2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN) | 979-8-3315-2042-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICMLCN64995.2025.11140497

Semantic Knowledge Distillation for Onboard Satellite Earth Observation Image Classification

Thanh-Dung Le, Vu Nguyen Ha, Ti Ti Nguyen, Geoffrey Eappen, Prabhu Thiruvasagam,
Hong-fu Chou, Duc-Dung Tran, Luis M. Garces-Socarras, Jorge L. Gonzalez-Rios,
Juan Carlos Merlano-Duncan, Symeon Chatzinotas

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg

***Abstract*—This study introduces a dynamic weighting knowledge distillation (KD) framework for efficient Earth observation (EO) image classification (IC) in resource-constrained environments. By leveraging EfficientViT and MobileViT as *teacher* models, this approach enables lightweight *student* models, specifically ResNet8 and ResNet16, to achieve over 90% accuracy, precision, and recall, meeting the confidence thresholds required for reliable classification. Unlike traditional KD with fixed weights, our dynamic weighting mechanism adjusts based on each *teachers* confidence, allowing the *student* model to prioritize more reliable knowledge sources. ResNet8, in particular, achieves substantial efficiency gains, with 97.5% fewer parameters, 96.7% fewer FLOPs, 86.2% lower power consumption, and 63.5% faster inference time compared to MobileViT. This significant reduction in complexity and resource demand makes ResNet8 an ideal choice for EO tasks, balancing high performance with practical deployment requirements. This confidence-driven, adaptable KD strategy demonstrates the potential of dynamic knowledge distillation to deliver high-performing, resource-efficient models for satellite-based EO applications. Reproducible codes are available from our shared Github repository [1].**

***Index Terms*—Earth Observation, Remote Sensing, Knowledge Distillation, Onboard Processing, Artificial Intelligence, ResNet.**

I. Introduction

The rapid increase in satellite deployments for EO and remote sensing (RS) missions reflects a growing demand for applications like environmental monitoring, disaster response, precision agriculture, and scientific research [1]. These applications rely on high-frequency, high-resolution data for timely and accurate decision-making. However, a significant bottleneck in Low Earth Orbit (LEO) satellite operations is reliance on ground stations for data transmission, which limits the availability of communication windows and results in frequent connectivity loss [2]–[4]. This delay can impede critical responses in situations requiring immediate data access.

The advent of Satellite Internet Providers, such as Starlink and OneWeb, offers the potential for continuous (24/7) connectivity to LEO satellites, facilitating on-demand data access [5]–[7]. Yet, seamless connectivity alone does not fully meet modern EO and RS requirements, which increasingly demand real-time, onboard decision-making. For optimal operations, onboard neural networks (NNs) must prioritize computational efficiency to autonomously analyze data, identify critical information, and make immediate adjustments, such as refocusing on a target area during subsequent satellite passes [2].

Historically, onboard NNs have been designed for efficiency, often relying on convolutional neural network (CNN) models to balance performance and resource constraints. For example, the Φ-Sat-1 mission used a CNN-based NN for onboard image segmentation using the Intel Movidius Myriad 2 vision processing unit (VPU), representing the first deployment of deep learning on a satellite [8]. Similarly, Φ-Sat-2 adopted a convolutional autoencoder for image compression to reduce transmission requirements, demonstrating the feasibility of lightweight models on hardware-constrained environments on three different hardware, including graphic processing unit (GPU) NVIDIA GeForce GTX 1650, VPU Myriad 2, and central processing unit (CPU) Intel Core i7-6700 [9].

Despite their efficiency, CNNs can be limited in performance, especially compared to the recent success of Vision Transformer (ViT) architectures. ViTs have gained popularity in computer vision due to their ability to capture global context via self-attention mechanisms, often surpassing traditional CNNs in performance. However, ViTs require significantly more computational power and memory as image resolution increases, which poses challenges for deployment on power-constrained satellite platforms [10]–[14].

To overcome these limitations, KD offers a viable approach for onboard processing. KD is a method where a smaller, simpler model (the *"student"*) learns from a larger, complex model (the *"teacher"*, such as a ViT). By transferring the *teacher*'s semantic knowledge (SK), KD allows the *student* to generalize more effectively with lower computational demands [15]. KD was initially introduced to reduce the computational burden of deep learning models [16], and recent studies indicate that KD can help *students* learn complex representations with strong performance even in simplified forms [17].

