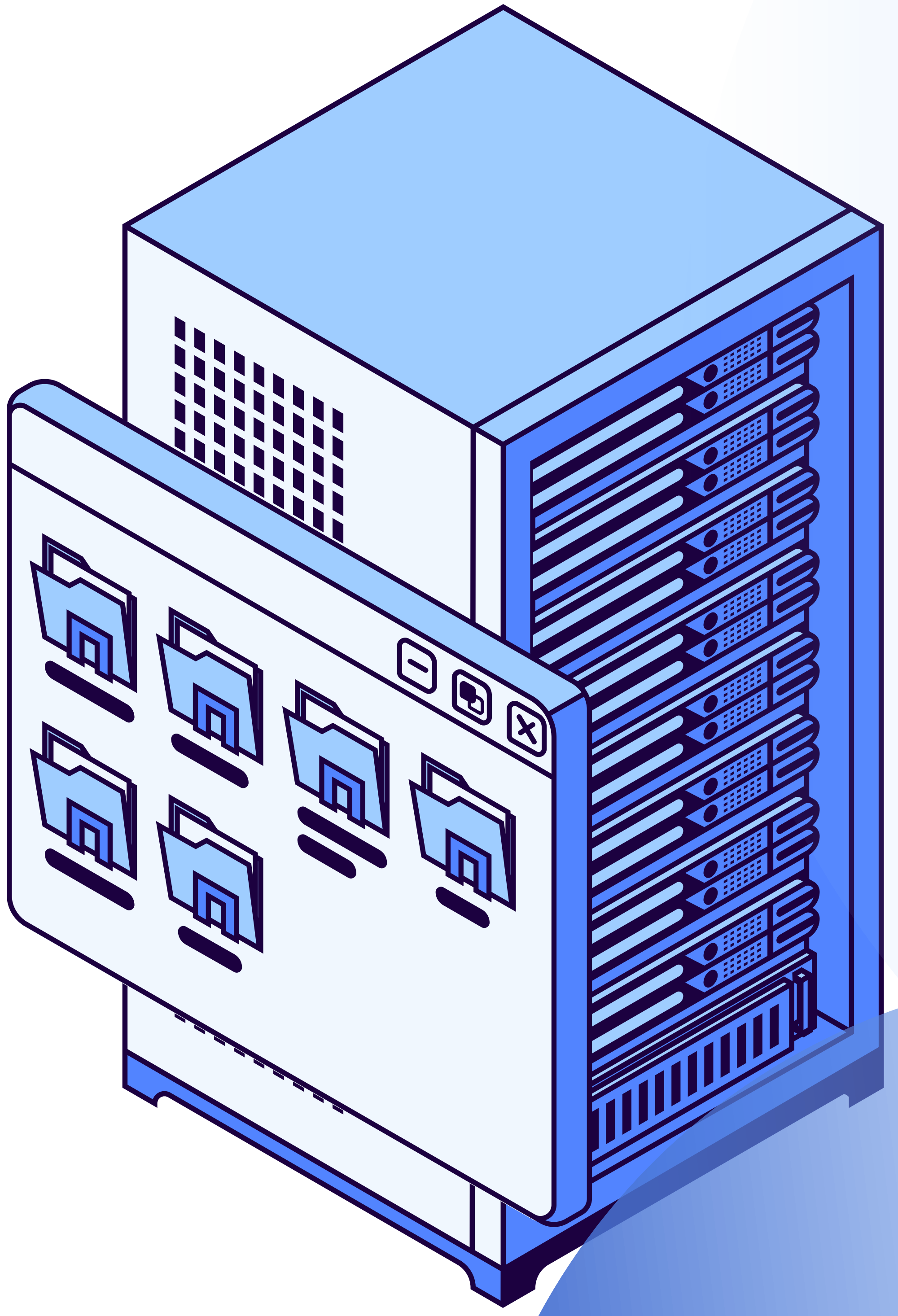
AUTOMATIC ESTIMATION OF PARKING LOT OCCUPANCY



FAORO ALBERTO EDOARDO (ID 5406685)
QUADRINI ELISA (ID 5412938)

AGENDA

1) Project overview

2) Dataset format

3) EDA and pre-processing

4) Neural network architecture

5) Regression Model

6) Classification Model

7) Conclusions

PROJECT OVERVIEW

Goal

The project focuses on the **automatic estimation of parking lot occupancy** using deep learning techniques.

Dataset

We worked with camera images of parking areas collected at the Universidade Federal do Paranà (Brazil) under **different weather conditions**, where individual parking spots were manually annotated through bounding boxes, originally intended for object detection tasks.

Problem Formulation

The final objective is divided into two tasks:

1) Regression: predict the number of occupied parking spaces in an image

2)Binary Classification: determine whether the parking lot is mostly occupied or mostly empty, based on a threshold of 50% occupancy.

https://public.roboflow.com/object-detection/pklot/2/preprocessing

DATASET FORMAT

PKLot format: Images + COCO annotations

Construction of project specific targets derived from COCO annotations

Dataset Classes:

- Regression → returns single (image, target variable (n_{occ}))
- Classification → returns single (image label, binary variable (based on $ratio_{occ}$))

Data Loader:

- Input: dataset, batch_size, shuffle
- Output: mini batches of (images, targets) to be used during training and evaluation

