GLUSE: Enhanced Channel-Wise Adaptive Gated Linear Units SE for Onboard Satellite Earth Observation Image Classification

Thanh-Dung Le, *Senior Member, IEEE*, Vu Nguyen Ha, *Senior Member, IEEE*, Ti Ti Nguyen, *Member, IEEE*, Duc-Dung Tran, *Member, IEEE*, Hung Nguyen-Kha, Luis M. Garces-Socarras, *Member, IEEE*, Juan Carlos Merlano-Duncan, *Senior Member, IEEE*, Symeon Chatzinotas, *Fellow, IEEE*

***Abstract*—This study introduces ResNet-GLUSE, a lightweight ResNet variant enhanced with Gated Linear Unit-enhanced Squeeze-and-Excitation (GLUSE), an adaptive channel-wise attention mechanism. By integrating dynamic gating into the traditional SE framework, GLUSE improves feature recalibration while maintaining computational efficiency. Experiments on EuroSAT and PatternNet datasets confirm its effectiveness, achieving exceeding 94% and 98% accuracy, respectively. While MobileViT achieves 99% accuracy, ResNet-GLUSE offers 33× fewer parameters, 27× fewer FLOPs, 33× smaller model size (MB), ≈6× lower power consumption (W), and ≈3× faster inference time (s), making it significantly more efficient for onboard satellite deployment. Furthermore, due to its simplicity, ResNet-GLUSE can be easily mimicked for neuromorphic computing, enabling ultra-low power inference at just 852.30 mW on Akida Brainchip. This balance between high accuracy and ultra-low resource consumption establishes ResNet-GLUSE as a practical solution for real-time Earth Observation (EO) tasks. Reproducible codes are available in our shared repository.**

***Impact Statement*– ResNet-GLUSE adds a lightweight, dynamic gated attention to a small ResNet, reaching high accuracy with a fraction of the computation, memory, and power required by existing models, enabling real-time image analytics directly on satellite edge hardware. This synergy of performance, scalability, and energy efficiency accelerates rapid decision-making in resource-constrained orbital environments, aiding critical tasks like near-real-time hazard detection and precision agriculture. Reproducible codes promote broad adoption and facilitate ongoing innovation, underscoring ResNet-GLUSE's potential as a transformative solution for next-generation EO missions.**

***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]. 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 [3]. 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 [4].

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 [5]. 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 [6].

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 [7], [8]. In line with this limitation, NASA has emphasized that the benefits of fine-tuning smaller, lightweight models for oriented tasks currently outweigh the costs and risks associated with large models [9].

ResNet architectures have emerged as a compelling solution by effectively addressing CNNs' vanishing gradient issues through skip connections, achieving a favorable balance between computational efficiency and performance [10]–[12].

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

Thanh-Dung Le, Vu Nguyen Ha, Ti Ti Nguyen, Duc-Dung Tran, Hung Nguyen-Kha, Luis M. Garces-Socarras, Juan Carlos Merlano-Duncan, Symeon Chatzinotas are with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg (Corresponding author. Email: thanh-dung.le@tamucc.edu).

This paper is a revised and expanded version of a paper entitled "Semantic Knowledge Distillation for Onboard Satellite Earth Observation Image Classification", which was presented at IEEE ICMLCN 2025, Barcelona, Spain, 26–29 May 2025.

Recent studies further illustrate that employing knowledge distillation (KD) [13] from pretrained ViT models can significantly enhance lightweight ResNet models by transferring semantic knowledge, thereby improving their performance to levels comparable to ViTs, while maintaining practicality for onboard satellite deployment [14].

Motivated by these findings, this study aims to further enhance the lightweight ResNet model through improved channel-wise feature recalibration. Specifically, we propose GLUSE, an adaptive channel-wise attention mechanism inspired by Gated Linear Units (GLU) [15] integrated into the Squeeze-and-Excitation (SE) framework [16]. GLUSE is designed to optimize the performance-complexity trade-off, maintaining suitability for on-the-air deployment.

Experimental validations demonstrate that the proposed ResNet-GLUSE consistently surpasses traditional SE methods across benchmark EO datasets (EuroSAT [17] and PatternNet [18]), both with and without KD from pre-trained ViT models. Remarkably, ResNet-GLUSE achieves over 94% and 98% overall evaluation metrics (accuracy, precision, and recall) on EuroSat and PatternNet datasets, respectively. Furthermore, it exhibits notable power efficiency, consuming approximately six times less energy (13.8 W) compared to pre-trained MobileViT (79.23 W) on GPU. Due to its structural simplicity and computational efficiency, the ResNet-GLUSE model is readily adaptable for neuromorphic Akida edge computing, achieving ultra-low inference power consumption (852.30 mW).

The contributions of this paper are three-fold:

- We propose GLUSE, an adaptive GLU-inspired channel-wise attention mechanism, achieving superior performance compared to traditional SE blocks.
- We comprehensively evaluate the robustness and effectiveness of lightweight ResNet-GLUSE in both standard training and KD scenarios, confirming its advantage.
- Leveraging its simplicity and efficiency, ResNet-GLUSE is highly suitable for ultra-low-power neuromorphic onboard satellite deployment.

II. Related Work

KD has been widely utilized to boost the performance of lightweight student models, such as ResNet, for onboard satellite processing by transferring knowledge from powerful pre-trained ViT teacher models [14]. Although KD significantly enhances the student model's generalization capabilities, a noticeable performance gap remains between the ResNet student and the large pre-trained ViT teachers, indicating a need for additional architectural improvements. The study [14] confirms that even when leveraging semantic information from the teacher model to make student predictions more effective, a considerable gap of approximately 6% still exists. Alternative strategies, such as increasing the student model's complexity by making it larger and deeper, have been proposed; however, these approaches come with the drawback of increased computational complexity and, consequently, power consumption.

Alternatively, channel-wise feature recalibration offers a promising strategy for enhancing student model capacity. The SE framework [16] is a notable example, as it boosts feature expressiveness without adding extra computational overhead. Specifically, SE blocks assign importance weights to each channel, recalibrating channel activations. However, this mechanism is static, applying the same learned attention weights uniformly across all spatial locations. As a consequence, critical features may be suppressed, limiting the model's adaptability to various spatial contexts [19].

Recent advancements have attempted to enhance SE attention mechanisms. Some studies integrate dense layers into SE blocks to strengthen global context representation [20]. In contrast, others propose lightweight global context modules, such as the Global Context block, to improve feature modeling [21]. Although these methods improve the learning capacity of lightweight models, they introduce additional computational complexity, making them less suitable for neuromorphic computing, which requires strict architectural simplicity for efficient conversion into spiking neural networks (SNN) [22], [23]. Furthermore, achieving ultra-low power consumption is essential for satellite-based communication, where energy efficiency is a critical constraint [24].

To address these limitations, we propose GLUSE. This GLUSE module retains the computational simplicity of SE while introducing adaptive gating for dynamic channel-wise recalibration. GLUSE balances computational efficiency and enhanced feature learning, making it well-suited for GPU-based execution and edge neuromorphic computing, enabling efficient onboard satellite deployment where power consumption and real-time processing are critical.

