DESiT-Lite: An optimized and quantized vision transformer model for wildfire detection in resource-constrained embedded environments

Paul Mahenina Randriamitsiry
Laboratory for Mathematical and Computer Applied to Development (LIMAD)
Fianarantsoa, Madagascar
mrandriamitsiry@gmail.com

Deric Claudio Vitasoa
Laboratory for Mathematical and Computer Applied to Development (LIMAD)
Fianarantsoa, Madagascar
dericclaudiovitasoa@gmail.com

William Germain Dimbisoa
Laboratory for Mathematical and Computer Applied to Development (LIMAD)
Fianarantsoa, Madagascar
dwilliamger@gmail.com

Hajarisena Razafimahatratra
Laboratory for Mathematical and Computer Applied to Development (LIMAD)
Fianarantsoa, Madagascar
hajarisena@yahoo.fr

Thomas Mahatody
Laboratory for Mathematical and Computer Applied to Development (LIMAD)
Fianarantsoa, Madagascar
tsmahatody@gmail.com

***Abstract*— Automatic fire detection from images is a major challenge for environmental disaster prevention, particularly in regions where monitoring infrastructure is limited. Vision Transformers (ViT) have recently demonstrated remarkable performance in visual classification tasks, but their computational complexity limits their use in resource-constrained systems. In this paper, we propose DESiT-Lite, an optimized and quantized version of the Data Efficient Satellite Image Transformer (DESiT) model, originally designed for fire detection from aerial and satellite images. To reduce the model's computational complexity and memory consumption, four complementary techniques are integrated: knowledge distillation, structural pruning, Quantization-Aware Training (QAT), and fine-tuning. Experimental results show that the distilled version of DESiT maintains acceptable accuracy after compression and quantization. Comparative experiments conducted with several lightweight reference architectures also highlight a significant reduction in computational requirements while maintaining satisfactory inference efficiency for resource-constrained embedded scenarios.**

Keywords*— *DESiT, DESiT-Lite, Fire detection, Knowledge distillation, Fine-tuning, Structural pruning, Quantization.

I. INTRODUCTION

Recent advances in computer vision models, particularly the emergence of visual Transformers, have significantly improved the accuracy of critical object and event detection systems [1, 2]. However, these models remain resource-intensive, limiting their deployment in complex computing environments, such as microcontrollers embedded in lightweight drones [3, 4]. In many developing countries, such as Madagascar, access to specialized computing hardware is limited, which hinders the adoption of Artificial Intelligence (AI) based solutions for environmental monitoring and early fire detection.

Image-based automated fire recognition is a crucial problem in environmental hazard prevention. Recent advances in deep learning have led to the emergence of increasingly powerful models, notably Vision Transformers (ViT), which outperform convolutional architectures in several demanding visual tasks [5]. However, the size of the model and its consumption of computing resources limit their deployment in constrained environments, particularly when detection must be performed in real time.

Faced with these challenges, research has recently focused on reducing, compressing, and optimizing Transformers to achieve lighter architectures while maintaining their representation capacity. Previous works [6 7, 8, 9] have demonstrated that it is possible to significantly reduce the size and complexity of models with minimal loss of accuracy. Post-training quantization, Quantization-Aware Training (QAT), dynamic pruning, and knowledge distillation are all techniques that have become essential for adapting vision models to strict computational constraints [10, 11].

Under these conditions, the Data Efficient Satellite Image Transformer (DESiT) model, a lightweight Transformer trained for fire detection in natural environments, already delivers high performance with 99.68% accuracy [12]. However, its current size remains too large to allow for its integration into embedded systems, particularly drones, Internet of Things (IoT) platforms, and edge AI devices, which generally have limited resources in terms of memory, computing power, and energy consumption.

In this context, deploying standard ViT remains challenging due to their high computational complexity and inference latency. Model compression therefore becomes essential to achieve an acceptable trade-off between detection accuracy, inference speed, and memory cost.

In this paper, the proposed framework combines several complementary techniques, including knowledge distillation, structured pruning, QAT, and fine-tuning to reduce model complexity while preserving high classification performance, , as in [13, 14].