In this study, we leverage KD to train deployable models for onboard EO tasks, explicitly focusing on IC. By distilling SK from ViTs into efficient *student* models (*StuMs*), we aim to boost onboard processing capabilities while maintaining computational efficiency suitable for satellite EO missions. Traditional KD approaches often struggle with training instability, mainly when exact prediction matches are enforced through Kullback-Leibler (KL) divergence from a single *teacher*, which can impair performance [18]. To address this, we propose a dynamic weighting mechanism for dual-*teacher* KD (DualKD), where the weight assigned to each *teacher* adapts based on their confidence level, enabling the *StuM* to

[1] https://github.com/ltdung/SnT-SENTRY

979-8-3315-2042-7/25/$31.00 ©2025 IEEE

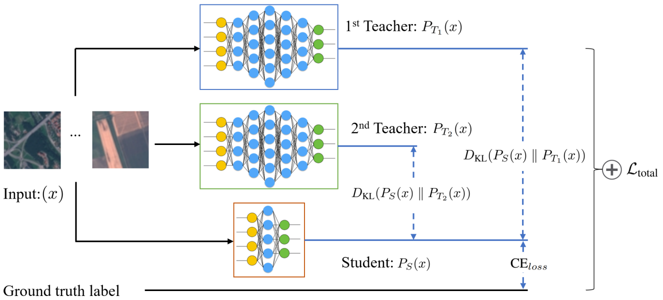
1st Teacher: $P_{T_1}(x)$
2nd Teacher: $P_{T_2}(x)$
$D_{\text{KL}}(P_S(x) \parallel P_{T_1}(x))$
$D_{\text{KL}}(P_S(x) \parallel P_{T_2}(x))$
$\mathcal{L}_{\text{total}}$
Input:(x)
...
Student: $P_S(x)$
CE_{loss}
Ground truth label

Fig. 1: The schematic workflow of DualKD.

prioritize the most reliable knowledge sources. This approach considers that one instance may have varying semantic similarities to different *teachers*, thus improving the *students* ability to generalize data representation.

II. Materials and Methods

A. Dataset

This study utilizes the EuroSAT dataset [19], a well-established land use and land cover classification benchmark specifically curated from Sentinel-2 satellite imagery. EuroSAT includes 27,000 labeled, geo-referenced images, each with a resolution of 64x64 pixels across 13 spectral bands, and is organized into 10 distinct classes. These classes cover various land types, such as industrial and residential buildings, annual and permanent crops, rivers, lakes, herbaceous vegetation, highways, pastures, and forests, representing diverse European landscapes. Each class contains between 2000 and 3000 images, providing a balanced model training and evaluation dataset. EuroSAT's compact image size and broad class diversity make it suitable for developing and assessing deep learning models intended for onboard processing in EO missions. This dataset is valuable for applications that demand real-time decision-making capabilities, such as environmental monitoring, disaster response, and precision agriculture.

B. Dual teachers Knowledge Distillation

Traditional KD methods often face training instability, especially when forced to closely match a single *teacher* model (*TeaM*)'s predictions using KL divergence, which can compromise performance [18]. We propose a DualKD framework with a dynamic weighting mechanism to overcome this challenge and enhance the *StuM*'s adaptability and effectiveness. Unlike traditional KD approaches that employ fixed weights, this method adjusts each *teacher*'s influence based on confidence levels, allowing the *StuM* to prioritize knowledge from the more reliable *teacher*. This dynamic weighting approach improves flexibility in handling multiple *teachers* with varying degrees of reliability, ultimately optimizing the semantic information from the KD process.

As shown in Fig. 1, given an input x, the semantic distillation process starts by computing softened probability distributions for the *TeaM* and the *StuM*. This is achieved by scaling their logits with a temperature parameter τ. For *TeaM* T_1, the softened probability distribution is:

$$P_{T_1}(x) = \text{softmax}\left(T_1(x)/\tau\right), \tag{1}$$

and similarly for *TeaM* T_2:

$$P_{T_2}(x) = \text{softmax}\left(T_2(x)/\tau\right), \tag{2}$$

with the *StuM* S:

$$P_S(x) = \text{softmax}\left(S(x)/\tau\right). \tag{3}$$

Confidence for each *teacher* is computed as the average of the maximum probabilities in their respective softened distributions:

$$C_{T_1} = \mathbb{E}\left[\max(P_{T_1}(x))\right], \quad C_{T_2} = \mathbb{E}\left[\max(P_{T_2}(x))\right]. \tag{4}$$

Based on these confidence scores, we dynamically adjust the weights α and β assigned to each *teacher* in the distillation loss KD_{loss}. If both confidence scores are significantly below a predefined threshold δ, both *teachers* are ignored ($\alpha = \beta = 0$). If either confidence score is close to the threshold, we prioritize the more reliable *teacher* by reducing the weight of the less reliable one, with minimum weights set by w_{min}. When both *teachers* are above the threshold, equal weights ($\alpha = \beta = 0.5$) are used.

