

Homework 2

Ping-Yeh Chou 113550901

1 Introduction

This project tackles the problem of digit recognition in images through a detection-based approach using Faster R-CNN [Ren *et al.*, 2015]. The task involves two key objectives: localizing and classifying individual digits within an image, and subsequently combining these detections to recognize complete multi-digit numbers. The proposed method employs Faster R-CNN as its foundation due to its strong performance in object detection tasks, modifying the architecture to specifically handle digit recognition. The model first identifies and classifies all digits present in an image with bounding box predictions, then processes these detections by spatial ordering to reconstruct the original numerical sequence. This approach effectively bridges object detection with sequence recognition, providing a unified solution that maintains spatial awareness while ensuring correct digit arrangement. By leveraging deep learning for both detection and recognition, my method achieves robust performance while adhering to the specified constraints of using only Faster R-CNN and avoiding external data sources. My code is available at https://github.com/cloud-zhoubingye/Faster_RCNN.

2 Method

2.1 Data Pre-processing

The data pre-processing pipeline employs color-based transformations to enhance model robustness while deliberately avoiding spatial transformations that could distort bounding box annotations. For training data, we apply a series of stochastic color augmentations including brightness, contrast, saturation, and hue adjustments with a maximum variation of 0.2, along with probabilistic grayscale conversion ($p=0.1$), Gaussian blur ($\sigma \sim [0.1, 2.0], p = 0.1$), and specialized image manipulations such as random equalization, posterization to 4 bits, auto-contrast adjustment, and inversion - each applied with 10% probability [Fedorova and others, 1968; Chiang *et al.*, 2014]. Validation and test sets undergo minimal processing, being converted to tensors and normalized to $[0,1]$ range without any augmentations to maintain evaluation integrity.

2.2 Model Architecture

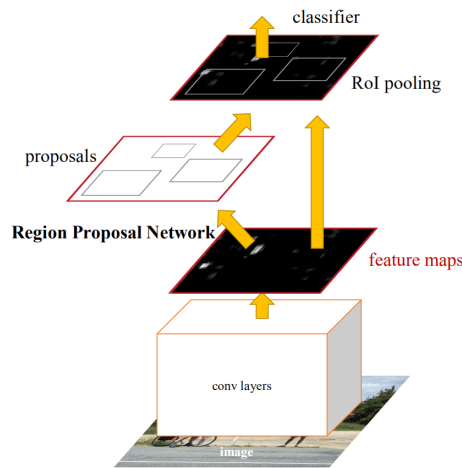


Figure 1: Illustration of the proposed model architecture, highlighting the integration of Faster R-CNN, ResNet-50, and FPN for end-to-end digit detection and recognition.

The proposed model architecture builds upon the Faster R-CNN framework, incorporating a ResNet-50 backbone [He *et al.*, 2016] paired with a Feature Pyramid Network (FPN) [Lin *et al.*, 2017]. This model seamlessly integrates feature extraction, region proposal generation, and object classification within a unified deep learning pipeline. Our overall structure is illustrated in Figure 1 [Ren *et al.*, 2015].

The backbone processes input images through residual blocks, generating hierarchical feature maps that combine low-level visual details with high-level semantic information, crucial for digit recognition. The FPN, serving as the neck of the architecture, constructs a top-down pathway with lateral connections to merge semantically rich deep features from higher layers with spatially precise shallow features from lower layers [Jin *et al.*, 2022]. This fusion results in a pyramid of feature maps (P2 to P6, each with 256 channels), with P6 created via stride-2 max-pooling of P5 for larger object handling. The design ensures

Category	Parameter	Value	Description
Training	Batch Size	4	Training samples per iteration
	Num Epochs	14	Total training epochs
	Val Epoch List	[1-14]	Epochs for validation
	Num Workers	4 (Linux)/0 (Windows)	Data loading threads
	Checkpoint Path	./checkpoints	Model save directory
	Device	CUDA/CPU	Training hardware
	Verbose	False	Print detailed logs
	Optimizer	AdamW	With weight decay
	Learning Rate	5×10^{-4}	Initial learning rate
Model	Weight Decay	5×10^{-4}	L2 regularization
	Backbone	ResNet50-FPN	Feature extractor
	Trainable Layers	3	Fine-tuned backbone layers
	Num Classes	11	10 digits + background
Detection	Pretrained Weights	ImageNet	Backbone initialization
	IoU Threshold	0.5	NMS criterion
	Score Threshold	0.55	Prediction filtering
	Input Resolution	Variable	Original image size

Table 1: Hyperparameter configurations for the Faster R-CNN digit detection model.

high spatial resolution for small digits and strong semantic precision for larger ones, enhancing robustness across varying digit sizes.

