Fine-Grained Feature Imitation for Efficient Object Detection Using Knowledge Distillation

1st Md Al Amin
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
ID: 1811904042

Dr. Nabeel Mohammed (NbM)
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
nabeel.mohammed@northsouth.edu

***Abstract*—State-of-the-art object detection models are powerful but have too many parameters, making them hard to use on low-end devices. Knowledge distillation is a way to solve this problem by letting a smaller ”student” model learn from a larger ”teacher” model. While this method works well for tasks like classification, it is not as effective for more complex tasks like object detection. To improve this, I used a fine-grained feature imitation approach. This focuses on areas near objects, as these regions carry important information about how the teacher model generalizes. My project tested this idea using datasets like BCCD, Lemon Disease, and Incorrect0mask-2, with a main focus on the Pascal VOC dataset. Results showed that my custom student model, with only 1.78M parameters, achieved better performance (mAP@50 of 0.707 and mAP@50-95 of 0.435) compared to the teacher model (6.2M parameters, mAP@50 of 0.644, and mAP@50-95 of 0.385). This shows that my method can improve performance while using a smaller and more efficient model.**

I. Introduction

Object detection has significantly advanced with the development of deep convolutional neural networks (CNNs). However, deploying state-of-the-art (SOTA) detectors, which are often computationally intensive and large in size, on resource-constrained devices remains challenging. To address these challenges, prior works have explored approaches like quantization [6, 22, 17] and network pruning [7, 8], which focus on reducing model size and computational cost. However, these methods often require specialized hardware or software to achieve practical deployment.

A promising approach to build compact yet efficient models is knowledge distillation. In this paradigm, a smaller student model learns to mimic the behavior of a larger, more powerful teacher model, enhancing its generalization performance. Despite success in tasks like image classification, applying knowledge distillation to object detection remains relatively unexplored. Challenges include distilling both localization and classification knowledge and addressing the foreground-background class imbalance.

In our project, we propose a modified knowledge distillation framework for object detection, inspired by the ”Distilling Object Detectors with Fine-Grained Feature Imitation” approach. Instead of relying on vanilla distillation, our method combines fine-grained feature imitation with a custom combined imitation loss function to guide the student model in mimicking the teacher’s responses effectively.

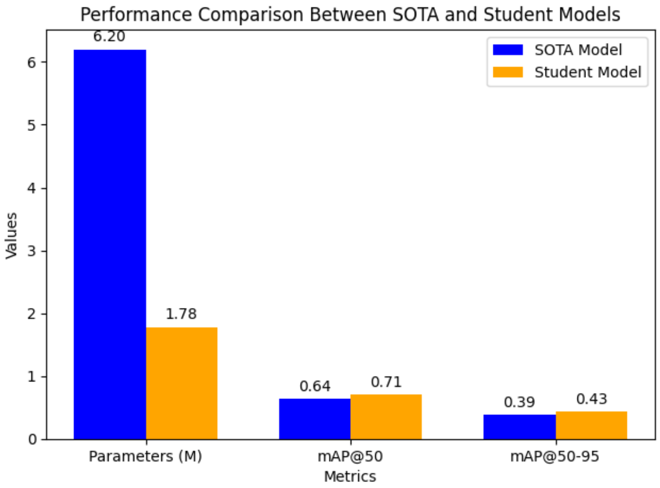

Fig. 1. Performance comparison between SOTA model and our proposed student model. The student model achieves higher mAP metrics with significantly fewer parameters.

We test our method using YOLOv5x as the teacher model and YOLOv5n as the student model, focusing primarily on datasets such as Pascal VOC, BCCD, Lemon Disease, and Incorrect Mask-2. The Pascal VOC dataset serves as the benchmark for performance comparison. Remarkably, our lightweight student model, with just 1.78M parameters, outperforms the teacher model (6.2M parameters) in both mAP@50 (0.707 vs. 0.644) and mAP@50-95 (0.435 vs. 0.385) while significantly reducing model size.

Figure 1 illustrates a comparison of parameters and detection performance between our student model and the SOTA teacher model, demonstrating the efficiency of the proposed method. **Key Highlights of Our Work:**

- **Compact yet Powerful Model:** The proposed student model reduces parameters by 71% compared to the teacher model while achieving superior detection accuracy.

