

# Report

## Section on (EDA) Exploratory Data Analysis in your report

### Distribution of Classification Classes (0-134)

The distribution of shape-pair classes shows that the dataset is approximately balanced across the 135 categories (Figure 1).

### Distribution of Counts per Shape Type

The distribution of counts per shape type shows that the dataset is approximately balanced across all of the 6 shapes (Figure 1).

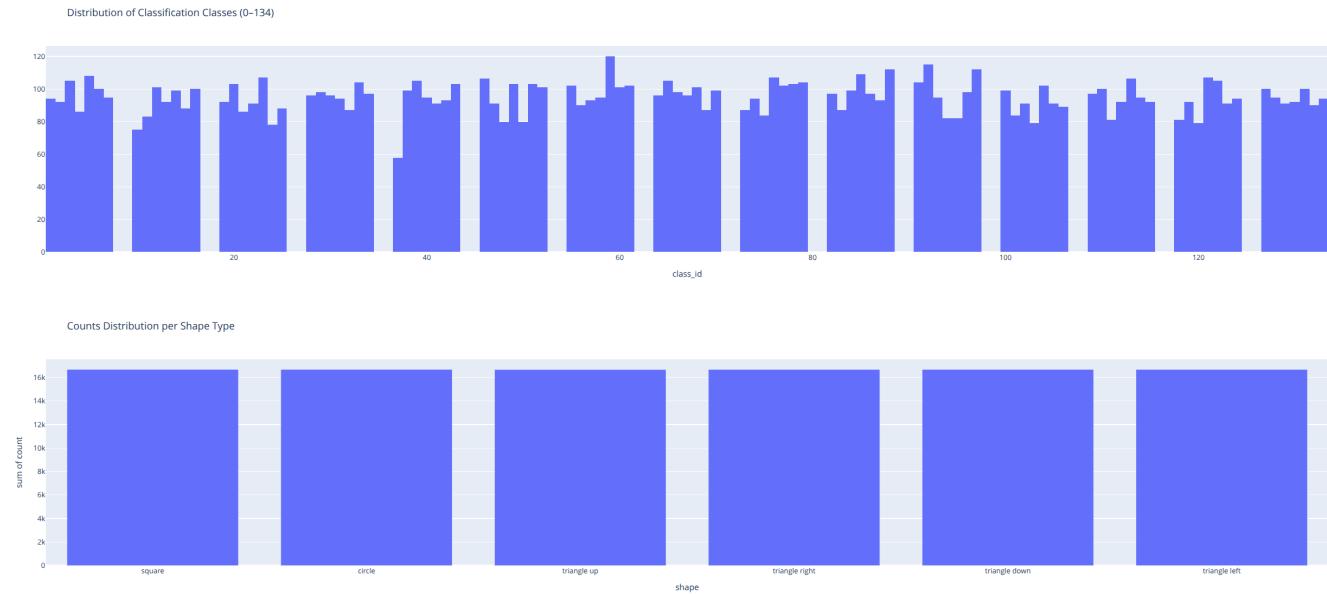


Figure 1: Distributions

## Model architecture

The model's backbone is known. Our model has two heads - one for classification task and one for regression task.

```
self.head_cls = nn.Sequential(  
    nn.Dropout(0.65),  
    nn.Linear(256, 135),  
    nn.LogSoftmax(dim=1)  
)  
self.head_cnt = nn.Sequential(  
    nn.Linear(256, 128),  
    nn.BatchNorm1d(128),  
    nn.ReLU(),  
    nn.Dropout(0.5),  
    nn.Linear(128, 6)  
)
```

## Description and justification of augmentations

### Horizontal flip ( $p\_flip=0.3$ )

The image is flipped horizontally, and labels corresponding to directional shapes are swapped to maintain correct labels.

This increases invariance to left-right orientation and helps the model learn that shapes may appear mirrored without changing the underlying class.

