From Collection to Classification: A Custom Dataset and Deep Learning Framework for Multi-Class Campus Scene Understanding

Owais Abusonaina, Fares Sara, Hala Jeries, Ibrahem Basem, and Huthaifa Alnaimat

***Abstract*—Scene classification is a very important task in computer vision that helps machines understand the environment around them. While there are many big datasets like ImageNet that help train strong models, these models often fail when tested on specific local environments, like university campuses, because of the different architectural styles and backgrounds. This paper presents a new custom dataset collected manually from Jordanian campuses and streets to solve this problem. The dataset contains about 2,400 images divided into five classes: Building, Car, Lab, Person, and Tree. To classify these images, we used the ConvNeXt Tiny model, which is a modern architecture that combines the best features of CNNs and Transformers. We also used specific techniques like data augmentation and WeightedRandomSampler to handle the small size of the dataset and the class imbalance. The results showed that our model achieved a high accuracy of over 99% on both training and validation sets. We also analyzed the loss values and used visualization tools like t-SNE and Grad-CAM to prove that the model is actually learning the correct features and not just memorizing the images.**

***Index Terms*—Scene Classification, Deep Learning, ConvNeXt, Transfer Learning, Custom Dataset, Class Imbalance, Campus Environment.**

I. Introduction

Deep learning techniques have had a profound impact and made significant contributions to the computer vision field, largely triggered by the breakthrough performance of AlexNet [1] on the ImageNet challenge [2]. Since then, a succession of neural network architectures, such as VGG [3], ResNet [4], and EfficientNet [5], have introduced state-of-the-art results on image and video datasets. An example of these applications is **Scene Understanding and Classification**, which remains a crucial component of autonomous navigation, surveillance, and robotics.

According to Zeng et al. [6], **Scene Classification** is defined as the task of assigning a scene image to one of several predefined categories (such as specific types of indoor or outdoor environments) by comprehending the entire image.

While there is actually some already done research and models applied and trained on big datasets such as Places365 and ImageNet, the problem is that they are so generic, meaning that, even though it is so robust and powerful, a model that is trained on European streets would be less accurate when tested on architectural on environmental features of a local campus, for example, in a country in the middle east.

The authors are with the College of IT, University of Jordan, Amman, Jordan.

Manuscript received December 22, 2025.

This discrepancy is formally recognized in computer vision research as the 'domain shift' problem. As discussed by Csurka [7], deep learning models assume that the training and testing data share the same statistical distribution. However, when a model trained on a source domain (e.g., standard Western datasets) is deployed in a different target domain (e.g., a Middle Eastern campus), the variations in architectural styles, vegetation types, and lighting conditions cause a significant degradation in performance. This creates a necessity for domain-specific data collection to bridge the gap between pre-trained generalizations and local reality.

Some institutions, such as universities and schools, might need specific scene detection, which they might use in security or student services. But the problem will be that there is no existing dataset for their environment.

As a solution for this, this paper introduces a custom-collected dataset of a local Jordanian campus and streets, and evaluates the performance of the ConvNeXt Tiny architecture on 5 distinct scene classes.

The main contributions of this paper are summarized as follows:

- The collection and annotation of a novel dataset for campus scene detection.
- The implementation and fine-tuning of the ConvNeXt Tiny model.
- A comprehensive evaluation of the model's performance in terms of accuracy and loss.

II. Problem Definition and Objectives

A. Problem Statement

Despite the remarkable success of deep learning models in scene classification tasks, their performance often degrades significantly when deployed in specific, localized environments. This phenomenon, commonly referred to as **dataset bias** or domain shift, was highlighted by Torralba and Efros [8], who demonstrated that models frequently overfit to the lighting conditions, camera distributions, and semantic contexts of their training datasets, resulting in poor generalization to unseen environments.

Beyond dataset bias, scene classification itself presents intrinsic challenges that distinguish it from object classification. As detailed in the comprehensive survey by Zeng et al. [6], one of the primary difficulties lies in **high intra-class variability** combined with **semantic ambiguity**. For example, a single category such as "Lab" within a campus environment may

encompass visually diverse scenes, ranging from computer laboratories filled with screens to chemistry labs equipped with specialized apparatus, yet all must be mapped to the same semantic label. At the same time, the task often suffers from **low inter-class variability**, where visually similar environments—such as outdoor building facades and indoor corridors—share overlapping structural and textural features, making them difficult to distinguish.