Identify applicable funding agency here. If none, delete this.

- **Improved Detection Performance:** Significant gains in mAP metrics (+6.3% in mAP@50 and +5% in mAP@50-95) validate the effectiveness of the proposed knowledge distillation approach.
- **Generalized Approach:** Extensive evaluations on diverse datasets demonstrate the broad applicability of the proposed method for object detection tasks.
- **Simplified Deployment:** The lightweight student model enables real-time deployment on resource-constrained devices without requiring specialized hardware.

II. Literature Review

Object Detection: The evolution of object detection models has been closely tied to advancements in convolutional neural networks (CNNs). Early breakthroughs, such as R-CNN , introduced region-based approaches for detecting objects by extracting and classifying regions of interest. This framework was further enhanced in Fast R-CNN [4] and Faster R-CNN [31], improving computational efficiency and detection accuracy. One-stage detectors, such as YOLO [29, 30] and SSD [14], emerged to meet the demand for real-time applications, bypassing region proposals and directly predicting detections in a single forward pass. These models trade off slightly lower accuracy for faster inference, making them suitable for mobile and embedded systems. FPN [13] extended detection capabilities by incorporating multi-scale features, significantly improving performance on small objects. Despite these advancements, deploying such models on resource-constrained devices remains challenging.

Knowledge Distillation: Knowledge distillation, introduced by Hinton et al. [9], transfers knowledge from a larger, pre-trained ”teacher” model to a smaller ”student” model. This technique enables the student model to mimic the teacher's predictions, achieving comparable accuracy with fewer parameters. Several extensions to this concept have been proposed. Hint learning [19] transfers intermediate feature representations from teacher to student. Attention transfer improves distillation by focusing on regions of interest highlighted by attention maps. Relationship-based distillation methods [3] utilize cross-sample similarities to enhance knowledge transfer. Distribution matching approaches [10] formalize distillation as aligning teacher and student output distributions. In the context of object detection, distillation faces unique challenges, such as region proposal inconsistency between teacher and student models [5]. Full feature imitation methods [2] and region-based distillation [12] have been explored but often encounter degraded performance or limited applicability to one-stage detectors.

Model Compression and Acceleration: Efforts to accelerate deep neural networks without sacrificing accuracy have gained significant traction. Quantization [6, 8, 17] reduces computation by representing model parameters with lower precision. This approach often requires specialized hardware for efficient inference. Weight pruning [7, 8, 16] eliminates redundant connections to reduce model size, but higher pruning ratios can severely degrade performance. Low-rank approximations [21] decompose large layers into smaller components, achieving theoretical reductions in computation but often falling short of practical speedups. Channel pruning methods [11, 15, 1] remove entire feature maps, offering better efficiency for dense computations. However, most compression techniques either rely on hardware-specific optimizations or struggle to balance model size and accuracy effectively.

Key Insights for Knowledge Distillation in Object Detection: While substantial progress has been made in object detection, knowledge distillation, and model compression, gaps remain in effectively combining these approaches for resource-constrained settings. Existing distillation techniques often fail to generalize to object detection tasks due to the unique requirements of localization and foreground-background imbalances. This paper addresses these limitations by proposing a fine-grained feature imitation method that leverages inter-location discrepancies to distill localization and classification knowledge more effectively.

III. Method

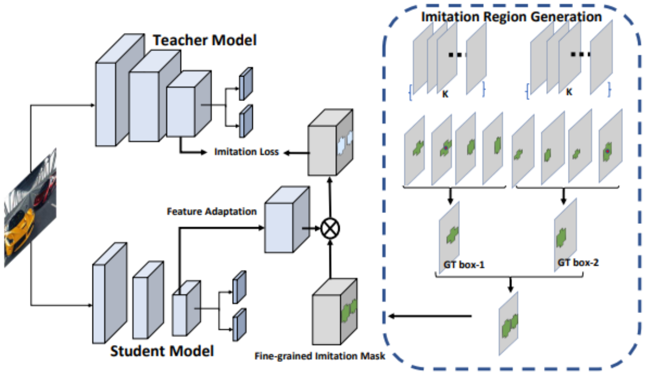

Fig. 2. Illustration of the proposed fine-grained feature imitation method

In this work, we develop a Combined Imitation Loss Function (CILF) to enhance lightweight object detectors by leveraging feature and probabilistic knowledge distillation. This approach builds on the teacher-student paradigm, with additional mechanisms for fine-grained imitation and region-specific masking, and is implemented in the context of YOLOv5. Our primary modification replaces the traditional mean squared error (MSE) loss with the proposed combined loss, allowing for richer knowledge transfer.

