Supervised Learning - Assignment 8

May 16, 2024

Annalisa Di Pasquali
858848

Simone Vaccari
915222

Abstract

The aim of this experiment is to detect and recognize the correct values of some Uno cards using Faster RCNN as object detector. Subsequently, we will provide a visualization of the results, using Intersection over Union (IoU) as a metric to evaluate the performances.

1 Dataset and pre-processing

The dataset consists of 8,992 images of Uno cards and 26,976 labeled examples on various textured backgrounds. The total number of classes is 16: classes from 0 to 9 represent the corresponding numeric cards, from 10 to 14 the special action cards ("+4", "+2", "reverse", "skip", "wild"), and then we have an extra class for the background. The initial dataset is split into training, validation, and test set in a 70:20:10 ratio to ensure a balanced representation of data across different phases of model evaluation. In particular we have:

```
Number of training samples: 6295
Number of validation samples: 1798
Number of test samples: 899
```

Each image is subjected to some initial transformations in order to fit the model requirements. Firstly, images are resized to a common size, and then converted from BGR to RGB color format. Secondly, the pixels' values are normalized by dividing them by 255.

Subsequently, data augmentation techniques, such as random flips, blurring, and rotations, are applied in order to generate new samples and enhance the model's ability to generalize from the training data.

In the preparation phase, we develop a `CustomDataset` class in which images are loaded, their corresponding annotations are read from XML files, and the bounding boxes' coordinates are adjusted based on the resized image dimensions.

In order to confirm that the bounding boxes and the labels correctly map to the Uno cards' values, we visualize several samples, some of which are reported in Figure 1.

Figure 1: Training samples with labels and bounding boxes

2 Model setup

In order to perform our detection and classification tasks, we adopt the **Faster R-CNN** model. This type of architecture differs from the Fast R-CNN in the introduction of a Region Proposal Network (RPN) after the last convolutional layer, which is trained to produce region proposals directly, avoiding the selective search. As in the case of a standard R-CNN the concept of transfer learning is applied: in our case the model is initialized with a *ResNet50* backbone from `torchvision.models.detection`, pretrained on the ImageNet dataset.

Finally, the RoI heads of the Faster R-CNN is modified by replacing the default box predictor with a `FastRCNNPredictor` designed to work with our dataset's number of classes, and the whole model is fine-tuned.

3 Model training and validation

In the training function the model's learning process is defined by feeding the network batches of eight images and their annotations, computing losses, and updating the model parameters to minimize these losses using the Adam optimizer with a learning rate of 0.0001. This iterative process is visually tracked using the `tqdm` progress bar and the loss is recorded. During the validation phase the model's ability to generalize is tested by computing losses on unseen validation data.

After one single training epoch the results are the following:

```
EPOCH 1
Training Loss:  0.1539:
 787/787 [19:14<00:00,  1.45s/it]

Validating Loss: 0.1551: 100%
 225/225 [02:39<00:00,  1.53it/s]
Epoch #1 train loss: 0.325
```

```
Epoch #1 validation loss: 0.164
Took 21.909 minutes for epoch 0

Best validation loss: 0.16376833001772562
Saving best model for epoch: 1
```

Since the validation loss is already sufficiently low, and considering the time required to complete an epoch, we have decided to stop the training.

4 Model testing

In the testing phase, we evaluate our trained Faster R-CNN model on the test set to assess its detection accuracy and efficiency. The model performs inference on each image and the bounding boxes with scores above the detection threshold are identified. A high detection threshold of 0.8 is set, meaning only detections with a confidence score of 0.8 or higher will be considered.
For each image, the bounding boxes for detected Uno cards are drawn, and the predicted class names are annotated, providing a clear visual verification of the model's performance. Figure 2 shows some examples.

Figure 2: Testing samples with detections and predicted classes

5 Conclusions: evaluation by Intersection over Union

At the end of the testing phase we calculate **Intersection over Union (IoU)** scores to evaluate the precision of our detections. This metric quantifies the overlap between the predicted bounding box and the ground truth bounding box by computing the ratio between the area of overlap and the area of union. Thus, for each processed image, we not only predict the locations and types of Uno cards but also match these predictions against the ground truths to compute IoUs. This step is crucial for understanding the model's ability to accurately localize and identify cards.

