International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.103 ISSN: 2581-4621

Autonomous Landing Scene Recognition Based on Transfer Learning for Drones

Quazi Mohammed Haroon[1], Bushra Tahseen[2]
[1]*MCA Student, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, Andhra Pradesh, India*
[2]*Associate Professor, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, Andhra Pradesh, India*

Abstract
In this work, we investigate drone autonomous landing scene detection via knowledge transfer. The challenges associated with aerial remote sensing—namely, the fact that various images have distinct representations at different altitudes or that some pictures are very similar—led us to use a deep convolutional neural network (CNN) that is based on knowledge transfer and fine-tuning to address the issue. Next, the seven classes comprise the LandingScenes-7 dataset is created. Furthermore, we use thresholding in the prediction step to take care of the classifier's ongoing novelty detection issue by excluding additional landing scenes. The adaptive momentum (ADAM) optimization technique is used in conjunction with the ResNeXt-50 backbone to facilitate our transfer learning approach. We also compare momentum stochastic gradient descent (SGD) optimizer with ResNet-50 backbone. ResNeXt50, which uses the ADAM optimization method, performs better, according to the experiment findings. It is possible for drones to autonomously learn landing scenes using this pre-trained model and finetuning, as it achieves 97.8450% top-1 accuracy on the LandingScenes-7 dataset.
Keywords: Drone, Landing, SGD, ADAM

Introduction

Our study delves into the autonomous landing scene identification of drones using knowledge transfer. We used a deep convolutional neural network (CNN) that is based on knowledge transfer and fine-tuning to address the issues related to aerial remote sensing, specifically the fact that different images have distinct representations at different altitudes or that some pictures are very similar. The LandingScenes-7 dataset is then constructed, consisting of seven classes. Moreover, we address the classifier's persistent novelty detection problem by removing extra landing scenes via thresholding in the prediction stage. To support our transfer learning strategy, we combine the ResNeXt-50 backbone with the adaptive momentum (ADAM) optimization technique. Additionally, we contrast the ResNet50 backbone with the momentum stochastic gradient descent (SGD) optimizer. The experiment results show that ResNeXt-50 works better and employs the ADAM optimization approach. Given that our pre-trained model achieves 97.8450% top-1 accuracy on the LandingScenes-7 dataset, drones may be able to independently learn landing scenes with some fine-tuning.

Related Work

Target categorization in remote sensing photos using an efficient distributed convolutional neural network architecture and pre-training It is becoming more difficult to identify objects with similar looks using remote sensing images (RSIs) in an effective and efficient manner. Convolutional neural networks (CNNs) are now the dominant method for classifying targets because of their superior performance and strong feature representation capabilities. However, CNN relies mostly on a single computer for testing and training. Because processing RSIs requires a lot of time and limited hardware resources, a single system naturally has limitations and becomes a bottleneck. Furthermore, because of the imbalance between the model structure and the RSI data, the CNN model faces the problem of overfitting. Overfitting happens and results in poor prediction

DOI:10.46647/ijetms.2025.v09i02.103 ISSN: 2581-4621

performance when a model is complicated or the training data is small. In order to tackle these issues, a distributed CNN architecture is suggested for RSIs target categorization, which significantly boosts the system's scalability and CNN's training performance. It enhances RSIs' processing effectiveness and storage capacity. Additionally, the CNN model is made more flexible and resilient by using the Bayesian regularization strategy to initialize the CNN extractor's weights. It assists in avoiding local optima brought on by inadequate RSI training pictures or an improper CNN structure, as well as overfitting. Furthermore, taking into account the effectiveness of the Naïve Bayes classifier, a distributed Naïve Bayes classifier is engineered to minimize the training expenses. The suggested system and approach work the best and improve recognition accuracy when compared to other algorithms. The results demonstrate that the suggested algorithms and distributed system architecture are appropriate for target categorization tasks in RSIs. Challenges, Approaches, Benchmarks, and Opportunities in Remote Sensing Image Scene Classification Combined with Deep Learning With a wide variety of applications, remote sensing image scene classification seeks to assign a set of semantic categories to remote sensing pictures based on their contents. Deep neural networks' potent feature learning capabilities have propelled the field of remote sensing picture scene categorization, which has garnered notable interest and yielded noteworthy advancements. Nonetheless, as far as we are aware, there hasn't been a thorough examination of current developments in deep learning for remote sensing picture scene categorization. This article offers a comprehensive overview of deep learning techniques for remote sensing picture scene categorization, including over 160 publications, in light of the field's rapid advancement. Specifically, we go over the three main approaches to remote sensing image scene classification and survey challenges: autoencoder-based, convolutional neural network-based, and generative adversarial network-based. Furthermore, we provide an overview of the benchmarks used in remote sensing picture scene categorization and provide a performance summary of over two dozen sample methods on three widely-used benchmark datasets. We conclude by talking about the exciting prospects for further study.