Architecture Overview

The architecture of our proposed method is illustrated in Figure 1. It consists of the following components:

1) **Teacher Model**: A computationally intensive, pre-trained **(YOLOv5x)** detector used to provide high-level feature maps and outputs for supervision.

2) **Student Model:** A lightweight YOLOv5n detector that learns both ground-truth object detection and fine-grained imitation of the teacher's outputs.
3) **Fine-grained Imitation Mask:** Guides the distillation process by focusing on regions of high importance in the teacher's feature maps, ensuring efficient knowledge transfer.
4) **Feature Adaptation Layer:** Aligns the feature maps of the student model with the teacher to ensure compatibility during the loss computation.
5) **Combined Imitation Loss Function (CILF):** A hybrid loss function combining KL divergence and mean squared error (MSE) for both probabilistic and feature-level alignment.

The student model is trained using both detection loss and the proposed imitation loss, enhancing its performance while maintaining efficiency.

Imitation Region Generation

To distill knowledge effectively, we focus the imitation process on specific regions of interest in the feature map using a fine-grained imitation mask I. This mask is generated as follows:

- **IoU Map Calculation:** Compute the IoU between each ground truth bounding box and all anchor priors to generate a $W \times H \times K$ map m, where W and H are the feature map dimensions, and K is the number of anchors.
- **Thresholding:** The maximum IoU value M is scaled by a factor ψ (default $\psi = 0.5$) to generate a threshold $F = \psi \cdot M$. Locations with IoU $\geq F$ are retained in the mask.
- **Mask Combination:** Masks from all ground-truth boxes are combined using a logical OR operation to generate the final mask I.

This process focuses the imitation loss computation on regions most critical for object detection.

Combined Imitation Loss Function (CILF)

- The Combined Imitation Loss Function integrates two components for effective knowledge transfer:
  - **KL Divergence Loss (L_{KL}):** Matches the probabilistic outputs of the teacher and student by minimizing the divergence between their softmax distributions.
  - **MSE Loss (L_{MSE}):** Aligns the teacher and student feature maps spatially and channel-wise at regions specified by the mask I.
- The combined loss is formulated as:

$$L_{\mathrm{CILF}} = \alpha L_{\mathrm{KL}} + (1-\alpha) L_{\mathrm{MSE}},$$

where α (default $\alpha = 0.5$) balances the contributions of the two loss components.

The implementation of the loss function is as follows:

```
def combined_imitation_loss(teacher, student,
    mask, temperature=3, alpha=0.5):
    """
    Combined KL divergence and MSE loss for
        imitation.
    """
    # KL Divergence
    teacher_soft = F.softmax(teacher /
        temperature, dim=-1)
    student_soft = F.log_softmax(student /
        temperature, dim=-1)
    kl_div = F.kl_div(student_soft,
        teacher_soft, reduction='none')
    kl_loss = (kl_div * mask).sum() / mask.sum
        ()

    # MSE Loss
    mse_loss = torch.pow(teacher - student, 2)
         * mask
    mse_loss = mse_loss.sum() / mask.sum()

    # Combine losses
    combined_loss = alpha * kl_loss + (1 -
        alpha) * mse_loss
    return combined_loss
```

Listing 1. Combined Imitation Loss Function Code

This loss focuses on regions specified by the mask and integrates temperature scaling in the KL divergence for smoother distributions.