Image + Bounding Boxes

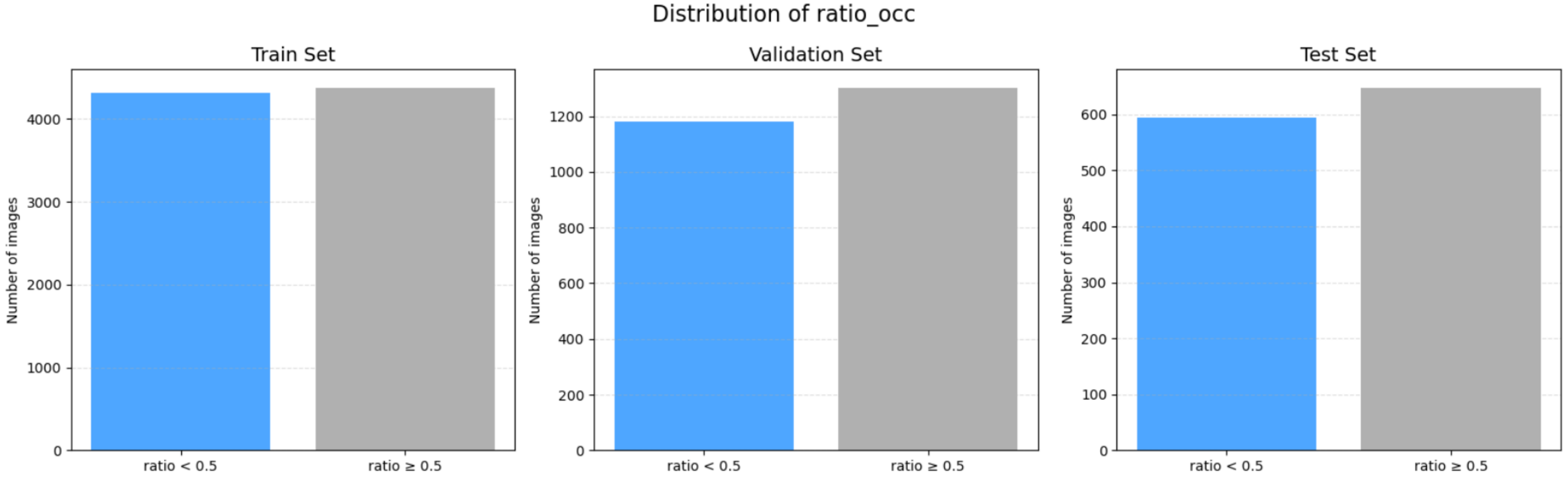

```
{0: {'file_name': '2013-04-10_12_00_06_jpg.rf.bef78f7879666a6d8997ce513fd8d124.jpg',
  'n_total': 40,
  'n_occ': 39,
  'n_empty': 1,
  'ratio_occ': 0.975,
  'label_binary': 1,
  'anns': [{'bbox': [304, 468, 83.5, 114], 'category': 2},
   {'bbox': [271, 390, 76.5, 89], 'category': 2},
   {'bbox': [245, 326, 67.5, 80], 'category': 2},
   {'bbox': [223, 270, 63.5, 73], 'category': 2},
   {'bbox': [204, 224, 58, 57], 'category': 2},
   {'bbox': [192, 176, 56.5, 58], 'category': 2},
   {'bbox': [171, 139, 53.5, 49], 'category': 2},
   {'bbox': [158, 101, 48.5, 50], 'category': 2},
   {'bbox': [151, 75, 46, 41], 'category': 2},
   {'bbox': [139, 46, 43, 41], 'category': 2},
   {'bbox': [365, 425, 67.5, 82], 'category': 2},
   {'bbox': [317, 353, 71, 78], 'category': 2},
   {'bbox': [302, 288, 60.5, 69.5], 'category': 2},
   {'bbox': [272, 235, 65, 62], 'category': 2},
   {'bbox': [254, 192, 61, 54], 'category': 2},
   {'bbox': [237, 148, 58, 51.5], 'category': 2},
   {'bbox': [222, 117, 49.5, 43.5], 'category': 2},
   {'bbox': [202, 78, 55.5, 49], 'category': 2},
   {'bbox': [192, 58, 44.5, 35.5], 'category': 2},
```

EDA AND PRE-PROCESSING

Class Imbalance

As part of the EDA, we performed a check for class imbalance. We examined the distribution of the binary labels derived from the occupancy ratio, defined as:

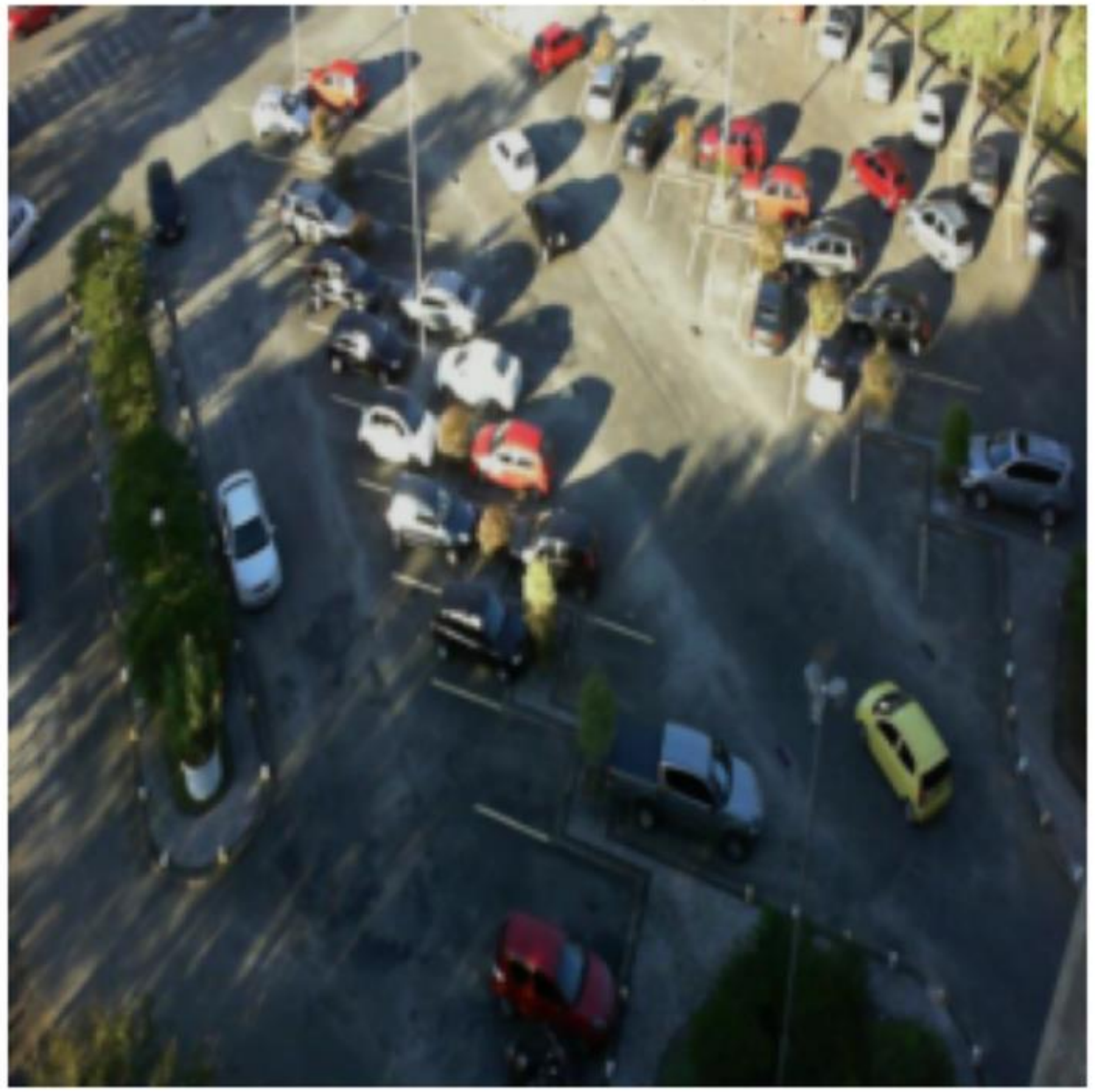
$$\mathrm{ratio}_{\mathrm{occ}} = \frac{n_{\mathrm{occ}}}{n_{\mathrm{total}}}$$



Pre-Processing

- Resize, tensor conversion, normalization
- ColorJitter to improve robustness to lighting and weather conditions