The foundation of GLUSE is inspired by GLU [25], [26], which selectively emphasizes or suppresses information based on task requirements. Gating mechanisms have been widely adopted in architectures such as Gated Transformer Networks [27] and Temporal Fusion Transformers [28], demonstrating their ability to improve feature selectivity. Mathematically, as described in [28], a GLU applies a component gating mechanism that modulates the contribution of an input η:

$$\mathrm{GLU}_{\omega(\eta)} = \sigma(W_{1,\omega}\eta + b_{1,\omega}) \odot (W_{2,\omega}\eta + b_{2,\omega}), \quad (1)$$

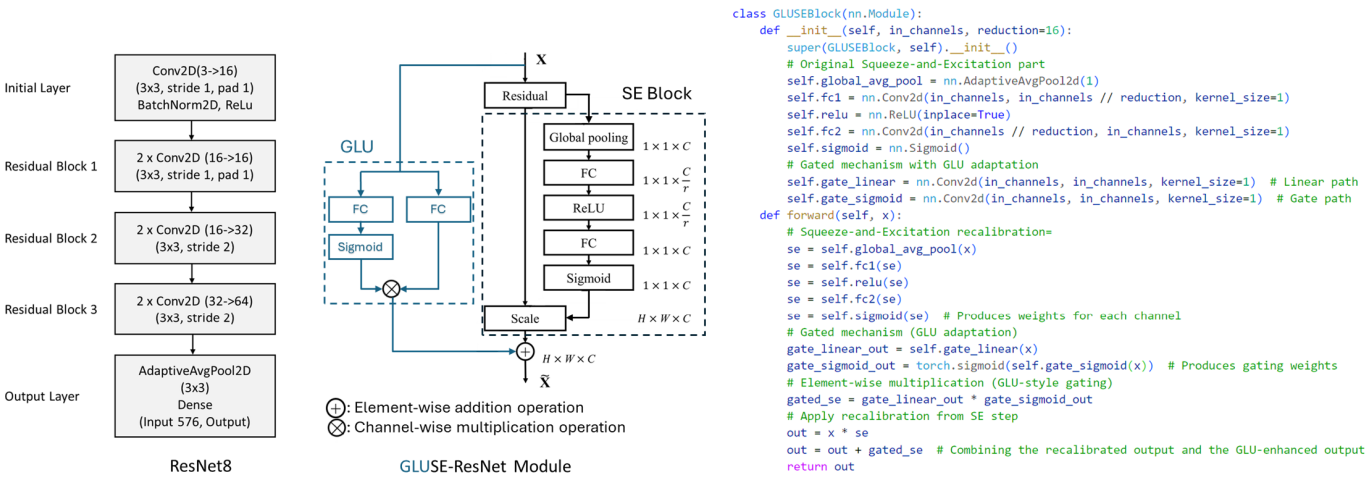
where $W_{(.)}$, $b_{(.)}$ are the weights and biases, $\odot$ is the element-wise Hadamard product, and $\sigma(\cdot)$ denotes the sigmoid activation function:

$$\sigma(x) = \frac{\exp(x)}{1+\exp(x)}. \quad (2)$$

GLUs allow selective suppression of irrelevant components by controlling their contribution, effectively skipping nonlinear transformations when necessary, especially enhancing the mutual information between hidden representations, confirmed by [26]. By integrating this concept into SE, GLUSE provides a more adaptive and efficient recalibration mechanism.

III. Proposed Approach

Let the input feature map be denoted by: $X \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are height, width, and channel dimensions, respectively.




```python
class GLUSEBlock(nn.Module):
    def __init__(self, in_channels, reduction=16):
        super(GLUSEBlock, self).__init__()
        # Original Squeeze-and-Excitation part
        self.global_avg_pool = nn.AdaptiveAvgPool2d(1)
        self.fc1 = nn.Conv2d(in_channels, in_channels // reduction, kernel_size=1)
        self.relu = nn.ReLU(inplace=True)
        self.fc2 = nn.Conv2d(in_channels // reduction, in_channels, kernel_size=1)
        self.sigmoid = nn.Sigmoid()
        # Gated mechanism with GLU adaptation
        self.gate_linear = nn.Conv2d(in_channels, in_channels, kernel_size=1)  # Linear path
        self.gate_sigmoid = nn.Conv2d(in_channels, in_channels, kernel_size=1)  # Gate path
    def forward(self, x):
        # Squeeze-and-Excitation recalibration=
        se = self.global_avg_pool(x)
        se = self.fc1(se)
        se = self.relu(se)
        se = self.fc2(se)
        se = self.sigmoid(se)  # Produces weights for each channel
        # Gated mechanism (GLU adaptation)
        gate_linear_out = self.gate_linear(x)
        gate_sigmoid_out = torch.sigmoid(self.gate_sigmoid(x))  # Produces gating weights
        # Element-wise multiplication (GLU-style gating)
        gated_se = gate_linear_out * gate_sigmoid_out
        # Apply recalibration from SE step
        out = x * se
        out = out + gated_se  # Combining the recalibrated output and the GLU-enhanced output
        return out
```

Fig. 1: A lightweight ResNet backbone, ResNet-GLUSE Module, and GLUSE snipet code in Pytorch.

A. Lightweigh ResNet

As shown in Fig. 1, the lightweight ResNet variant begins with an initial convolutional layer, followed by three residual blocks that progressively expand the feature channels from 16 to 64. These blocks utilize convolutional layers with varying strides (1 or 2) to balance feature extraction and downsampling. The architecture concludes with adaptive average pooling and fully connected layers. With its shallow depth, the proposed ResNet in this study is optimized for computationally constrained environments, ensuring efficiency while maintaining representational capacity.

B. Squeeze-and-Excitation (SE) Block

Channel-wise feature recalibration techniques, particularly the SE block, have proven highly effective in enhancing learning models' representational power, including CNN [29], and ResNet [30]. SE blocks explicitly model inter-channel dependencies by first applying global average pooling to compress spatial information, followed by two fully connected layers to dynamically recalibrate channel importance. The resultant scalar weights refine feature representations, improving model performance with minimal additional complexity.

The SE block recalibrates channel-wise features through two main steps. First, the **Squeeze** step to compute the global average pooling (GAP):

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j), \quad \mathbf{z} \in \mathbb{R}^C. \tag{3}$$

Then, the **Excitation** step will perform recalibration through two fully connected convolutional layers:

$$\mathbf{s} = \sigma(W_2(\delta(W_1\mathbf{z}))). \tag{4}$$

where, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$, $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$, $\delta(\cdot)$ is ReLU activation, $\sigma(\cdot)$ is sigmoid. And, r denotes the reduction ratio used in the SE blocks.

After that, the final recalibration will follow by:

$$\tilde{x}_c = s_c \odot x_c, \quad s_c \in (0,1). \tag{5}$$

This recalibration from Eq. 5 produces channel-wise static recalibration from the SE block.

C. Gated Linear Units SE (GLUSE) Block

Despite SE's effectiveness, their recalibration weights are static during inference, potentially limiting adaptation to varying spatial contexts [19]. This study introduces additional adaptive gating operations, enabling recalibration weights to respond dynamically to spatially varying feature distributions.

It is inspired by recent advances in adaptive gating mechanisms, particularly the success of GLU [15] in capturing complex feature interactions. Unlike traditional gating methods, GLU employs two parallel convolutional pathways: one linear and one gating path, which effectively learn complementary representations, as shown in Fig. 1. Integrating GLU-inspired gating with the SE framework, the proposed GLUSE approach combines static recalibration from SE with dynamic feature refinement via GLU, enhancing recalibration with greater adaptivity. Initially, GLUSE employs the standard SE procedure. This step provides a channel-wise recalibration that emphasizes informative channels while suppressing redundant ones, precisely as the same SE block from Eq. 5.

$$\mathbf{s} = \sigma(W_2(\delta(W_1\mathbf{z}))), \quad x_{se} = \mathbf{x} \odot \mathbf{s}. \tag{6}$$

Then, GLUSE introduces a gating mechanism inspired by GLU to enhance feature refinement adaptivity further. Specifically, two parallel convolutional operations are performed directly on the original feature map $\mathbf{x}$:

Linear transform path:

$$\mathbf{h} = W_h * \mathbf{x} \tag{7}$$

where W_h is a convolutional kernel.

Gating path:

$$\mathbf{g} = \sigma(W_g * \mathbf{x}) \tag{8}$$

where W_g is another convolutional kernel.

