URBAN AERIAL IMAGE CLASSIFICATION WITH DEEP LEARNING

Matt De Filippo, Jeremy Sugimoto, Gi-E Thang, Taran Bains, Shadman Sajid

University of Calgary, Department of Electrical and Computer Engineering

ABSTRACT

Identifying aerial images represents a practical application of image classification through the utilization of deep learning models. These models can be used for various applications, such as urban planning, environmental monitoring, emergency response management, and aerial surveillance. This study explored the use of deep learning for classifying aerial image data of cities into ten distinct classes. The images were tested on five deep learning classification models, including pre-trained VGG11-BN, ResNet-18, EfficientNet, ShuffleNet, and MobileNetV3 models. Our findings showed consistently high accuracy and robustness across all tested models, with EfficientNet having the highest accuracy, precision, and recall. Through comprehensive model evaluation metrics, this study indicated that deep learning methodologies are applicable for aerial image classification.

Keywords — Machine Learning, Deep Learning, Aerial Image Classification, Urban Planning

1. INTRODUCTION

The accurate categorization of aerial images plays a pivotal role in enabling informed decision-making in key areas such as urban planning, environmental monitoring, emergency response management, and aerial surveillance. This project focused on the classification of aerial images in urban landscapes. These images were classified into ten distinct categories: bridge, commercial, industrial, intersection, landmark, park, parking, playground, residential, and stadium. The *Aerial Images of Cities* dataset hosted on Kaggle were used for our analysis [1].

By leveraging deep learning techniques and transfer learning, we aimed to develop an accurate and efficient classification model capable of analyzing aerial scenes and identifying relevant geographic features. The target metrics used for our analysis were accuracy, precision, recall, and F1-score. A total of five deep learning models were explored and compared for this application: ResNet-18, VGG11-BN, EfficientNet, MobileNetV3, and ShuffleNet.

2. RELATED WORK

Several studies have used deep learning models for similar applications. Pritt and Chen's application of deep learning applications for satellite image classification is an example that stands out [2]. They leveraged convolutional neural networks (CNNs) to classify multi-spectral satellite imagery into 63 distinct classes, with an emphasis on accuracy leveraging satellite metadata alongside image features. Achieving an accuracy of 83% and an F1 score of 0.797, their approach highlights the potential of deep learning in parsing satellite imagery for various applications, including environmental monitoring and law enforcement, making a significant step forward in the field.

Dimitrovski, Kitanovski, Kocev, and Simidjievki introduced *AiTLAS: Benchmark Arena* [3], a comprehensive open-source suite for evaluating deep learning models in Earth Observation (EO) image classification, comparing over 500 models across various architectures on 22 datasets. Emphasizing the benefit of transfer learning, especially with pre-trained models, this study underscored deep learning's adaptability for remote sensing tasks. Ensuring reproducibility, it makes all models, configurations, and dataset details publicly available, promoting further research and application in EO image classification.

The *TernausNET* [4] is an example of a U-net model with a VGG11 encoder. The *TernausNET* was evaluated using the Jaccard index as a performance metric on the Inria Aerial Image Labeling Dataset [5].

3. MATERIALS AND METHODS

3.1 Dataset

The *Aerial Images of Cities* dataset from Kaggle was used for this study [1]. The dataset featured 8000 images with 800 images for each of the following categories: Bridge, Commercial, Industrial, Intersection, Landmark, Park, Parking, Playground, Residential, and Stadium. Each image could be categorized into one of these classes. This therefore represented a multiclass problem. Example images from the dataset are provided in Figure 1 below.

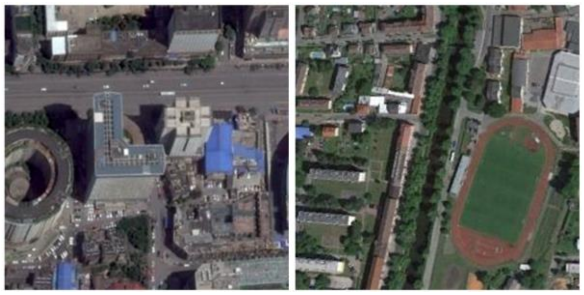
Figure 1. Examples of dataset images for commercial class image (left) and playground class (right).

