The true vs predicted counts scatter plot 8 shows a clear diagonal trend, indicating a working model, although quite a lot of inacuracy is present.

3.3 Experiment 3

3.3.1 Training

In this experiment, we trained the model for both classification and regression tasks simultaneously, with regression loss weight $\lambda_{cnt} = 0.4$. Early stopping was employed based on a combined validation loss. The training concluded after 37 epochs. Training curves are presented in Figure 9.

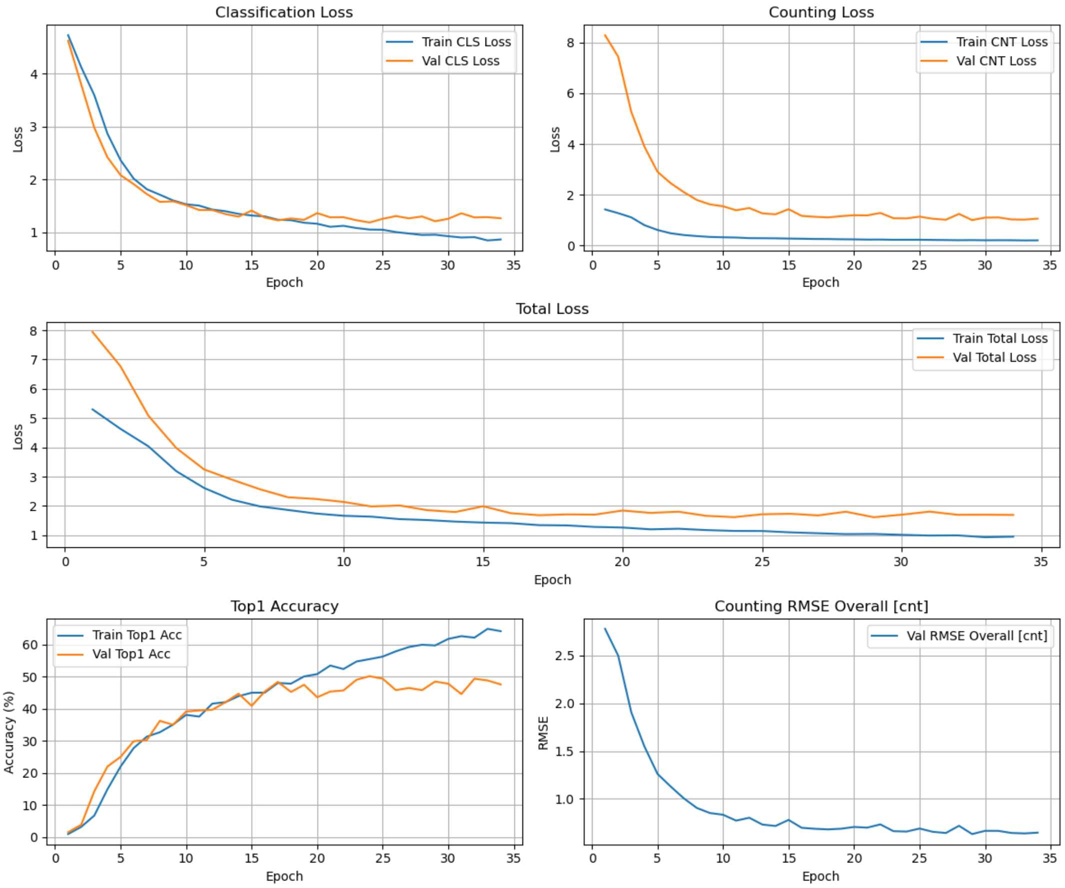

Figure 9: Training and Validation metrics for Experiment 3 (Multitask).

Both the classification and counting metrics improved steadily. However, the validation counting loss did not approach the training loss as closely as the classification loss did. This, together with the results from Experiment 2, suggests that the counting task is more difficult than classification for this dataset.

3.3.2 Evaluation

The exact same metrics and plots as in Experiment 1 and 2 are presented here.