Consequently, this dual-path approach allows GLUSE to dynamically adapt to local feature patterns by generating precise gating masks. Then, the linear path $\mathbf{h}$ and gating path $\mathbf{g}$ outputs are combined via element-wise multiplication, following GLU principles:

$$x_{glu} = \mathbf{h} \odot \mathbf{g}. \tag{9}$$

The GLU operation enables adaptive selection and transformation of features based on their localized importance and

relevance, thereby providing richer contextual representation and improved discrimination.

Finally, the SE recalibration output (x_{se}) and the adaptive GLU gating output (x_{glu}) are summed to form the final enhanced feature representation $\hat{x}$:

$$\hat{x} = x_{se} + x_{glu}. \tag{10}$$

This final step ensures that both global channel-wise information (from SE) and adaptive local gating (from GLU) collaboratively improve the feature quality. This approach enhances feature selection by prioritizing significant channels while reducing the influence of less relevant ones. The network can concentrate on critical information by dynamically modulating channel importance, leading to improved feature representation and effective extraction from each channel.

D. Gated SE Block

To effectively compare and validate the improvements introduced by GLUSE, we also examine the Gated SE block as an intermediate enhancement to the standard SE mechanism. Motivated by the ECA-Net [31], the Gated SE block introduces an additional gating mechanism via a 1×1 convolution, dynamically recalibrating the weights based on spatial context. This modification enhances SE's adaptability, making feature modulation more flexible and responsive to input variations.

Improving upon the SE block by adding adaptive gating. First, as in the SE block, the recalibration weights $\mathbf{s}$ are computed as:

$$\mathbf{s} = \sigma(W_2(\delta(W_1\mathbf{z}))). \tag{11}$$

where W_1 and W_2 are fully connected layers, $\delta(\cdot)$ is a ReLU activation, and $\sigma(\cdot)$ is the sigmoid function.

Instead of directly applying $\mathbf{s}$ to scale the feature map $\mathbf{x}$, the Gated SE introduces an adaptive gate $\mathbf{g}$ via a 1×1 convolution:

$$\mathbf{g} = \sigma(W_g * \mathbf{s}), \tag{12}$$

where W_g is a learnable 1×1 convolution kernel. This additional gating modulates the SE-derived recalibration, allowing dynamic adjustments based on local spatial variations.

Finally, the recalibrated feature map is then obtained by applying the gating function to the original input. This makes the recalibration dynamic, ensuring that channel importance is adjusted contextually rather than being statically assigned.

$$\hat{x} = \mathbf{x} \odot \mathbf{g}. \tag{13}$$

Table I summarizes the differences between three block SE, Gated SE, and GLUSE. The SE block performs static channel-wise recalibration using GAP followed by fully connected layers. The Gated SE block is an intermediate step between the traditional SE block and the proposed GLUSE mechanism. While Gated SE introduces dynamic gating through a single convolution, it cannot fully adapt feature transformations, as it only modifies the SE weights. The proposed GLUSE further enhances adaptivity by incorporating GLU-inspired gating, utilizing parallel convolutional paths to capture more intricate feature interactions. Consequently, while SE recalibration is static, Gated SE offers moderate adaptability, and GLUSE achieves highly adaptive feature recalibration.

The computational complexity analysis, summarized in Table II, demonstrates that while the proposed GLUSE slightly increases computational overhead compared to the plain ResNet, SE, and Gated SE blocks, this increase remains modest and manageable. Specifically, the GLUSE structure introduces two additional lightweight convolutional operations (linear and gating paths), resulting in an approximate complexity of $\mathcal{O}(HWC + \frac{C^2}{r} + 2C^2)$, compared to $\mathcal{O}(HWC + \frac{C^2}{r} + C^2)$ for Gated SE, and $\mathcal{O}(HWC + \frac{C^2}{r})$ for standard SE blocks.

IV. Experiments

A. Datasets

This study addresses a spectrum of EO tasks - from image classification to image retrieval—while exploring lower and higher spatial resolutions. To this end, we employ two widely recognized benchmark datasets, EuroSAT [17] and PatternNet [18], summarized in Table III.

EuroSAT is a Sentinel-2-based benchmark for land use and land cover classification, comprising about 27,000 labeled, geo-referenced images at 64x64 pixel resolution across 13 spectral bands. It is divided into 10 classes and offers a balanced dataset for onboard deep learning methods targeting real-time tasks like environmental monitoring and precision agriculture. In contrast, PatternNet targets image retrieval through 30,400 high-resolution images (256x256 pixels) spanning 38 categories. This diverse coverage supports the development of retrieval-focused approaches for tackling more complex, fine-grained RS challenges.

B. Training Strategy

To validate the effectiveness and robustness of GLUSE, we evaluate its performance under two distinct training scenarios: (1) **Standard training**, where ResNet-GLUSE is trained conventionally without additional supervision, and (2) **Dual-teacher KD training**, where knowledge is transferred from large pretrained ViTs to enhance the model's generalization.

Standard training, in this approach, ResNet-GLUSE is trained directly using conventional supervised learning. While this setup ensures a fair baseline comparison against existing SE and Gated SE models, its performance is inherently limited by the available training data and model capacity. With KD training, to further enhance model performance while maintaining computational efficiency for onboard satellite deployment, we employ KD - a technique where a compact student model learns from larger, more expressive teacher models [32]. Proposed initially to reduce deep learning models' computational burden [13], KD has since evolved as a powerful method for enabling lightweight models to acquire complex representations while achieving competitive accuracy [33].

In this study, we adopt dual-teacher KD to distill semantic knowledge from ViTs into ResNet-GLUSE. Traditional KD methods, which enforce strict alignment with a single teacher's predictions via Kullback-Leibler (KL) divergence [34], often suffer from training instability and suboptimal performance when the teacher model is uncertain [35]. To mitigate these limitations, we introduce dynamic weighting in dual-teacher

TABLE I: Detailed comparison among SE variants.

Operation	**SE Block**	**Gated SE Block**	**GLUSE**
Squeeze	$z = \text{GAP}(x)$	same as SE	same as SE
Excitation	$\sigma(W_2\delta(W_1 z))$	same as SE	same as SE
Gating Mechanism	none	$\sigma(W_g * s)$	$\mathbf{h} = W_h * x,\ \mathbf{g} = \sigma(W_g * x)$
Final Recalibration	$x \odot s$	$x \odot g$	$(x \odot s) + (\mathbf{h} \odot \mathbf{g})$
Dynamic Gating	No	Yes	Enhanced (GLU-based)
Feature Interaction	Static	Moderate	Highly adaptive

TABLE II: Computational complexity comparison of plain ResNet with SE, Gated SE, and GLUSE.

Structure	**Computational Steps**	**Complexity (approx.)**
Plain ResNet	Conv: $3 \times 3 \times C \times C$	$\mathcal{O}(HWC^2)$
SE	Global Avg Pool: $\mathcal{O}(HWC)$ FC Layers: $\frac{C^2}{r} + \frac{C^2}{r} = \frac{2C^2}{r}$ Scaling: $\mathcal{O}(HWC)$	$\mathcal{O}(HWC + \frac{C^2}{r})$
Gated SE	Gate Conv: $C \times C \times 1 \times 1$ Gating Operation: $\mathcal{O}(HWC)$	$\mathcal{O}(HWC + \frac{C^2}{r} + C^2)$
GLUSE	Linear Conv: $C \times C \times 1 \times 1$ Gate Conv: $C \times C \times 1 \times 1$ Gating Operation: $\mathcal{O}(HWC)$ Combination (addition): $\mathcal{O}(HWC)$	$\mathcal{O}(HWC + \frac{C^2}{r} + 2C^2)$

TABLE III: Comparison of two satellite imagery datasets for EO oriented task.