The main contributions of this paper are summarized as:

- Proposal of a lightweight ViT architecture for embedded wildfire detection;
- Joint integration of knowledge distillation, structured pruning, and quantization;
- Comparative evaluation against lightweight Convolutional Neural Network (CNN) and Transformer architectures under identical experimental conditions;
- Analysis of trade-offs between accuracy, recall, latency, and memory efficiency for low-resource embedded systems.

II. Literature Review

Vision Transformers and their hybrid variants offer better modeling of spatial context, which is essential for fire and smoke detection, but their complexity limits their embedded deployment without optimization. We add a critical comparison of the strengths and weaknesses of each approach, and then deduce our choice of model.

In [15], (Sandler et al., 2018) propose MobileNetV2, a lightweight CNN architecture designed for embedded devices. This model has been used in several studies on fire detection from aerial images, achieving accuracies between 85% and 90%. Although MobileNetV2 is well suited to computational constraints, its limited ability to model the overall context reduces its performance in complex scenarios containing diffuse smoke or heterogeneous backgrounds.

In [16], (Tan et al., 2019) propose EfficientNet, a family of CNN by composite scaling. The EfficientNet-B0 model achieves high accuracy (over 93% on ImageNet) while maintaining relatively low complexity. This architecture has been widely adopted for the classification of aerial and satellite images. However, its operation relies exclusively on local convolutions, which limits its ability to capture global spatial relationships, particularly important for the detection of widespread phenomena such as smoke or fire fronts.

In [17], (Dosovitskiy et al., 2020) introduce the ViT, which applies self-directed attention mechanisms to image classification. The results demonstrate an excellent ability to model global dependencies, outperforming traditional CNNs in several vision tasks. However, ViT requires large volumes of data and significant computational resources, which limits its deployment on embedded or resource- constrained platforms.

In [18], (Touvron et al., 2021) propose Data-efficient Image Transformer (DeiT), an optimized variant of the ViT based on knowledge distillation. This approach allows transformers to be trained efficiently with less data, while maintaining competitive accuracy. Although DeiT improves learning efficiency, its computational cost remains high for direct deployment on real-time embedded systems, particularly on lightweight drones.

In [19], (Liu et al., 2021) explore a post-training quantization approach aimed at reducing the memory and computational costs of Vision Transformers for constrained scenarios. Experimental results show that this strategy maintains high performance while significantly reducing model complexity, achieving 81.29% accuracy with DeiT-B on ImageNet under near 8-bit quantization. However, the study highlights that the effectiveness and robustness of quantization remain highly dependent on the architecture of the transformer in question.

In [20], (Lu et al., 2023) study a hybrid architecture combining a convolutional network and a transformer for land cover classification based on high-resolution Unmanned Aerial Vehicle (UAV) images. The proposed model, called DE-UNet, uses a dual encoder where local features extracted by a CNN backbone are enriched by a Swin Transformer module to capture global context. However, although these results are promising for semantic classification tasks, the complexity of the model and the training time remain significant constraints for real-time implementation on embedded platforms, particularly in resource-constrained environments.

In [21], (Jangirova et al., 2025) propose a real-time aerial fire detection approach targeting resource-constrained embedded platforms. The system develops a lightweight model combining convolutional networks and vision transformers, optimized by knowledge distillation. Experimental results indicate that the distilled model is suitable for execution on constrained devices. However, the computational resources required for inference deployment on constrained devices remain a limitation, as the study does not provide a detailed analysis of on-device inference time, memory footprint, or energy consumption on low-power hardware.

In [22], (Nielsen et al., 2024) introduce BitNet b1.58, a large language model fully quantized using ternary arithmetic, in which each parameter is encoded using 1.58 bits. This approach significantly reduces memory consumption and computational cost while maintaining performance comparable to that of full-precision models. However, it is designed exclusively for natural language processing tasks and does not support structured visual inputs such as aerial imagery.