The distillation loss KD_{loss}, a weighted KL divergence between the *students* and each *teachers* softened probabilities, is then computed, with the weights α and β reflecting each *teacher*'s confidence.

$$\text{KD}_{\text{loss}} = \alpha \cdot D_{\text{KL}}(P_S(x) \parallel P_{T_1}(x)) + \beta \cdot D_{\text{KL}}(P_S(x) \parallel P_{T_2}(x)), \tag{5}$$

where the KL divergence D_{KL} for each "*teacher-student*" pair is scaled by the temperature squared, τ^2, to stabilize training

$$D_{\text{KL}}(P_S(x) \| P_{T_i}(x)) = \frac{1}{\tau^2} \sum_j P_{T_i}(x)_j \log\left(\frac{P_{T_i}(x)_j}{P_S(x)_j}\right). \tag{6}$$

The total distillation loss is calculated as a combination of the classification loss, CE_{loss}, and the distillation loss KD_{loss}. A classification loss CE_{loss} between the *students* predictions and the true labels is calculated to ground the *students* learning in *teacher* guidance and actual labels, where

$$\text{CE}_{\text{loss}} = -\sum_i y_i \log\left(P_S(x)_i\right). \tag{7}$$

Then, the final combined loss, $\mathcal{L}_{\text{total}}$, integrates these components: a weighted combination of CE_{loss} and KD_{loss}. This framework allows the *student* to leverage insights from both *teachers* selectively, focusing on the most reliable sources for improved generalization and adaptability across instances during training,

$$\mathcal{L}_{\text{total}} = \left(1 - \frac{\alpha + \beta}{2}\right) \cdot \text{CE}_{\text{loss}} + \frac{\alpha + \beta}{2} \cdot \text{KD}_{\text{loss}} \tag{8}$$

The pseudo Algorithm 1 presents the core steps for implementing the proposed DualKD framework with a dynamic weighting mechanism that prioritizes the SK from two *TeaMs*. This adaptive approach is designed to maximize the *StuM*'s

Algorithm 1 DualKD with Dynamic Weighting

```
Require: Input data $x$, true labels $y$, model_student, model_teacher_1, model_teacher_2, temperature $\tau$, confidence threshold ($\delta$), minimum weight ($w_{min}$)
Ensure: Combined loss $\mathcal{L}_{total}$ for backpropagation
Forward pass:
    $T_1(x) \leftarrow$ model_teacher_1$(x)$
    $T_2(x) \leftarrow$ model_teacher_2$(x)$
    $S(x) \leftarrow$ model_student$(x)$
Calculate softened probabilities with temperature $\tau$:
    $P_{T_1}(x) \leftarrow \text{softmax}(T_1(x)/\tau)$
    $P_{T_2}(x) \leftarrow \text{softmax}(T_2(x)/\tau)$
    $P_S(x) \leftarrow \text{softmax}(S(x)/\tau)$
Calculate confidence scores for both teachers:
    $C_{T_1} \leftarrow \mathbb{E}\left[\max(P_{T_1}(x))\right]$
    $C_{T_2} \leftarrow \mathbb{E}\left[\max(P_{T_2}(x))\right]$
Dynamically set weights $\alpha$ and $\beta$ based on confidence scores:
if $C_{T_1} < 0.4$ and $C_{T_2} < 0.4$ then
    $\alpha, \beta \leftarrow 0.0, 0.0$                // Ignore both teachers
else if $C_{T_1} < \delta$ and $C_{T_2} < \delta$ then
    $\alpha \leftarrow \max(0.5 - (\delta - C_{T_1}), w_{min})$
    $\beta \leftarrow \max(0.5 - (\delta - C_{T_2}), w_{min})$
else if $C_{T_1} < \delta$ then
    $\alpha, \beta \leftarrow 0.3, 0.7$                // Reduce $\alpha$, prioritize teacher 2
else if $C_{T_2} < \delta$ then
    $\alpha, \beta \leftarrow 0.7, 0.3$                // Reduce $\beta$, prioritize teacher 1
else
    $\alpha, \beta \leftarrow 0.5, 0.5$        // Equal weighting for both confident teachers
end if
Compute Distillation Loss (KL Divergence with weighted sum):
    $\text{loss}_1 \leftarrow D_{KL}(P_S(x) \parallel P_{T_1}(x))$
    $\text{loss}_2 \leftarrow D_{KL}(P_S(x) \parallel P_{T_2}(x))$
    $\text{KD}_{loss} \leftarrow (\alpha \cdot \text{loss}_1 + \beta \cdot \text{loss}_2) \cdot \tau^2$
Compute Classification Loss (Cross-Entropy):
    $\text{CE}_{loss} \leftarrow -\sum_i y_i \log\left(P_S(x)_i\right)$
Combine losses to calculate Total Loss:
    $\mathcal{L}_{total} \leftarrow \left(1 - \frac{\alpha+\beta}{2}\right) \cdot \text{CE}_{loss} + \frac{\alpha+\beta}{2} \cdot \text{KD}_{loss}$
Backpropagate using $\mathcal{L}_{total}$
```

