Performance Analysis of Deep Learning Models for Campus Scene Recognition: A Comparative Study of Attention Mechanisms

Yarah Al-Hindi, Malek Alsaudi, Sahar Abdulhay, Tasneem Al-Dweiri

January 2026

Abstract

This paper presents a comprehensive benchmarking of deep learning architectures for scene recognition within a university campus environment. We evaluate five distinct model configurations, ranging from heavyweight feature extractors (DenseNet121, ResNet18) to mobile-optimized architectures (MobileNetV2, EfficientNet-B0). To address the challenge of limited data, we employ a transfer learning strategy utilizing a frozen backbone. Furthermore, we investigate the impact of integrating advanced attention mechanisms, specifically the Convolutional Block Attention Module (CBAM) and Coordinate Attention (CA). Our experimental results reveal a distinct architectural split: while heavyweight models achieve peak performance (99.15%) with CBAM, lightweight models suffer performance degradation with CBAM but achieve optimal results (98.03%) with Coordinate Attention. This study validates that Coordinate Attention is the superior choice for edge-device deployment, offering a better balance of spatial preservation and computational efficiency.

Keywords: Campus Scene Recognition; Coordinate Attention; CBAM; EfficientNet; DenseNet; Edge AI; Transfer Learning.

1 Introduction

The mental well-being and security of students are paramount concerns for modern university campuses [1]. Automated Scene Recognition systems can serve as non-intrusive tools for monitoring general campus activity, detecting unauthorized access, and identifying potential security threats (e.g., distinguishing between a "Person" and a static "Tree" in low light). However, deploying these systems in real-world campus settings involves a difficult trade-off: models must be computationally lightweight enough to run on surveillance cameras or edge devices [2], yet robust enough to capture subtle environmental cues.

Traditional heavy architectures like DenseNet121 [3] or ResNet18 [4], while accurate, are often unsuitable for edge deployment due to their high parameter count and latency. Conversely, state-of-the-art (SOTA) mobile architectures such as MobileNetV3 [5] rely heavily on Squeeze-and-Excitation (SE) mechanisms [6]. While SE blocks effectively model inter-channel relationships, they typically utilize Global Average Pooling (GAP), which compresses global spatial information into a channel descriptor. In the context of scene recognition, this is a significant limitation because spatial location is critical—the precise arrangement of objects often defines the scene class.

To address this "spatial gap," we introduce **Coordinate Attention (CA)** [7] into two dominant lightweight families: MobileNetV2 [8] and EfficientNet-B0 [9]. Unlike SE or the Convolutional Block Attention Module (CBAM) [10], CA encodes channel relationships along two independent spatial directions (vertical and horizontal). This allows the model to capture long-range dependencies across the image while preserving precise positional information, effectively letting the network answer not just "what" feature is present, but "where" it is located.

2 Related Work

2.1 Deep Learning on the Edge

The demand for mobile-friendly deep learning has led to the development of efficient architectures. MobileNetV1 **howard2017mobilenets** introduced depth-wise separable convolutions, significantly reducing computational cost. MobileNetV2 improved this with inverted residual blocks and linear bottlenecks. EfficientNet utilized compound scaling to balance depth, width, and resolution. While these models excel at general object detection (e.g., ImageNet), their performance often degrades on tasks requiring fine-grained spatial discrimination when resources are constrained.

2.2 Transfer Learning Strategy

Given the limited size of our scene dataset compared to large-scale benchmarks [11], training deep Convolutional Neural Networks (CNNs) from scratch presents significant risks of overfitting [12]. To address this, we employed a **Transfer Learning** strategy utilizing a **Frozen Backbone** approach [13]. This is supported by Yosinski et al. [14], who demonstrated that initial layers learn generic features (edges, textures) transferable across domains, while deeper layers become increasingly specific to the original training classes [15].