In this context, DESiT-Lite, by applying global attention through Transformer, is an effective compromise. Experimental comparisons conducted in our study show that it offers the best balance between accuracy and generalization ability. The quantified DESiT-Lite version thus aims to preserve these advantages while respecting the computational and memory constraints imposed by real-time embedded systems.

III. Proposed Methodology

This section presents the methodology adopted for the design, optimization, and evaluation of the proposed model. It details, in turn, the initial architecture of DESiT, the simplification strategy used to obtain the optimized DESiT-Lite version, the quantification process applied, and as the algorithm representing its mode of operation.

A. Description of the initial architecture of DESiT

The DESiT model is a hybrid architecture that combines the global dependency modeling techniques offered by ViT. It is based on a three-step sequence found in [12]:

1) Data preprocessing: Input images, derived from multitemporal satellite data, are resized and normalized using random rotation, cropping, brightness/contrast variation (ColorJitter), and pixel normalization. Pre-training on large datasets allows the model to learn generic visual features, promoting better generalization.

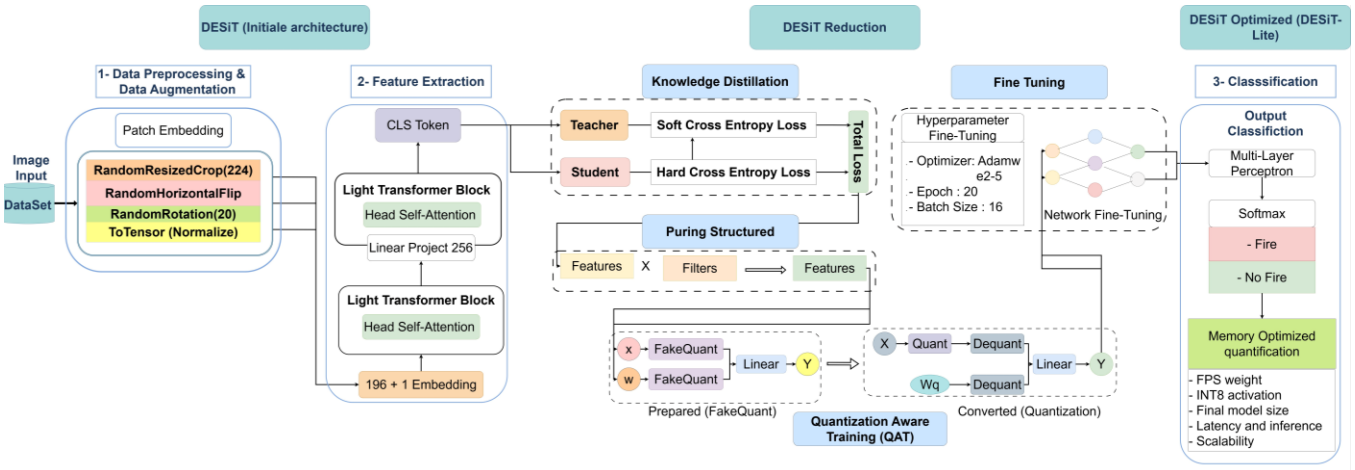

Fig. 1. Overall architecture of the proposed DESiT Optimized system, illustrating the main stages from satellite image preprocessing to feature extraction, classification, and final forest fire detection.

2) Feature extraction: The feature maps extracted by the convolutional layers are then cut into patches and projected into a latent space adapted to multi-head attention. This step preserves local details while preparing the data for global processing.

3) Classification: A compact transformer module analyzes long-range spatial relationships between patches. The final representations apply layer normalization, linear projection, and a 3-class softmax. The results are then aggregated and sent to a classification head, which produces the final prediction (fire/no fire, smoke/no smoke).

This architecture allows DESiT to achieve high accuracy of up to 99.68% during initial training while maintaining a structure compatible with subsequent optimizations.

B. *DESiT-Lite optimization strategy*

Fig. 1 illustrates the overall architecture of the proposed system, including knowledge distillation, structural pruning, quantization, fine-tuning, and final model classification. The optimization strategy is based on four complementary levers.