Training Framework

- The training pipeline integrates the Combined Imitation Loss Function into a standard object detection framework. The forward pass of the student model includes:
  - **Feature Extraction:** The teacher model generates feature maps and outputs, while the student produces its own corresponding features.
  - **Loss Computation:**
    - The detection loss L_{det} and the imitation loss L_{CILF} are computed.
    - The imitation loss uses the teacher's feature maps and outputs as references, while the fine-grained imitation mask I ensures computation is focused on relevant areas.
- The total loss for training the student model is:

$$L_{\mathrm{total}} = L_{\mathrm{det}} + \lambda L_{\mathrm{CILF}},$$

where λ is a hyperparameter to control the weight of the imitation loss.

Implementation Details:

- **Framework:** Experiments are conducted using PyTorch.
- **Model Pair:** The teacher model is a computationally heavy detector, while the student is YOLOv5.
- **Hyperparameters:** Defaults include:
  - $T = 3$
  - $\alpha = 0.5$
  - $\psi = 0.5$
  - $\lambda = 0.01$
- **Datasets:** Evaluated on PASCAL VOC datasets.

IV. Experiment

A. *Experiment Setup*

We evaluated the proposed method using the Pascal VOC 2007 and 2012 datasets, which are official and widely used benchmarks for object detection. These datasets consist of 20 object categories, with comprehensive annotations for each instance. Pre-processing steps included resizing images to 640×640 and applying data augmentation techniques, such as horizontal flipping and scaling.

Hardware and Software Configuration
The experiments were conducted on a system with the following specifications:

- **GPU:** NVIDIA RTX 4050 (6GB VRAM)
- RAM: 16GB
- **Processor:** Intel i5 12th Gen
- **Frameworks:** PyTorch 2.1 and Python 3.12

Hyperparameter
The training process was guided by the following hyperparameters:

TABLE I
HYPERPARAMETERS AND THEIR VALUES

Hyperparameter	Value
Learning Rate (lr_0)	0.01
Final LR Factor (lrf)	0.01
Momentum	0.937
Weight Decay	0.0005
Warmup Epochs	3.0
Warmup Momentum	0.8
Warmup Bias LR	0.1
Batch Size	8
Input Image Size	640×640
Optimizer	SGD

Additional augmentation parameters included probabilities for flipping (`fliplr=0.5`), scaling (`scale=0.5`), and HSV color transformations (`hsv_h=0.015`, `hsv_s=0.7`, `hsv_v=0.4`).

Teacher and Student Models:
We employed the YOLOv5x model as the teacher and two variants of YOLOv5n as student models, detailed in Table II below:

TABLE II
DETAILS OF TEACHER AND STUDENT MODELS

Model Type	Model Name	Trainable Parameters
Teacher Model	YOLOv5x	86.7M
Our-Student Model	YOLOv5n	1.78M
Our-Student Model	YOLOv5n Custom	0.58M

B. Methodology

Knowledge Distillation Framework:
The student models were trained using a fine-grained feature imitation approach. The teacher model generated feature responses and fine-grained imitation masks based on anchor locations near object instances. These masks guided the student model to mimic the teacher's feature response in specific regions. A custom Combined Imitation Loss function was employed, integrating KL Divergence and Mean Squared Error (MSE) to better distill knowledge from the teacher to the student.

Training Procedure:
The training process consisted of 200 epochs, with the teacher model initialized using pre-trained YOLOv5x weights. The following workflow was followed:

1) Forward pass through both teacher and student models to generate predictions and features.
2) Generation of imitation masks by the teacher model.
3) Computation of the Combined Imitation Loss function to distill knowledge.
4) Optimization of the student models using SGD.

Evaluation Metrics:
The following standard object detection metrics were utilized for evaluation:

- **Mean Average Precision (mAP)**: Evaluated at thresholds $IoU = 0.5$ (`mAP@50`) and across a range of thresholds (`mAP@[0.5:0.95]`).
- **Precision and Recall**: Measured to evaluate the quality of predictions.
- **Per-Class Performance**: Metrics for each object category in Pascal VOC were analyzed.