2.3 Evolution of Attention Mechanisms

Attention mechanisms have become a cornerstone of modern computer vision [16]. The Squeeze-and-Excitation (SE) block [6] was a pioneer, adaptively recalibrating channel-wise feature responses. However, SE blocks rely on Global Average Pooling, which discards spatial distri-

bution information. To mitigate this, the Convolutional Block Attention Module (CBAM) [10] was introduced, inferring attention maps along both channel and spatial dimensions. However, CBAM's spatial attention is computed via large-kernel convolutions, which can be computationally expensive. Coordinate Attention (CA) [7] addresses these limitations by embedding positional information into channel attention, a technique we validate in this study.

3 Materials and Methods

3.1 Dataset and Pre-processing

The study utilizes a custom dataset collected from a university campus environment, specifically curated to represent the student demographic and infrastructure. The dataset focuses on five key classes: *Car, Building, Lab, Person, and Tree*. These classes were selected based on their relevance to campus security and monitoring. To ensure data quality and model suitability, a rigorous pre-processing pipeline was implemented:

- **Normalization:** All images were resized to 224×224 pixels and pixel values were normalized using ImageNet statistics (Mean: [0.485, 0.456, 0.406], Std: [0.229, 0.224, 0.225]).
- **Data Augmentation:** To improve generalization [17], we applied random horizontal flips, rotations ($\pm 10^{\circ}$), and zooms. This simulates varying CCTV camera angles and distances typical in surveillance footage [18].

3.2 Proposed Architectures

We categorize our experiments into two tracks to address the spectrum of deployment scenarios:

3.2.1 Track A: Heavyweight Architectures (Server-Side)

For maximum accuracy, we employed **DenseNet121** [3]. Its dense connectivity pattern maximizes feature propagation, allowing deep supervision. To refine these features, we integrated the **CBAM** module at the bottleneck.

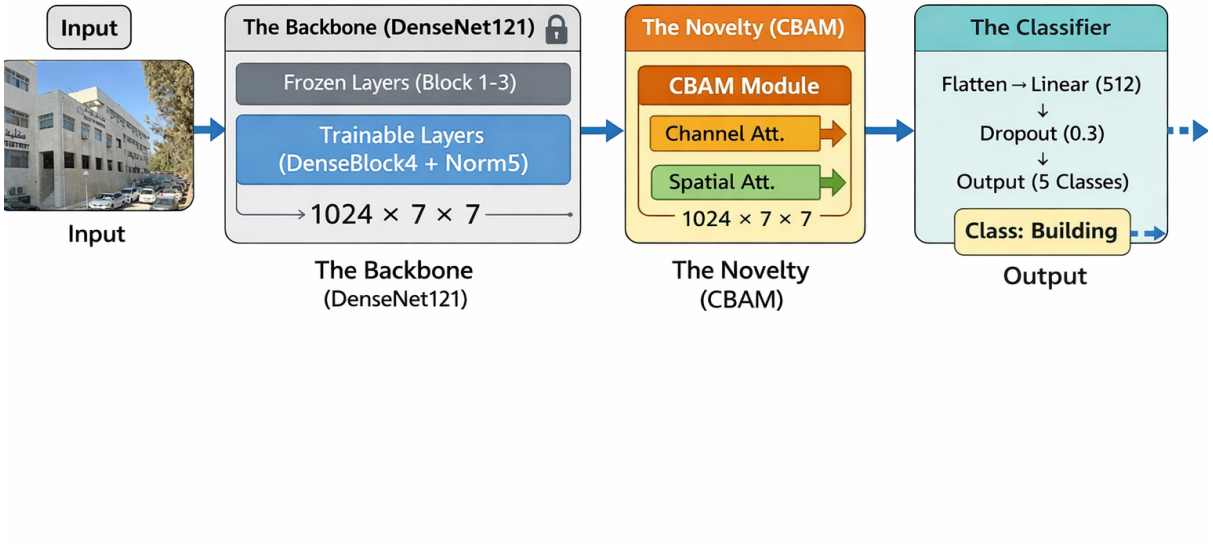

Figure 1: Server-Side Architecture: DenseNet121 integration with CBAM. The backbone (Layers 1-3) remains frozen, while the attention module and custom classifier are trained to refine feature selection.