1) Knowledge Distillation: The system applies a limitation on the number of attention blocks while maintaining their accuracy performance [23]:

- The number of Transformer blocks is reduced;
- The size of embeddings and the number of attention heads are reduced;
- The student model learns not only to predict the final classes, but also to imitate the intermediate outputs of the teacher model.

2) Structural Pruning: Structural pruning was applied with an average pruning rate of 30% to the focus heads and certain fully connected layers of the Transformer. This reduction helps minimize the number of parameters while maintaining satisfactory classification performance [24]:

- Removal of redundant attention heads;
- Layer reduction transformation.

3) Quantization-Aware Training: The model operates in reduced size (also called INTeger 8-bit: INT8) on the weight, activation, and attention mechanism [11]:

- Reduction of memory size from 32 bits to 8 bits (int8);
- Transformation into a version compatible with TFLite.

4) Fine Tuning: DESiT-Lite is trained to reproduce the predictions of the original DESiT model, which preserves much of the accuracy while significantly reducing complexity, and also corrects the cumulative losses introduced by distillation, pruning, and quantization [25]:

- Reduction of 16x16 patches to 8x8;
- Final adjustment of quantified weights.

This multi-level optimization strategy ensures an optimal compromise between accuracy, computational complexity, and memory consumption, making the model particularly suitable for low-cost, embedded deployment scenarios while maintaining high detection performance.

C. *Quantization process*

Quantization aims to reduce the numerical precision of the model's parameters and activations in order to decrease its memory size, inference latency, and energy consumption, while preserving classification performance as much as possible.

In this work, quantization is a key step in making the DESiT model compatible with embedded platforms with limited resources.

In order to reduce the loss of accuracy induced by quantization, a QAT approach is adopted, simulating the effects of quantization during the learning phase [10]. Fig. 2 shows the QAT process.

During training, the weights, biases, and activations of each layer are subjected to a low-precision quantization operation (INT8) during the forward pass.

These factors allow floating values to be projected onto discrete space while limiting quantization error [26].

The gradient calculation (backward pass) is performed exclusively in floating point precision, ensuring optimization

stability and avoiding performance degradation due to numerical approximation [27].

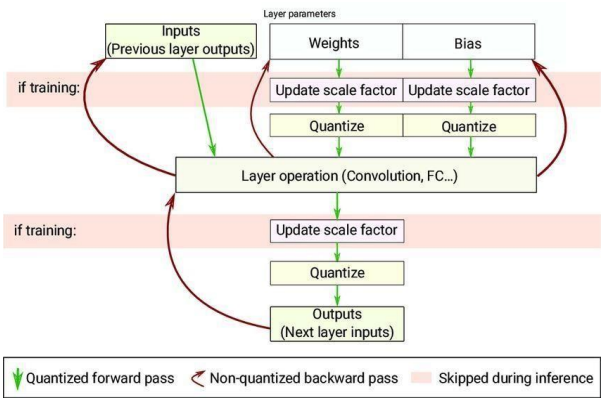

Fig. 2. QAT process applied to the DESiT optimized model to reduce computational complexity. [26]

D. Model operating algorithm

Algorithm 1 illustrates the overall operation of the DESiT-Lite model quantified from an input image.

Algorithm 1: DESiT-Lite Training and Inference Pipeline.

Input: Aerial image $I \in \mathbb{R}^{HxWx3}$

Output: Predicted class $\hat{y} \in$ {fire, no_fire}

```
Preprocessing
    I_aug ← Resize(I) + RandomFlip(I)
    I_pr ← Normalize(ViTImageProcessor(I_aug))
    → pixel_values ∈ ℝ³×224×224
Teacher–Student Initialization
    Teacher ← DESiT-FP32 (initial DeiT-based model)
    Student ←Lightweight DESiT architecture
knowledge Distillation
    Train the student model
Structured Pruning
    Remove low-importance attention heads and Channels
    Student_Pruned ← compact Student network
Quantization-Aware Training
    Insert fake-quantization modules (INT8)
    Train the model under quantization constraints
Fine-tuning
    Fine-tune on the training set
    Select best model based on validation accuracy
    → Model_DESiT-Lite
Inference & Evaluation
    ŷ ← argmax (Model_DESiT-Lite(I_proc))
Evaluate
Return: ŷ
```

IV. Validation Strategy

All experiments were conducted on Google Colab, on an instance equipped with an NVIDIA T4 GPU, 51 GB of RAM, and 15 GB of GPU memory, optimized for deep learning. The execution environment is based on Transformers 4.52.4, TorchVision, and Python 3.8.5, on a 64-bit machine.

