Convolutional Neural Networks for Scene Recognition using ResNet-34

Sumesh Gurudas Chodankar (6853084)

Rajnish Kumar (6846168),

Nuha Alshhamari B (6836507),

David Slaughter (6214785)

Abstract- This Study presents the comparative analysis of Convolutional Neural Network (CNN) models for scene recognition and urban scene classification, specifically focusing on the ResNet-34 architecture. The evaluation is conducted using the PLACES2 dataset, which offers a diverse collection of labeled images across various categories, facilitating a comprehensive assessment of model capabilities.

In addition to model comparison, the study delves into the impact of decaying learning rates on the performance of ResNet-34. Decaying learning rates play a crucial role in training deep learning models by aiding in convergence and mitigating overfitting issues. By incorporating a decaying learning rate strategy during training, the research aims to explore how this approach influences the accuracy, robustness, and generalizability of ResNet-34 specifically in the context of scene recognition and urban scene classification tasks.

The inclusion of decaying learning rates in the study adds a dynamic aspect to the training process, potentially enhancing the model's adaptability and optimization capabilities. This investigation contributes to understanding the optimal training strategies for ResNet-34 and provides insights into improving its performance for complex image classification tasks. The findings from this research can inform future developments and practical implementations of ResNet-34 in the field of computer vision and deep learning, particularly for scene analysis and recognition applications.

I. Introduction

1.1 Background and Motivation

This study is related to Scene Recognition as Scene recognition is fundamental to the operation of autonomous systems, enabling robots and autonomous vehicles to identify and interpret their environments accurately and efficiently. This capability is crucial for tasks such as navigating complex urban environments, performing search and rescue operations, and ensuring safe interactions between autonomous systems and human users. As such, scene recognition has become a focal point in robotics and autonomous vehicle research, driving the need for highly accurate and real-time processing technologies (Krizhevsky, Sutskever, & Hinton, 2012). Convolutional Neural Networks (CNNs) have emerged as the backbone of modern scene recognition systems due to their exceptional ability to process and analyze image data. Inspired by the biological visual systems of animals, CNNs are structured in layers that each detect different features of an input image—from simple edges in the initial layers to complex objects in deeper layers. This hierarchical processing mimics how the brain processes visual information, allowing CNNs to assemble a detailed understanding of visual scenes from raw pixel data (He, Zhang, Ren, & Sun, 2016). A significant breakthrough in training these networks for scene identification tasks is the creation of the Places365 Standard dataset, which provides a comprehensive collection of photos that depict a wide range of real-life locations. This extensive resource aids in refining the accuracy and generalizability of CNNs by exposing them to a wide array of environmental contexts (Zhou, Lapedriza, Khosla, Oliva, & Torralba, 2017)

While conventional machine learning has made significant strides in image classification, it often falls short when dealing with complex image patterns. Deep learning techniques, particularly Convolutional Neural Networks (CNNs), rise to the challenge. However, applying CNNs and transfer learning to scene recognition tasks presents obstacles. These include distinguishing visually similar scenes, handling occlusions, and adapting to varying illumination conditions. Additionally, hyperparameter choices, data preprocessing techniques, and augmentation strategies significantly impact model performance. A thorough investigation of these factors is essential for effective scene recognition

1.2 Problem Statement and Objectives

This study delves into leveraging Convolutional Neural Networks (CNNs), particularly focusing on the ResNet-34 architecture, and transfer learning techniques for scene recognition using the "Places2 simp" dataset, comprising 40,000 images across 40 diverse scene categories. The core objectives of this investigation are as follows:

- Improve model performance by strategically preprocessing data.
- Customize ResNet-34 scene classification models and assess their efficacy.
- Train CNN models with a keen emphasis on optimizing hyperparameters for superior performance.
- Evaluate model performance using classification accuracy metrics and confusion matrices.
- Analyse the impact of hyperparameter choices and visually depict the classification behaviour of each model.

- Analyse the impact of hyperparameter choices and visually depict the classification behaviour of each model.

Through achieving these objectives, this research aims to advance the understanding of CNNs and transfer learning in scene recognition tasks, offering valuable insights into refining hyperparameters, data preprocessing methodologies, and augmentation strategies to enhance the effectiveness of various ResNet architectures.