C. Comparisons and Ablations

1) Comparison with State-of-the-Art Models: The proposed models were compared with YOLOv7-tiny, a state-of-the-art lightweight object detection model. Results are summarized in Table III:

TABLE III
COMPARISON WITH STATE-OF-THE-ART MODELS

Model	Parameters (M)	mAP@50	mAP@[0.5:0.95]
YOLOv7-tiny (SOTA)	6.2	0.644	0.385
Our-YOLOv5n (Student1)	1.78	0.708	0.435
Our-YOLOv5n Custom	0.58	0.550	0.293

2) Loss Function Ablation: Table IV highlights the performance of the student model (YOLOv5n) using different loss functions:

TABLE IV
LOSS FUNCTION ABLATION STUDY

KD Loss Function	mAP@50	mAP@[0.5:0.95]
Combined Loss ($\alpha = 0.75$)	0.704	0.435
Combined Loss ($\alpha = 0.5$)	0.709	0.435
Combined Loss ($\alpha = 0.25$)	0.705	0.428
Attention Transfer Loss	0.708	0.435
Smooth L1 Loss	0.701	0.430
Smooth MSE Loss	0.707	0.433

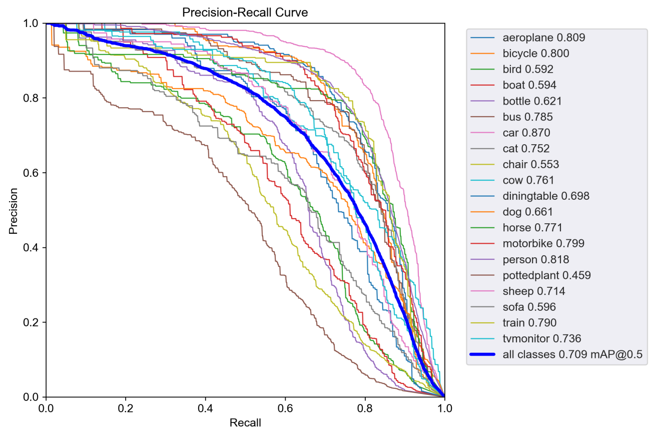

Fig. 3. the precision-recall curves for the YOLOv5n model on Pascal VOC datasets

D. Precision-Recall Curve on Pascal VOC datasets

E. Generalization to Other Datasets

The trained student model was evaluated on additional datasets to assess generalization. Results are shown in Table V:

TABLE V
GENERALIZATION RESULTS ON ADDITIONAL DATASETS

Dataset	P (Precision)	R (Recall)	mAP@50	mAP@[0.5:0.95]
Pascal VOC	0.722	0.644	0.707	0.433
Road	0.791	0.658	0.722	0.412
BCCD	0.873	0.881	0.916	0.628
Lemon Disease	0.928	0.888	0.925	0.693
Incorrect-Mask-2	0.919	0.804	0.877	0.583

F. Training Curves

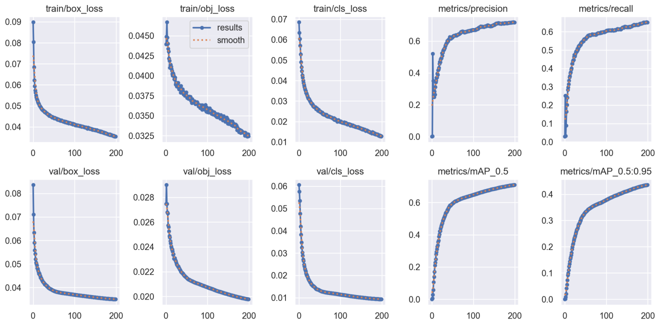

Fig. 4. Training and Validation Curves. The figure illustrates various performance metrics, including overall model performance, train/box_loss, train/obj_loss, train/class_loss, metrics/precision, metrics/recall, val/box_loss, val/obj_loss, val/class_loss, metrics/mAP_0.5, and metrics/mAP_0.5:0.95.

V. RESULTS AND ANALYSIS

This section summarizes the key findings from our experiments, highlighting the performance of our proposed method, comparisons with state-of-the-art (SOTA) models, and insights derived from ablation studies.

A. Performance on Pascal VOC Dataset

The evaluation of our student models on the Pascal VOC 2007 and 2012 datasets shows significant performance improvements while maintaining computational efficiency:

- **Student Model YOLOv5n**: Achieved a mAP@50 of **0.707** and mAP@50-95 of **0.433**, demonstrating its ability to accurately detect objects across various categories.
- **Student Model YOLOv5n Custom**: Obtained a mAP@50 of **0.550** and mAP@50-95 of **0.293**, highlighting its efficiency with only **0.58M** parameters, making it a viable choice for resource-constrained environments.