Transformed Image



NEURAL NETWORK CONSTRUCTION

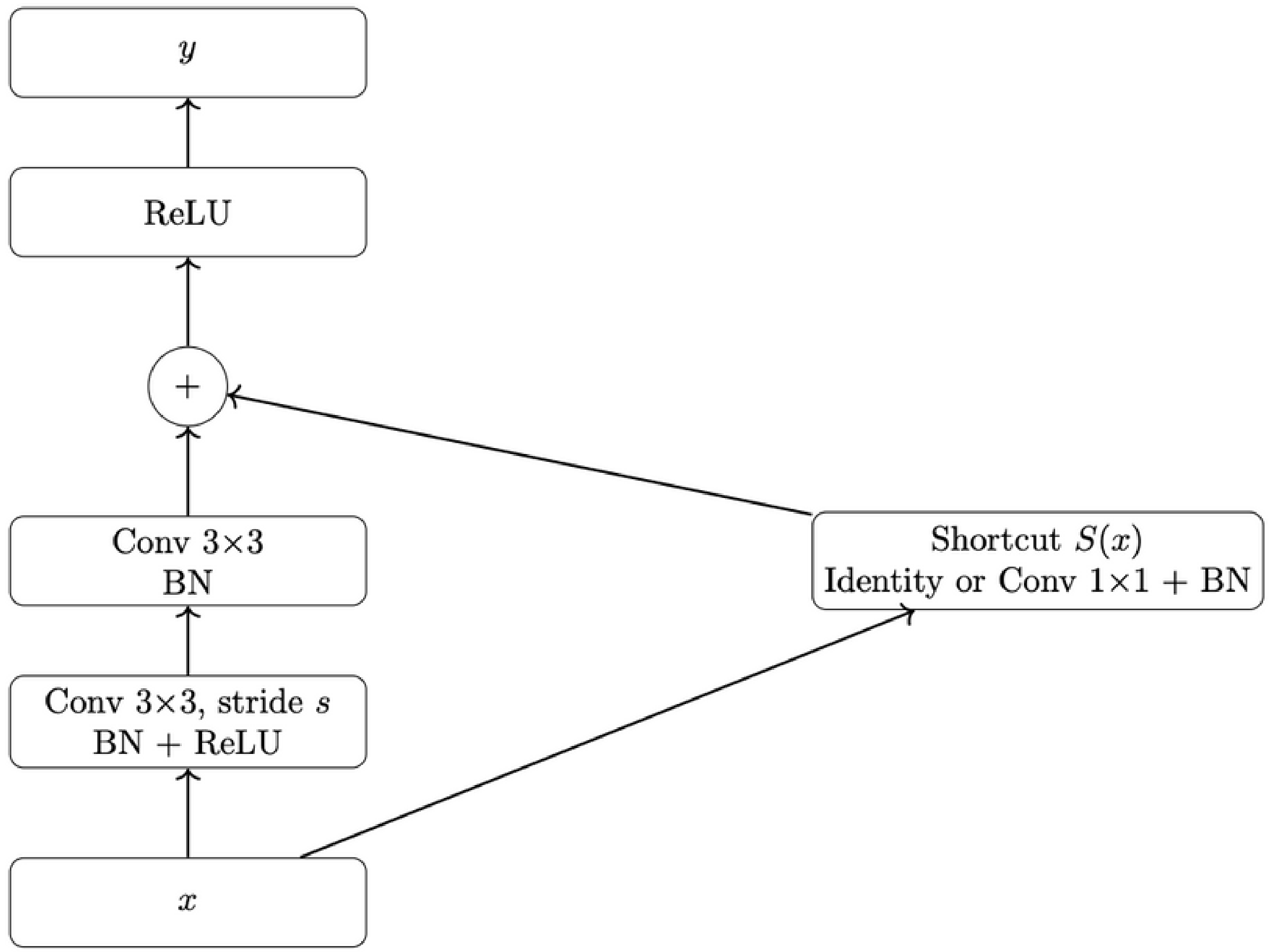


Figure 1: ResNet BasicBlock architecture.

Residual block with two 3×3 convolutions

- Shortcut connection: identity or 1×1 conv + batch norm (for dimension matching)
- Output = input + residual mapping $y = ReLU\left(F\left(x\right) + S\left(x\right)\right)$

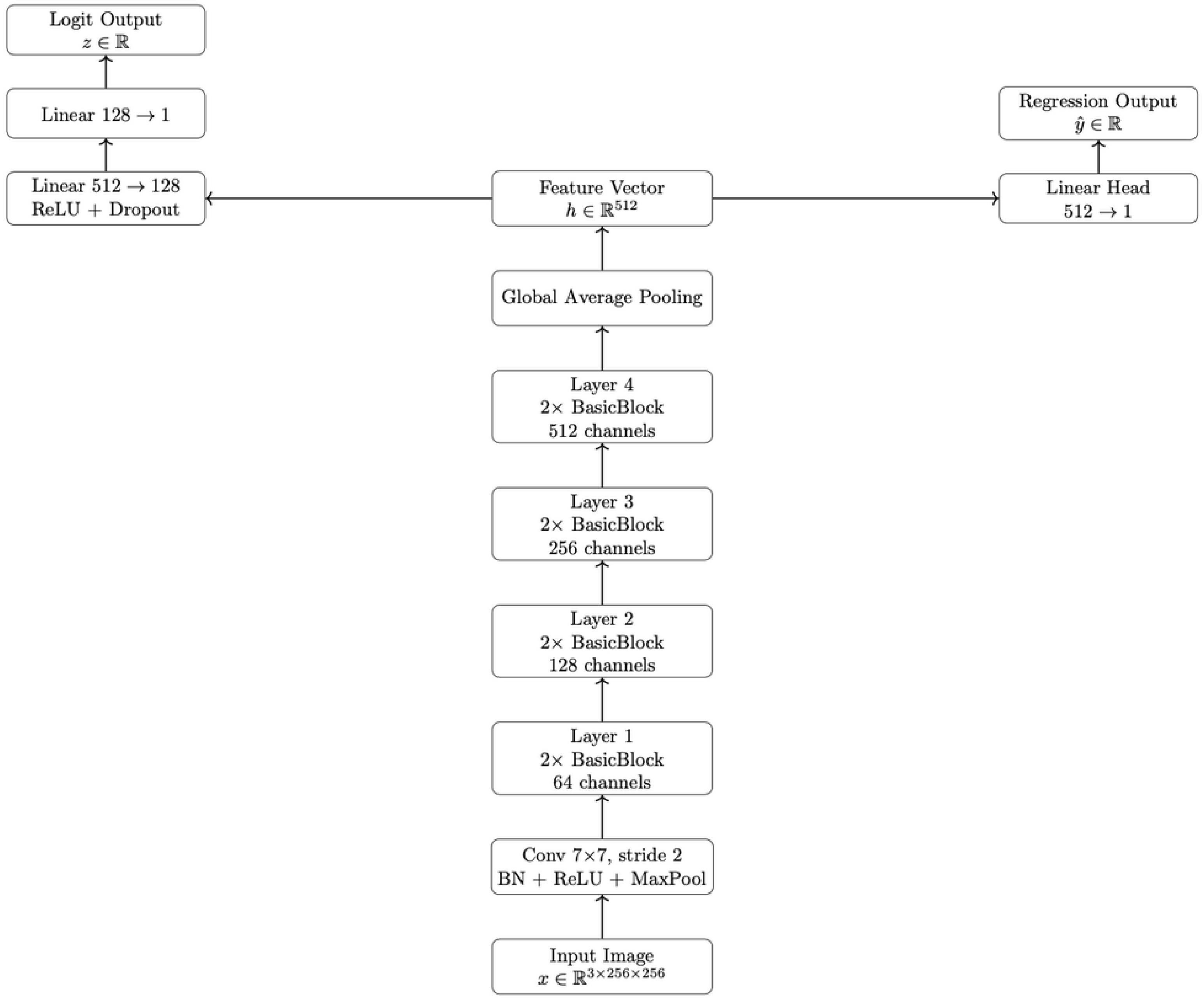


Figure 2: Overview of the proposed architecture.