The Region Proposal Network (RPN) operates on these FPN outputs, generating candidate object regions through anchor boxes at multiple scales and aspect ratios [Qing *et al.*, 2024]. Each spatial location accommodates 9 anchors (3 scales \times 3 aspect ratios), processed by shared 3×3 convolution followed by parallel 1×1 branches for objectness classification and bounding box regression. This mechanism efficiently narrows down potential digit locations, outputting approximately 2,000 top-ranked proposals per image after applying non-maximum suppression.

The detection head employs RoI alignment to pool features from the FPN outputs based on proposal sizes [Ding *et al.*, 2019]. Using a scale-aware formula, it selects the appropriate feature level for each region, then extracts fixed-size features (7×7 spatial dimensions). These pooled features pass through two fully connected layers before splitting into classification and regression branches. The classification branch, equipped with a modified FastRCNNPredictor, outputs probabilities for 10 digit classes plus background via a softmax layer. Meanwhile, the regression branch adjusts bounding box coordinates with class-specific refinements.

Among various architectures, this design was chosen for its ability to integrate detection and recognition within a unified framework while offering flexibility for domain-specific adaptations. Its strengths include robust multi-scale feature representation, precise proposal generation, and effective handling of spatial coherence for digit sequences. However, limitations arise from computational complexity and reliance on extensive training data, which may affect scalability for resource-constrained applications. Despite these trade-offs, the architecture strikes a balance between accuracy, efficiency, and adaptability to digit recognition tasks [He *et al.*, 2017].

Additionally, for task 2, we refine the final prediction results using a systematic post-processing approach based on the outputs of Task 1. First, low-confidence predictions are filtered out, ensuring only reliable candidates are considered. Next, Non-Maximum Suppression (NMS) is applied to resolve overlapping detections and eliminate redundant bounding boxes. This step is crucial for isolating individual digits accurately within complex arrangements. Finally, the surviving predictions are sorted along the horizontal axis to establish their left-to-right ordering, which mimics the natural reading sequence. This sequential refinement process ensures both spatial coherence and interpretive accuracy, resulting in robust predictions tailored to the requirements of the task.

2.3 Hyperparameter

We have detailed the hyperparameter settings used in the project in Table 1.

3 Result

The experimental results demonstrate the effectiveness of our Faster R-CNN-based digit detection model, achieving competitive performance while maintaining computational efficiency. The model was trained for 14 epochs on single NVIDIA P100 GPU, with a total training time of 23 hours utilizing mixed precision training (combining float16 and float32 operations), which significantly reduced memory consumption to approximately 10GB while maintaining numerical stability.



Figure 2: Convergence of the training and validation loss curve.


	113550901	1	2025-04-08 11:11	261514	113550901	0.39	0.81
---	-----------	---	------------------	--------	-----------	------	------

Figure 3: Scores of task1 and task2.

The training loss curve in Figure 2 exhibited a steady and rapid convergence pattern, indicating effective optimization of the multi-task loss function that combines classification cross-entropy and smooth L1 regression losses. This convergence behavior can be attributed to several factors: the AdamW optimizer’s adaptive learning rate mechanism with weight decay (5×10^{-4}) effectively regularized the model, while the cosine learning rate schedule (initialized at 5×10^{-4} and annealed to 8×10^{-6}) enabled stable fine-tuning of the pretrained ResNet50-FPN backbone. What’s more, as observed in the figure, both the training and validation losses exhibit slight fluctuations, attributed to the cosine annealing learning rate scheduler, which aids the model in escaping local optima.

For task evaluation in Figure 3, the model achieved a mean Average Precision (mAP) of 0.39 in the object detection task (task1), which reflects the challenging nature of digit localization in varying scales and orientations. The moderate mAP score can be explained by several architectural characteristics: while the Feature Pyramid Network (FPN) enabled multi-scale feature extraction, the limited diversity in anchor box configurations (3 scales \times 3 aspect ratios) may have constrained the model’s ability to precisely localize digits with extreme aspect ratios. Additionally, the absence of spatial augmentations (to preserve bounding box integrity) potentially reduced the model’s robustness to geometric variations.

The model demonstrated strong performance in the digit recognition task (task2), achieving 0.81 accuracy. This superior performance stems from our optimized post-processing pipeline, which combines non-maximum suppression (IoU threshold=0.5) with confidence score filtering (threshold=0.3) and spatial ordering of predictions by horizontal coordinates. This three-stage refinement effectively eliminates redundant detections while preserving valid digit instances and maintaining proper sequence alignment.

4 Additional Experiments

To further investigate the performance characteristics of our digit detection model, we conducted two complementary experiments that provide insights into architectural choices and post-processing optimization.