When applied to real-world campus environments, generic scene classification models trained on large-scale datasets (e.g., Places365 or ImageNet) struggle to capture environment-specific visual patterns, including distinctive architectural textures and complex indoor–outdoor spatial layouts. Moreover, existing approaches frequently rely on high-quality web-scraped imagery or controlled surveillance data, which fail to reflect the in-the-wild conditions of mobile photography. Such conditions include variations in image resolution, sensor noise, illumination, and device heterogeneity, all of which are common in practical, student-centric usage scenarios. The absence of localized and representative benchmarks under these realistic conditions creates a significant bottleneck for effective campus scene understanding and monitoring applications.

B. Research Objective

To address the aforementioned challenges, the primary objective of this project is to develop a robust, environment-aware scene classification system tailored to real-world campus environments.

III. Literature Review and Background

Scene classification has evolved significantly over the past two decades, transitioning from hand-crafted feature extraction to end-to-end deep learning.

A. Traditional Approaches

Early research relied on local invariant features to describe scenes. Methods such as **Scale-Invariant Feature Transform (SIFT)** [9] and **Histogram of Oriented Gradients (HOG)** [10] were widely used. These features were typically aggregated using **Bag-of-Visual-Words (BoVW)** [11] models and classified using **Support Vector Machines (SVMs)** [12]. While effective for simple textures, these shallow architectures struggled to capture the high-level semantic complexity of cluttered campus environments.

B. CNN-Based Approaches

The advent of Convolutional Neural Networks (CNNs) revolutionized the field. **Zhou et al.** [13] introduced the Places365 dataset, demonstrating that deep CNNs could learn high-level scene attributes. **He et al.** [4] proposed Residual Networks (ResNet), which utilized skip connections to train deeper networks without vanishing gradients, becoming a standard backbone for scene tasks. Other architectures like **Inception** [14] and **DenseNet** [15] further optimized feature propagation. **King et al.** [16] investigated these architectures specifically for large-scale scene classification, highlighting the efficacy of transfer learning from large datasets.

C. The Shift to Transformers

Recently, the focus has shifted towards Vision Transformers (ViT). **Dosovitskiy et al.** [17] demonstrated that pure transformers could outperform CNNs on massive datasets by leveraging global self-attention mechanisms. However, ViTs often lack the inductive biases of CNNs, making them data-hungry. This limitation led to hybrid architectures like **ConvNeXt**, which we utilize in this work. ConvNeXt modernizes the standard CNN design by incorporating Transformer-like components, such as larger kernel sizes and **Layer Normalization** [18], to achieve superior robustness on mid-sized datasets without the extreme data requirements of pure ViTs.

IV. Dataset and Pre-processing

A. Dataset Collection & Overview

The dataset we worked on was a completely custom, manually-collected dataset. It was collected by taking pictures from a local campus and streets of Jordan, and this was done using the phones of students at the university.

We didn't use any generic dataset for our work (e.g., ImageNet) because such big datasets don't capture the specific architectural style, lighting, and environmental context of specific Jordanian streets and campus. And what we aimed for in the first place was to build a robust model that can recognize exactly this kind of images for our university environment and be more familiar with it.

The total number of images was roughly 2,400, formatted in the standard format of RGB and organized in a directory structure (Root/Class/Image).

B. Class Distribution & Challenges

TABLE I
Dataset Composition: Class Names and Image Counts

Class Name	Number of Images
Tree	580
Car	556
Lab	448
Person	441
Building	380

The dataset has 5 classes, distributed as shown in Table I. As shown, there is a significant class imbalance, with 'Tree' being the majority class and 'Building' being the minority.

As noted by **Johnson and Khoshgoftaar** [19], such imbalance can severely bias deep learning models towards majority classes, necessitating intervention strategies.

In addition to this imbalance problem, most of the images were "noisy" in different ways. There was high variance due to the photos being taken at different times of the day (morning, afternoon, evening). And occlusion, which happened because there was always a chance of objects blocking each other (e.g., a student standing in the front of a building, a car that is parked beside a tree, etc.). And also clutter, which is represented by the background noise (e.g., trash cans) that is not part of the main class.