ResNet-18 backbone → 512-D feature vector

Two heads:

- Regression head: single linear layer → scalar output
- Classification head: 2 FC layers + ReLU + dropout → scalar logit

FIRST MODEL: REGRESSION

Loss function: Mean Squared Error

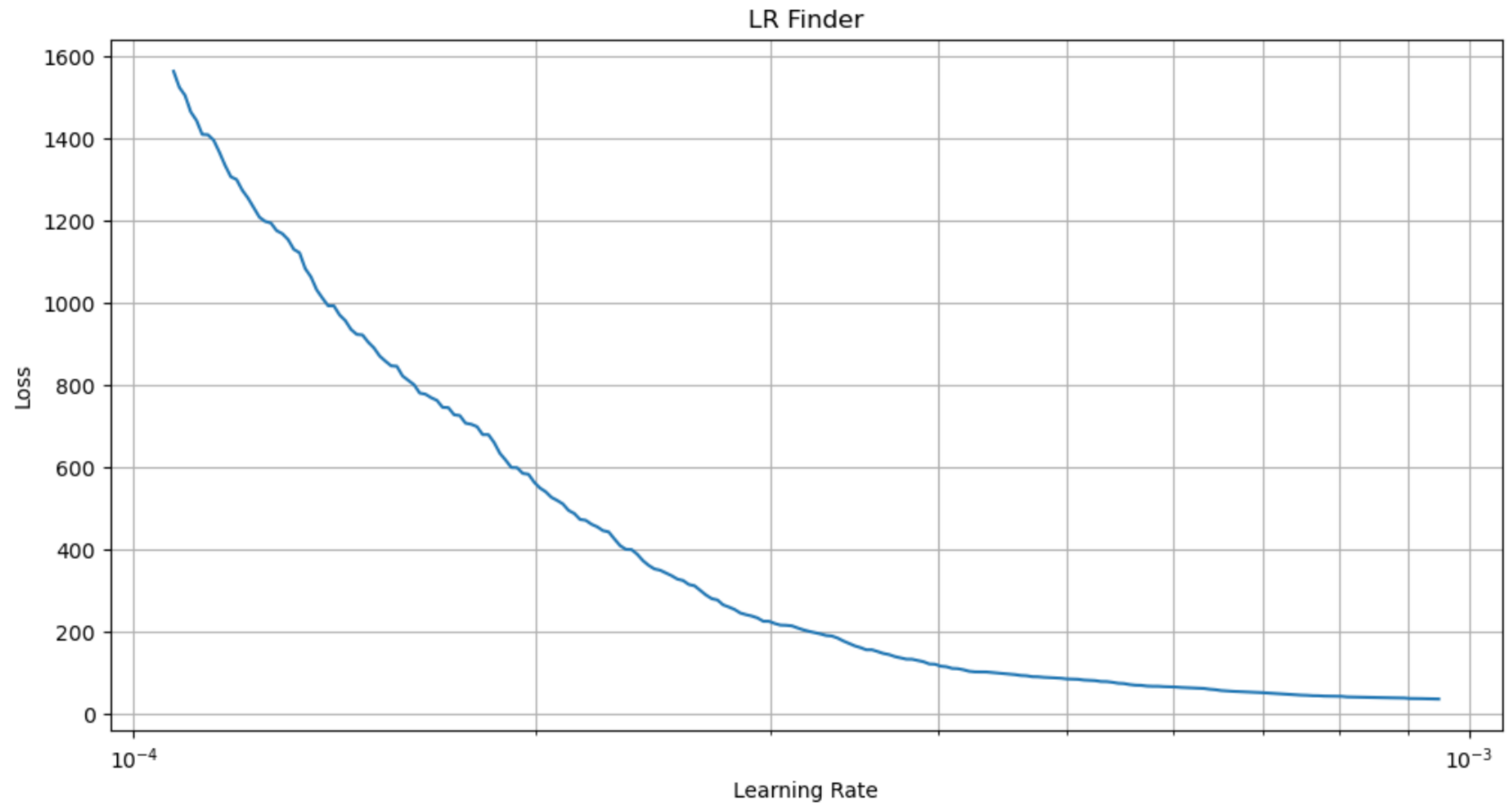
TRAINING INITIALIZATION

Learning Rate Finder (Smith, 2017)

- Exponentially increases the learning rate over 700 iterations:

$$lr_i = lr_{start} \times \left(\frac{lr_{end}}{lr_{start}} \right)^{\frac{i}{N}}$$

- The selected learning rate corresponds to the region of steep loss decrease before divergence.
- Final choice: $lr = 10^{-3}$



Weight Initialization (Kaiming / He)

- Convolutional and linear layers initialized with Kaiming normal initialization (ReLU): $Var\,(w) = \frac{2}{fan\_in}$
- Linear biases initialized to zero