2.Background Study:

2.1 Related study on other Neural Network architectures

(Sarfaraz Masood, 2020) proposed an automated deep-learning method for classification and scene recognition. The aim of the study was to achieve high performance in scene recognition and compare different CNN-based models. The PLACES2 dataset, which contains 7 million labelled pictures across 15 different categories, was utilized in this study. Data augmentation techniques such as zooming, shearing, horizontal shifting, and vertical shifting were employed to expand the dataset. The study reported that the AlexNet, VGG-16, and Inception v3 models attained accuracies of 79.8%, 87.3%, and 89.3% respectively. Further testing of the Inception-V3 model on the SUN397 dataset showed an accuracy of 94.6%. The conclusion drawn from this research suggests that the Inception-V3 model was highly effective for tasks involving scene recognition and identification. (Davari Majd, 2022) demonstrated that pre-trained Convolutional Neural Networks (CNNs) can be effectively utilized for urban scene recognition. This research aimed to evaluate the performance of pre-trained Convolutional Neural Networks for the recognition of urban scenes, specifically buildings, and to test the models' generalizability across different scenarios. The dataset comprised a wide variety of urban scenes sourced from various platforms such as Vaihingen Aerial Images and Google Earth. Unlike previous studies that randomly divided the dataset into training and testing sets, this study employed a unique approach by dividing the data based on different countries. Data augmentation techniques, including rotations, zoom, weight and height shifts, brightness adjustments, and horizontal shifts, were utilized to enhance the dataset. Among the models tested, the ResNet50 model outperformed others and was subsequently chosen for the second task. The ResNet50 model demonstrated excellent generalizability across four different datasets, achieving an impressive F1 score of 91% on the test datasets.

A method employing AlexNet for the classification of different scenes was proposed by (Romanuke, 2018). The research aimed to evaluate AlexNet's performance and compare it with TLCNN size reduction. The dataset consisted of 15 categories with 4485 grayscale images, split into a training set (4037 images) and a validation set (448 images). Data augmentation techniques, including horizontal and vertical shifts, were applied during training and testing. Stochastic gradient descent with momentum and L2 regularization were used for optimization. AlexNet achieved 91.07% accuracy, while TLCNN achieved 93.3%. However, further study is needed for real-world scenarios with a larger number of categories.

2.2 Learning rate Variations

YANZHAO WU and LING LIU propose the importance of learning rate (LR) policies in deep neural network (DNN) training. LR is a function of the training iteration 't' and plays a crucial role in determining how model parameters are updated during training. LR policies define concrete parameter settings for LR functions and can significantly impact training performance.

The paper categorizes LR functions into three families: fixed LRs, decaying LRs, and cyclic LRs. Fixed LRs maintain a constant LR throughout training while decaying LRs gradually reduce the LR during training iterations. Cyclic LRs cyclically vary the LR within a specified range during training. The paper presents various LR policies within these families and discusses their advantages and limitations. We have used the weight decay here by reference in this paper. Decaying LRs gradually decreases LR values during training, aiming to balance training speed and convergence.

3.Methodology

Our method to solve the scene recognition challenge is highly experimental and mostly uses the ResNet-34 CNN architecture. We have experimented with various hyperparameters, such as different learning rates, Optimizers, Batch-Sizes and in addition we also experimented on similar architectures. We have also briefly experimented with different neural network architectures such as Swin Transformer and ResNet-50.

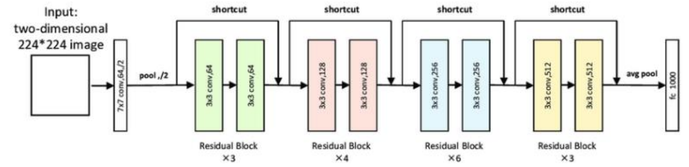

Figure 1:Resnet 34 architecture

3.1 Data Preparation

The dataset “Places2 Simp” was provided by Surrey University, which comprises 40 different categories of scenes (including playground, highway, and museum), each with 1,000 images totaling 40,000 images. Images were resized from 128x128 to 224x224 to match the input size expected by ResNet-34. The folders in which the images are stored serve as their labels. The Places2_simp dataset is a simplified version of the MIT Explore Places dataset and is designed to facilitate the deep learning of scene features for various scene recognition tasks, to achieve good performances on benchmarks.