3.2 Data Pre-Processing

Several data pre-processing steps were performed to prepare the aerial image dataset for training and evaluation using the deep learning models.

The dataset was loaded by extracting image paths and corresponding labels, which were then converted to integer representations. The dataset was then split into development and test sets. The development set was further split into training and validation sets. Stratification was used to maintain the overall class distribution for all sets.

The dataset was further refined using various transformations. The images were resized to 224x224 to make them consistent and compatible with the convolutional neural network (CNN) models. Data augmentation techniques including random horizontal and vertical image flips were used to artificially increase the diversity of the dataset; this supported model generalization and reduced the risk of overfitting by exposing the model to various perspectives of the same aerial image [6]. Images were converted to PyTorch tensors for processing and then normalized. Data normalization was applied using the mean and standard deviation of ImageNet [7].

3.3 Model Architecture and Implementation

Five models were utilized for the classification of the refined image dataset: ResNet-18, VGG11-BN, EfficientNet, MobileNetV3, and ShuffleNet. Details for the implementation of each model are summarized below.

The pre-trained ResNet-18 as described in the paper *Deep Residual Learning for Image Recognition* was the first model assessed [8]. The implemented ResNet-18 model had 11,181642 trainable parameters and 10 classes. No pretrained weights were used.

VGG11-BN is a variant of VGG (Visual Geometry Group) convolutional neural network that includes batch normalization layers. This model had a total number of 11 layers [9]. Batch normalization is a technique used to stabilize and accelerate the training of deep neural networks. The implemented VGG11-BN model had 128,812,810 trainable parameters. No pre-trained weights were used.

EfficientNet introduced a compound scaling method that uniformly scaled all dimensions of depth, width, and resolution using a set of fixed scaling coefficients [10]. The implemented EfficientNet B0 model had a total of 5,301,358 trainable parameters.

MobileNetV3 was a convolution neural network optimized for mobile phone CPUs [11]. The implementation of MobileNetV3 had 1,528,106 trainable parameters.

ShuffleNet was a convolutional neural network architecture that reduced the computational cost of convolutions while maintaining competitive accuracy by exploiting channel shuffle operations and group convolutions [12]. The implementation of ShuffleNet used a total of 352,042 trainable parameters.

3.4 Training Procedure

For each model, the training loop iterated over a set number of epochs. A limit of 20 epochs was used for all models except ShuffleNet, which required 40 epochs to achieve sufficient convergence. It is worth noting that hyperparameters were not optimized due to computational constraints; therefore, the choice of epochs and other hyperparameters may not be optimal for the models under evaluation. Additionally, the training loss was calculated using cross-entropy loss, a common approach to achieve high-accuracy results in multiclass classification problems [13]. It was optimized using stochastic gradient descent (SGD). Learning rate scheduling was applied using ExponentialLR. Validation loss was then computed after each epoch to track the model performance, with the best model based on validation loss being saved. Early stopping was implemented with a patience limit of 5. Therefore, if the validation results did not improve after 5 consecutive epochs, the training stage would be completed.

3.5 Evaluation Metrics

The best model (based on the validation loss) was loaded and evaluated against the test set, to calculate the classification accuracy. The accuracy, precision, recall and F1 scores were computed, and a confusion matrix was generated to assess performance by class.

For the ResNet-18 model, Gradient-weighted Class Activation Mapping (Grad-CAM) analysis was used to produce heatmaps of sample images highlighting important regions that contribute the most to the neural network's prediction [14]. Both the untrained model and the model trained with the aerial image dataset were analyzed to further explore the impact of model training.

4. RESULTS

4.1 ResNet-18

Figure 2 graphs the training and validation loss for the ResNet-18 model against the number of epochs (a max of 20). Figure 3 graphs the training and validation accuracy.