As illustrated in Figure 1, the DenseNet backbone functions as a static feature extractor, with only the final dense block and attention module being trainable. We also evaluated **ResNet18** [4] with the same integration to serve as a residual baseline.

3.2.2 Track B: Lightweight Architectures (Edge-Side)

For real-time applications, we focused on **EfficientNet-B0** [9] and **MobileNetV2** [8]. Initial experiments with CBAM on these models showed suboptimal results. To address this, we introduced **Coordinate Attention (CA)**.

Unlike SE blocks that crush spatial dimensions via global pooling, CA factorizes channel attention into two 1D feature encoding processes. Specifically, it aggregates features along the vertical (H) and horizontal (W) directions independently, generating direction-aware feature maps. These maps are subsequently concatenated and transformed to generate attention weights that highlight *where* an object is, not just *what* it is. This mechanism provides the spatial awareness of CBAM without the computational overhead of large kernels, making it ideal for mobile networks.

4 Experimental Setup

All models were implemented using PyTorch on a Google Colab environment equipped with an NVIDIA T4 GPU. We utilized the Adam optimizer [19] with an initial learning rate of 0.001. A "ReduceLROnPlateau" scheduler was employed to halve the learning rate if the validation loss did not improve for consecutive epochs. We selected Accuracy and F1-Score as primary evaluation metrics to account for potential class imbalances. Batch Normalization [20] and ReLU activations [21] were utilized in the custom classification heads to prevent overfitting.

5 Results and Analysis

5.1 Track A: Heavyweight Model Performance

Table 1 presents the performance of the server-side models. The integration of CBAM proved highly effective for these high-capacity networks. DenseNet121 + CBAM achieved the highest overall accuracy of **99.15%**. The large capacity of DenseNet allows it to fully utilize the complex spatial maps generated by CBAM without suffering from information bottlenecks [22].

Table 1: Track A: Heavyweight Architectures (Server-Side)

Model Configuration	Accuracy	F1-Score
ResNet18 (Baseline)	93.52%	0.93
DenseNet121 (Baseline)	95.77%	0.95
ResNet18 + CBAM	98.59%	0.98
DenseNet121 + CBAM	**99.15%**	**0.99**

5.2 Track B: Lightweight Model Performance (The Edge Solution)

A critical finding emerged in the lightweight track (Table 2). Applying CBAM to EfficientNet-B0 resulted in a performance drop (-0.28%), likely due to the "feature suppression" phenomenon where aggressive spatial pooling discards critical information from the already-compressed feature maps of lightweight models.

In contrast, **Coordinate Attention (CA)** successfully solved this bottleneck. It boosted EfficientNet-B0 to **98.03%** and MobileNetV2 to **96.90%**. Notably, our MobileNetV2+CA outperformed the architectural reference MobileNetV3 (96.34%) [5]. This proves that intelligently customizing a model for a specific task can outperform general-purpose models designed by automated algorithms.

Table 2: Track B: Lightweight Architectures (Edge-Side)

Model Architecture	Accuracy	F1-Score
MobileNet Family		
MobileNetV2 (Baseline)	93.24%	0.9324
MobileNetV2 + CBAM	94.37%	0.9437
MobileNetV2 + CoordAtt (Ours)	**96.90%**	**0.9690**
SOTA Reference		
MobileNetV3 Large	96.34%	0.9634
EfficientNet Family		
EfficientNet-B0 (Baseline)	96.07%	0.9607
EfficientNet-B0 + CBAM	95.79%	0.9579
EfficientNet-B0 + CoordAtt (Ours)	**98.03%**	**0.9803**