Data augmentation was to generate an extensive set of images from each image in the original dataset, which helps mitigate overfitting during the training phase and enhance the robustness of the model. Specific transformations such as random shifts along the X and Y axes, rotations varying from -20 to 20 degrees, and random size adjustments were applied. These augmentations enhance the model’s ability to adapt to new, unseen data by offering a wider variety of training instances. They also ensure the model’s stability against changes in the objects’ location, angle, and scale within the images. The dataset was normalized and transformed to tensor format. We split the dataset into training (80%) and validation (20%) sets, which is a standard ratio to use to test the model’s performance on unseen data.

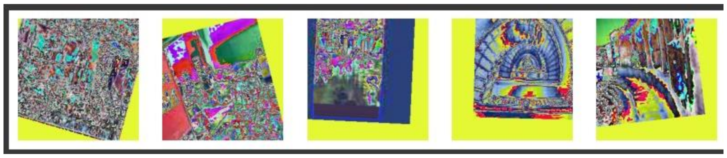

Figure 2 Augmentation applied on dataset.

The model used for most of our experimentation was the pre-trained ResNet-34 model from the torchvision Python module. We changed the final fully connected layer (originally designed for 1,000 classes) to match the 40 classes in our dataset. We used transfer learning, keeping most of the ResNet-34 pre-trained layers.

Most of our experiments used the Adam optimizer, but we also tried SGD. Our loss function used was always cross-entropy loss, which is standard for multi-classification problems.

Hyperparameters such as learning rate, batch size, and number of epochs were experimented with to gain insight into hyperparameter tuning, learn more about the ResNet-34 architecture, and to find optimal settings for better performance (as judged by our evaluation metrics). We also experimented with L2 regularization, which penalizes large weights in the model, to help reduce model overfitting.

We have also experimented on changing the LR rates of each layers. Keeping higher at the starting layers and lower at the inner layer was the approach.

3.2Model Evaluation and Results

The model’s performance was evaluated using top-1 and top-5 classification accuracies. We also used a confusion matrix to visualize the model’s performance across different categories, and easily spot any stand-out high performers (where the model correctly identified most of the scenes).

Randomly selected images from the validation set were used to display the top-5 scores along with the actual and predicted class names, providing insight into the model’s classification decisions.

We also used a custom test set (which contained scene images from Guildford). The model’s accuracy was evaluated on this set, and top-5 scores were displayed for selected images.

Sno.	Resnet 34	Epochs	Batch size	optm	results
	Lr				
1	0.0005	10	32	adam	Top1 acc: 33.8000 Top-5 accuracy: 67.9000
2	0.005	10	32	Adam	Top-1 accuracy: 35.6250 Top-5 accuracy: 69.5875
3	0.00005	10	32	Adam	Top-1 accuracy: 35.6000 Top-5 accuracy: 68.6250
4	0.001	10	32	Adam	Top-1 accuracy: 34.6875 Top-5 accuracy: 67.9375
5	0.005	10	64	Adam	Top-1 accuracy: 33.8500 Top-5 accuracy: 67.0250
6	0.005	50	32	Adam	No improvement in accuracy
7	0.0009	50	32	Adam	Top-1 accuracy: 36.2250 Top-5 accuracy: 68.7125
8	1.00E-06	30	32	Adam	Top-1 accuracy: 39.0625 Top-5 accuracy: 71.8750

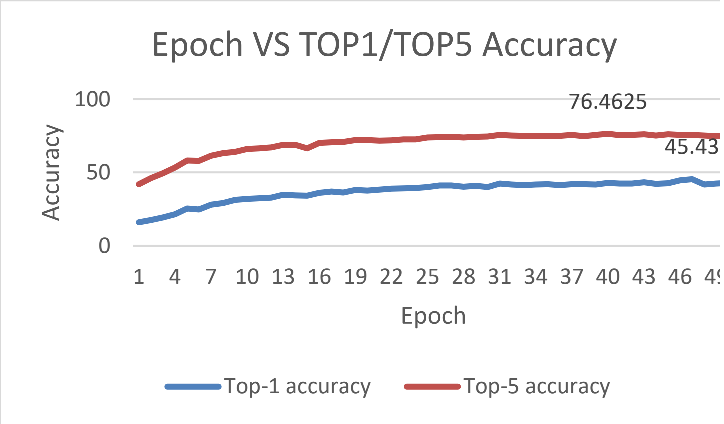

Figure 3Epoch vs Accuracy.