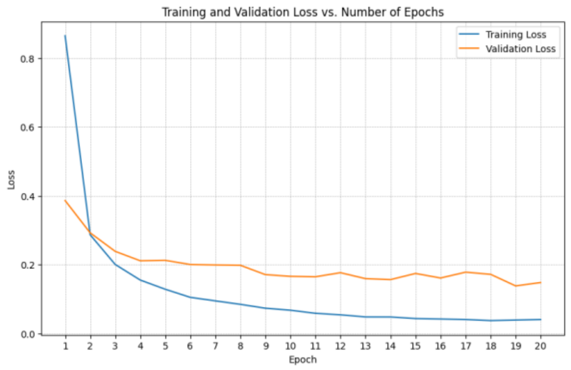

Figure 2. (ResNet-18) Train and val. loss vs epochs.

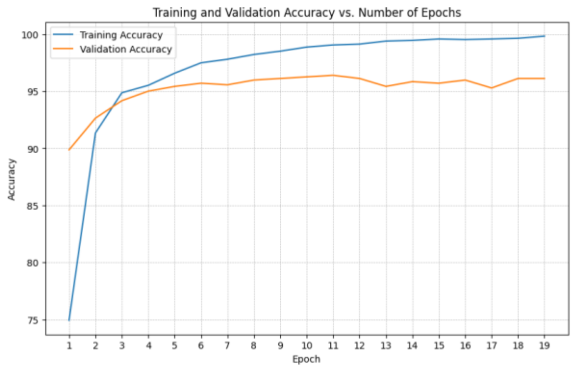

Figure 3. (ResNet-18) Train and val. accuracy vs epochs.

4.2 VGG11-BN

Figure 4 graphs the training and validation loss for the VGG11-BN model against a max of 20 epochs. Figure 5 graphs the training and validation accuracy for 20 epochs.

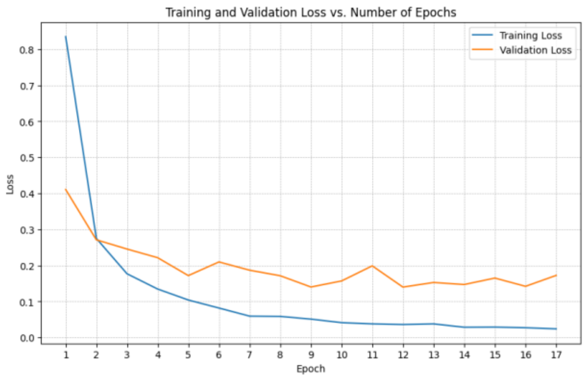

Figure 4. (VGG11-BN) Train and val. loss vs epochs.

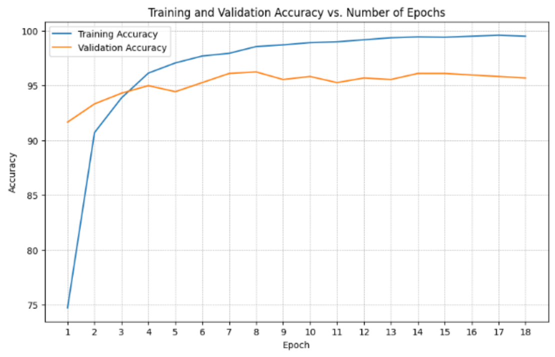

Figure 5. (VGG11-BN) Train and val. accuracy vs epochs.

4.3 EfficientNet

Figures 6 and 7 graph the training and validation loss and accuracy versus 20 epochs for the EfficientNet model.

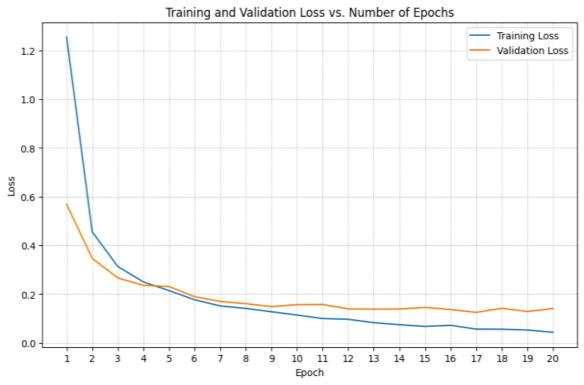

Figure 6. (EfficientNet) Train and val. loss vs epochs.