A. Case studies

For experimental validation, a combined dataset containing approximately 47 992 images was constructed from multiple public wildfire detection datasets [28, 29, 30]. The data were grouped into two main classes: “fire” and “no_fire”. The “Fire” dataset contains images captured in controlled environments, while the “Forest Fire” dataset includes more diverse experimental scenes.

The flame dataset provides UAV aerial images simulating real-time conditions with varying textures, illumination levels, and complex backgrounds. To ensure experimental reproducibility, the dataset was divided using a 70% training, 15% validation, and 15% testing split strategy.

In addition, five-fold cross-validation was performed to improve the statistical robustness of the evaluation. To reduce the impact of class imbalance, a weighted cross-entropy loss function was adopted during training.

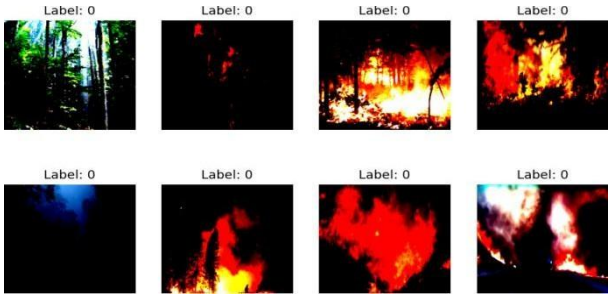

Fig. 3. Representative image extracts from the real dataset with different textures.

The visual extracts shown in Fig. 3 illustrate the intra- class diversity of representative images from the simulated real-world dataset.

B. Performance evaluation

The DESiT-Lite model is trained over 20 epochs, using an early stopping strategy to avoid overfitting [31]. The AdamW optimizer is used for its stability when fine-tuning transformer-based models, while a weighted Cross-Entropy loss function is adopted to correct the imbalance between the fire and no_fire classes.

1) Experimental parameters: All models compared were trained under identical experimental conditions, as described in [11, 25]. The main training and quantization parameters are summarized in Table I.

TABLE I. EXPERIMENTAL PARAMETERS

Parameters	Value and description
Basic architecture	DeiT
Quantization type	INT8 [26]
Input image size	224 × 224 pixels
Patch size	Reduced from 16×16 to 8×8 pixels
Loss function	CrossEntropyLoss
Optimizer	AdamW e2-5 [32]
Transformations applied	Rotation, ColorJitter, RandomResizedCrop, Normalize
Number of epochs	20
Batch size	16

2) Visualization of the validation curve: Fig. 4 and Fig. 5 illustrate, for each fold, the cross-validation of the validation loss and the validation accuracy of the quantified DESiT-Lite model.

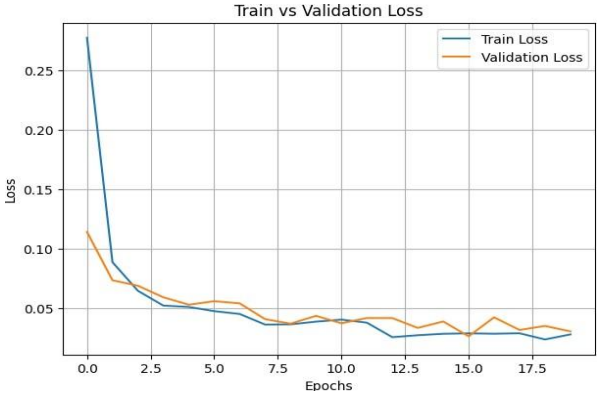

Fig. 4. Cross-validation of model performance showing the evolution of validation loss.

Fig. 5. Cross-validation of model performance showing the evolution of validation accuracy.

The final average values are summarized in Table II.