Dataset	**Classes**	**Images/Class**	**Images**	**Resolution (m)**	**Size**	**Task**
EuroSAT [17]	10	2000–3000	27,000	0.3	64×64	Classification
PatternNet [18]	38	800	30,400	0.062–4.693	256×256	Retrieval

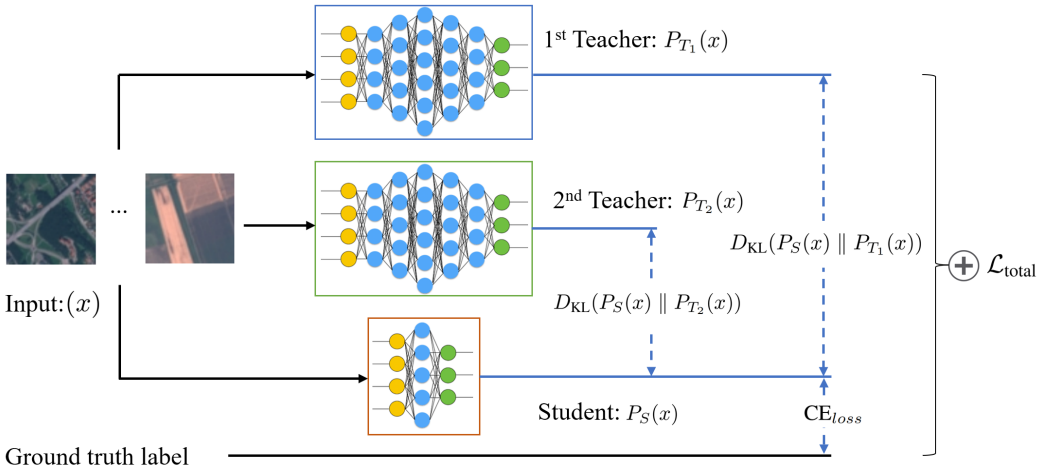


Fig. 2: The schematic workflow of dynamic weighting in dual-teacher KD.

KD (Algorithm 1), where the weight assigned to each teacher is adjusted based on confidence scores. This adaptive strategy enables the student to prioritize the more reliable knowledge source, improving generalization across data distributions.

As shown in Fig. 2, given an input x, the semantic distillation process starts by computing softened probability distributions for the teacher models and the student model. This is achieved by scaling their logits with a temperature parameter τ. For teacher model T_1, the softened probability distribution is:

$$P_{T_1}(x) = \text{softmax}\left(\frac{T_1(x)}{\tau}\right), \tag{14}$$

and similarly for teacher model T_2:

$$P_{T_2}(x) = \text{softmax}\left(\frac{T_2(x)}{\tau}\right), \tag{15}$$

with the student model S:

$$P_S(x) = \text{softmax}\left(\frac{S(x)}{\tau}\right). \tag{16}$$

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

$$C_{T_1} = \mathbb{E}\left[\max(P_{T_1}(x))\right], \quad C_{T_2} = \mathbb{E}\left[\max(P_{T_2}(x))\right]. \quad (17)$$

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 student's and each teacher's softened probabilities, is then computed, with the weights α and β reflecting each teacher's confidence.

$$\text{KD}_{\text{loss}} = \alpha \cdot D_{\text{KL}}(P_S(x) \parallel P_{T_1}(x)) + \beta \cdot D_{\text{KL}}(P_S(x) \parallel P_{T_2}(x)), \quad (18)$$

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

$$D_{\text{KL}}(P_S(x) \parallel P_{T_i}(x)) = \frac{1}{\tau^2} \sum_j P_{T_i}(x)_j \log\left(\frac{P_{T_i}(x)_j}{P_S(x)_j}\right) \quad (19)$$

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 student's predictions and the true labels is calculated to ground the student's learning in teacher guidance and actual labels, where:

$$\text{CE}_{\text{loss}} = -\sum_i y_i \log\left(P_S(x)_i\right), \quad (20)$$

Then, the final combined loss, $\mathcal{L}_{\text{total}}$, integrates these components: a weighted combination of the CE_{loss} and the 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.

$$\mathcal{L}_{\text{total}} = \left(1 - \frac{\alpha + \beta}{2}\right) \cdot \text{CE}_{\text{loss}} + \frac{\alpha + \beta}{2} \cdot \text{KD}_{\text{loss}} \quad (21)$$

Recent work [8], [36] has identified EfficientViT and MobileViT as the two most effective ViTs for EO image classification tasks. Accordingly, we select EfficientViT [37] and MobileViT [38] as our teacher models. By combining ResNet-GLUSE with dynamic dual-teacher KD, we aim to maximize accuracy and resource efficiency.

C. Implementation

The provided PyTorch implementation GLUSE from Fig. 1 demonstrates a simple, modular, and easily integrable design. The code separates the original SE recalibration from the additional GLU-inspired gating mechanism, utilizing standard PyTorch operations and minimal custom code. With a few lightweight convolutional layers and straightforward tensor operations, this block can effortlessly be incorporated into existing neural network architectures. Its modular nature ensures minimal impact on overall computational complexity while enhancing the adaptivity of channel-wise recalibration. Consequently, the GLUSE module is efficient and readily applicable to various neural network backbones due to its simplicity and flexibility.

Algorithm 1 Dynamic Weighting Dual-Teacher KD

Require: Input batch (x, y), student model S, teacher models T_1 and T_2, temperature τ, confidence threshold δ, minimum weight w_{min}
Ensure: Combined loss $\mathcal{L}_{total}$ for backpropagation

Forward Pass:
  Compute logits: $l_S \leftarrow S(x)$, $l_{T1} \leftarrow T_1(x)$, $l_{T2} \leftarrow T_2(x)$
  Compute softened predictions using temperature τ:
  $P_S \leftarrow \text{softmax}(l_S/\tau)$, $P_{T1} \leftarrow \text{softmax}(l_{T1}/\tau)$, $P_{T2} \leftarrow \text{softmax}(l_{T2}/\tau)$
Compute Teacher Confidence Scores:
  $C_{T1} \leftarrow \text{mean}(\max(P_{T1}))$, $C_{T2} \leftarrow \text{mean}(\max(P_{T2}))$
Dynamic Weight Adjustment:
if $C_{T1}, C_{T2} < 0.4$ **then**
  $\alpha, \beta \leftarrow 0.0, 0.0$ // Both teachers ignored
else if $C_{T1} < \delta$ **and** $C_{T2} < \delta$ **then**
  $\alpha \leftarrow \max(0.5 - (\delta - C_{T1}), w_{min})$
  $\beta \leftarrow \max(0.5 - (\delta - C_{T2}), w_{min})$
else if $C_{T1} < \delta$ **then**
  $\alpha, \beta \leftarrow 0.3, 0.7$ // Prioritize confident Teacher 2
else if $C_{T2} < \delta$ **then**
  $\alpha, \beta \leftarrow 0.7, 0.3$ // Prioritize confident Teacher 1
else
  $\alpha, \beta \leftarrow 0.5, 0.5$ // Equal weighting
end if
Compute Knowledge Distillation (KD) Loss:
  $\text{KD}_{loss} \leftarrow \tau^2 \left[\alpha \cdot D_{KL}(P_S \parallel P_{T1}) + \beta \cdot D_{KL}(P_S \parallel P_{T2})\right]$
Compute Classification Loss (Cross-Entropy):
  $\text{CE}_{loss} \leftarrow -\sum_i y_i \log(P_S)_i$
Combine Losses:
  $w \leftarrow \frac{\alpha+\beta}{2}$
  $\mathcal{L}_{total} \leftarrow (1-w)\text{CE}_{loss} + w\text{KD}_{loss}$
Backpropagate $\mathcal{L}_{total}$

TABLE IV: Experiment parameters setting

Parameter	Value
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

All the experiments are conducted on GPU NVIDIA RTX™ 6000 Ada Generation, 48 GB GDDR6. Experiments were implemented using the Scikit-learn library [39], and PyTorch. For the inference, we also tested the proposed model running on the in-lab Akida BrainChip edge neuromorphic computing. The data was divided into 70% training and 30% testing. The data transformation pipeline applies standard preprocessing steps to ensure compatibility with various neural network backbones [40]. First, it resizes images to 64×64 pixels, converts them to tensors, and normalizes pixel values using the mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$, which aligns with the input requirements of widely used pretrained models, including ViTs and ResNets. Finally, the detailed experiment parameters setting are summarized in Table IV.