We ensure a fair comparison by matching predicted and actual boxes based on both label consistency and detection confidence. This approach allows us to focus on high-confidence detections, filtering out less reliable ones. The highest IoU for each predicted box is recorded, along with corresponding image and box details.

In the final stage of our evaluation, we display some images with the highest and lowest IoU scores, displaying the predicted bounding boxes in red and the actual bounding boxes in green.
In Figure 3 we report some images with the corresponding value of IoU score:

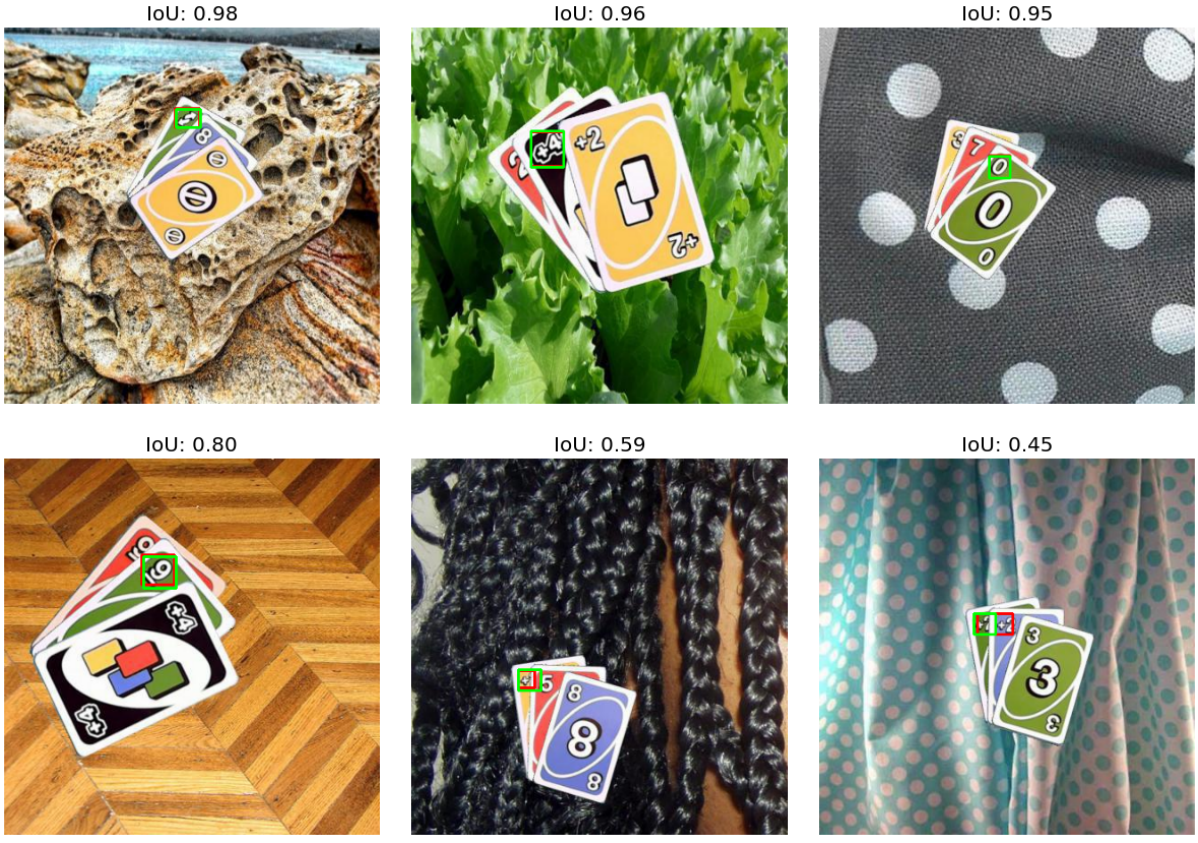

Figure 3: Predicted vs ground truth bounding boxes in test images with corresponding value of IoU

The images with the lowest IoU scores provide valuable insights into the conditions under which the model struggled. Several factors may contribute to these difficulties, for example complex backgrounds:

for some images the backgrounds have similar colors or patterns, making it hard for the model to distinguish edges and features accurately. Moreover, it seems like the model has some issues in detecting cards that are partially obscured by other objects or overlapping with other cards, as it may not have enough visible features to make a confident detection.