C. Data Pre-processing Pipeline

In order to turn the raw images into model-ready tensors, all images were resized to the fixed size that ConvNeXt expects, which is 224×224 pixels.

In addition, all pixel values were normalized using ImageNet's mean and standard deviation to make the training stable.

D. Augmentation Strategy

According to Zhang et al. [20], because the dataset is small, deep learning models might memorize the images exactly, which leads to overfitting. The robustness recipe we used here before starting the training phase was to apply an augmentation strategy that would completely solve the problem.

We used the following augmentation techniques for the **Training Set**, and each one of them did something different to enhance the model and make it more robust:

- **RandomResizedCrop:** This was our main spatial augmentation. Instead of just resizing the whole image, this technique crops a random part of the image (between 80% and 100% of the original size) and then resizes it to 224×224. This solves the "squashing" issue and effectively creates "new" variations of the images, helping the model handle different scales.
- **RandomHorizontalFlip:** This is standard for scene detection. Since a tree or a car looks like a tree or a car regardless of which way it faces, flipping the images horizontally doubles the diversity of our orientation data.
- **ColorJitter:** This was crucial given the "low resolution" and "noisy" nature of our dataset. We randomly adjusted brightness, contrast, and saturation by 0.2. This forces the model to ignore specific lighting conditions (like sunny vs. cloudy days) and focus on the actual shapes and features of the objects.
- **RandomErasing (p=0.1):** We implemented Random Erasing [21], which is a more advanced regularization trick. It randomly blacks out small squares in the image. We used this to simulate **occlusion** (e.g., a person is walking in front of a car, a branch is blocking a building, etc.). It teaches the model to recognize the object even if it can't see the whole thing.

On the other hand, for the **Validation Set**, we kept things strict to ensure a fair evaluation. We did not apply any random augmentations there. Instead, we used a standard "Resize and CenterCrop" strategy: we resized images to 256 and then cropped the center 224 pixels. This preserves the original aspect ratio so we aren't testing the model on "squashed" or distorted images.

E. Handling Class Imbalance

One of the tricky challenges we faced with this dataset was the difference in the number of images between classes. While the difference between 580 "Tree" images and 380 "Building" images might not look huge at first glance, in a small dataset like ours, this gap is actually significant. If we trained the model normally, it would see "Trees" much more often than "Buildings." This would make the model lazy, so it would learn to just guess "Tree" most of the time because that's the safe bet to get a lower loss, ignoring the minority classes like Buildings.

To fix this without having to go out and collect hundreds of new photos, we used a specific technical solution called the **WeightedRandomSampler**, which is based on the weighted random sampling algorithms for data streams proposed by Efraimidis [22].

The way it works is actually quite smart but simple. We calculated the count for each class and then assigned an **inverse weight** to them. Basically, the "Building" class (the minority) gets a high weight, and the "Tree" class (the majority) gets a lower weight. Then, during training, we tell the Data Loader to draw samples based on these weights, not just randomly. This tricks the model into thinking the classes are perfectly balanced because, within every batch of data it processes, it sees "Building" images just as often as "Tree" images.

V. Methodology, Implementation, and Results

A. Methodological Framework

We selected the ConvNeXt architecture [23] as the backbone for our campus scene detection model. This choice was motivated by the architecture's ability to bridge the gap between standard Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs).

ConvNeXt modernizes the standard ResNet architecture by integrating design choices typical of Transformers. Specifically, it replaces standard Batch Normalization with Layer Normalization [18] and utilizes GELU activations [24] to stabilize training dynamics. (Figure 1). This results in a network that benefits from the global receptive fields of Transformers without their high data dependency. This characteristic is particularly critical for our study, as pure Transformers typically require massive datasets to avoid overfitting. By utilizing ConvNeXt, we leverage robust feature extraction capabilities suitable for our mid-sized dataset.

Drawing on the findings of Yosinski et al. [25], we leverage the generalizable features learned from ImageNet to accelerate convergence on our smaller target domain.

We utilized a **ConvNeXt-Tiny** model pretrained on ImageNet, removing the original 1,000-class classification head and replacing it with a custom linear layer designed for our 5 specific classes (Building, Tree, Car, Person, Lab).