Table 2 in the Experiment section provides a per-class analysis, showing consistent detection performance across multiple categories, with particularly high scores in classes like *car* (mAP@50: 0.870) and *person* (mAP@50: 0.821). This reflects the model's strong capability to generalize across object types.

B. Comparison with SOTA Models

Our approach is compared to YOLOv7-tiny, a lightweight SOTA model, in Table 3 of the Experiment section:

- **YOLOv7-tiny**: Achieves a mAP@50 of **0.644** and mAP@50-95 of **0.385** with **6.2M** parameters.
- **YOLOv5n Student Model**: Achieves a higher mAP@50 of **0.708** and mAP@50-95 of **0.435** with only **1.79M** parameters.
- **YOLOv5n Custom Model**: Offers an ultra-lightweight solution (0.58M parameters) with reasonable performance (mAP@50: 0.550, mAP@50-95: 0.293).

These results emphasize the effectiveness of our Combined Imitation Loss function and fine-grained feature imitation approach in achieving superior performance while reducing model size.

C. Ablation Studies: Impact of Different KD Loss Functions

We experimented with various knowledge distillation (KD) loss functions to evaluate their impact on student model performance. The results, as shown in Table 4 of the Experiment section, indicate that:

- The **Combined Loss Function** (with $\alpha = 0.5$) achieved the best balance between mAP@50 (**0.709**) and mAP@50-95 (**0.435**).
- Other loss functions like Attention Transfer Loss and Smooth MSE Loss also showed competitive performance but did not surpass the Combined Loss Function in terms of overall accuracy and robustness.

These findings confirm the advantage of our proposed Combined Imitation Loss function in guiding the student model effectively.

D. Generalization Across Datasets

To evaluate the generalizability of the student models, we tested them on various datasets beyond Pascal VOC, including Road, BCCD, Lemon Disease, and Incorrect-Mask-2. As summarized in Table 5 of the Experiment section:

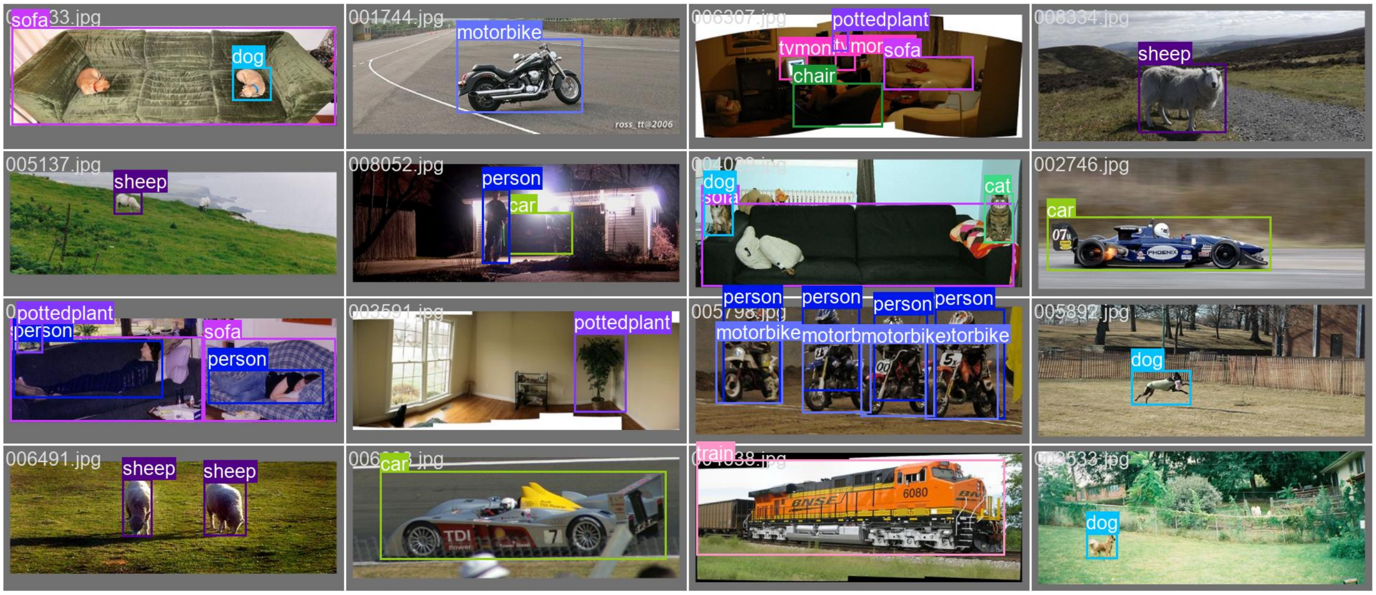

Fig. 5. Ground truth labels from the validation dataset (val_batch0_label).

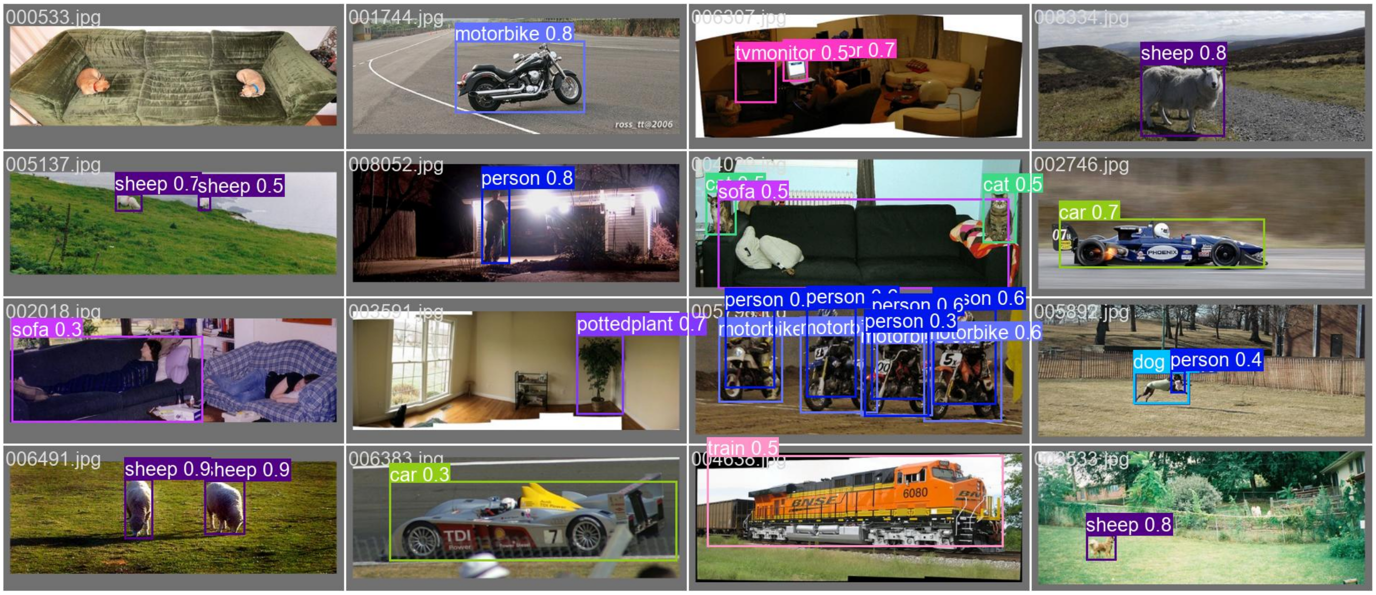

Fig. 6. Comparison between ground truth labels and predicted results on the validation dataset.

- The **BCCD dataset** yielded the highest mAP@50 of **0.916**, demonstrating the model's ability to handle domain-specific tasks.
- The **Lemon Disease dataset** achieved a mAP@50 of **0.925**, highlighting its adaptability to agricultural applications.
- Even on challenging datasets like **Incorrect-Mask-2**, the model achieved a strong mAP@50 of **0.877**, showing its robustness.

These results demonstrate that our approach generalizes well across diverse domains, making it suitable for a wide range of applications.