Methodology

a) Proposed Work:

This study proposes an autonomous landing scene recognition system for drones, leveraging knowledge transfer learning with ResNeXt-50[13]. The system will undergo fine-tuning on the LandingScenes-7 dataset, a specialized dataset curated for landing scene classification tasks. By adapting pre-trained weights from ResNeXt-50, the model can efficiently learn discriminative features relevant to landing environments. Incorporating a novelty detection module, the system aims to address challenges related to
unexpected environmental conditions or anomalies during the landing process. Through thresholding techniques applied to model confidence scores, the system will assess the certainty of scene recognition predictions, enhancing reliability in diverse scenarios. To optimize model training, the system integrates the ADAM[18] optimizer, known for its effectiveness in training deep neural networks. By comparing the
performance of ResNeXt-50 with ResNet-50[12], the study will evaluate the efficacy of the proposed architecture in achieving accurate and robust landing scene recognition. By focusing on mitigating issues such as background interference and novelty detection, the proposed system seeks to enhance flight safety and improve recognition accuracy, particularly in emergency landing situations. Through experimental validation and performance analysis, this research aims to contribute to the advancement of autonomous drone technologies for enhanced operational capabilities and safety measures.

b) System Architecture:

The proposed system architecture for landing scene recognition comprises several key components. Initially, landing scene images from the dataset are inputted into the system. These images undergo preprocessing and visualization steps to enhance their quality and extract relevant features. Subsequently, the dataset is split into training and test sets for model development and evaluation. The system employs three different algorithms for classification: ResNet50 with the ADAM[18] optimizer, ResNet50 with ADAM[12] optimizer, and ResNet50 combined with a Random Forest[12] classifier. Each algorithm undergoes training on the training split of the dataset to learn discriminative features for landing scene recognition. After training, the models are evaluated on the test split to assess their performance using metrics such as accuracy, precision, recall, and F1 score. These metrics provide insights into the effectiveness and robustness of each algorithm in accurately classifying landing scenes. Finally, the classification model with the highest performance is selected for landing scene recognition in real-world scenarios. By leveraging state-of-the-art algorithms and comprehensive performance evaluation, the proposed system architecture aims to achieve accurate and reliable landing scene recognition, contributing to enhanced safety and efficiency in drone operations.

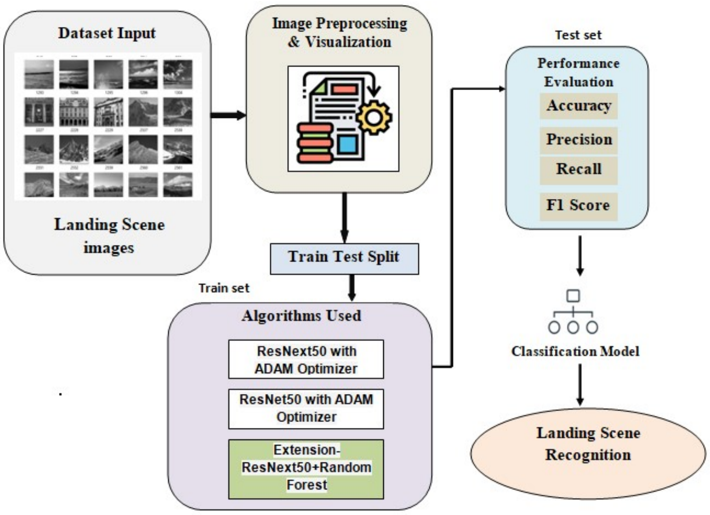


Fig 1 Proposed Architecture

c) Dataset:

The LandingScenes-7 dataset is custom-built for the specific task of emergency landing by drones, encompassing approximately 5,300 images across seven distinct categories. These categories include crowded_place, lawn, road, vehicle_intensive_place, wilderness, wheat_field, and water_area, each representing diverse landing environments. The dataset is further organized into three safety levels: "safe", "general", and "dangerous", based on the landing scene category's inherent risks. Specifically, lawn is classified as "safe", wilderness and wheat_field as "general", while crowded_place, vehicle_intensive_place, road, and water_area are marked as "dangerous". Safety levels are encoded numerically for classification purposes, with "safe" represented by "00", "general" by "01", and "dangerous" by "11". This classification schema allows for effective distinction between landing scenes based on their safety implications, aiding in the development of robust drone landing systems capable of making informed decisions in emergency situations.