We also adjusted standard optimization and loss techniques to enhance performance:

- **Optimizer:** We selected **AdamW** [26] due to its superior handling of weight decay in modern architectures compared to standard SGD.
- **Loss Function:** We employed Cross-Entropy Loss with **Label Smoothing**. This technique modifies the hard target labels to prevent the model from becoming overconfident on noisy data, a technique originally proposed to stabilize Inception models [27], thereby improving generalization. [28].

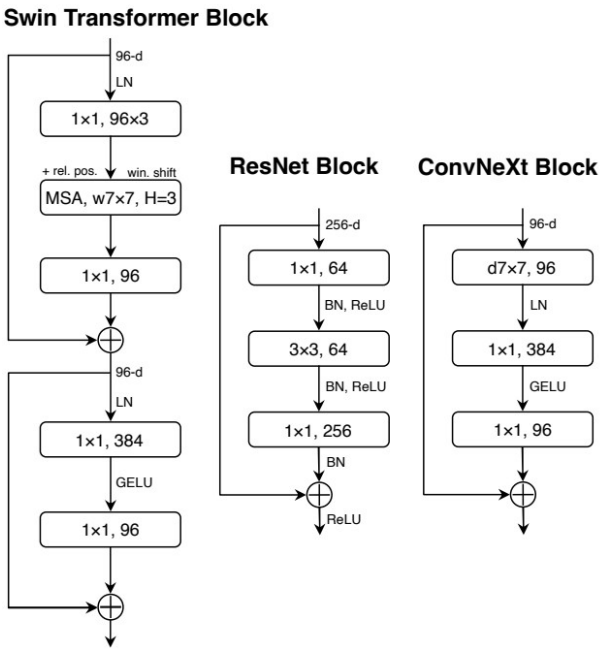

Fig. 1. The ConvNeXt Architecture [23], illustrating the modernized block design.

B. Implementation Details

1) Experimental Setup: The implementation was developed using **PyTorch** [29] with **Scikit-learn** [30] used for metric calculation and **Matplotlib** for visualization. All experiments were conducted on a workstation equipped with an NVIDIA RTX 3060 GPU, which provided sufficient computational power for training and inference.

2) Training Hyperparameters: The specific hyperparameters used during training are detailed in Table II.

TABLE II
TRAINING HYPERPARAMETERS

Parameter	Value
Batch Size	**32**
Epochs	**20** (utilizing Early Stopping [31] to prevent overfitting)
Learning Rate	5×10^{-4}
Weight Decay	1×10^{-4}
Optimizer	**AdamW**
Loss Function	**CrossEntropy** (With Label Smoothing=0.1)

C. Experimental Results

After training for 20 epochs with the aforementioned configuration, the model demonstrated robust performance across all evaluation metrics.

1) Quantitative Analysis: The model achieved a final **Training Accuracy of 99.43%** and a **Validation Accuracy of 99.12%**.

Regarding loss, the final **Training Loss** was **0.4439** and the **Validation Loss** was **0.4583**. While these values may appear numerically higher than standard Cross-Entropy results, they are consistent with the mathematical properties of Label Smoothing. By setting the smoothing factor to 0.1, the target probability distribution changes, introducing a mathematical lower bound known as the **Entropy Floor**.

Derivation of the Entropy Floor: With a label smoothing factor $\alpha = 0.1$ and number of classes $K = 5$, the target probability distribution y is transformed. The probability for the correct class (y_{true}) and incorrect classes (y_{false}) becomes:

$$y_{true} = 1 - \alpha + \frac{\alpha}{K} = 0.92, \quad y_{false} = \frac{\alpha}{K} = 0.02 \tag{1}$$

The theoretical minimum loss (L_{min}) occurs when the predicted distribution p perfectly matches this smoothed target distribution y. Substituting these values into the Cross-Entropy formula:

$$\begin{aligned} L_{min} &= -\sum_{i=1}^{K} y_i \ln(y_i) \\ &= -\left[0.92 \ln(0.92) + 4 \cdot (0.02 \ln(0.02))\right] \\ &\approx -\left[-0.0767 - 0.313\right] \\ &\approx \mathbf{0.39} \end{aligned} \tag{2}$$

Consequently, our final training loss of 0.4439 indicates that the model has converged to within ≈ 0.05 of the absolute mathematical limit, representing a near-perfect fitting of the smoothed targets.