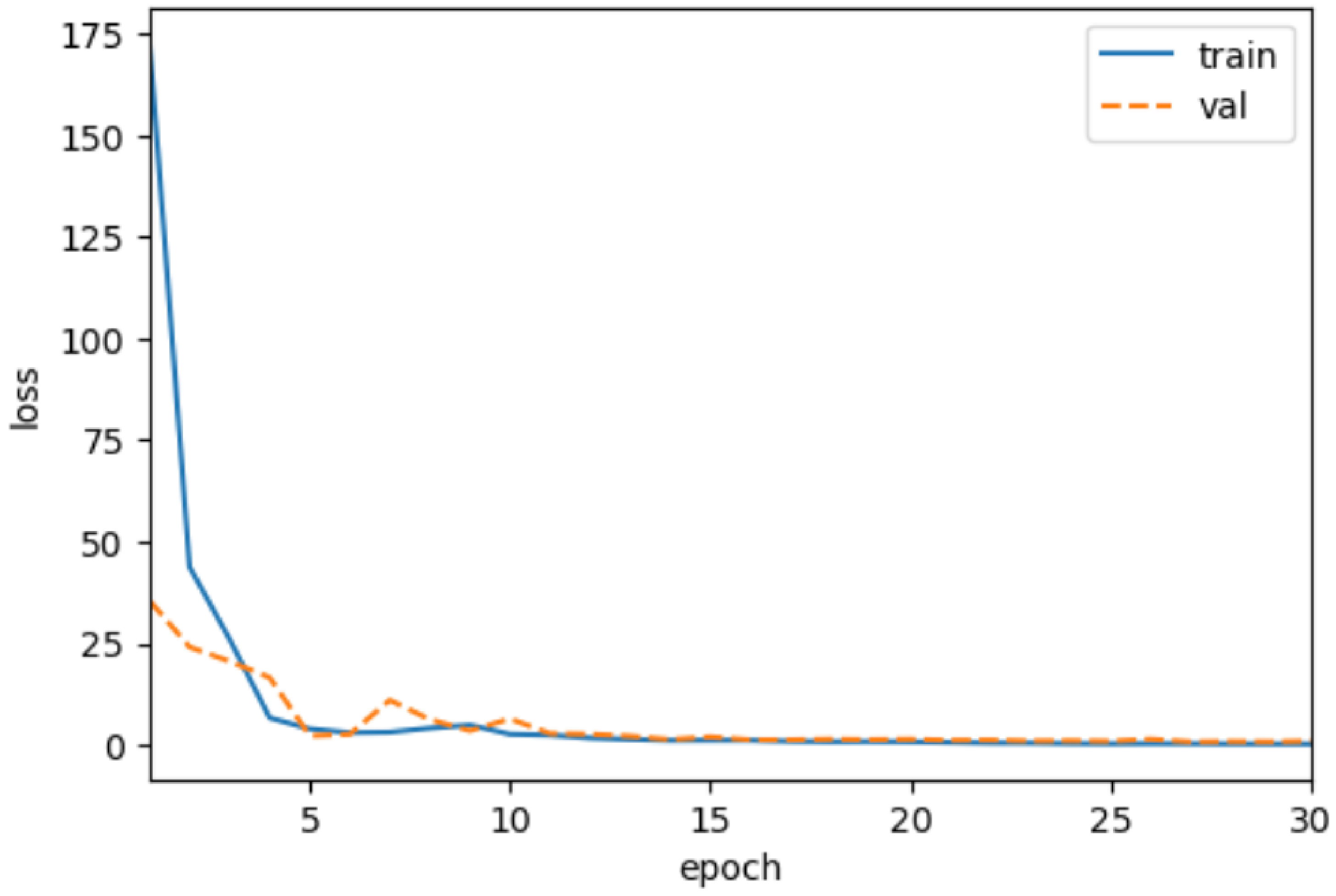
Optimizer (Adam)

Adam optimizer, using first and second moments of gradients

Learning Rate Scheduler

The learning rate policy combines warm-up and cosine annealing:

- Warm-up over the first 3 epochs
- Cosine annealing for the remaining epochs




```
[REG] Epoch 30/30 | Train: 0.2000 | Val: 0.9295
Restored best model with val loss = 0.7352
```

Model Selection

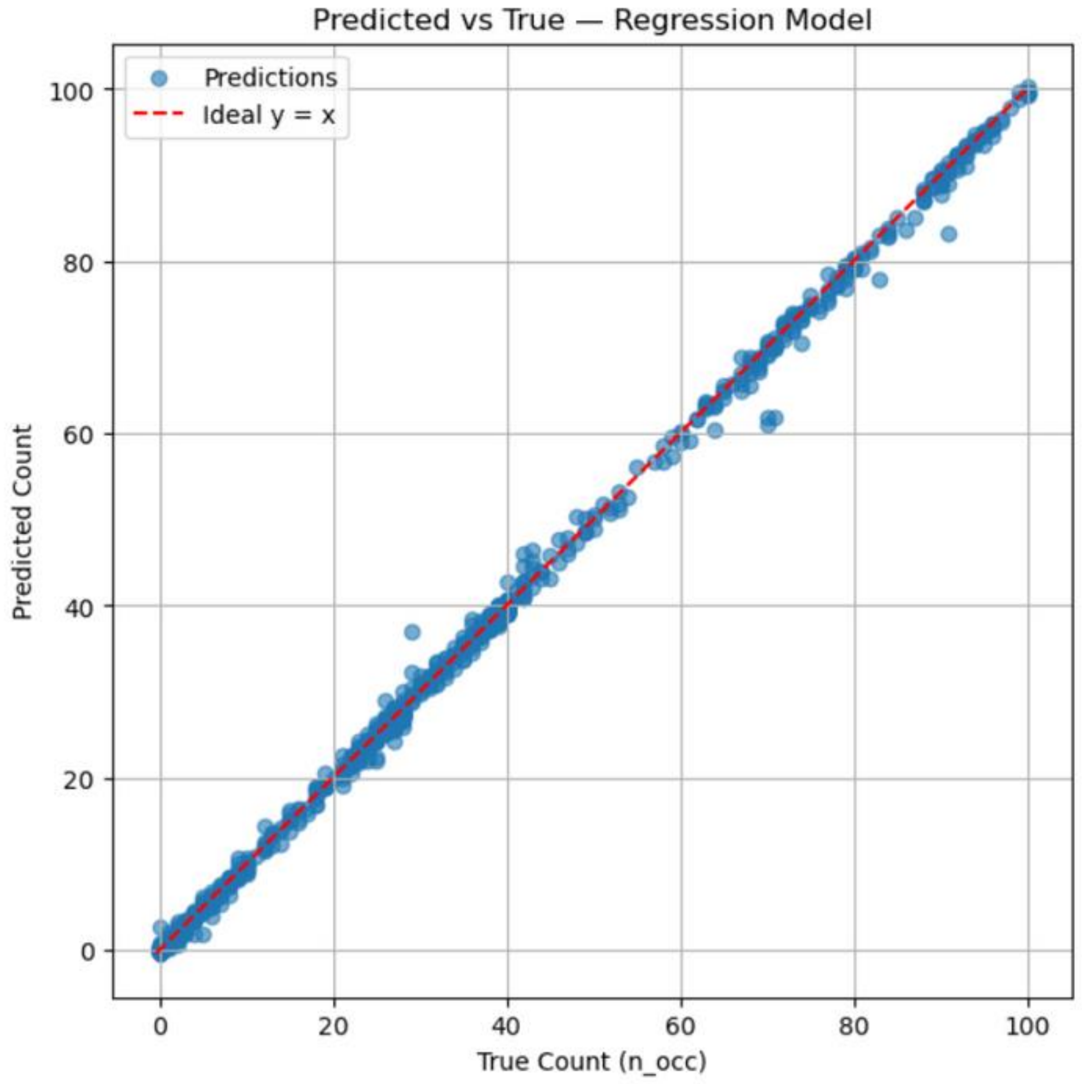
- Early stopping based on validation loss with fixed patience = 8.
- The model achieving the minimum validation loss is saved and restored
- Max_epochs = 30

TEST EVALUATION

Metrics used

We computed different absolute measures and a relative one, the ROE:

```
[TEST RESULTS]
 Skipped images (n_total=0): 2
 MSE  = 0.8203
 RMSE = 0.9057
 MAE  = 0.4933
 ROE  = 0.0096
```