This Graph is for the best hyperparameters optimizer Adam, BatchSize=64, Decaying, and the best Learning rates of high learning rates and small learning rates in the middle layers.

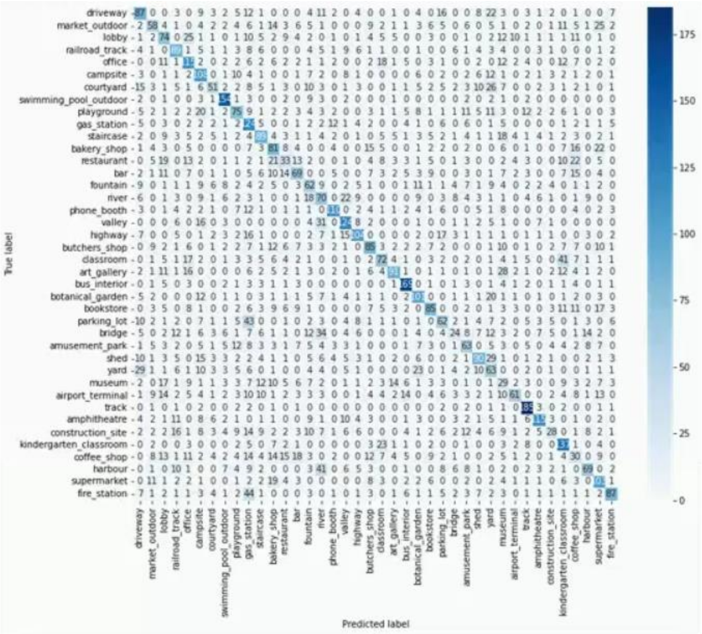

Figure 4:Confusuion Matrix of best Parameters

ResNet32: Two variations of Resnet 32, each using different optimizers, show significant differences in accuracy. Despite training for only half the number of epochs compared to the SGD version, the Adam optimizer version achieves superior Top-1 and Top-5 accuracy. Applying the Adam optimizer to a third ResNet 32 setup, trained with a learning rate of 0.001 for 24 epochs, yields even better results, achieving a Top-1 accuracy of 40.7625% and a Top-5 accuracy of 74.1625%. These findings indicate that Adam may be more effective for this specific job and dataset.

Parameters	Top-1 accuracy	Top-5 accuracy
High Learning rate on layer 1=0.01 and low inside=0.0009	29.4875	63.1875
Very High Learning rate =0.05 and very low learning rate inner layers 0.0005	2.6375	12.8375
High Learning rate on layer 1=0.05 and low inside=0.0005	39.275	73.2125
Weight Decaying	45.43	76.4625

Comparing ResNet 32 and ResNet 50: When using the Adam optimizer, the ResNet 50 model is not more accurate than the ResNet 32 model, even though it is a more complex network and should work better in theory. Both configurations provide the same accuracy, but the performance did not improve by increasing the model complexity from ResNet 32 to ResNet 50, possibly due to the limited number of epochs.

The Swin_v2_b transformer-based model is not as good as CNN models (ResNets) because it is more complicated, requires more data, and takes longer to train. It recorded a Top-1 accuracy of 21.49% and a Top-5 accuracy of 54.09%.

Furthermore, the Adam optimizer on Resnet 32, along with a learning rate of 0.001 and 24 training epochs, makes it the fastest and most accurate model in both the Top-1 and Top-5 measures, and it does all of this with fewer training runs.

Our methodology is structured to provide us insight into how various hyperparameter tuning affects model performance. The detailed steps from data preparation to model evaluation provide a clear pathway for replicating and understanding the project’s workflow. Using ResNet-34 for most of our experiments allowed us to easily compare different hyperparameters and our evaluation metrics made it simple to compare model performance. Our visualization techniques, such as generating confusion matrices, helped to quickly see model performance and which categories were highly correlated.