learning efficiency by allowing it to concentrate on information from the most reliable *teacher* in each instance. Specifically, each *TeaM*'s confidence score is computed by averaging the maximum probabilities of their softened outputs across a batch, reflecting each *teachers* reliability. These scores are then used to dynamically adjust the weights, α and β, for each *teacher*, guiding the *student* to selectively emphasize knowledge from the *teacher* with higher semantic value in each scenario. This dynamic weighting mechanism enables the *student* to effectively distill the most meaningful and relevant SK, ultimately enhancing its performance and robustness across tasks.

- **Low confidence in both *teachers***: If both confidence scores fall significantly below a threshold δ, both *teachers* are disregarded
- **Moderate confidence in both *teachers***: Both *teachers* are assigned reduced weights, ensuring some influence without complete reliance.
- **Low confidence in one *teacher***: Lower weight is assigned to the less confident *teacher*, prioritizing the more reliable one.
- **High confidence in both *teachers***: Both *teachers* receive equal weighting.

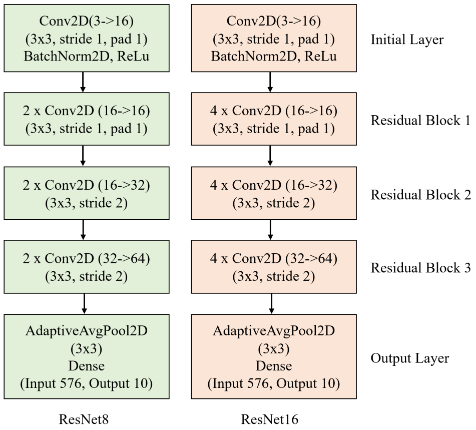

Fig. 2: Variants of ResNet-based *StuMs* network structure.

C. Machine Learning Models

A recent study [13] provides a comprehensive analysis of ViTs performance and robustness in EO-IC, identifying EfficientViT and MobileViT as the two most effective models. Therefore, we select EfficientViT and MobileViT as our *TeaMs* for this study. Technically, the EfficientViT model combines convolutional layers with local window attention mechanisms to optimize the balance between performance and computational efficiency, featuring approximately 4 million parameters [20]. The MobileViT model integrates convolutional and transformer-based processing, using depthwise separable convolutions and self-attention mechanisms for high accuracy and efficient image classification [21].

ResNet has been shown to outperform standard CNNs for IC because of its ability to address the vanishing gradient problem through skip connections, allowing for deeper architectures and improved feature learning, as demonstrated in a comparative analysis [22]. Additionally, ResNet is widely recognized for its effectiveness in KD training [23], [24]. Therefore, we select a ResNet-based architecture as the *StuM* for this study. We utilize two variants of ResNet—ResNet8 and ResNet16—as *StuMs* for KD, as shown in Fig. 2. These models are designed with progressively deeper architectures to balance performance with computational efficiency, making them suitable for deployment in resource-constrained environments, such as onboard satellite processing.