d) Data Processing:

In the data processing phase, normalization of image training features is crucial to ensure consistency and improve the convergence of machine learning models during training. Normalization involves scaling the pixel values of images to a standardized range, typically between 0 and 1 or -1 and 1. This process enhances the stability of the training process by minimizing the impact of varying pixel intensity ranges across different images. To normalize image training features, each pixel value in the image is divided by the maximum pixel value (e.g., 255 for 8-bit images) to rescale it to the range [0, 1]. Alternatively, the pixel values can be centered around zero by subtracting the mean pixel value and then dividing by the standard deviation of pixel values across the entire dataset. Normalization helps mitigate issues such as vanishing or exploding

International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.103 ISSN: 2581-4621

gradients, which can hinder the training process and result in poor model performance. By ensuring that all input features are on a similar scale, normalization promotes more stable and efficient optimization, enabling machine learning models to effectively learn from the data and generalize well to unseen samples.

e) Visualization:

In the visualization process using Seaborn and Matplotlib, a bar plot is created to display the distribution of landing scene images in the dataset. The x-axis corresponds to the names of the landing scenes, while the y-axis represents the count of images associated with each scene category. Each scene category is depicted as a separate bar on the plot, with the height of the bar indicating the number of images belonging to that specific scene. By visually inspecting the bar plot, viewers can gain insights into the dataset's composition and the relative abundance of different landing scene categories. This visualization helps researchers and practitioners understand the dataset's diversity and balance, which is crucial for training machine learning models effectively. Furthermore, it facilitates the identification of any class imbalances or biases present in the dataset, informing potential strategies for data augmentation or class weighting during model training. Overall, the Seaborn and Matplotlib visualization provides a clear and intuitive representation of the distribution of landing scene images, aiding in the exploratory analysis and interpretation of the dataset's characteristics.

f) Feature Selection:

Feature selection in landing scene recognition involves identifying the most informative visual attributes essential for accurate classification while reducing computational complexity. This process entails leveraging domain knowledge to pinpoint relevant features like texture, color, and spatial relationships extracted from images using techniques such as Histogram of Oriented Gradients (HOG) or deep learning-based methods like convolutional neural networks (CNNs). Additionally, feature selection methods like Principal Component Analysis (PCA) or Recursive Feature Elimination (RFE) can automatically identify and retain the most discriminative features, enhancing model interpretability and generalization performance while mitigating the curse of dimensionality.

By selecting pertinent features and discarding redundant ones, feature selection optimizes model efficiency and robustness, crucial for the deployment of reliable autonomous drone landing systems. This approach not only streamlines computational resources but also improves the model's ability to accurately classify diverse landing environments, contributing to enhanced safety and efficiency in drone operations.

g) Training & Testing:

In the training phase, the LandingScenes-7 dataset is split into training and testing sets to facilitate model development and evaluation. Two deep learning models, ResNext50 and ResNet50, are trained using the training data. These models leverage convolutional neural networks (CNNs), known for their effectiveness in extracting hierarchical features from images, and the ADAM optimizer, which adaptively adjusts learning rates for efficient parameter updates. During training, the models iteratively process batches of training samples, adjusting their parameters to minimize the prediction error. The training process involves forward propagation to compute predictions, followed by backward propagation to calculate gradients and update model weights accordingly. This iterative optimization process continues until the models converge to a state where further training does not significantly improve performance on the training data.

Following training, the trained models are evaluated using the testing set to assess their performance in accurately recognizing landing scenes. Performance metrics such as accuracy, precision, recall, and F1 score are computed to quantify the models' effectiveness in classifying landing scene images. This evaluation phase helps determine the models' generalization capabilities and their suitability for real-world deployment in autonomous drone landing systems.

International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.103 ISSN: 2581-4621

h) Algorithms:

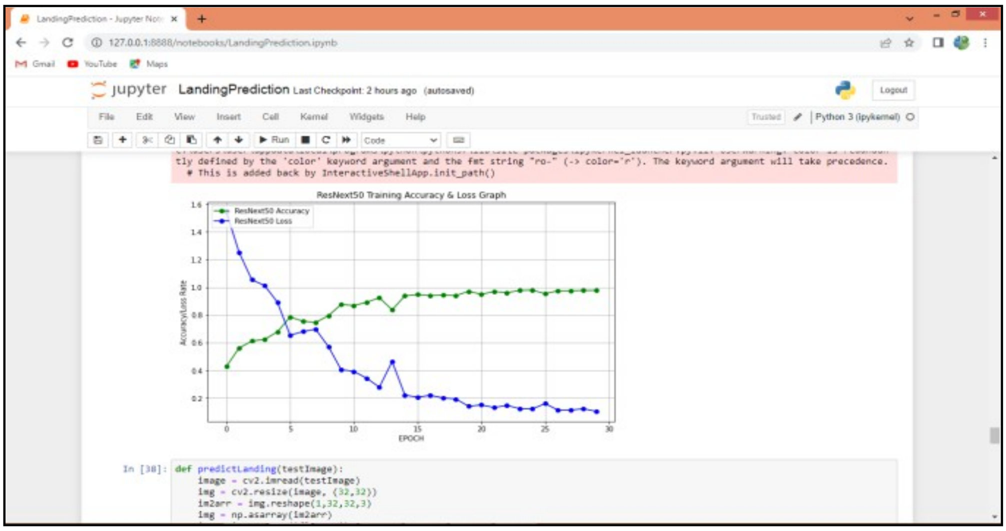
ResNext50 with ADAM: ResNext50 extends the ResNet architecture by introducing the concept of cardinality, enabling the simultaneous learning of diverse features through multiple paths. This deep CNN is adept at capturing complex patterns and variations within images, making it well-suited for landing scene recognition. Paired with the ADAM[18] optimizer, ResNext50 dynamically adjusts its learning rates, facilitating efficient convergence during training. Its ability to grasp intricate details makes ResNext50 with ADAM a compelling choice for achieving high accuracy in classifying diverse landing environments.

ResNet50 with ADAM: ResNet50 leverages residual learning to effectively train very deep neural networks. Residual connections enable the learning of residual functions, addressing issues like the vanishing gradient problem. With its deep architecture, ResNet50 captures hierarchical features essential for image recognition tasks, including landing scene recognition. By utilizing the ADAM optimizer, ResNet50[12] adjusts learning rates based on the optimization landscape, contributing to efficient training. In the context of landing scene recognition, ResNet50 with ADAM emerges as a potent combination for robustly learning and classifying intricate scene patterns.

ResNext50+Random Forest:

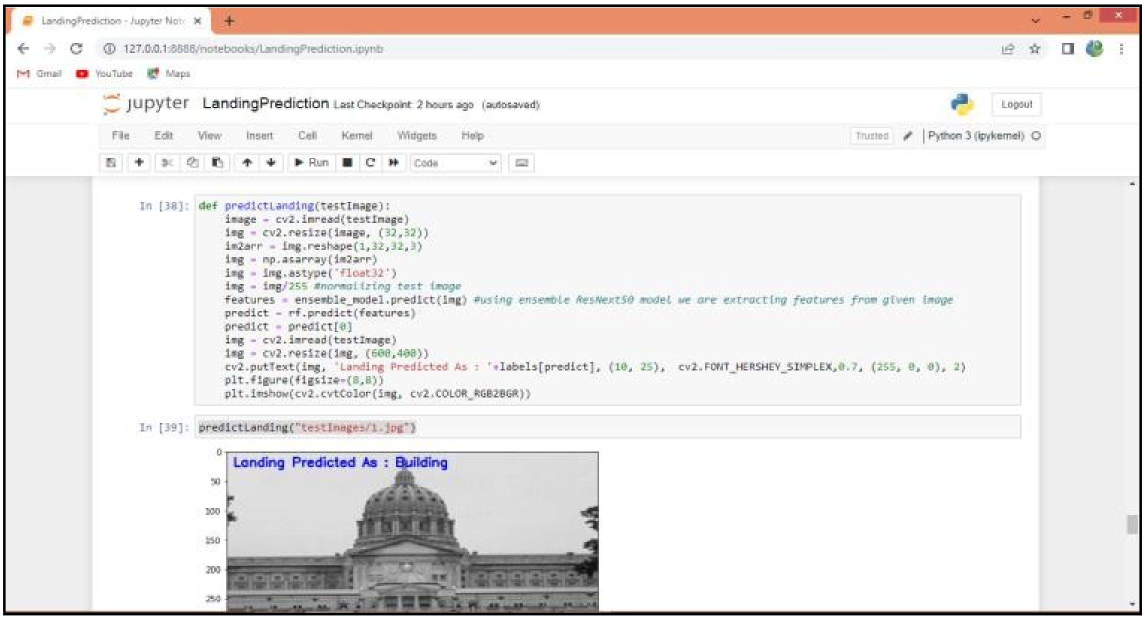
In the ResNext50+Random Forest approach, features extracted by the ResNext50 model are passed to a Random Forest classifier for final decision making. ResNext50[13], with its cardinality concept and deep architecture, extracts rich features from landing scene images. These features are then fed into the Random Forest classifier, which utilizes an ensemble of decision trees to make predictions. This combination harnesses the strengths of both deep learning and traditional machine learning methods, leveraging the powerful feature extraction capabilities of ResNext50 and the robustness of Random Forest for classification tasks. In the domain of landing scene recognition, the ResNext50+Random Forest ensemble offers a versatile and effective solution for accurately classifying diverse landing environments.