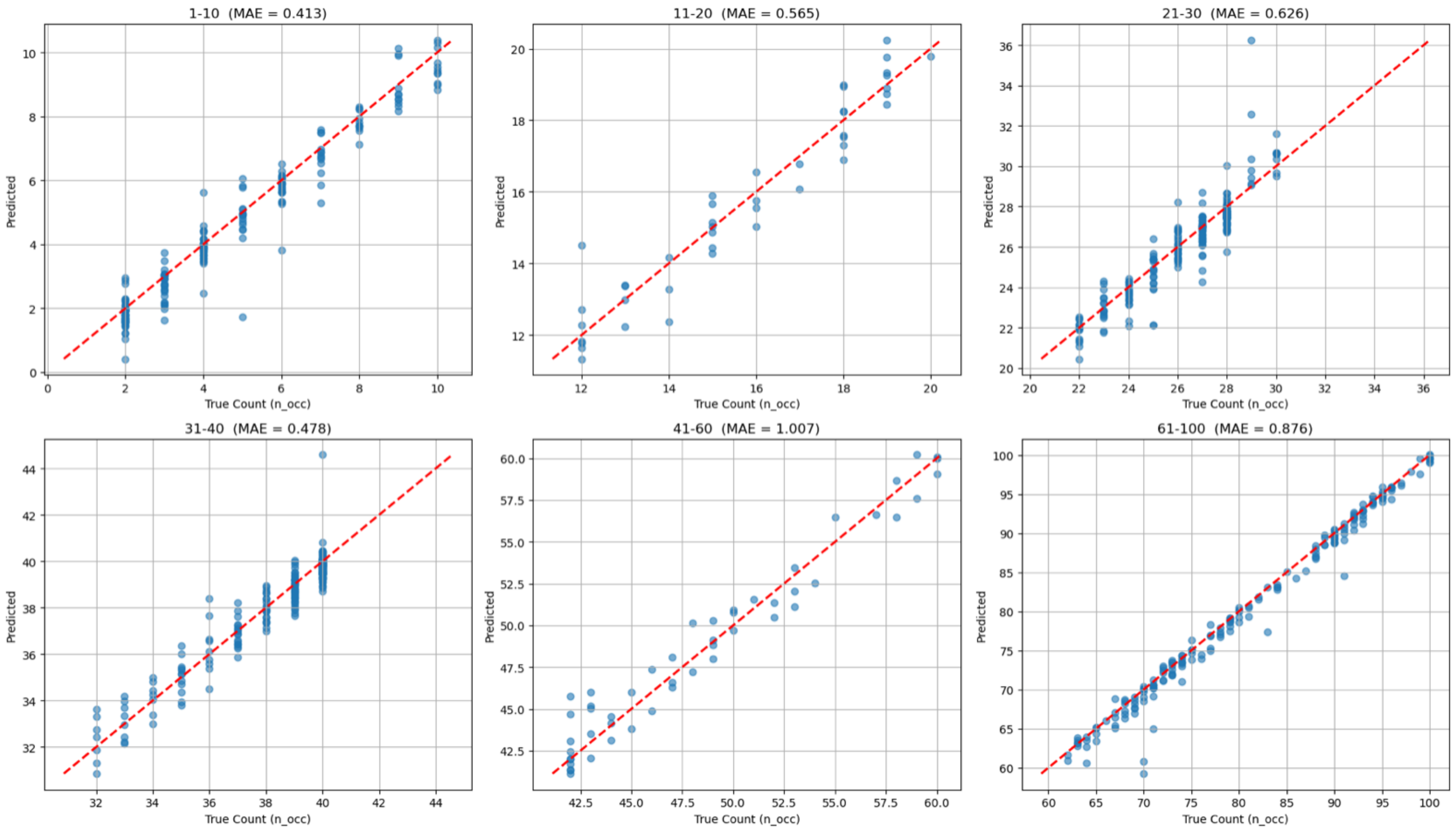


MAE's ranges

- Dataset with highly variable occupancy levels
- MAE evaluated across different occupancy ranges to assess performance robustness

```
=== MAE for bins of n_occ ===

Bin     0 ( 0 –   0) | Example:  251 | MAE = 0.123
Bin  1–10 ( 1 –  10) | Example:  251 | MAE = 0.359
Bin 11–20 (11 –  20) | Example:   44 | MAE = 0.545
Bin 21–30 (21 –  30) | Example:  221 | MAE = 0.609
Bin 31–40 (31 –  40) | Example:  235 | MAE = 0.463
Bin 41–60 (41 –  60) | Example:   52 | MAE = 0.959
Bin 61–100 (61 – 100) | Example:  188 | MAE = 0.855
```



SECOND MODEL: CLASSIFICATION

Loss function: Binary Cross-Entropy

TRAINING INITIALIZATION

Backbone reuse

- The classification model reuses the backbone trained for regression.
- The regression head is replaced by a new binary classification head.

Frozen backbone

- Backbone parameters are frozen.
- Only the classification head is trained.

Optimizer and learning rate

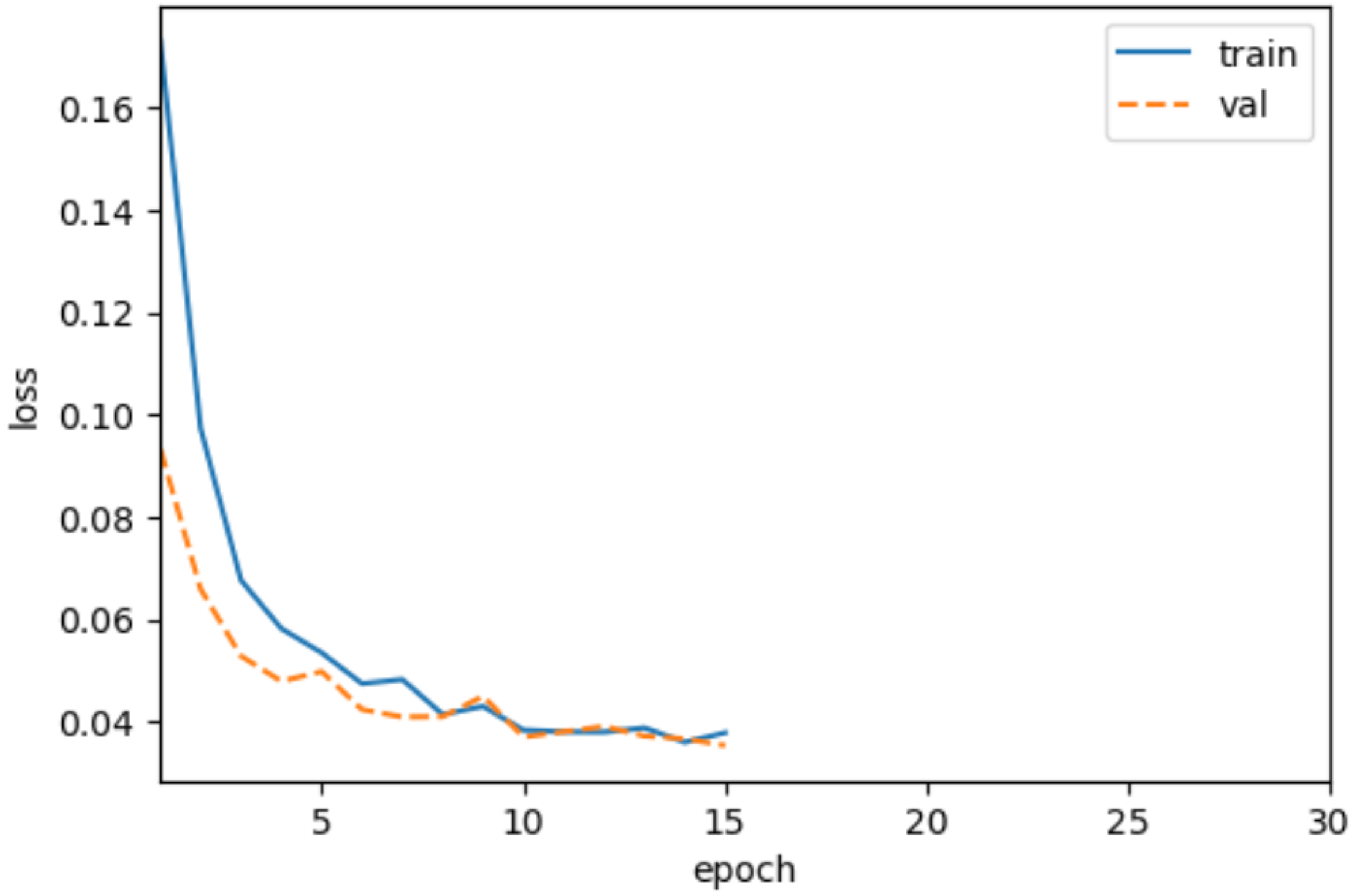
- Adam optimizer with cosine annealing schedule.
- No warm-up phase is used.