To comprehensively evaluate the performance of our multiclass classification model across classes, we employ three key metrics: accuracy, precision, and recall (sensitivity) [41]. These metrics are calculated for each class individually and then aggregated using macro-averaging to assess the model's performance as follows:

$$\text{Accuracy} = \sum_{k=1}^{K} \frac{\text{TP}_k}{N} \tag{22}$$

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^{K} N_k \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k} \tag{23}$$

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^{K} N_k \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \tag{24}$$

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.

V. Results and Discussion

The experimental learning curves from Fig. 3, and 4 demonstrate that the proposed ResNet-GLUSE consistently outperforms the plain ResNet, ResNet-SE, and ResNet-gated-SE architectures, both with and without KD. Without KD, ResNet-GLUSE achieves faster convergence, higher validation accuracy ($\approx 91\%$), and lower validation loss, highlighting the efficacy of the adaptive gating mechanism in feature recalibration. With KD, all models significantly improve; however, ResNet-GLUSE shows notably superior performance, quickly converging and maintaining validation accuracy around 94-95%, surpassing other models by a clear margin. The consistently lowest loss values further confirm that GLUSE effectively leverages the distilled knowledge from large pretrained ViT teachers. Additionally, it is vital to note that an early stopping strategy was employed to mitigate overfitting, revealing that training without KD converged after 25 epochs, while with KD, achieving stable convergence required 100 epochs.

TABLE V: Comparison of model performance on EuroSat.

Models	Accuracy (↑)	Precision (↑)	Recall (↑)
Pretrained Models			
EfficientViT	98.76	98.77	98.76
MobileViT	**99.09**	**99.09**	**99.09**
w/o KD			
ResNet	87.76	87.70	87.76
ResNet-SE	90.54	90.50	90.54
ResNet-gated-SE	90.19	90.19	90.19
ResNet-GLUSE	91.05	91.1	91.05
with KD			
ResNet	92.88	93.07	92.88
ResNet-SE	93.49	93.47	93.49
ResNet-gated-SE	93.07	93.04	93.07
ResNet-GLUSE	**94.63**	**94.61**	**94.63**

Bold denotes the best values.
Bold and underline denote the second best values.

TABLE VI: Comparison of model performance on PatternNet.

Models	Accuracy (↑)	Precision (↑)	Recall (↑)
EfficientViT	99.52	99.52	99.52
MobileViT	**99.66**	**99.66**	**99.66**
w/o KD			
ResNet	80.18	79.42	80.18
ResNet-SE	86.34	86	86.34
ResNet-gated-SE	86.02	85.80	86.02
ResNet-GLUSE	88.16	87.93	88.16
with KD			
ResNet	97.73	97.70	97.73
ResNet-SE	98.02	98	98.02
ResNet-gated-SE	97.98	97.98	97.98
ResNet-GLUSE	**98.09**	**98.09**	**98.09**

Bold denotes the best values.
Bold and underline denote the second best values.

Furthermore, across the EuroSat and PatternNet datasets, as shown in Table V and VI, the proposed ResNet-GLUSE consistently outperforms the standard ResNet, ResNet-SE, and ResNet Gated SE architectures. Specifically, without KD, the ResNet-GLUSE achieves higher accuracy, precision, and recall, demonstrating that the adaptive gating mechanism inspired by GLU significantly enhances the traditional SE framework by providing more effective channel-wise feature recalibration. When leveraging KD, ResNet-GLUSE further boosts its performance substantially, achieving metrics closely comparable to the top-performing, with only a slight dip in accuracy. Particularly on PatternNet, ResNet-GLUSE reaches 98.09% accuracy (99.66% for MobileViT), and on EuroSat, it attains 94.63% accuracy (99.09% for MobileViT).

As shown in Table VII, the proposed ResNet8-GLUSE exhibits remarkable computational efficiency compared to large pretrained ViT-based models (EfficientViT and MobileViT). Although ResNet8-GLUSE is slightly larger and more computationally demanding than the plain ResNet8, ResNet8-SE, and ResNet8-gated-SE variants (with parameters at 131,565 and FLOPs at 66.56M), it remains significantly simpler, smaller, and more efficient than MobileViT (4.39M parameters, 1.84G FLOPs). Specifically, ResNet8-GLUSE achieves approximately 33 times fewer parameters and 27 times fewer FLOPs than MobileViT, translating into substantially lower inference power consumption (13.80W vs. 79.23W). These results underline that ResNet-GLUSE, especially combined with KD, can achieve competitive performance levels while offering significant computational efficiency, making it highly suitable for onboard satellite image classification and retrieval.

More importantly, due to its architectural simplicity and compactness, the ResNet-GLUSE model can easily be mimicked, adapted, and deployed on the Akida Brainchip neuromorphic computing platform [42]. Experimental deployment on Akida hardware further highlights its exceptional efficiency, with an extremely low inference power consumption averaging 877 mW, achieving an inference energy consumption of just 182.42 mJ/frame, and maintaining a practical frame rate of 4.81 fps. Such results highlight the potential of the ResNet-GLUSE model to operate efficiently on neuromorphic hardware, enabling energy-efficient onboard image analysis in resource-constrained environments. From the boxplot accuracy distribution, depicted in Fig. 5, the model's accuracy ranges