### **Vertical Flip (`p_vflip = 0.3`)**

The image is flipped vertically, and labels for top/bottom-oriented shapes are swapped.

This increases invariance to up-down orientation and helps the model learn that shapes may appear mirrored without changing the underlying class.

### **Rotation (`p_rot = 0.3`)**

The image is rotated 90deg counterclockwise, and labels for directional shapes are updated (`up -> left -> down -> right`).

### **Gaussian Noise (`p_noise = 0.1, noise_std = 0.1`)**

Random Gaussian noise is added to pixel values and clipped between 0 and 1.

For this dataset, which consists of clean synthetic binary images, this augmentation is likely not very beneficial, but it was included to test whether small perturbations improve model robustness.

### **Random Erasing (`p_erase = 0.2`)**

Small random rectangles of pixels are zeroed out in the image. Labels remain unchanged.

Simulates occlusions or missing parts of shapes, forcing the model to focus on global features rather than individual pixels.

## **Results table (loss, accuracy, RMSE for all runs)**

### **Classification only**

Loss	Top-1 Accuracy	Macro F1-score	RMSE overall	MAE overall	Runtime
1.3432	46.0000	0.3137	3.1157	1.8375	4min 40s

#### **Per-pair Accuracy (15 classes):**

```
(square, circle): 57.1429,  (square, up): 44.2857,  (square, right): 47.6190
(square, down): 48.4375,  (square, left): 42.0290,  (circle, up): 45.3333
(circle, right): 45.0704,  (circle, down): 47.5410,  (circle, left): 53.1250
(up, right): 44.6154,  (up, down): 47.7612,  (up, left): 43.0769
(right, down): 41.9355,  (right, left): 46.9697,  (down, left): 37.3333
```

#### **RMSE/MAE per class:**

```
Class square: 3.0108/1.7913,  Class circle: 3.1260/1.8253,  Class up: 3.3232/1.9383
Class right: 3.0703/1.8740,  Class down: 3.0530/1.8273,  Class left: 3.1011/1.7691
```

### **Regression only**

Loss	Top-1 Accuracy	Macro F1-score	RMSE overall	MAE overall	Runtime
0.1399	0.6000	0.0024	0.5664	0.3263	18min 35s

#### **Per-pair Accuracy (15 classes):**

```
(square, circle): 0.0000,  (square, up): 5.7143,  (square, right): 0.0000
(square, down): 0.0000,  (square, left): 0.0000,  (circle, up): 0.0000
(circle, right): 0.0000,  (circle, down): 0.0000,  (circle, left): 0.0000
(up, right): 0.0000,  (up, down): 2.9851,  (up, left): 0.0000
```

(right, down): 0.0000, (right, left): 0.0000, (down, left): 0.0000

RMSE/MAE per class:

Class square: 0.4714/0.3035, Class circle: 0.4651/0.2621, Class up: 0.5872/0.3288  
Class right: 0.6195/0.3442, Class down: 0.6169/0.3402, Class left: 0.6140/0.3792

## Multitask

Loss	Top-1 Accuracy	Macro F1-score	RMSE overall	MAE overall	Runtime
1.3110	46.7000	0.3277	0.6109	0.3493	5min 10s

Per-pair Accuracy (15 classes):

(square, circle): 58.7302, (square, up): 44.2857, (square, right): 50.7937  
(square, down): 53.1250, (square, left): 62.3188, (circle, up): 41.3333  
(circle, right): 50.7042, (circle, down): 42.6230, (circle, left): 46.8750  
(up, right): 47.6923, (up, down): 52.2388, (up, left): 35.3846  
(right, down): 53.2258, (right, left): 30.3030, (down, left): 33.3333

RMSE/MAE per class:

Class square: 0.5639/0.3576, Class circle: 0.5107/0.2896, Class up: 0.6012/0.3310  
Class right: 0.6627/0.3793, Class down: 0.6252/0.3562, Class left: 0.6848/0.3823

## Discussion on multitask effects

An interesting observation is that, although in the specific run presented here the classification-only model converged slightly faster than the multitask model, this appears to be mostly due to randomness in initialization and sampling. In earlier experiments with reproducibility disabled (several repeated runs), the multitask setup typically reached the target classification accuracy about 10 epochs earlier than the classification-only network. This supports the original hypothesis:

the auxiliary regression task helps guide the learning process, making the feature representation more informative and improving convergence speed on average.