- **ResNet8**: This lightweight ResNet variant consists of an initial convolutional layer followed by three residual blocks. Each block is structured to gradually increase the number of feature channels, from 16 to 64, through convolutional layers with either stride 1 or 2. The model concludes with an adaptive average pooling and dense layers. ResNet8's shallow architecture makes it highly efficient for scenarios with limited computation power.
- **ResNet16**: This model builds on ResNet8's structure by incorporating additional convolutional layers within each residual block, doubling the network's depth. The increased depth allows ResNet16 to capture more complex features, making it a more capable model for handling detailed image classification tasks. Like ResNet8, it uses

TABLE I: Experiment Parameters Setting

Parameter	Value
Epoch	50
Batch size	64
Optimizer	AdamW
Learning rate	0.00025
Weight decay	0.0005
Scheduler	ReduceLRonPlateau
Threshold (δ)	0.6
Temperature (τ)	5
Min weight (w_{min})	0.1

an adaptive average pooling layer and a dense layer.

Both models are configured with Batch Normalization and ReLU activation functions to enhance training stability and convergence. Their structural differences offer a range of performance and efficiency trade-offs, providing flexibility for different use cases in the knowledge distillation framework.

D. Evaluation Metrics

To comprehensively evaluate the performance of our multi-class classification model across 10 classes, we employ three key metrics: accuracy, precision, and recall (sensitivity) [25]. These metrics are calculated for each class individually and then aggregated using macro-averaging to assess the model's performance as follows [26]–[28],

$$\text{Accuracy} = \sum_{k=1}^{K} \frac{\text{TP}_k}{N} \tag{9}$$

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^{K} N_k \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k} \tag{10}$$

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^{K} N_k \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \tag{11}$$

where N is the total number of data points across all classes. K is the total number of classes. N_k is the number of data points in class k. TP_k is True Positives, FP_k is False Positives, FN_k is False Negatives for class k, respectively. We use weighted precision and recall to ensure that each class is given equal importance, thereby providing a balanced evaluation of the model's classification capabilities across the entire dataset. These macro-averaged evaluation metrics will select the best models in the final analysis.

III. Experimental Setup

All the experiments are conducted on GPU NVIDIA RTX™ 6000 Ada Generation, 48 GB GDDR6. Experiments were implemented using the Scikit-learn library [29], and Pytorch. The data was divided into 70% training and 30% testing. In addition, we also applied batch normalization [30] are employed for models' stability. Experiment parameters setting are summarized in Table I.

IV. Results and Discussions

The performance comparison, as shown in Table II, clearly illustrates the substantial improvements gained by applying KD, especially with a DualKD approach. Initially, employing KD with single *teachers* like EfficientViT and MobileViT boosts ResNet8's accuracy from 87.76% (base model) to 91.77% and 91.06%, respectively, demonstrating increases of approximately 4% over the baseline. Precision and recall also reflect similar enhancements. However, the dual-*teacher* setup achieves the best results, increasing accuracy to 92.88% - a total improvement of 5.12% over the base model. Precision and recall also reach peak values of 93.07% and 92.88%, showing the most significant gains. This highlights how leveraging semantic insights from both *teachers* allows the model to capture richer, more nuanced features, resulting in a marked increase in performance across all metrics.

TABLE II: Performance Comparison of ResNet8

Model	Accuracy (↑)	Precision (↑)	Recall (↑)
ResNet8 (Base)	87.76	87.7	87.76
ResNet8 (EfficientViT)	91.77	91.74	91.77
ResNet8 (MobileViT)	91.06	91	91.06
ResNet8 (Dual)	**92.88**	**93.07**	**92.88**

Bold denotes the best values.

TABLE III: Performance Comparison of ResNet16

Model	Accuracy (↑)	Precision (↑)	Recall (↑)
ResNet16 (Base)	92.9	92.91	92.9
ResNet16 (EfficientViT)	94.49	94.52	94.49
ResNet16 (MobileViT)	93.29	93.33	93.29
ResNet16 (Dual)	**96.46**	**96.52**	**96.46**

Bold denotes the best values.

Similarly, the performance comparison for ResNet16 in Table III demonstrates the significant gains achieved through KD, particularly with a DualKD approach. Utilizing single-*teacher* KD models like EfficientViT and MobileViT already enhances ResNet16's performance, with EfficientViT increasing accuracy from 92.9% (base model) to 94.49%—an improvement of 1.59%—and MobileViT achieving a slight increase to 93.29%. However, the dual-*teacher* configuration yields the highest performance across all metrics, boosting accuracy to 96.46%, a total improvement of 3.56% over the base model. Precision and recall are similarly elevated, reaching 96.52% and 96.46%, respectively, marking the most substantial gains. Once again, this result underscores the DualKD strategy's effectiveness in enhancing ResNet16's predictive capability, demonstrating that combining semantic insights from two *teachers* allows for a more comprehensive knowledge transfer, significantly improving the model's overall performance.