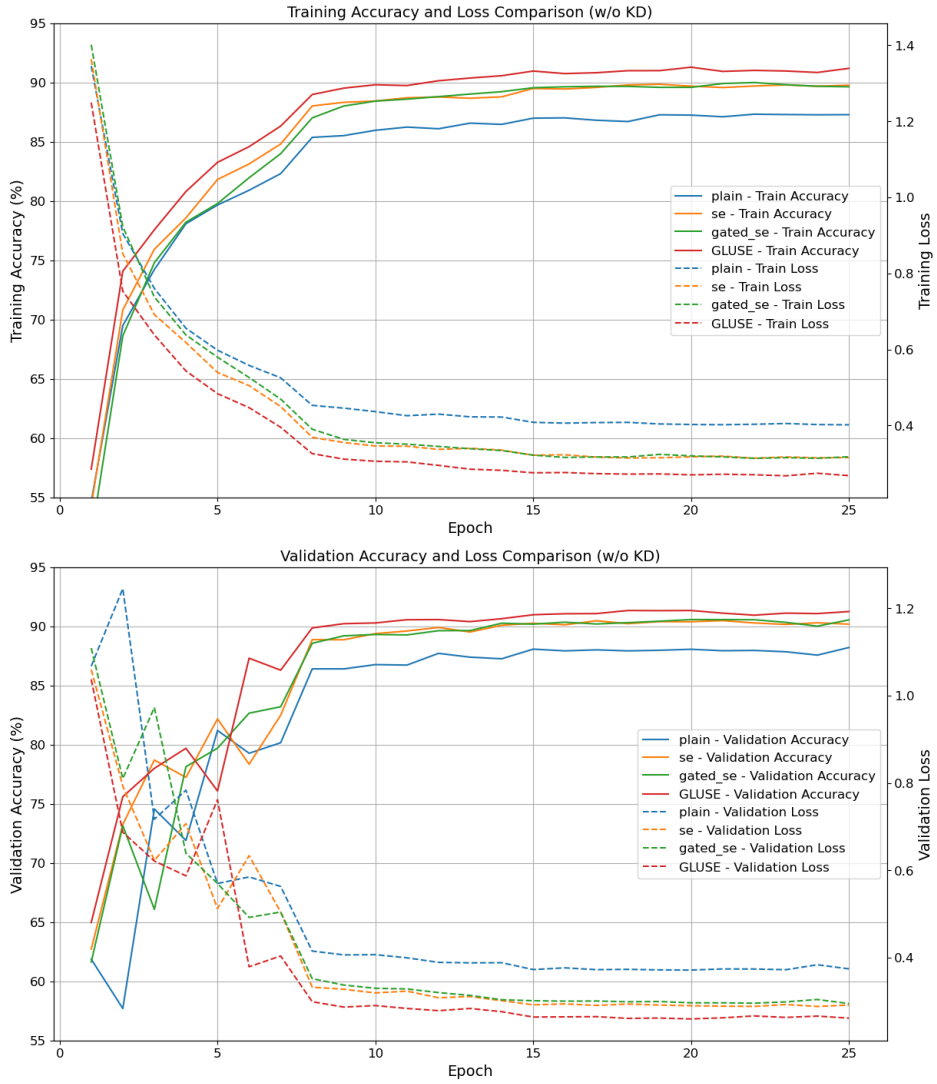


Fig. 3: Training (Top) and validation (Bottom) learning curve from standard training strategy.

TABLE VII: 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**
ResNet8-SE	120,589	60,151,840	6.04	5.87	11.51 ± 1.58
ResNet8-gated-SE	126,077	60,271,904	6.06	5.99	13.05 ± 0.74
ResNet8-GLUSE	131,565	66,557,984	7.92	6.01	13.80 ± 1.39
EfficientViT	3,964,804	203,533,056	38.19	10	29.04 ± 0.96
MobileViT	4,393,971	1,843,303,424	259.30	16	79.23 ± 1.45

Bold denotes the best values.

from about 93% to nearly 96%, with a median of approximately 94.7%, demonstrating stable and high performance across multiple runs, further confirming its robust inference performance under varied operational conditions.

Furthermore, the Grad-CAM [43] visualization qualitatively confirms the effectiveness of the proposed ResNet-GLUSE compared to plain ResNet, SE, and Gated-SE models, as shown in Fig. 6. Across diverse classes (“highway,” “annual crop,” “airplane,” and particularly “christmas tree farm”), GLUSE consistently produces more transparent and more precise activation maps aligned closely with the ground truth. Notably, in the “christmas tree farm” scenario, the GLUSE model distinctly captures individual tree distributions, highlighting fine-grained features clearly, while other methods yield significantly noisier and less structured activations. The adaptive gating mechanism within GLUSE effectively recalibrates channel-wise attention, enhancing the interpretability and specificity of the learned features. This qualitative evidence further validates GLUSE’s ability to identify relevant features accurately. This hybrid approach from GLU and SE significantly enhances adaptivity and recalibration precision, improving representation clarity and classification.

Lastly, comparing the diagonal entries in the two confusion matrices reveals that ResNet-GLUSE produces $\approx 8\%$ higher overall accuracy than the baseline ResNet, translating to an increase of roughly 660+ correct predictions on this 9,000-

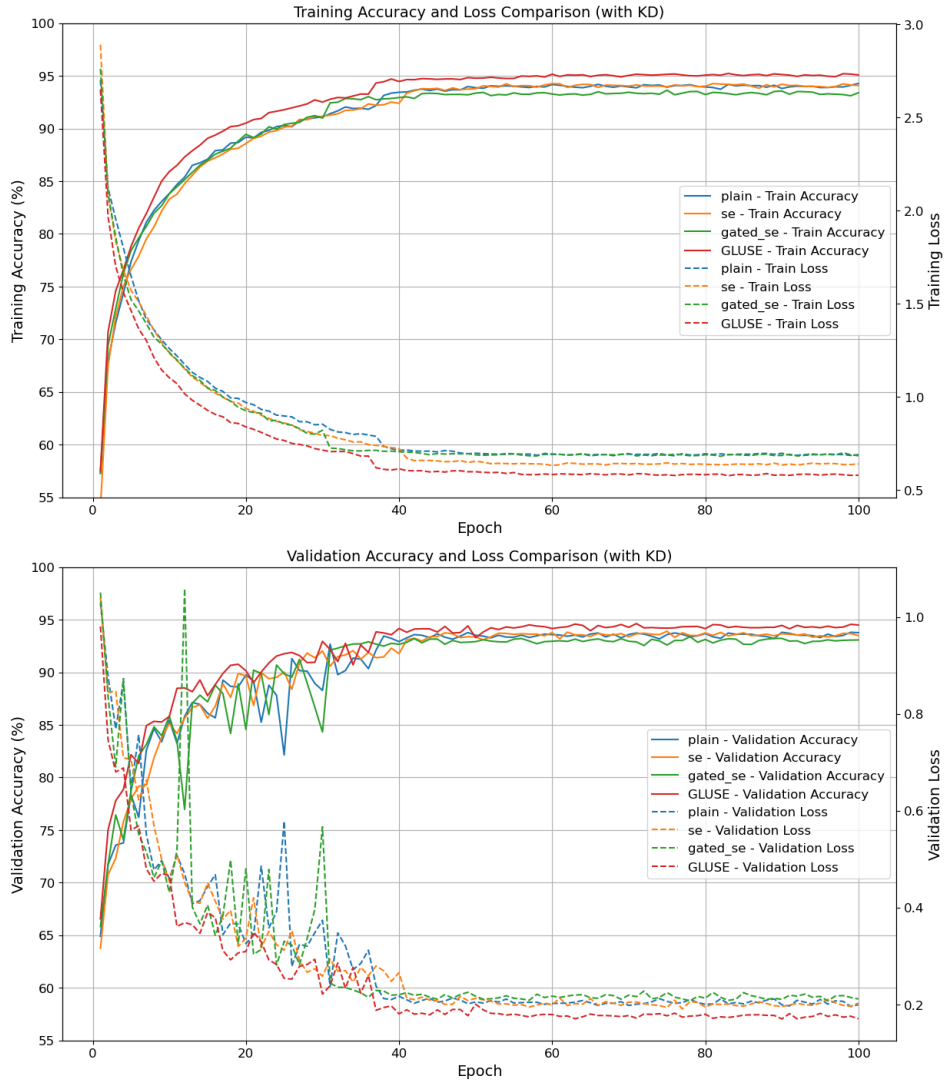


Fig. 4: Training (Top) and validation (Bottom) learning curve with KD training strategy.

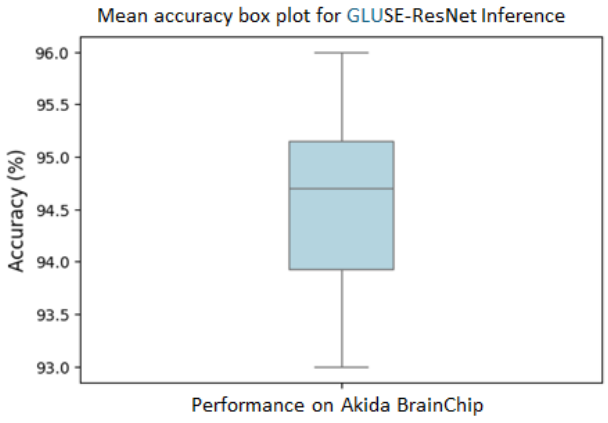


Fig. 5: Performance of the ResNet-GLUSE during the inference on Akida BrainChip neuromorphic computing hardware.

image test set, shown in Fig. 7. Notably, performance in challenging classes such as Highway improves substantially, with correct classifications rising from 438 to 652, and River also sees a significant jump from 536 to 678. In contrast, classes already well-modeled by the baseline (e.g., SeaLake) show similar high performance under ResNet-GLUSE. These results underscore the model's capacity to discriminate more effectively among visually similar categories, boosting both recall and overall accuracy. A similar trend is also observed in the PatternNet dataset, where ResNet-GLUSE achieves outstanding performance across nearly all classes, illustrated in Fig. 8. From the PatternNet confusion matrix, each class is predicted with near-perfect accuracy—for instance, airplane achieves 239 correct predictions (99.6%) out of 240 samples. By combining knowledge distillation with a more capable student architecture, ResNet-GLUSE demonstrates clear and consistent gains across a wide range of EO classes.