E. Training and Convergence Behavior

Training curves for metrics like box loss, classification loss, and object loss indicate stable convergence over 200 epochs. As shown in **Figure 4**:

- Both box and classification losses steadily decrease, indicating effective learning.
- The mAP@50 and mAP@50-95 improve consistently, with no signs of overfitting, validating the effectiveness of the training process.

F. Visualization of Predictions

The qualitative results in **Figure 6** showcase accurate bounding box predictions across various object categories. The student models effectively localize and classify objects with high precision, even in cluttered scenes.

VI. Conclusion

In this work, we proposed an appropriate fine-grained feature imitation method for knowledge distillation, utilizing a **Combined Imitation Loss Function** to enhance the performance of lightweight student object detection models. By leveraging both KL divergence and MSE in the imitation loss, our approach effectively distilled the knowledge from a larger teacher model (YOLOv5x) to significantly smaller student models (YOLOv5n and YOLOv5n-Custom) without compromising much on accuracy.

Experimental results on the Pascal VOC 2007 and 2012 datasets demonstrated the superiority of our method over baseline models and loss functions. Specifically, the student model trained with our combined loss achieved a higher mAP@50 (0.709) compared to other loss functions, while maintaining computational efficiency and reduced model size. Furthermore, when benchmarked against state-of-the-art lightweight models like YOLOv7-tiny, our approach outperformed them with a smaller model size and fewer parameters.

Through detailed ablation studies, we verified the effectiveness of the **Combined Imitation Loss** with varying alpha values and observed optimal performance at $\alpha = 0.5$. Additionally, our approach was validated on various datasets, including Road, BCCD, Lemon Disease, and Incorrect-Mask-2, showing consistent performance across diverse tasks and classes. The confusion matrix and class-wise mAP further highlight the robustness of our student model, particularly in detecting smaller objects and addressing class imbalances.

The proposed method not only simplifies the feature imitation process but also provides a generalized framework for improving the efficiency of object detection models. This makes it a promising technique for real-world applications where computational resources are limited, such as autonomous systems, medical imaging, and smart surveillance.

In summary, our work paves the way for future research in efficient knowledge distillation and lightweight object detection models. Future work may focus on exploring additional datasets, enhancing imitation strategies, and extending this method to transformer-based detection architectures.

References

[1] D. Anisimov and T. Khanova, "Towards lightweight convolutional neural networks for object detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, 2017, pp. 1–8.

[2] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Advances Neural Information Processing Systems (NeurIPS)*, 2017, pp. 742–751.

[3] Y. Chen, N. Wang, and Z. Zhang, "DarkRank: Accelerating deep metric learning via cross-sample similarities transfer," *arXiv preprint arXiv:1707.01220*, 2017.

[4] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2015, pp. 1440–1448.

[5] P. Gysel, M. Motamedi, and S. Ghiasi, "Hardware-oriented approximation of convolutional neural networks," *arXiv preprint arXiv:1604.03168*, 2016.

[6] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Advances Neural Information Processing Systems (NeurIPS)*, 2015, pp. 1135–1143.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[8] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *arXiv preprint arXiv:1707.01219*, 2017.

[9] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," *arXiv preprint arXiv:1608.08710*, 2016.

[10] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2017, pp. 7341–7349.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[12] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2016, pp. 21–37.

[13] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," *arXiv preprint arXiv:1707.06342*, 2017.

[14] J. Park *et al.*, "Faster CNNs with direct sparse convolutions and guided pruning," *arXiv preprint arXiv:1608.01409*, 2016.

[15] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2016, pp. 525–542.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Advances Neural Information Processing Systems (NeurIPS)*, 2015, pp. 91–99.

[17] A. Romero *et al.*, "FitNets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[18] C. Tai, T. Xiao, Y. Zhang, and X. Wang, "Convolutional neural networks with low-rank regularization," *arXiv preprint arXiv:1511.06067*, 2015.

[19] W. Wen *et al.*, "Coordinating filters for faster deep neural networks," *arXiv preprint arXiv:1703.09746*, 2017.

[20] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2016, pp. 4820–4828.

[21] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," *arXiv preprint arXiv:1702.03044*, 2017.