Results



The training graph for the ResNext50 is shown above. The x-axis shows the training epoch, and the y-axis shows the accuracy and loss values. The accuracy line is represented by a green line, while the loss line is represented by a blue line. As the epoch progressed, the accuracy increased and approached 1, while the loss decreased.

International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.103 ISSN: 2581-4621



The predict function is defined in the above graph. It takes an input picture path and, using an extended ensemble object, classifies the provided image scenes; in the example scene above, it is classified as a building.

Conclusion

The authors of this research investigate drone knowledge transfer for autonomous landing scene detection. We use a deep convolution neural network (CNN) based on knowledge transfer and fine-tuning to address the issue, taking into account the challenges in aerial remote sensing, particularly the fact that some pictures are very similar or the same scene has distinct representations at various altitudes. Next, a dataset called LandingScenes-7 is created and classified into seven classifications. Additionally, the classifier still struggles with novelty detection, which we solve by eliminating additional landing scenes using the thresholding method in the prediction step. Using the adaptive momentum (ADAM) optimization technique, we use the transfer learning approach based on the ResNeXt-50 backbone. The momentum stochastic gradient descent (SGD) optimizer and the ResNet-50 backbone are also contrasted.

References

1. Li BQ, Hu XH. Effective distributed convolutional neural network architecture for remote sensing images target classification with a pre-training approach. J Syst Eng Electron. 2019;30(2):238-244. doi:10.21629/JSEE.2019.02.12.
2. Xia GS, Bai X, Ding J, *et al*. DOTA: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018:3974-3983. doi:10.1109/CVPR.2018.00417.
3. Cheng G, Xie XX, Han JW, *et al*. Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities. IEEE J Sel Top Appl Earth Obs Remote Sens. 2020;13:3735-3756. doi:10.1109/JSTARS.2020.3002216.
4. Anand AK, Barman S, Prakash NS, *et al*. Vision based automatic landing of unmanned aerial vehicle. In: Proceedings of the International Conference on Information Technology and Applied Mathematics. 2019:102-113. doi:10.1007/978-3-030-48392-2_12.
5. Tian C, Huang CB. An algorithm for unmanned aerial vehicle landing scene recognition based on deep learning and computational verbs. In: Proceedings of the IEEE International Conference on Civil Aviation Safety and Information Technology. 2019:180-184. doi:10.1109/ICCAVSIT.2019.8945648.

6. Lu CW, Tsougenis E, Tang C. Improving object recognition with the l-channel. Pattern Recognit. 2016;49:187-197. doi:10.1016/j.patcog.2015.09.029.
7. Zhou BL, Lapedriza A, Khosla A, *et al*. Places: a 10 million image database for scene recognition. IEEE Trans Pattern Anal Mach Intell. 2018;40(6):1452-1464. doi:10.1109/TPAMI.2017.2723009.
8. Xiao JX, Hays J, Ehinger KA, *et al*. SUN database: large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2010:3485-3492. doi:10.1109/CVPR.2010.5539970.
9. Xiao JX, Ehinger KA, Hays J, *et al*. SUN database: exploring a large collection of scene categories. Int J Comput Vis. 2016;119(1):3-22.
doi:10.1007/s11263-015-0815-8.
10. Patterson G, Xu C, Su H, *et al*. The SUN attribute database: beyond categories for deeper scene understanding. Int J Comput Vis. 2014;108(1-2):59-81. doi:10.1007/s11263-013-0695-z.
11. Yang T, Li PQ, Zhang HM, *et al*. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. Electronics. 2018;7(5):73. doi:10.3390/electronics7050073.
12. He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:770-778. doi:10.1109/CVPR.2016.90.
13. Xie SN, Girshick R, Dollar P, *et al*. Aggregated residual transformations for deep neural networks. In: Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2017:5987- 5995. doi:10.1109/CVPR.2017.634.
14. Xie L, Lee FF, Liu L, *et al*. Scene recognition: a comprehensive survey. Pattern Recognit. 2020;102:107205.
doi:10.1016/j.patcog.2020.107205.