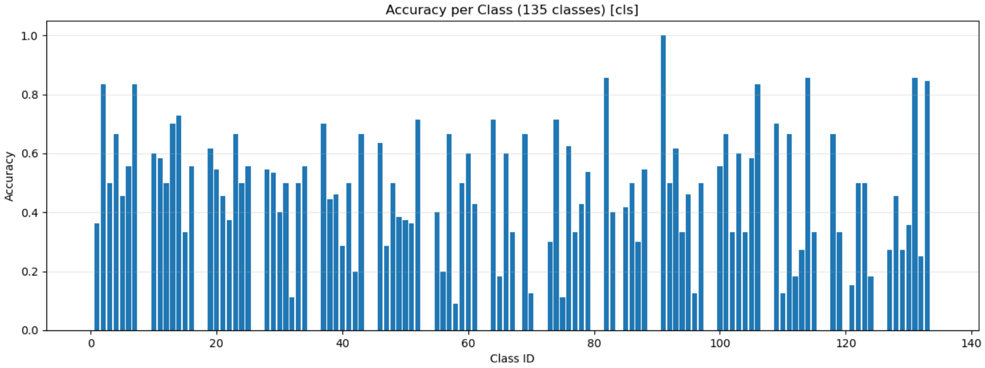

Figure 10: Per-class Accuracy for Experiment 3 (Classification Only).

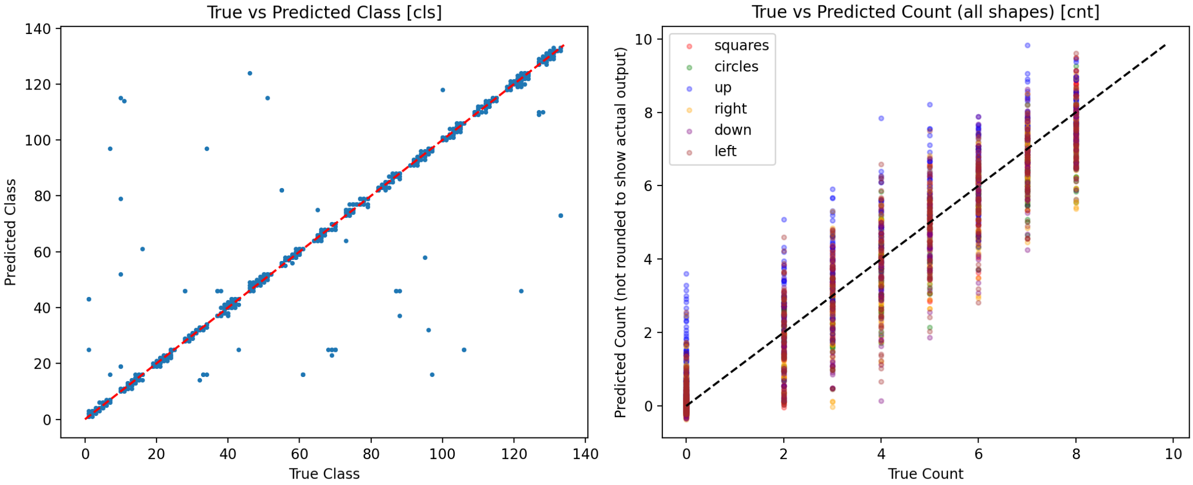

Figure 11: Scatter Plots of Predicted vs True Labels and Counts (Experiment 3).

Table 6: Classification Metrics (Exp 3)

Metric	Value
Top-1 Accuracy	48.50%
Macro F1-score	0.3579
Per-pair Accuracy	94.40%

Table 7: Regression Metrics (Exp 3)

Class	RMSE	MAE
squares	0.5075	0.2796
circles	0.5496	0.3041
up	0.7994	0.4925
right	0.6565	0.3767
down	0.6244	0.3507
left	0.6938	0.3900
Overall	**0.6456**	**0.3656**

The classification metrics are comparable to those from Experiment 1, indicating that multitask learning did not significantly harm classification performance. The regression metrics are slightly worse than those from Experiment 2, suggesting a small trade-off when training both tasks simultaneously. In conclusion, the results seem to combine the strengths of both single-task experiments with just slight trade-offs.

4 Discussion

The results from our experiments suggest that multitask learning can effectively balance the demands of both classification and regression tasks in the context of geometric shape analysis. We managed to train a single model that performs reasonably well on both tasks, although with some trade-offs compared to single-task baselines. The results could be improved further with more extensive hyperparameter tuning and potentially more sophisticated architectures, but for the scope of this assignment, the findings are promising.

From the expirments done outside of this report, we found out that indroducing data augmentations significantly improved the model's results on the testing set. Augmentations such as horizontal and vertical flips, as well as rotations, helped the model become invariant to shape orientations, which is crucial given the nature of the dataset. Adding Gaussian noise also contributed to robustness against minor pixel-level variations.

Introducing dropout layers in the task-specific heads proved beneficial in mitigating overfitting, especially given the relatively small size of the dataset (10,000 images). The dropout layers helped regularize the model, leading to better generalization on the validation and test sets.