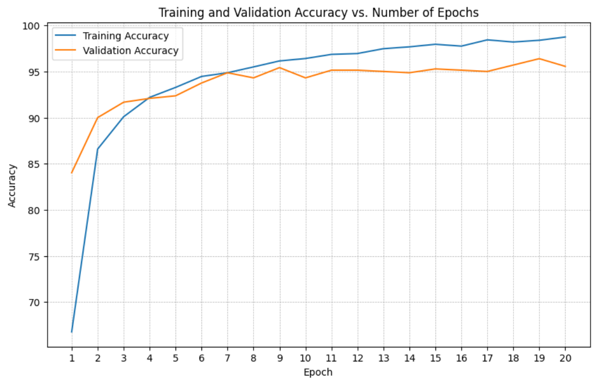

Figure 7. (EfficientNet) Train and val. accuracy vs epochs.

4.4 MobileNetV3

Figures 8 and 9 graph the training and validation loss and accuracy, versus 20 epochs for the MobileNetV3 model respectively.

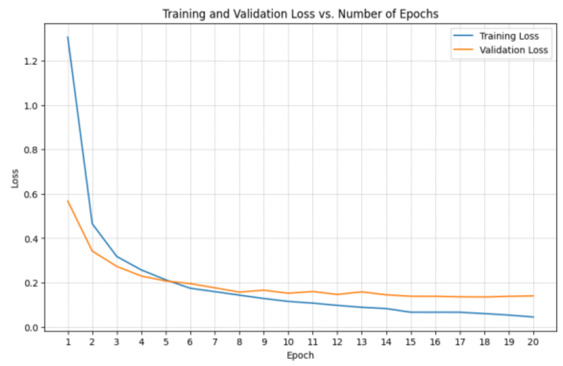

Figure 8. (MobileNetV3) Train and val. loss vs epochs.

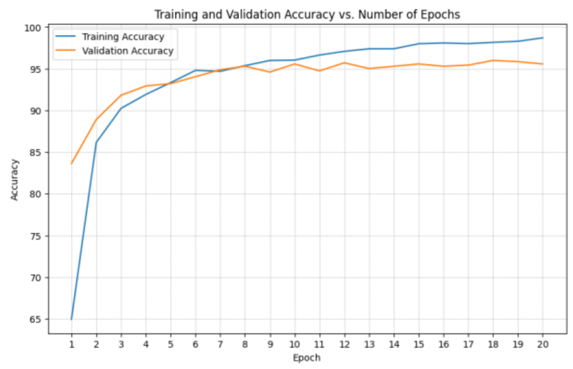

Figure 9. (MobileNetV3) Train and val. accuracy vs epochs.

4.5 ShuffleNet

Figures 10 and 11 show the training and validation loss and accuracy over 40 epochs for the ShuffleNet model.

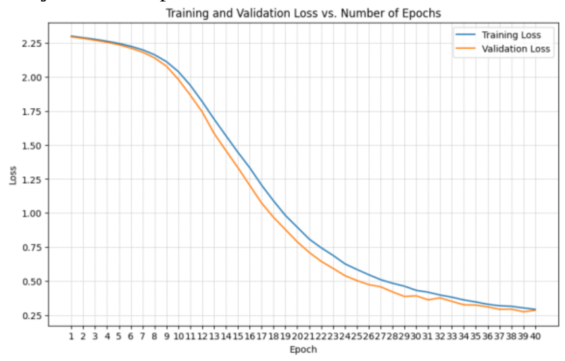

Figure 10. (ShuffleNet) Train and val. loss vs epochs.

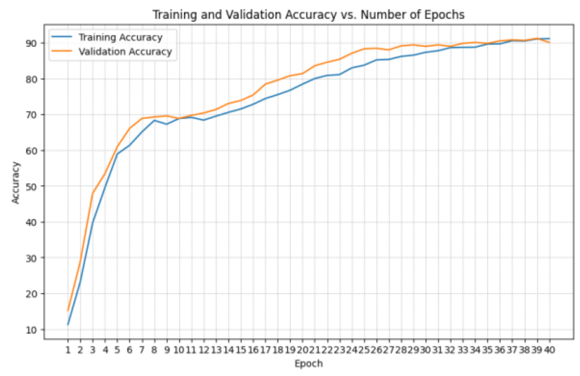

Figure 11. (ShuffleNet) Train and val. accuracy vs epochs.

