Object classification in a domain shift problem

Gaetano De Angelis[1] deangelisgaetano99@email.com
Giuseppe Leonardi[2] leonardigiuseppej@gmail.com

1 Introduction

The problem addressed in this project is the classification of images into eight categories, given a dataset composed of 1600 training samples and 800 test samples. A key challenge lies in the presence of a *domain shift*: while the training images exhibit heterogeneous backgrounds, the test set was acquired under a uniform background. This discrepancy makes it difficult for a model trained naively on the training distribution to generalize effectively.

The dataset also provides bounding box annotations for training images. These annotations allow us to reduce the influence of the background by focusing on the object of interest, mitigating the domain gap between training and test conditions. Another constraint of the competition is that pre-trained classification models cannot be used. Therefore, a custom model had to be trained entirely from scratch. Furthermore, the project requires the integration of an interpretability method to validate which image regions drive the model's predictions.

To tackle these challenges, our solution combines three main elements. First, we exploit the bounding box information during preprocessing to emphasize the relevant object regions and reduce background variability. Second, we design and train from scratch a deep convolutional architecture based on *ResNet18*, adapted for our classification task by replacing its final fully connected layer. This choice allows us to benefit from a well-established and robust residual design without relying on pre-trained weights. Finally, we integrate the Gradient-weighted Class Activation Mapping (Grad-CAM) technique, which provides visual explanations of the network's decisions, ensuring transparency and interpretability.

At a high level, this approach addresses the competition's requirements by (i) mitigating the domain shift through bounding-box-based preprocessing, (ii) employing a deep residual architecture trained entirely from scratch, and (iii) validating predictions with an interpretability method. Together, these strategies aim to achieve both strong classification accuracy and model explainability.

2 Model Description

The proposed system is composed of two main components: a detector, which estimates the position of the object within the image, and a classifier, which assigns the cropped image to one of the eight target categories. Both models share the same backbone architecture, namely a ResNet-18, adapted to their specific tasks. The detector employs ResNet-18 up to the global average pooling layer, and the resulting feature vector is passed to a fully connected layer with two output units corresponding to the normalized coordinates $(\hat{x}, \hat{y})$ of

the object center. A linear activation is used at the output, and two alternative loss functions are considered: the Mean Squared Error loss,

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)}\|_2^2,$$

and the Smooth L1 loss,

$$\mathcal{L}_{\text{SmoothL1}}(r) = \begin{cases} 0.5\, r^2 & \text{if } |r| < 1, \\ |r| - 0.5 & \text{otherwise,} \end{cases} \qquad r = \hat{y} - y.$$

The classifier also uses ResNet-18, with its final fully connected layer replaced by a layer with 8 output units producing the class logits. The optimization is performed with the standard cross-entropy loss,

$$\mathcal{L}_{\text{CE}}(\mathbf{z}, y) = -\log\left(\frac{e^{z_y}}{\sum_{j=1}^{C} e^{z_j}}\right),$$

with optional label smoothing to improve generalization and training stability. Both models are trained using the Adam optimizer with an initial learning rate of 10^{-3}, reduced to 10^{-4} in some detector runs. A ReduceLROnPlateau scheduler decreases the learning rate by a factor of 0.5 when the validation metric does not improve for three consecutive epochs, and early stopping is applied with a patience of 10 epochs to prevent overfitting. Regularization within the networks is obtained implicitly through weight decay in Adam and explicitly through data augmentation strategies applied during training, including Random Erasing as an input-level regularization technique to increase robustness against background bias and overfitting.

3 Dataset

Data Source

The dataset is released by the competition organizers in a compressed archive. It contains a `train` folder with labeled images organized into class-specific subfolders, a `test` folder with 800 unlabeled images, an example `submission.csv` file, and a `train.csv` file with bounding box annotations provided in the format $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$.

Data Size

The training set consists of 1,600 images divided into eight balanced categories, while the test set contains 800 images without labels. Each training sample is associated with its ground-truth class, and bounding boxes are provided to localize the objects within the images. A major challenge arises from the domain shift: training images exhibit diverse backgrounds, whereas test images are captured under a uniform background setting.

Preprocessing

All images are resized to 224×224 pixels and normalized using ImageNet mean and standard deviation to match the requirements of pre-trained convolutional backbones. To increase

robustness and mitigate overfitting, data augmentation strategies are applied during training, including random horizontal flips, random rotations, color jittering, Gaussian blur, and random erasing. These transformations reduce the risk of overfitting to background patterns and improve generalization under domain shift conditions.

1. Evaluation Metrics

The primary metric used to evaluate the model performance is **classification accuracy**. This metric measures the percentage of correctly predicted labels over the total number of samples in the test set.

2. Training Procedure

Epochs: The model was trained for a total of 30 epochs.
Batch Size: A batch size of 64 was used during training.
Validation Strategy: The validation set was obtained by splitting the training dataset into training and validation subsets with a (20/80)% ratio. This ensured that the model's generalization ability could be monitored during training.
Stopping Criteria: Early stopping was applied based on the validation accuracy with a patience of 15 epochs to avoid overfitting. Model checkpoints were also saved whenever an improvement in validation accuracy was observed.

3. Training Details

Optimizer: The model was optimized using the *Adam* optimizer with a learning rate of $\alpha = 0.001$ and default momentum parameters ($\beta_1 = 0.9, \beta_2 = 0.999$).
Hyperparameters: The main hyperparameters include a learning rate schedule that reduces the learning rate by a factor of 0.1 after 3 epochs without improvement, a batch size of 64, and 30 total epochs. Hyperparameter tuning was performed using a grid/random search on the validation set.
Initialization: Model weights were initialized randomically using He inizialization.

Experimental Results

Training Curves

Figure 1 illustrates the evolution of the training and validation loss, as well as the accuracy, across the training epochs. The curves show the learning progress of the model and highlight how well it generalizes over time.

Performance

The final model achieved a **classification accuracy of 79%** on the test set. This result demonstrates that the model successfully generalizes to the evaluation domain despite the domain shift between training and test datasets.

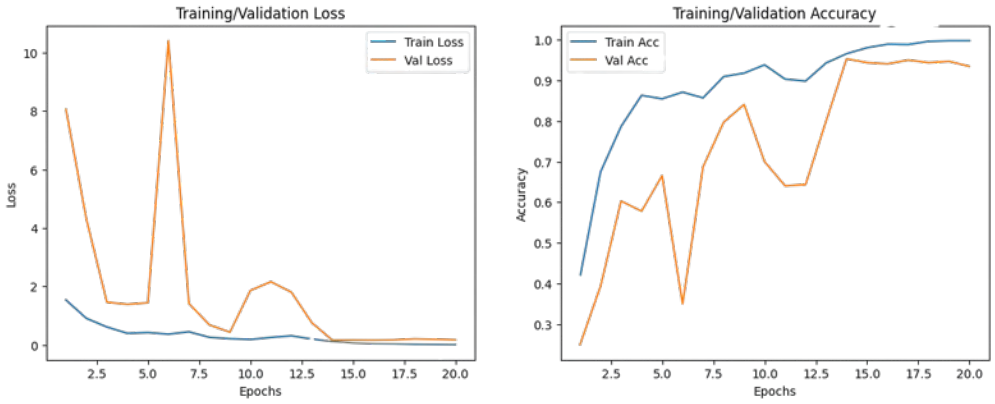

Figure 1: Training and validation loss/accuracy over epochs.