6 Discussion

The comparative analysis reveals a distinct architectural dichotomy. Heavyweight models (DenseNet/ResNet) benefit from the global, aggressive filtering of CBAM, utilizing their vast parameter space to refine features effectively. In contrast, lightweight models (MobileNet/EfficientNet) are sensitive to spatial compression. For these compact networks, Coordinate Attention proves superior because it preserves positional information (vertical/horizontal awareness) without adding the computational burden of large kernels. This finding suggests a design rule for future campus security systems: use DenseNet+CBAM for centralized server analysis and EfficientNet+CA for distributed edge cameras.

7 Conclusion

This study presented a robust scene recognition framework designed for campus environments. By integrating Coordinate Attention into lightweight backbones, we demonstrated that it is possible to achieve high accuracy without sacrificing the efficiency required for edge deployment. Our **EfficientNet-B0 + CA** model achieved 98.03% accuracy, identifying it as the optimal candidate for real-world deployment. Furthermore, we showed that manual optimization with Coordinate Attention allows legacy architectures like MobileNetV2 to outperform state-of-the-art baselines like MobileNetV3.

References

[1] B. C. Ko, "A brief review of facial emotion recognition based on visual information," *Sensors*, 2018.

[2] W. Shi et al., "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, 2016.

[3] G. Huang et al., "Densely connected convolutional networks," *CVPR*, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CVPR*, 2016.

[5] A. Howard et al., "Searching for mobilenetv3," *ICCV*, 2019.

[6] J. Hu et al., "Squeeze-and-excitation networks," *CVPR*, 2018.

[7] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," *CVPR*, 2021.

[8] M. Sandler et al., "Mobilenetv2: Inverted residuals and linear bottlenecks," *CVPR*, 2018.

[9] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ICML*, 2019.

[10] S. Woo et al., "Cbam: Convolutional block attention module," *ECCV*, 2018.

[11] J. Deng et al., "Imagenet: A large-scale hierarchical image database," *CVPR*, 2009.

[12] R. K. Samala et al., "Targeted transfer learning to improve performance in small medical physics datasets," *Scientific Reports*, 2020.

[13] N. Vrielynck et al., "Top-tuning: A study on transfer learning for an efficient alternative to fine tuning," *Computer Vision and Image Understanding*, 2022.

[14] J. Yosinski et al., "How transferable are features in deep neural networks?" *NIPS*, 2014.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE TKDE*, 2010.

[16] M.-H. Guo et al., "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, 2022.

[17] Y. Mao et al., "Fucitnet: Improving the generalization of deep learning networks by the fusion of learned class-inherent transformations," *Neural Networks*, 2021.

[18] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, 2019.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ICML*, 2015.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *ICML*, 2010.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[23] M. Hassan et al., “Image classification using deep and classical machine learning models on small datasets: A complete comparative,” *Journal of Big Data*, 2023.

[24] C. Szegedy et al., “Going deeper with convolutions,” *CVPR*, 2015.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NIPS*, 2012.

[26] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *CVPR*, 2017.

[27] Q. Wang et al., “Eca-net: Efficient channel attention for deep convolutional neural networks,” *CVPR*, 2020.

[28] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, 2015.

[29] N. Srivastava et al., “Dropout: A simple way to prevent neural networks from overfitting,” *JMLR*, 2014.

[30] R. R. Selvaraju et al., “Grad-cam: Visual explanations from deep networks via gradient-based localization,” *ICCV*, 2017.

[31] J. Redmon et al., “You only look once: Unified, real-time object detection,” *CVPR*, 2016.

[32] S. Ren et al., “Faster r-cnn: Towards real-time object detection with region proposal networks,” *NIPS*, 2015.

[33] O. Russakovsky et al., “Imagenet large scale visual recognition challenge,” *IJCV*, 2015.

[34] X. Zhang et al., “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” *CVPR*, 2018.