2) Visual Evaluation: To further validate the model's robustness, we analyzed learning curves and confusion matrices.

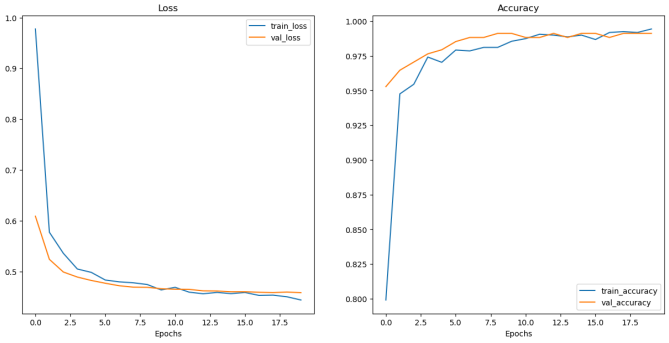

Fig. 2. Training and Validation Learning Curves.

As shown in Figure 2, both training and validation accuracy increased consistently, while loss decreased in tandem. The model exhibited rapid convergence within the first 3 epochs, with no significant divergence between training and validation curves, indicating an absence of overfitting.

The Confusion Matrix (Figure 3) highlights the model's precision. Misclassifications were minimal and restricted to visually ambiguous samples, such as a "Car" being predicted as a "Tree" (likely due to occlusion) or a "Building" as a "Tree" (due to dense foliage in the scene).

3) Explainability and Feature Analysis: We employed Grad-CAM and t-SNE to verify the model's decision-making process.

Grad-CAM Analysis: Gradient-weighted Class Activation Mapping (Grad-CAM) [32] allows for the visualization of

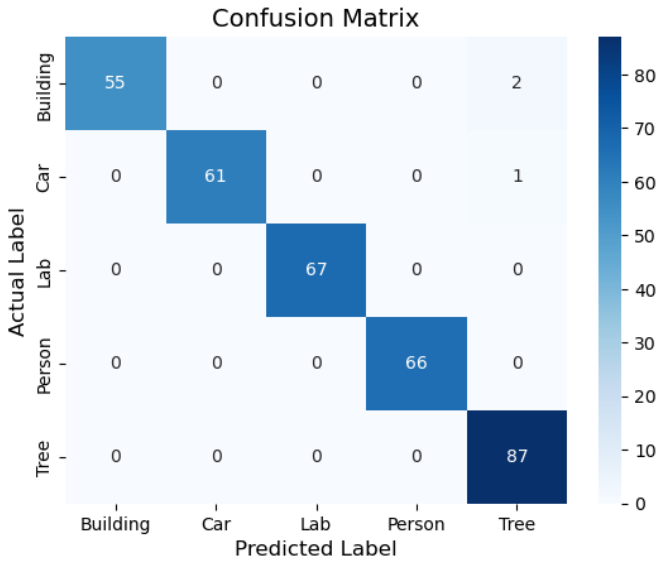

Fig. 3. Confusion Matrix of the test set predictions.

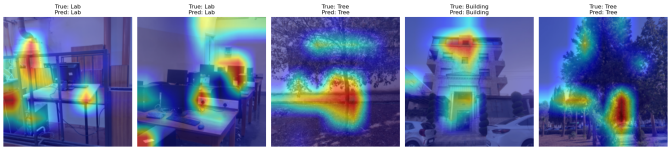

Fig. 4. Grad-CAM heatmaps showing the model's focus on relevant object features.

regions of interest within an image. As seen in Figure 4, the model correctly focused on relevant features, such as tree trunks for the "Tree" class and equipment for the "Lab" class.

t-SNE Visualization: To evaluate feature separability, we utilized t-distributed Stochastic Neighbor Embedding (t-SNE) [33].