However, despite these advancements, the *StuMs* still fall short of matching the performance levels of the *TeaMs*, as summarized from [13]. The EfficientViT *TeaM* achieves an impressive accuracy of 98.76%, precision of 98.77%, and recall of 98.76%. MobileViT, the highest-performing model in this comparison, reaches an accuracy, precision, and recall of 99.09%. This difference highlights the gap between the *StuMs* and their *teacher* counterparts, illustrating that while KD with dual *teachers* substantially narrows the performance gap, the *StuMs* have yet to achieve the full predictive capacity exhibited by the *teachers*.

TABLE IV: Model Comparison on Parameters, FLOPs, Size, Inference Time, and Power Consumption

Models	**Total Parameters (↓)**	**FLOPs (↓)**	**Size (MB) (↓)**	**Inference time (s) (↓)**	**Power (W) (↓)**
ResNet8	**98,522**	**60,113,536**	**5.95**	**5.84**	**10.94 ± 0.83**
ResNet16	195,738	117,883,520	10.01	6.7	24.63 ± 1.63
EfficientViT [13]	3,964,804	203,533,056	38.19	10	29.04 ± 0.96
MobileViT [13]	4,393,971	1,843,303,424	259.30	16	79.23 ± 1.45

Bold denotes the best values.

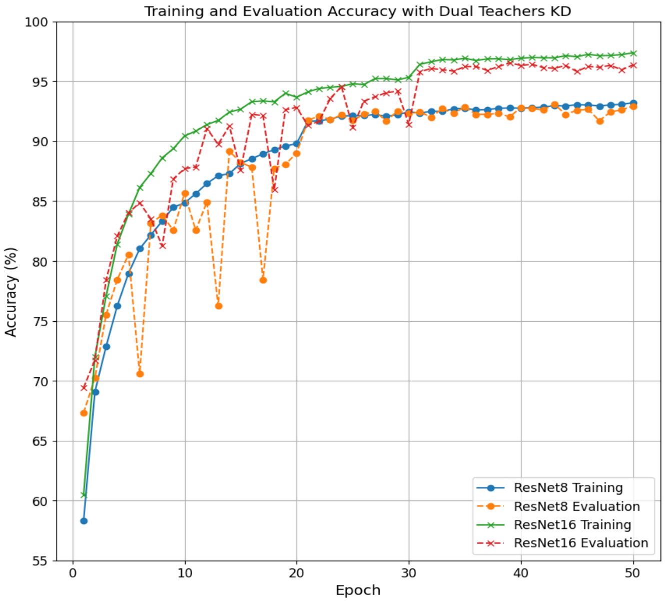

Fig. 3: Training and evaluation accuracy of ResNet variants with DualKD.

Besides, ResNet8 achieves over 90% in all evaluation metrics (accuracy, precision, and recall), meeting the required confidence level for reliable predictions. This strong performance underscores its effectiveness as a *StuM* under the DualKD approach. The key advantage of using ResNet8 lies in its significantly lower complexity compared to its *TeaMs*, EfficientViT and MobileViT, making it exceptionally suitable for deployment in resource-constrained environments.

As summarized in Table IV, with a parameter count of only 98,522, ResNet8 is 97.5% smaller than EfficientViT (3,964,804 parameters) and 97.8% smaller than MobileViT (4,393,971 parameters). It requires just 60,113,536 FLOPs, representing a 70.5% reduction compared to EfficientViT and an impressive 96.7% reduction compared to MobileViT. Additionally, ResNet8's model size is only 5.95 MB, making it 84.4% smaller than EfficientViT (38.19 MB) and 97.7% smaller than MobileViT (259.30 MB). The inference time is equally optimized, with ResNet8 achieving 5.84 seconds, which is 41.6% faster than EfficientViT and 63.5% faster than MobileViT. Its power consumption is also considerably lower at 10.94 W ± 0.83 W, which is 62.3% less than EfficientViT and 86.2% less than MobileViT. These reductions in complexity, size, and energy demands highlight ResNet8's suitability for real-world applications where computational resources and power efficiency are critical. By maintaining high performance with low complexity, ResNet8 demonstrates the effectiveness of DualKD in creating a lightweight model that meets both predictive confidence and deployment constraints.