This effect arises because the two tasks are strongly related. Predicting the number of each shape type (regression) reinforces the internal representation required for identifying shape pairs (classification). The auxiliary task therefore acts as an additional learning signal.

It is also important to note that the multitask setup remains effective thanks to regularization. The total loss becomes larger due to the regression component, which could cause the network to “learn too fast” and destabilize training. However, the use of dropout and batch normalization in task-specific heads prevents overfitting and keeps learning stable.

Finally, we observe that the multitask model achieves performance on both tasks that is comparable to the dedicated single-task networks—classification quality similar to the classification-only model and regression accuracy similar to the regression-only model. Moreover, we obtain this with a similar number of epochs for classification, and a significantly smaller number of epochs needed for regression, all in a single training run. This demonstrates that when tasks are related, multitask learning can effectively solve them simultaneously at lower overall cost, providing both efficiency and improved learning dynamics.

## Learning curves

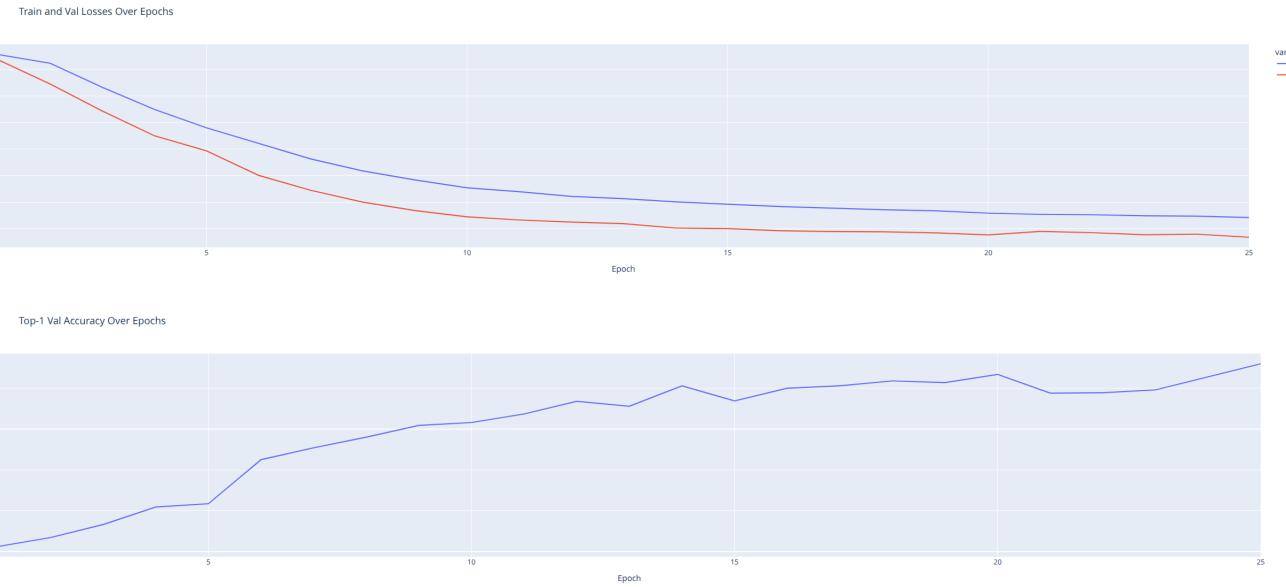


Figure 2: Classification only learning curves

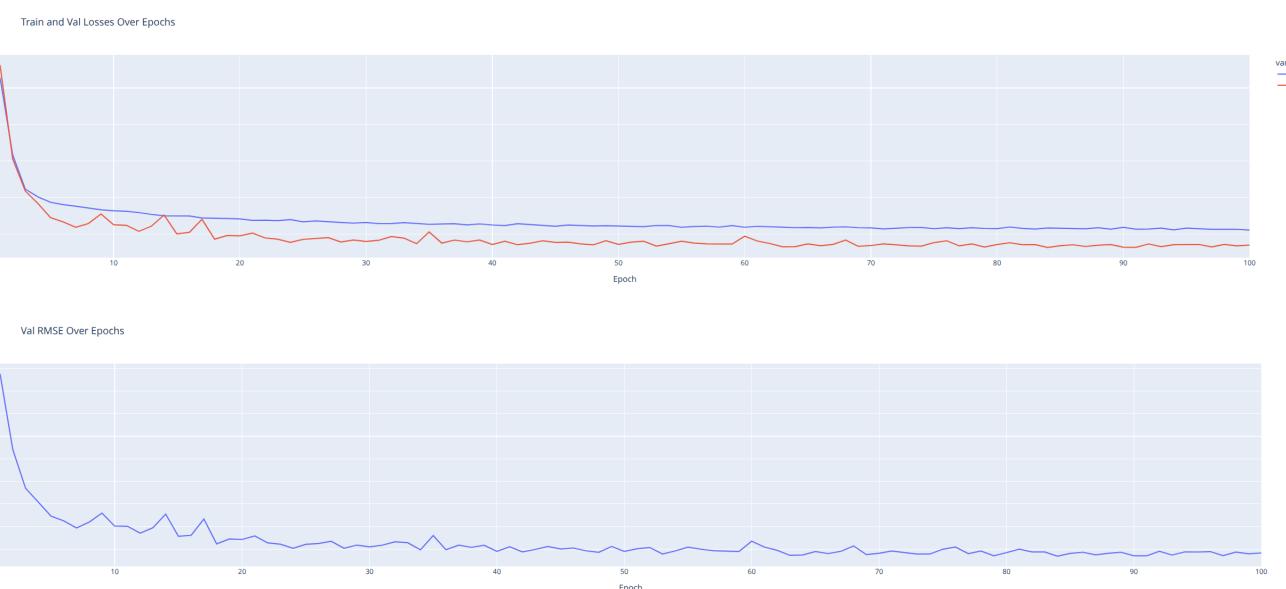


Figure 3: Regression only learning curves

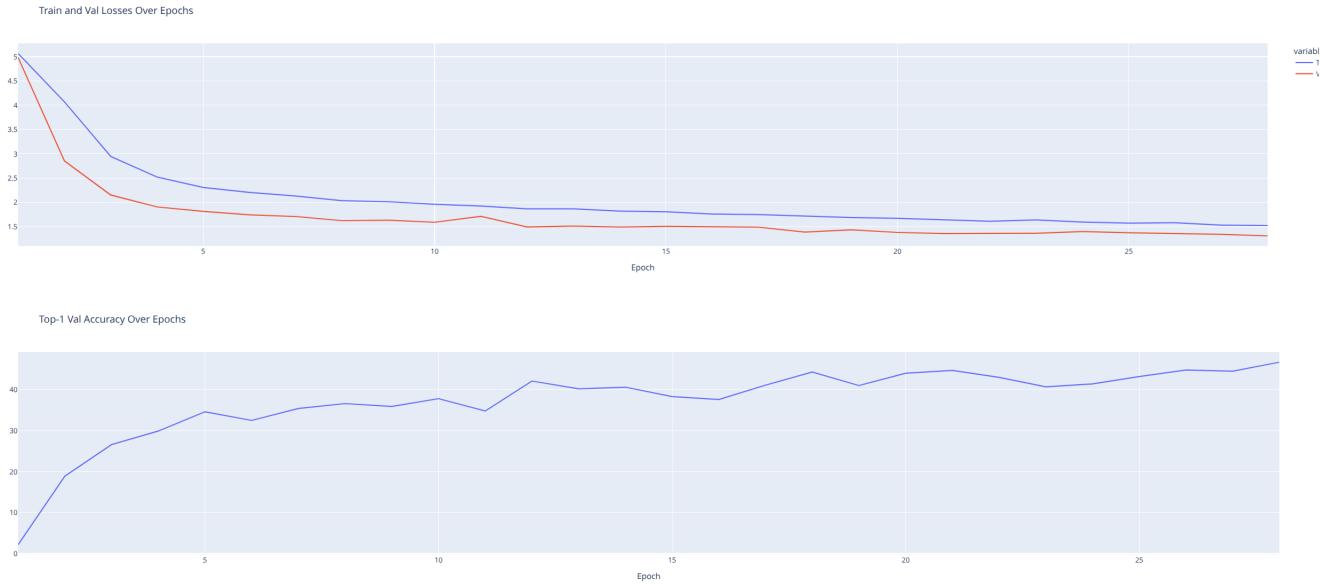


Figure 4: Multitask loss plot and top1 accuracy plot



Figure 5: Multitask rmse plot

## Confusion matrix

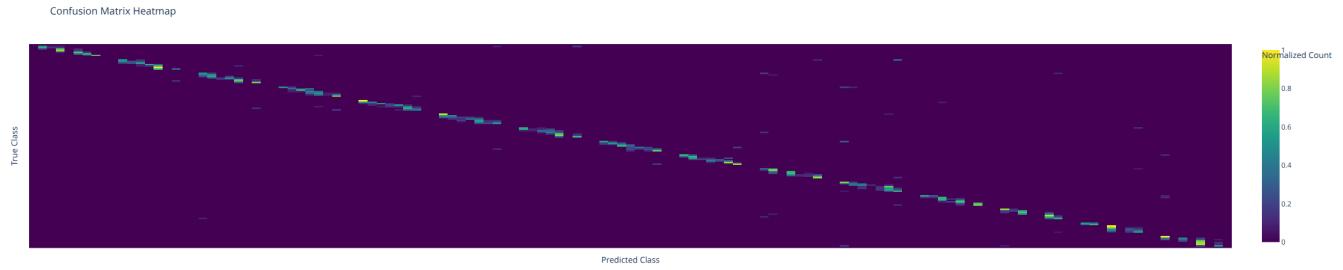


Figure 6: Classification only confusion matrix

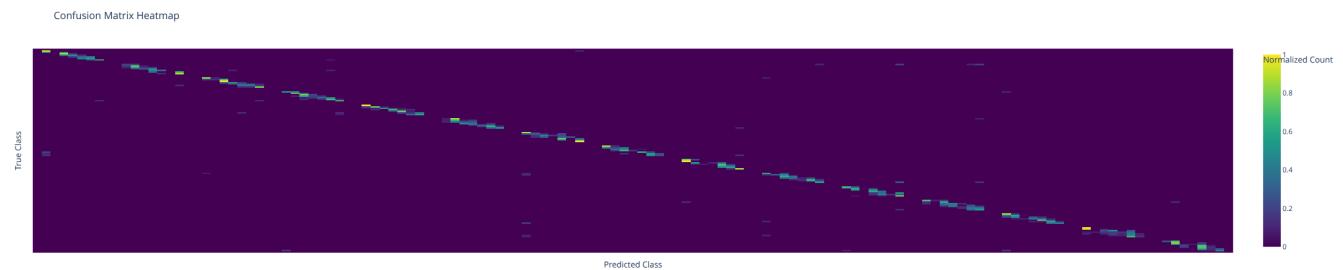


Figure 7: Multitask confusion matrix