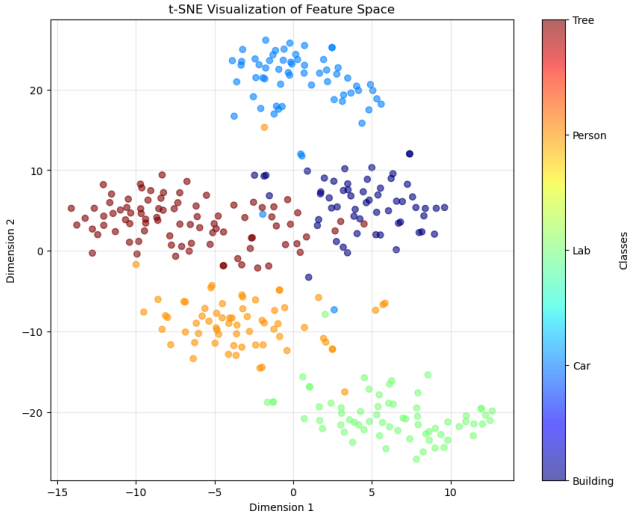

Fig. 5. The t-SNE visualization of the learned feature space reveals well-defined and separated clusters for the five scene categories.

Figure 5 demonstrates that the model successfully learned to map inputs into distinct clusters based on semantic class, confirming the effectiveness of the feature extraction process.

VI. Future Work

Although the outcomes of this study demonstrate strong performance, several key areas require further exploration to enhance both the model's capabilities and its applicability. A critical focus for future development is expanding the dataset. Increasing the diversity and volume of images will not only improve the model's generalization across different environments but also reduce overfitting. As highlighted by Zeng et al. [6], enriching the dataset is essential for bridging domain gaps and ensuring that the model remains robust when encountering unseen data.

Additionally, incorporating more scene classes could significantly improve the model's ability to recognize a broader range of environments. Adding more categories, especially those that capture the complexity and variety of campus and street scenes, would make the model more versatile and reliable. This aligns with the findings of Torralba and Efros [8], who emphasize the importance of expanding the range of categories to enhance performance in domains with unique visual characteristics.

Furthermore, optimizing the model for faster processing is another promising direction. Techniques such as pruning and quantization could help reduce computational overhead, improving inference speed and making the model more suitable for deployment in resource constrained environments. These advancements would be particularly valuable for applications like campus surveillance or autonomous systems, where rapid decision making is critical.

By adopting strategies such as dataset expansion, class diversification, and performance optimization, the model's robustness and adaptability would be significantly enhanced, facilitating its seamless integration into a wide range of dynamic applications.

VII. Conclusion

This study addressed the critical gap in scene classification for localized environments by introducing a dedicated pipeline tailored to Jordanian campus and street environments. While generic models often struggle with domain shifts, our findings confirm that constructing a representative, manually collected dataset—even if moderate in size—is more effective than relying on massive but semantically noisy global datasets.

The implementation of the ConvNeXt Tiny architecture, strengthened by targeted augmentation and weighted sampling, demonstrated that modern lightweight models can achieve strong competitive performance on distinct local classes without demanding excessive computational resources. Furthermore, interpretability analysis using Grad-CAM provided empirical evidence that the model successfully learned to identify meaningful structural and environmental features rather than background noise.

Overall, this work validates the feasibility of deploying efficient, environment-aware deep learning solutions for campus security and monitoring, offering a scalable blueprint for similar applications in other under-represented environments.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[6] D. Zeng, M. Liao, M. Tavakolian, Y. Guo, B. Zhou, D. Hu, M. Pietikäinen, and L. Liu, "Deep learning for scene classification: A survey," *arXiv preprint arXiv:2101.10531*, 2021.

[7] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[8] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[11] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Learning representations by predicting bags of visual words," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6928–6938.

[12] J. M. Moguerza and A. Muñoz, "Support vector machines with applications," 2006.

[13] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[16] J. King, V. Kishore, and F. Ranalli, "Scene classification with convolutional neural networks," *Stanford CS231n Reports*, 2017.

[17] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[18] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[19] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[20] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International conference on learning representations*, 2017.

[21] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[22] P. S. Efraimidis, "Weighted random sampling over data streams," in *Algorithms, probability, networks, and games: Scientific papers and essays dedicated to paul g. spirakis on the occasion of his 60th birthday*. Springer, 2015, pp. 183–195.

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[24] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, vol. 27, 2014.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Kalenichenko, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[28] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[31] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "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.

[33] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.