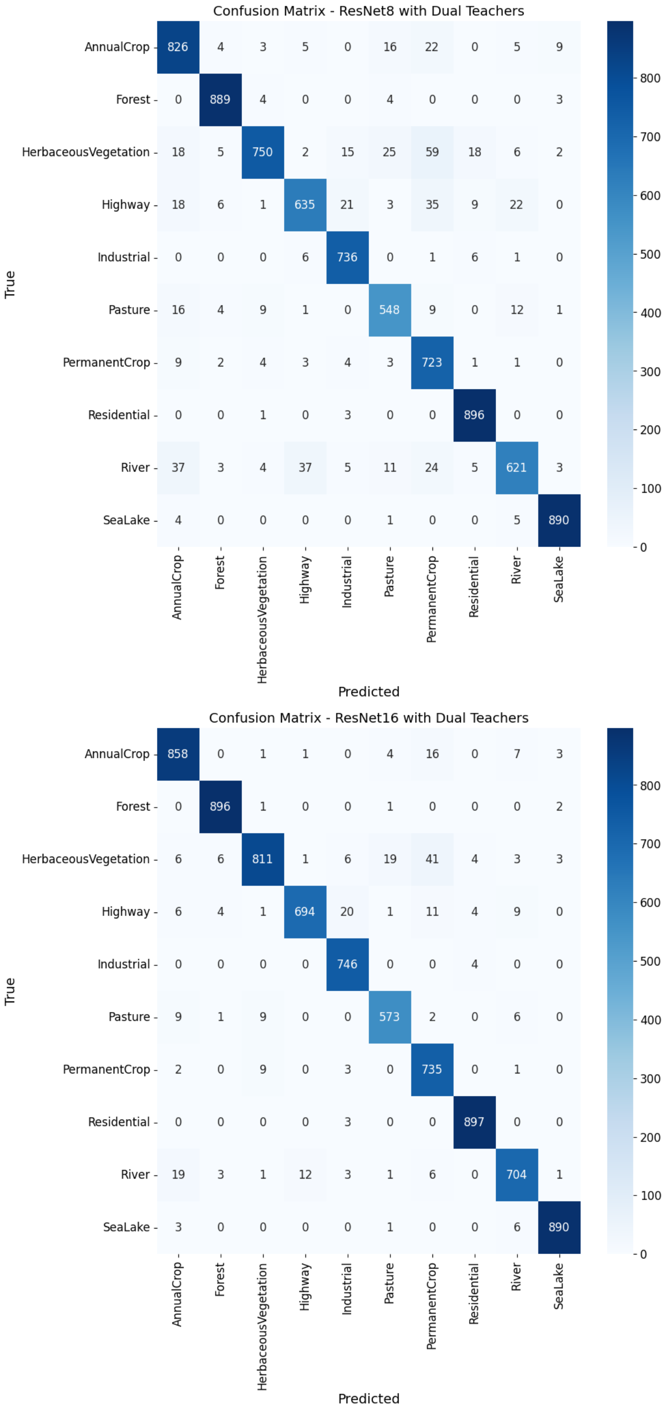

Fig. 4: Confusion matrix from ResNet8 (top) and ResNet16 (bottom) with DualKD.

Figure 3 shows the training and evaluation accuracy for

ResNet8 and ResNet16 with DualKD shows an initial phase of fluctuations, particularly in the evaluation metrics. During the first 20 epochs, both models experience notable variability, reflecting the model's adaptation to the dual-*teacher* signals. However, after 20 epochs, both ResNet8 and ResNet16 curves smooth out and converge, indicating stabilized learning and consistent improvement. Notably, ResNet8 exhibits a better convergence pattern than ResNet16, as evidenced by the smaller gap between its training and evaluation accuracy. This narrower difference suggests that ResNet8 generalizes more effectively, maintaining closer alignment between its training and evaluation performance. While ResNet16 ultimately achieves higher overall accuracy, comparing the confusion matrices of Fig. 4, it does so with a larger training-evaluation gap and at the cost of increased complexity—approximately double that of ResNet8. Given the marginal performance gain relative to its added computational cost, ResNet16 may not be worth the additional complexity, making ResNet8 the more efficient for applications requiring a balanced trade-off between accuracy and power consumption.

V. Conclusions

In conclusion, this study demonstrates the effectiveness of DualKD in enhancing the performance of lightweight *StuMs*, specifically ResNet8 and ResNet16. Both models achieve over 90% accuracy, precision, and recall, meeting the required confidence level for reliable predictions and showing substantial improvements over baseline performances. ResNet8, in particular, strikes an optimal balance between high accuracy and efficiency, with significantly lower parameter counts, FLOPs, inference time, and power consumption.