5. DISCUSSION

When evaluated on the *Aerial Images of Cities Dataset* [1], the five models tested all had strong results. However, the best performing model was EfficientNet, with an accuracy score of 99.63%, a precision score of 99.63%, a recall score of 99.63%, and an F1 score of 99.63%. EfficientNet's slight advantage over the other models could be a result of the model's computational efficiency thanks to its innovative scaling method and AutoML-based optimization.

Table 1. Accuracy, Precision, Recall, & F1 scores

Model	Accuracy	Precision	Recall	F1 Score
ResNet-18	99.50%	99.51%	99.50%	99.50%
VGG11-BN	99.25%	99.26%	99.25%	99.25%
EfficientNet	99.63%	99.63%	99.63%	99.63%
MobileNetV3	99.13%	99.14%	99.13%	99.12%
ShuffleNet	94.50%	94.48%	94.50%	94.46%

Notably the ResNet-18, VGG11-BN and MobileNetV3 models had scores for accuracy, precision, recall, and F1 above 99%.

With EfficientNet, ResNet-18, VGG11-BN and MobileNetV3 demonstrating competitive results, with only slight differences in their performance metrics, it fueled discussion about tradeoffs between each of these models. Considering factors other than performance metrics such as computational efficiency, model size, and training/inference time were essential when choosing which model to use for aerial image classification applications.

Grad-CAM was used to generate heatmaps to highlight sections of images that were most relevant to the model's decision-making process. This technique was used to compare an untrained ResNet-18 model against the ResNet-18 model that was trained using the aerial image data set in the study. Heat map visualizations for untrained and trained ResNet-18 models are provided in Figures 12.

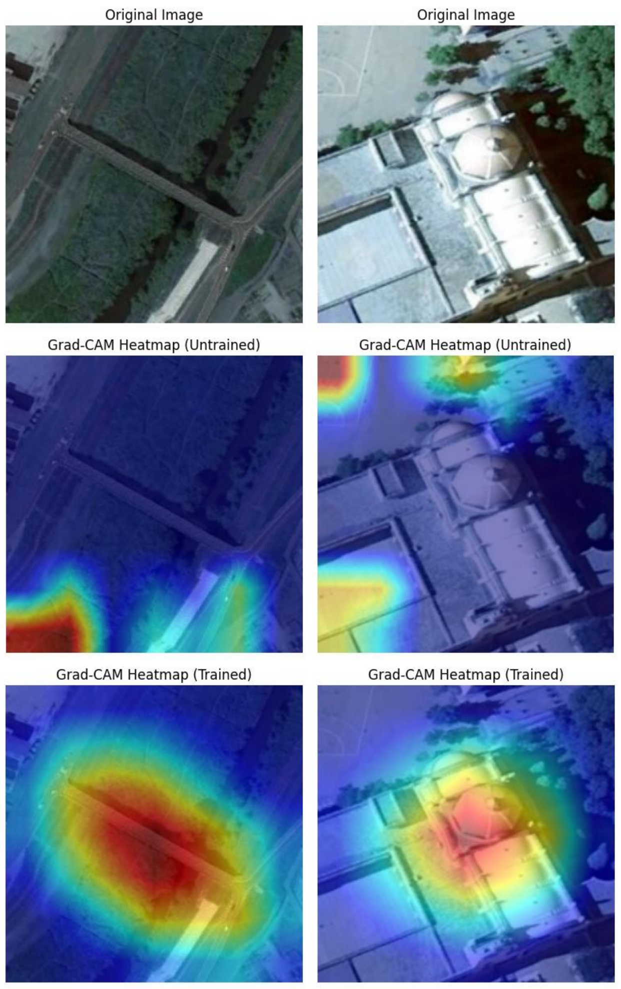
Original Image
Original Image
Grad-CAM Heatmap (Untrained)
Grad-CAM Heatmap (Untrained)
Grad-CAM Heatmap (Trained)
Grad-CAM Heatmap (Trained)

Figure 12. Grad-CAM heatmap visualizations for an image of a bridge (left) and landmark (right) using ResNet-18.