TABLE II. FINAL AVERAGE VALUES

Metric	Average value (%)
Accuracy	99.48
Precision	99.47
Recall	84.50
F1-score	91.37

3) Analysis of the Overall Confusion Matrix: Fig. 6 shows the overall confusion matrix of the DESiT-Lite model.

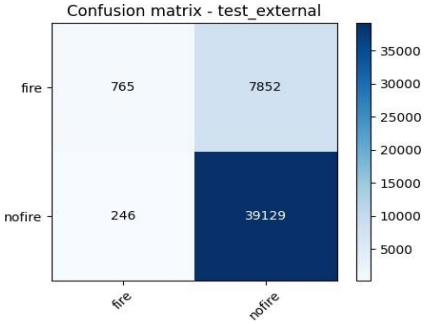

Fig. 6. Overall confusion matrices for the DESiT-Lite model illustrating the classification performance between the fire and no_fire classes.

C. Identify the Headings

In order to better evaluate the performance of DESiT-Lite, three state-of-the-art architectures were retrained and evaluated under the same experimental conditions, with a view to their performance, as noted in [7, 15, 16].

Fig. 7 presents an overall comparison of the performance of the four lightweight models for forest fire detection. The comparison of the values obtained during the experiment is presented in Table III.

TABLE III. EXPERIMENTAL COMPARISON

Model	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)	Latency (%)	FPS (%)
DESiT-Lite	99.48	99.47	91.37	84.50	93	92
MobileViT	85.89	86.02	84.62	83.26	90	89
Mobile NetV2	80.38	80.30	80.11	79.92	86	87
EfficientNet-B0	80.00	82.00	79.95	78.00	88	90

V. DISCUSSION

Experimental results show that DESiT-Lite preserves high accuracy after compression and INT8 quantization while significantly reducing computational complexity and memory consumption.

However, the recall remains relatively lower, which can be attributed to the cumulative effects of knowledge distillation, structured pruning, and quantization, potentially reducing model sensitivity to small fire regions or visually ambiguous smoke patterns a critical limitation for safety-oriented wildfire detection systems.

Despite this, DESiT-Lite offers an interesting trade-off between accuracy, memory efficiency, and computational cost, making it suitable for near real-time embedded inference rather than strict real-time detection (0.92 FPS).

Although no validation has yet been conducted on physical platforms such as Raspberry Pi, Jetson Nano, or Coral Edge, the parameter reduction and INT8 quantization suggest potential compatibility with such architectures.

Future directions include recall improvement strategies, energy consumption evaluation, and the exploration of neuromorphic vision sensors for event-based fire detection.

VI. VCONCLUSION

This paper presented DESiT-Lite, an optimized and quantized version of the DESiT model designed for resource-constrained embedded environments for wildfire detection from aerial and satellite imagery. The proposed framework combines several complementary compression techniques, including knowledge distillation, structured pruning, QAT, and fine-tuning.

Experimental results demonstrate that DESiT-Lite achieves high accuracy after compression and INT8 quantization while significantly reducing model complexity compared to the original DESiT architecture.

However, despite satisfactory overall performance, the obtained recall remains relatively lower for a safety-critical application such as wildfire detection, emphasizing the

importance of further reducing false negatives under complex real-world conditions.

Future work will focus on energy consumption evaluation, real embedded hardware deployment, recall improvement through class balancing and focal loss strategies, as well as the exploration of advanced Transformer compression techniques for edge AI and UAV applications.

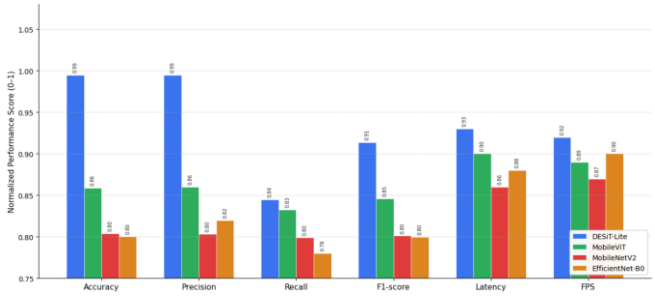

Fig. 7. Diagram showing overall performance comparison of the four lightweight models, to assess their respective effectiveness.