VI. Conclusions

This study presents GLUSE, an adaptive channel-wise calibration mechanism that enhances lightweight ResNet models for onboard EO image classification. Extensive experiments on EuroSAT and PatternNet confirm that ResNet-GLUSE outperforms SE and Gated SE models, consistently exceeding

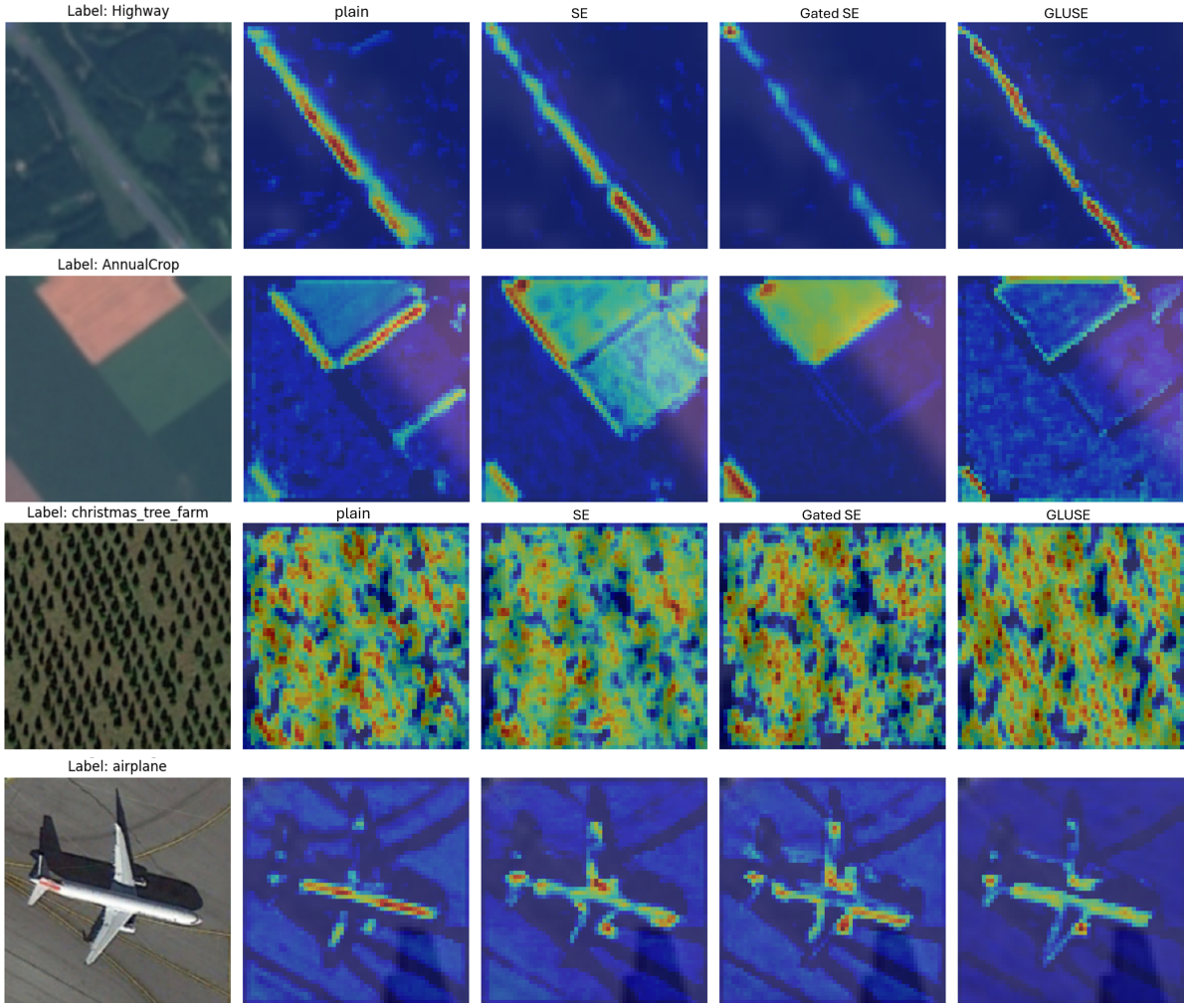


Fig. 6: Qualitative results of different channel-wise attention for the EuroSat, and PatternNet data by using Grad-CAM.

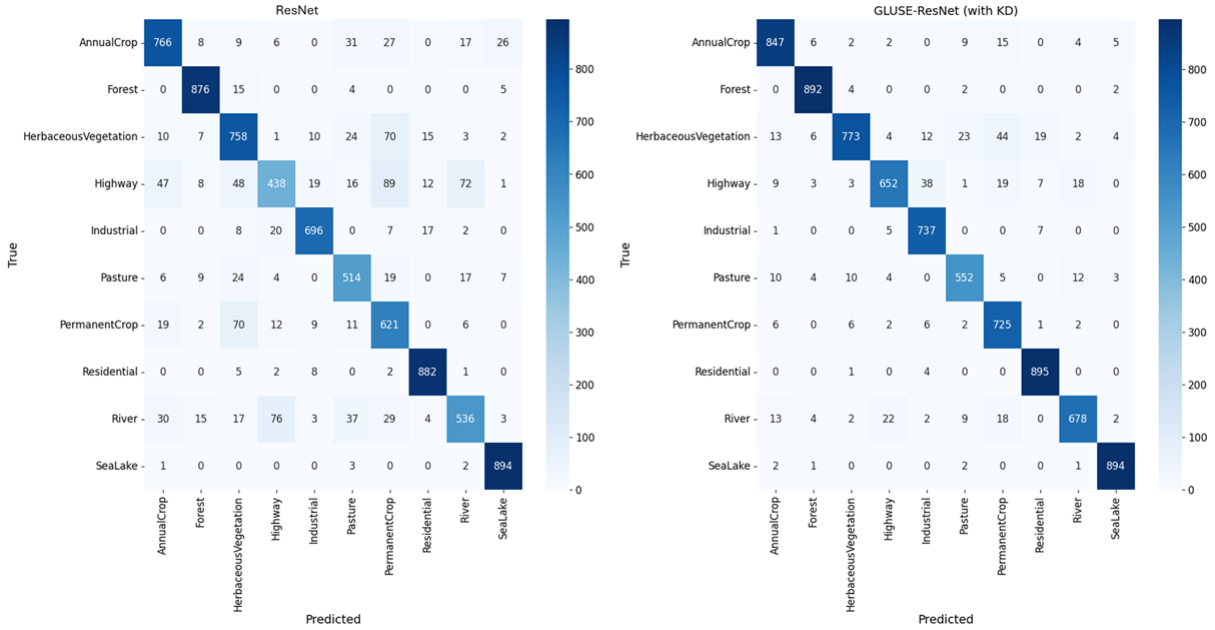


Fig. 7: Confusion matrix from ResNet (left) and ResNet-GLUSE with KD (right) on the EuroSat dataset.

94% accuracy, precision, and recall across different training scenarios. With KD, it approaches ViT-level performance achieving over 98% accuracy while maintaining a significantly lower computational footprint.