These images demonstrated the impact of the additional training dataset in improving the overall model performance. Before training, the model failed to focus on the aspects of the image that are critical to classification. After training, the model detected the specific features of the bridge and landmark to accurately classify them.

6. CONCLUSIONS

We have presented 5 different deep learning CNN models that classify aerial images into 10 different categories. On the *Aerial Images of Cities Dataset* [1], EfficientNet performed best classifying 7 classes with 100% accuracy and 3 classes with 98.75% accuracy. This resulted in an accuracy score of 99.63%, precision score of 99.63%, recall score of 99.63%, and an F1 score of 99.63%.

EfficientNet stood out as the top performer in our evaluation; ResNet-18, VGG11-BN, and MobileNet have also demonstrated remarkable performance, each achieving scores exceeding 99% across all metrics. This indicated that any of these models could serve as viable options for supporting the deep learning task under consideration. However, the decision between these models ultimately hinged on a thorough examination of the tradeoffs inherent in each model. Factors such as computational efficiency, model size, and training/inference time played pivotal roles in real-world applications. By carefully weighing these considerations, practitioners can make informed choices to optimize their models for specific use cases. As the field of deep learning continues to evolve, understanding and navigating these tradeoffs will remain essential for driving advancements in artificial intelligence research and its practical applications, especially for aerial image classification.

REFERENCES

[1] Yessica Tuteja. *SkyCity: The City Landscape Dataset*. Kaggle. Retrieved March 19, 2024 from: https://www.kaggle.com/datasets/yessicatuteja/skycity-the-city-landscape-dataset.

[2] Mark Pritt & Gary Chen. *Satellite Image Classification with Deep Learning*. Arxiv. Retrieved April 6, 2024 from: https://arxiv.org/ftp/arxiv/papers/2010/2010.06497.pdf

[3] Ivica Dimitrovski, Ivan Kitanovski, Dragi Kocev ,& Nikola Simidjievski. *Current Trends in Deep Learning for Earth Observation: An Open-source Benchmark Arena for Image Classification*. Arxiv. Retrieved April 6, 2024 from: https://arxiv.org/pdf/2207.07189v2.pdf

[4] Vladimir Iglovikov & Alexey Shvets. *TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation*. Arxiv. Retrieved April 6, 2024 from: https://arxiv.org/pdf/1801.05746.pdf

[5] Inria. *Intria Aerial Image Labelling Dataset*. Inria Retrieved April 6, 2024 from: https://project.inria.fr/aerialimagelabeling/

[6] Connor Shorten & Taghi M. Khoshgoftaar. *A survey on Image Data Augmentation for Deep Learning*. SpringerOpen Retrieved April 6, 2024 from: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[7] shahidedu7. *How to normalize images in PyTorch ?*. Geeks for Geeks. Retrieved April 6, 2024 from: https://www.geeksforgeeks.org/how-to-normalize-images-in-pytorch/

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*. Cornell University. Retrieved April 6, 2024 from: https://arxiv.org/abs/1512.03385

[9] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Cornell University. Retrieved April 6, 2024 from: https://arxiv.org/abs/1409.1556

[10] Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Retrieved April 6, 2024 from: https://arxiv.org/abs/1905.11946

[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam. *Searching for MobileNetV3*. Cornell University. Retrieved on April 6, 2024 from: https://arxiv.org/abs/1905.02244

[12] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, Jian Sun. *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*. Cornell University. Retrieved on April 6, 2024 from: https://arxiv.org/abs/1707.01083

[13] Valeria Andreiva, Nadiia Shvai. *Generalization of Cross-Entropy Loss Function for Image Classification*. Research Gate. Retrieved April 6, 2024 from: https://www.researchgate.net/publication/349850933_Generalization_of_Cross-Entropy_Loss_Function_for_Image_Classification

[14] Jacobgil. *Introduction: Advanced Explainable AI for Computer Vision*. Github. Retrieved April 7, 2024 from: https://jacobgil.github.io/pytorch-gradcam-book/introduction.html