The first experiment in Table 2 explored the impact of backbone architecture selection by comparing ResNet-50-FPN against MobileNetV3-Large-320-FPN configurations [Huang *et al.*, 2017]. Our hypothesis posited that the deeper residual connections and feature pyramid integration in ResNet-50 would yield superior performance compared to the lightweight MobileNetV3 architecture, particularly for small digit detection where multi-scale feature representation is crucial. This expectation was based on ResNet’s proven capacity to preserve spatial information through skip connections and its ability to construct rich hierarchical features through stacked residual blocks. The experimental results confirmed this hypothesis, with ResNet-50-FPN achieving significantly higher metrics (0.39 mAP and 0.81 recognition accuracy) compared to MobileNetV3 (0.33 mAP

Category	Model	mAP	Accuracy
Performance	ResNet-50-FPN	0.39	0.81
	MobileNetV3	0.33	0.72

Table 2: Comparison of performance metrics between ResNet-50-FPN and MobileNetV3.

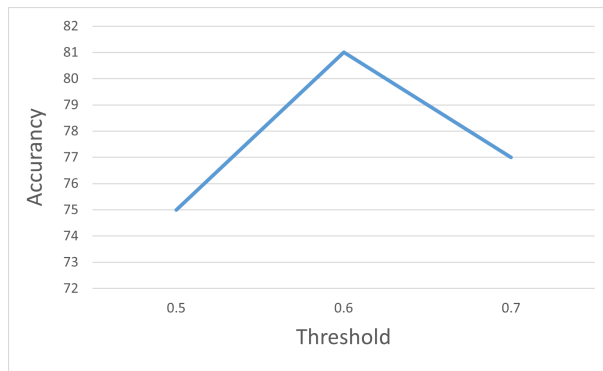


Figure 4: Analysis of confidence threshold sensitivity.

and 0.72 accuracy). ResNet-50’s larger receptive fields and deeper feature hierarchies better capture contextual relationships between adjacent digits, while its FPN implementation enables more effective handling of scale variation through top-down feature fusion. MobileNetV3’s depthwise separable convolutions, while computationally efficient, may lose subtle digit features critical for both localization and classification, particularly in low-contrast scenarios. The accuracy difference in task2 (0.81 vs 0.72) specifically suggests that MobileNetV3’s compressed feature space struggles to maintain discriminative power for similar-looking digits (e.g., ‘3’ vs ‘8’), whereas ResNet’s richer features provide more robust embeddings for final classification.

The second experiment systematically evaluated the impact of confidence threshold selection on task2 performance, revealing a complex relationship between threshold strictness and recognition accuracy [Lee *et al.*, 2017]. We hypothesized that an optimal threshold exists that balances false positive suppression with true positive retention, and that this balance would manifest as a peak in accuracy at intermediate threshold values. In Figure 4, testing thresholds of 0.5, 0.6, and 0.7 yields accuracies of 0.75, 0.81, and 0.77 respectively. The improvement from 0.5 to 0.6 threshold indicates that the model generates numerous low-confidence false positives (likely background noise or partial digit fragments) that are effectively filtered at the higher threshold. However, the subsequent accuracy drop at 0.7 threshold suggests the model’s confidence calibration has limitations - some true digit detections fall below this stringent cutoff, particularly for challenging cases like occluded or blurred digits. The threshold sensitivity analysis implies that our post-processing pipeline would benefit from adaptive thresholding techniques or additional confidence calibration in the future, particularly for deployment scenarios requiring robust operation across diverse image quality conditions.

References

- [Chiang *et al.*, 2014] Jen-Shiun Chiang, Chih-Hsien Hsia, Hao-Wei Peng, Chun-Hung Lien, et al. Color image enhancement with saturation adjustment method. *Journal of Applied Science and Engineering*, 17(4):341–352, 2014.
- [Ding *et al.*, 2019] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2849–2858, 2019.
- [Fedorova and others, 1968] VI Fedorova et al. Changes in hue, saturation and brightness of spectral stimuli as a result of chromatic adaptation. *Optometry and Vision Science*, 45(9):595–604, 1968.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [Huang *et al.*, 2017] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.
- [Jin *et al.*, 2022] Zhenchao Jin, Dongdong Yu, Luchuan Song, Zehuan Yuan, and Lequan Yu. You should look at all objects. In *European conference on computer vision*, pages 332–349. Springer, 2022.
- [Lee *et al.*, 2017] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.
- [Lin *et al.*, 2017] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

- [Qing *et al.*, 2024] Chen Qing, Tong Xiao, Shuzhuang Zhang, and Peng Li. Region proposal networks (rpn) enhanced slicing for improved multi-scale object detection. In *2024 7th International Conference on Communication Engineering and Technology (ICCET)*, pages 66–70. IEEE, 2024.
- [Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.