Acknowledgment

This work was funded by the Luxembourg National Research Fund (FNR), with granted SENTRY project corresponding to grant reference C23/IS/18073708/SENTRY.

References

[1] S. Sadek, "New satellite market forecast anticipates 1,700 satellites to be launched on average per year by 2030 as new entrants and incumbents increase their investment in space," [Online]. Available: https://shorturl.at/6PVma, 2021.

[2] G. Fontanesi, *et al.*, "Artificial intelligence for satellite communication and non-terrestrial networks: A survey," *arXiv preprint arXiv:2304.13008*, 2023.

[3] T. S. Abdu, *et al.*, "Demand-aware flexible handover strategy for leo constellation," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2023, pp. 978–983.

[4] H. Nguyen-Kha, V. N. Ha, E. Lagunas, S. Chatzinotas, and J. Grotz, "Seamless 5g automotive connectivity with integrated satellite terrestrial networks in c-band," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 2024, pp. 1–5.

[5] ——, "Leo-to-user assignment and resource allocation for uplink transmit power minimization," in *WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding*. VDE, 2023, pp. 1–6.

[6] ——, "Joint two-tier user association and resource management for integrated satellite-terrestrial networks," *IEEE Transactions on Wireless Communications*, 2024.

[7] V. N. Ha, *et al.*, "User-centric beam selection and precoding design for coordinated multiple-satellite systems," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2024, pp. 1–6.

[8] G. Giuffrida, *et al.*, "The ϕ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2021.

[9] G. Guerrisi, F. Del Frate, and G. Schiavon, "Artificial intelligence based on-board image compression for the ϕ-sat-2 mission," *IEEE J. Sel. Top. Appl. Earth. Obs. Remote. Sens.*, 2023.

[10] H.-f. Chou, *et al.*, "Edge ai empowered physical layer security for 6g ntn: Potential threats and future opportunities," *arXiv preprint arXiv:2401.01005*, 2023.

[11] ——, "On-air deep learning integrated semantic inference models for enhanced earth observation satellite networks," *arXiv preprint arXiv:2409.15246*, 2024.

[12] H. F. Chou, *et al.*, "Semantic inference-based deep learning and modeling for earth observation: Cognitive semantic augmentation satellite networks," *arXiv preprint arXiv:2409.15246*, 2024.

[13] T. D. Le, *et al.*, "On-board satellite image classification for earth observation: A comparative study of vit models," *arXiv preprint arXiv:2409.03901*, 2024.

[14] T. T. Nguyen, *et al.*, "A semantic-loss function modeling framework with task-oriented machine learning perspectives," in *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 2025, pp. 1–6.

[15] L. Papa, P. Russo, I. Amerini, and L. Zhou, "A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024.

[16] G. Hinton, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[17] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, and A. G. Wilson, "Does knowledge distillation really work?" *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 6906–6919, 2021.

[18] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 33 716–33 727, 2022.

[19] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Top. Appl. Earth. Obs. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, 2019.

[20] X. Liu, *et al.*, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 14 420–14 430.

[21] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *arXiv preprint arXiv:2206.02680*, 2022.

[22] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International Conference on Disruptive Technologies for Multi-disciplinary Research and Applications*, vol. 1, 2021, pp. 96–99.

[23] Z. Huang, *et al.*, "Revisiting knowledge distillation: An inheritance and exploration framework," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3579–3588.

[24] Q. Guo, *et al.*, "Online knowledge distillation via collaborative learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 020–11 029.

[25] C. Goutte and et. al., "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European Conference on Information Retrieval*. Springer, 2005, pp. 345–359.

[26] T.-D. Le, C. Macabiau, K. Albert, P. Jouvet, and R. Noumeir, "A novel transformer-based self-supervised learning method to enhance photoplethysmogram signal artifact detection," *IEEE Access*, 2024.

[27] T.-D. Le, T. T. Nguyen, and V. N. Ha, "The impact of lora adapters for llms on clinical nlp classification under data limitations," *arXiv preprint arXiv:2407.19299*, 2024.

[28] T.-D. Le, "Boosting transformer's robustness and efficacy in ppg signal artifact detection with self-supervised learning," *arXiv preprint arXiv:2401.01013*, 2024.

[29] F. Pedregosa and et. al, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[30] N. Bjorck and et. al., "Understanding batch normalization," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.