REFERENCES

[1] A. B. Amjoud et M. Amrouch, « Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review », *IEEE Access*, vol. 11, p. 35479-35516, 2023, doi: 10.1109/ACCESS.2023.3266093.

[2] Y. Li, H. Mao, R. Girshick, et K. He, « Exploring Plain Vision Transformer Backbones for Object Detection », in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, et T. Hassner, Éd., Cham: Springer Nature Switzerland, 2022, p. 280-296. doi: 10.1007/978-3-031-20077-9_17.

[3] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, et M. Shah, « Transformers in Vision: A Survey », *ACM Comput. Surv.*, vol. 54, nº 10s, p. 1-41, janv. 2022, doi: 10.1145/3505244.

[4] H. Kheddar, Y. Habchi, M. C. Ghanem, M. Hemis, et D. Niyato, « Recent Advances in Transformer and Large Language Models for UAV Applications », 15 août 2025, *arXiv*: arXiv:2508.11834. doi: 10.48550/arXiv.2508.11834.

[5] K. Zheng et X. Yang, « Design and Implementation of Lightweight Vision Transformer for Low-Power Edge Devices », in *2025 5th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, févr. 2025, p. 1-4. doi: 10.1109/ICCECE65250.2025.10985633.

[6] J. Lin, W.-M. Chen, Y. Lin, john cohn, C. Gan, et S. Han, « MCUNet: Tiny Deep Learning on IoT Devices », in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, p. 11711-11722. Consulté le: 29 décembre 2025.

[7] S. Mehta et M. Rastegari, « MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer », 4 mars 2022, *arXiv*: arXiv:2110.02178. doi: 10.48550/arXiv.2110.02178.

[8] K. Wu *et al.*, « TinyViT: Fast Pretraining Distillation for Small Vision Transformers », in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, et T. Hassner, Éd., Cham: Springer Nature Switzerland, 2022, p. 68-85. doi: 10.1007/978-3-031-19803-8_5.

[9] Z. Chen, F. Zhong, Q. Luo, X. Zhang, et Y. Zheng, « EdgeViT: Efficient Visual Modeling for Edge Computing », in *Wireless Algorithms, Systems, and Applications*, L. Wang, M. Segal, J. Chen, et T. Qiu, Éd., Cham: Springer Nature Switzerland, 2022, p. 393-405. doi: 10.1007/978-3-031-19211-1_33.

[10] H. Wu, P. Judd, X. Zhang, M. Isaev, et P. Micikevicius, « Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation », 20 avril 2020, *arXiv*: arXiv:2004.09602. doi: 10.48550/arXiv.2004.09602.

[11] N.-D. Ho et I. J. Chang, « O-2A: Ourlier-Aware Compression for 8-bit Post-Training Quantization Model », *IEEE Access*, vol. PP, p. 1-1, janv. 2023, doi: 10.1109/ACCESS.2023.3311027.

[12] P. M. Randriamitsiry, D. C. Vitasoa, W. G. Dimbisoa, H. Razafimahatratra, et T. Mahatody, « Data Efficient Satellite Image Transformer (DESiT): A Hybrid Transformer Model for Robust Satellite Image Classification in Forest Monitoring », in *2025 5th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, oct. 2025, p. 1-8. doi: 10.1109/ICECCME64568.2025.11277915.

[13] P. Zhang, J. Wei, J. Zhang, J. Zhu, et J. Chen, « Accurate INT8 Training Through Dynamic Block-Level Fallback », 9 juin 2025, *arXiv*: arXiv:2503.08040. doi: 10.48550/arXiv.2503.08040.

[14] N. Andalib et M. Selimi, « Effective Quantization Technique for Enhancing Model Performance on Resource-Constrained Devices », in *2025 14th Mediterranean Conference on Embedded Computing (MECO)*, juin 2025, p. 1-4. doi: 10.1109/MECO66322.2025.11049283.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, et L.-C. Chen, « MobileNetV2: Inverted Residuals and Linear Bottlenecks », présenté à Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, p. 4510-4520.