Warm-up

- Warm-up was tested but increased loss variability.

Model selection

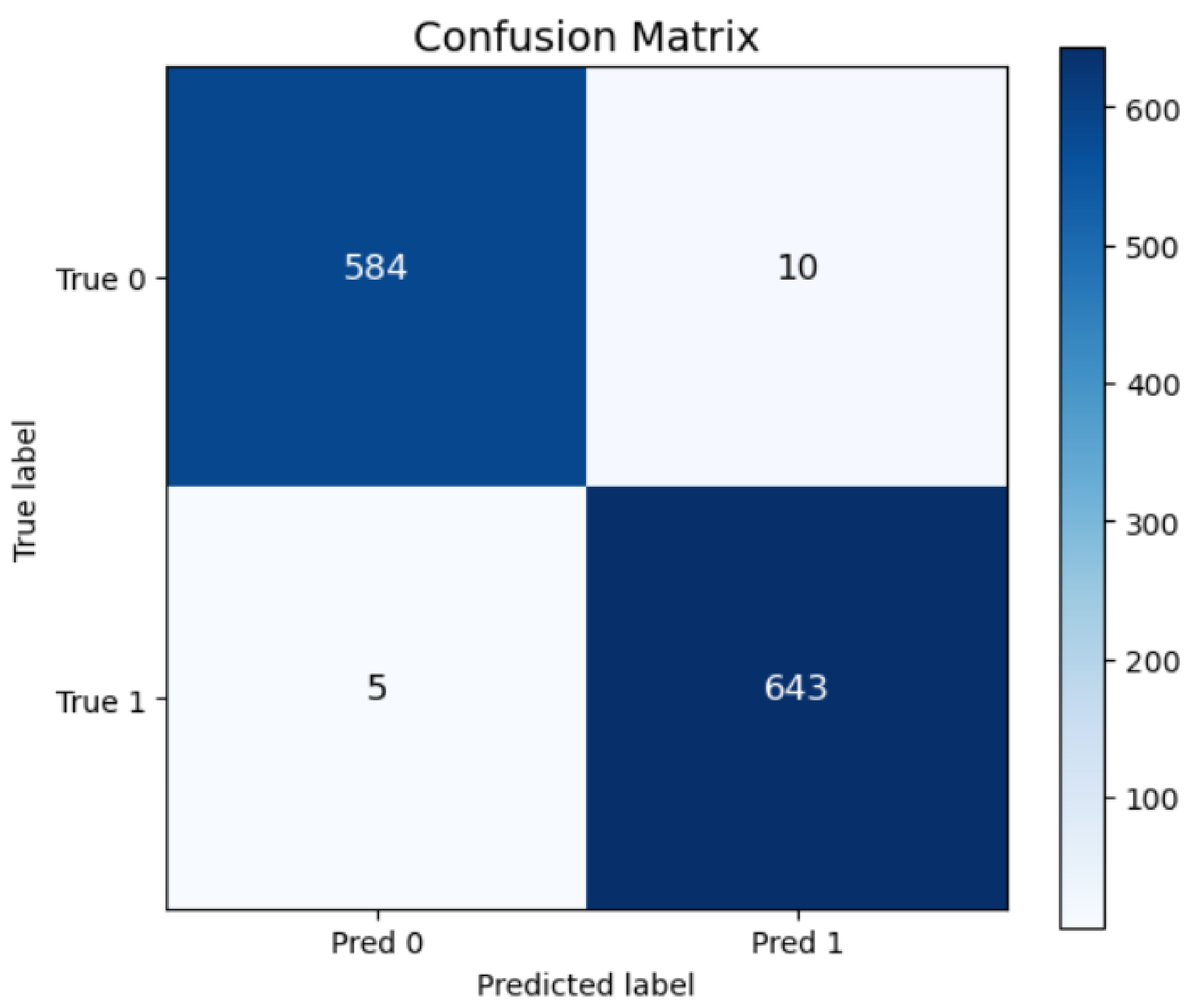
- Early stopping based on validation accuracy.