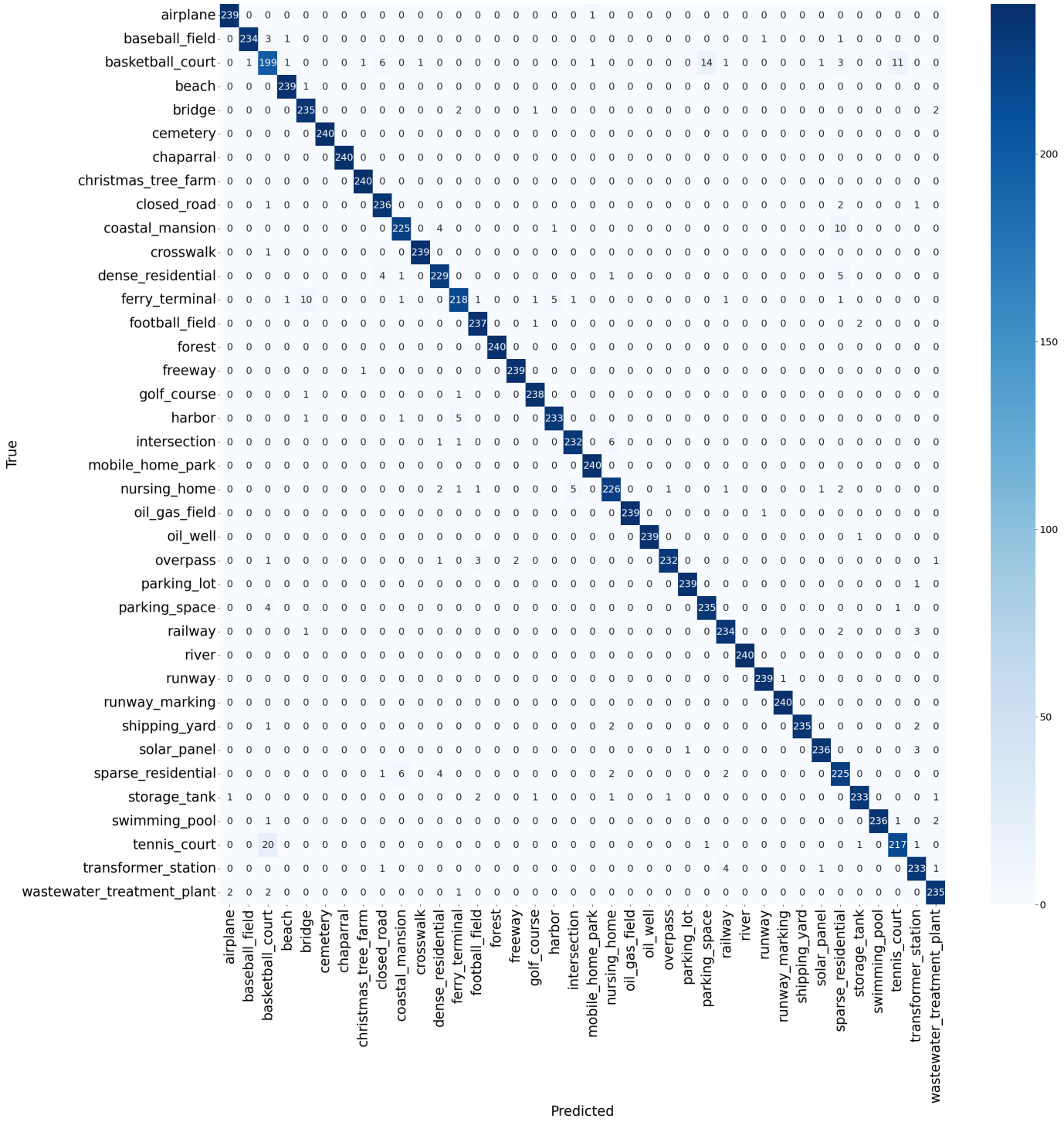


Fig. 8: Confusion matrix from ResNet-GLUSE with KD on the PatternNet dataset.

ResNet-GLUSE optimizes accuracy, efficiency, and resource consumption. It reduces energy consumption by up to 6× compared to MobileViT on GPUs and enables ultra-low power inference (852.30 mW) on Akida Brainchip. Despite a slight complexity increase over SE, its adaptive gating mechanism delivers substantial performance gains with minimal overhead.

For future work, we will extend ResNet-GLUSE to semantic segmentation and object detection tasks, enabling its applicability to more complex EO vision-based problems. Additionally, we plan to integrate multimodal data, including synthetic aperture radar (SAR) and hyperspectral imaging, to enhance further its robustness and generalization in diverse satellite-based remote sensing applications.

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] H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "A survey on nongeostationary satellite systems: The communication perspective," *IEEE Commun. Surveys & Tuts.*, vol. 25, no. 1, pp. 101–132, 2022.

[3] H. Chougrani, *et al.*, "Connecting space missions through ngso constellations: feasibility study," *Frontiers in Commun. Netw.*, vol. 5, p. 1356484, 2024.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] R. Ramachandran and K. Bugbee, "Balancing practical uses and ethical concerns: The role of large language models in scientific research," *Perspectives of Earth and Space Scientists*, vol. 6, no. 1, p. e2024CN000258, 2025.

[10] 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.

[11] M. Goldblum, *et al.*, "Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[12] Y. Haruna, *et al.*, "Exploring the synergies of hybrid convolutional neural network and vision transformer architectures for computer vision: A survey," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110057, 2025.

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

[14] T.-D. Le, *et al.*, "Semantic knowledge distillation for onboard satellite earth observation image classification," *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2025.

[15] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[17] 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.

[18] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 197–209, 2018.

[19] S. Pereira, *et al.*, "Adaptive feature recombination and recalibration for semantic segmentation with fully convolutional networks," *IEEE transactions on medical imaging*, vol. 38, no. 12, pp. 2914–2925, 2019.

[20] M. Narayanan, "Aggregated dense layer in squeeze and excitation networks," in *Intelligent Systems Conference*. Springer, 2024, pp. 510–525.

[21] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[22] J. K. Eshraghian, *et al.*, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, 2023.

[23] J. Yik, *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature Communications*, vol. 16, no. 1, p. 1545, 2025.

[24] F. Ortiz, *et al.*, "Energy-efficient on-board radio resource management for satellite communications via neuromorphic computing," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 169–189, 2024.

[25] Y. N. Dauphin and et. al., "Language modeling with gated convolutional networks," in *International Conference on ML*, 2017, pp. 933–941.

[26] T.-D. Le, *et al.*, "Transformer meets gated residual networks to enhance picu's ppg artifact detection informed by mutual information neural estimation," *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[27] M. Liu and et. al., "Gated transformer networks for multivariate time series classification," *arXiv preprint arXiv:2103.14438*, 2021.

[28] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[29] N. V., Y. G., N. N. B., M. R., and P. P., "Empirical analysis of squeeze and excitation-based densely connected cnn for chili leaf disease identification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 4, pp. 1681–1692, 2024.

[30] S. K. Roy, S. Chatterjee, S. Bhattacharyya, B. B. Chaudhuri, and J. Platoš, "Lightweight spectral–spatial squeeze-and- excitation residual bag-of-features learning for hyperspectral classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5277–5290, 2020.

[31] Q. Wang, *et al.*, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 534–11 542.

[32] 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.

[33] 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.

[34] T. Van Erven and P. Harremos, "Rényi divergence and kullback-leibler divergence," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[35] 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.

[36] T. T. Nguyen, *et al.*, "A semantic-loss function modeling framework with task-oriented machine learning perspectives," *arXiv preprint arXiv:2503.09903*, 2025.

[37] 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.

[38] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *Trans. Mach. Learn. Res.*, vol. 2023, 2023. [Online]. Available: https://openreview.net/forum?id=tBl4yBEjKi

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

[40] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[41] 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.

[42] Akida Brainchip. Accessed: March 13, 2025. [Online]. Available: https://brainchip.com/akida-neural-processor-soc/

[43] R. R. Selvaraju, *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.