[16] M. Tan et Q. Le, « Efficientnet: Rethinking model scaling for convolutional neural networks », in *International conference on machine learning*, PMLR, 2019, p. 6105-6114.

[17] A. Dosovitskiy, « An image is worth 16x16 words: Transformers for image recognition at scale », *ArXiv Prepr. ArXiv201011929*, 2020.

[18] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, et H. Jegou, « Training data-efficient image transformers & distillation through attention », in *International Conference on Machine Learning*, PMLR, juill. 2021, p. 10347-10357.

[19] Z. Liu, Y. Wang, K. Han, W. Zhang, S. Ma, et W. Gao, « Post-Training Quantization for Vision Transformer », in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, p. 28092-28103.

[20] T. Lu, L. Wan, S. Qi, et M. Gao, « Land Cover Classification of UAV Remote Sensing Based on Transformer–CNN Hybrid Architecture », *Sensors*, vol. 23, nº 11, p. 5288, janv. 2023, doi: 10.3390/s23115288.

[21] S. Jangirova, B. Jankovic, W. Ullah, L. U. Khan, et M. Guizani, « Real-time aerial fire detection on resource-constrained devices using knowledge distillation », *Int. J. Appl. Earth Obs. Geoinformation*, vol. 142, p. 104665, août 2025, doi: 10.1016/j.jag.2025.104665.

[22] J. Nielsen et P. Schneider-Kamp, « BitNet B1.58 Reloaded: State-of-the-Art Performance Also on Smaller Networks », in *Deep Learning Theory and Applications*, vol. 2172, A. Fred, A. Hadjali, O. Gusikhin, et C. Sansone, Éd., in Communications in Computer and Information Science, vol. 2172. , Cham: Springer Nature Switzerland, 2024, p. 301-315. doi: 10.1007/978-3-031-66705-3_20.

[23] A. Moslemi, A. Briskina, Z. Dang, et J. Li, « A survey on knowledge distillation: Recent advancements », *Mach. Learn. Appl.*, vol. 18, p. 100605, déc. 2024, doi: 10.1016/j.mlwa.2024.100605.

[24] J. Wang, G. Li, et W. Zhang, « Combine-Net: An Improved Filter Pruning Algorithm », *Information*, vol. 12, p. 264, juin 2021, doi: 10.3390/info12070264.

[25] A. Heikal, A. El-Ghamry, S. Elmougy, et M. Z. Rashad, « Fine tuning deep learning models for breast tumor classification », *Sci. Rep.*, vol. 14, nº 1, p. 10753, mai 2024, doi: 10.1038/s41598-024-60245-w.

[26] P.-E. Novac, G. Boukli, A. Pegatoquet, B. Miramond, et V. Gripon, *Quantization and Deployment of Deep Neural Networks on Microcontrollers*. 2021. doi: 10.48550/arXiv.2105.13331.

[27] J. Lee, J. Jeon, D. Kim, et B. Ham, « Scheduling Weight Transitions for Quantization-Aware Training », présenté à Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, p. 23466-23475.

[28] K. Akyol, « An innovative hybrid method utilizing fused transformer-based deep features and deep neural networks for detecting forest fires », *Adv. Space Res.*, 2025.

[29] A. Khan, B. Hassan, S. Khan, R. Ahmed, et A. Abuassba, « DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection », *Mob. Inf. Syst.*, vol. 2022, p. 1-14, avr. 2022, doi: 10.1155/2022/5358359.

[30] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, et P. Fulé, « Aerial images for pile fire detection using drones (UAVs) ». IEEE DataPort, 19 novembre 2020. doi: 10.21227/QAD6-R683.

[31] H. Wang, T. Li, Z. Zhuang, T. Chen, H. Liang, et J. Sun, « Early Stopping for Deep Image Prior », 11 décembre 2023, *arXiv*: arXiv:2112.06074. doi: 10.48550/arXiv.2112.06074.

[32] D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », 30 janvier 2017, *arXiv*: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.