3.3Additional experimentation:

Model	Epoch	BatchSize	Optimizer	Accuracy
Resnet50	20	32	Adam	38.5750
Ir = 0.001				72.5250
Resnet18	20	32	Adam	33.3750

lr=0.005				67.2500
Swin_v2_b	52	64	Adam	21.49%
lr = 0.003				54.09%

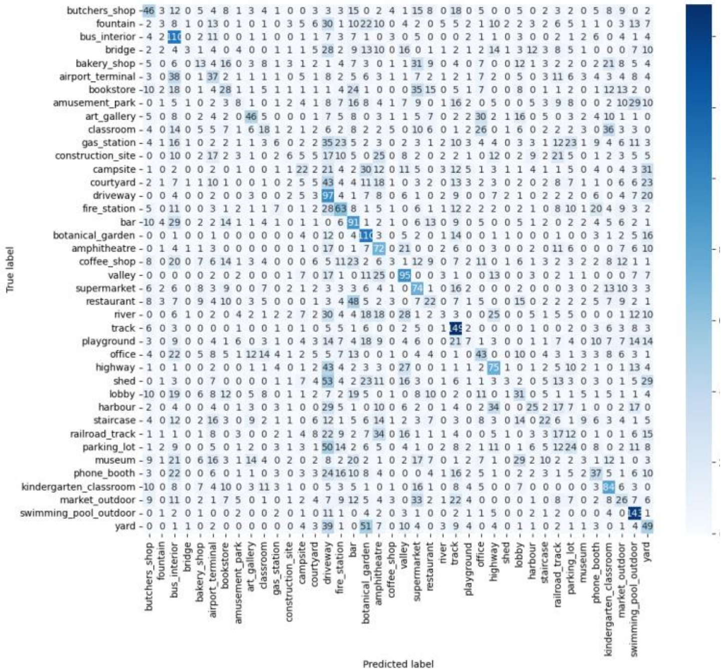

Figure5 swin_v2_b

Conclusion and Future work

This work investigates the application of deep learning techniques, notably ResNet architectures, to problems involving the categorization of scenes. Our findings show that the ResNet 32 model, trained with the Adam optimizer, yielded the best results. It achieves the maximum accuracy in both Top-1 and Top-5 metrics when trained for 24 epochs with a learning rate of 0. 001. Interestingly, this model outperformed the more complex ResNet 50 model under similar conditions. This suggests that increasing model complexity does not necessarily lead to better performance for specific tasks or datasets unless accompanied by appropriate adjustments in training duration and hyperparameters.

We utilized many techniques, including data augmentation and hyperparameter optimization, to enhance the performance of the models. Despite these efforts, we anticipate further work.

Further research will utilize sophisticated methods, such as evolutionary algorithms or reinforcement learning, to methodically explore the hyperparameter space. Using this method might lead to better arrangements than manual or grid search methods, especially for complicated models like Swin_v2_b. Furthermore, to incorporate the complexities of a scene, it is crucial to integrate a variety of techniques. also, train models to identify similarities and adapt to unfamiliar environments. Produce more training instances with advanced technology and techniques. Implement precise techniques to guarantee the model's proper recognition of scenes.

References

1. Hossain, M. A., & Alam Sajib, M. S. (2019, May 18). Classification of Image using Convolutional Neural Network (CNN). Global Journal of Computer Science and Technology, 13–18. https://doi.org/10.34257/gjcstdvol19is2pg13
2. Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015, June 3). Implementation of training convolutional neural networks.
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.
4. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
5. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: An image database for deep scene understanding. Pattern Recognition, 50, 15-25.
6. Davari Majd, R. M. (2022). Generalizability in convolutional neural networks for various types of building scene recognition in High-Resolution imagery. Retrieved from https://doi-org.surrey.idm.oclc.org/
7. Li, Y., Luo, C., Yang, H., & Wu, T. (2018). Convolutional Neural Networks for Scene Image Recognition. In: Sun, X., Pan, Z., Bertino, E. (eds) Artificial Intelligence and Security. ICAIS 2019. Lecture Notes in Computer Science(), vol 11632. Springer. Retrieved from https://doi-org.surrey.idm.oclc.org/10.1007/978-3-030-24274-9_42
8. Rafael Pires de Lima, K. M. (2019). Convolutional Neural Network for Remote-Sensing.
9. Sarfaraz Masood, U. A. (2020). Scene Recognition from Image Using Convolutional Neural Network. Retrieved from https://doi.org/10.1016/j.procs.2020.03.400
10. 10 LIU, Y. W. (n.d.). Selecting and Composing Learning Rate Policies for Deep Neural Networks. Georgia Institute of Technology, USA.
11. Lin Xie, F. L. (n.d.). Scene recognition: A comprehensive survey, Retrieved from https://doi.org/10.1016/j.patcog.2020.107205.