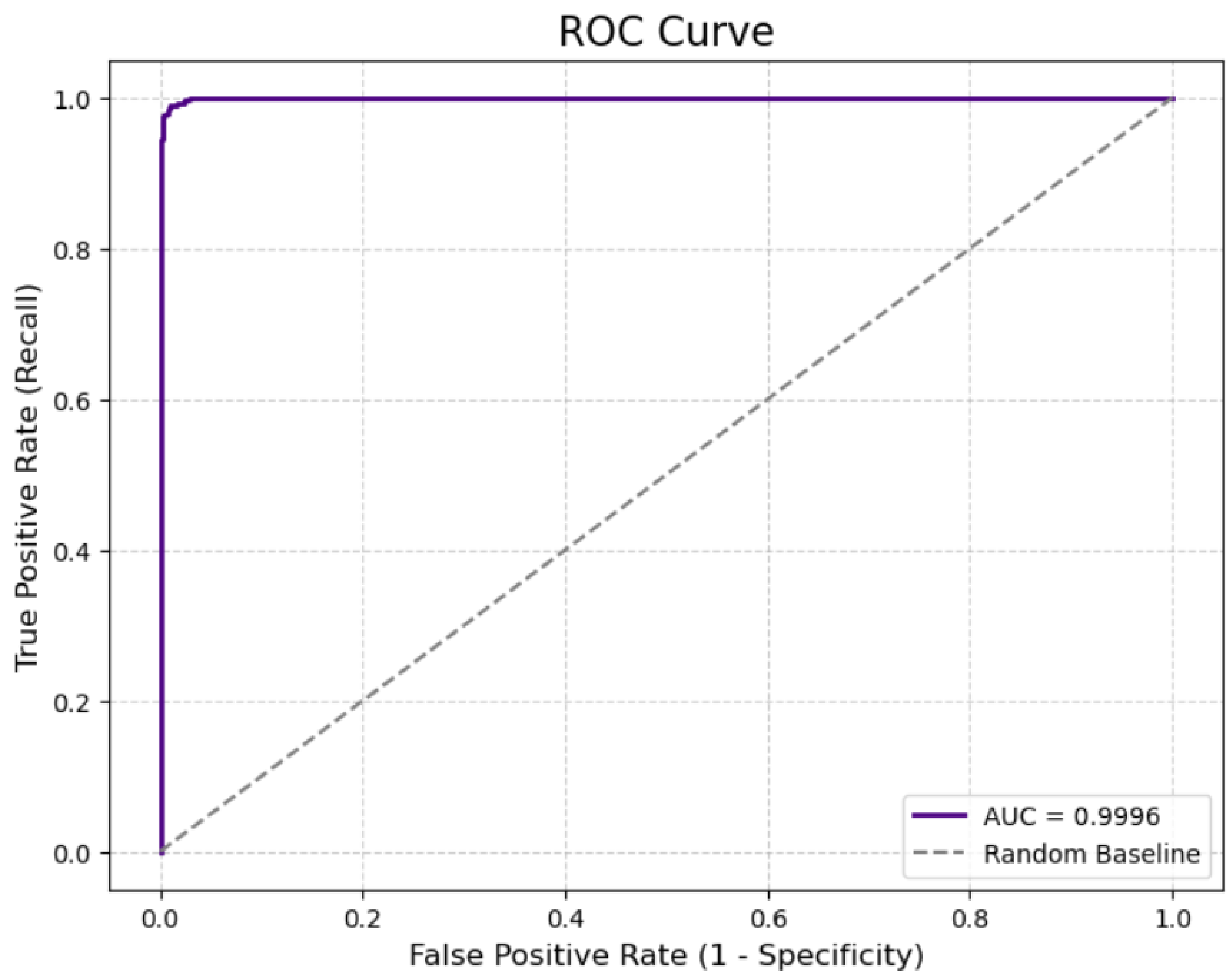



```
[CLS] Epoch 15/30 | Train Loss: 0.0379 | Val Loss: 0.0354 | Acc: 0.9863 | Prec: 0.9877 | Recall: 0.9862 | F1: 0.9869
Early stopping triggered at epoch 15.

Best model saved at: checkpoints/best_cls_model.pth | Best Accuracy = 0.9875
```

TEST EVALUATION

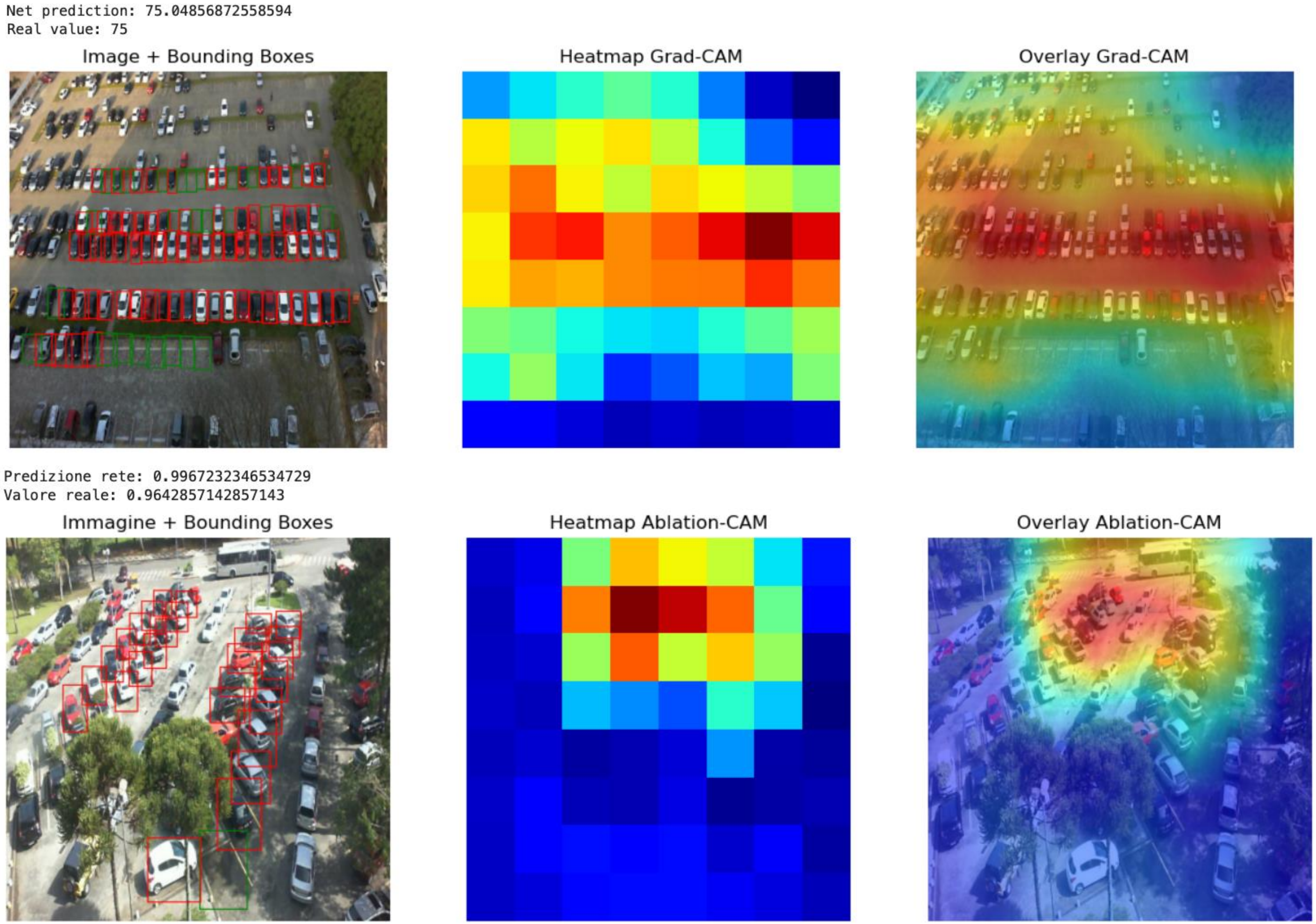





```
AUC: 0.9995921145612503
```

```
[CLASSIFICATION TEST RESULTS]
Loss BCE:   0.0330
Accuracy:   0.9879
Precision:  0.9847
Recall:     0.9923
F1-score:   0.9885
NIR:        0.5217
```

GRAD-CAM AND ABLATION-CAM



CONCLUSION

Overall, the proposed approach achieves **strong performance**, as confirmed by the metric evaluation on the test set.

This is likely due to the controlled nature of the dataset, which includes only a **limited variety of parking layouts**, allowing the model to learn consistent visual patterns effectively.

Further analysis could be performed by testing the model on parking images from **different locations** to assess generalization.

THANK YOU!