Scene Recognition using Convolution Neural Network

Vishva Gor
Purdue University Northwest
Hammond, USA
vgor@pnw.edu

Shivam Pandya
Purdue University Northwest
Hammond, USA
pandya23@pnw.edu

***Abstract*— This study delves into the use of Convolutional Neural Networks (CNNs) in scene recognition for computer vision applications. Scene recognition is critical for visual understanding, with implications for various fields such as autonomous systems, surveillance, and image indexing. This project aims to create an effective CNN-based model that can efficiently and accurately classify scenes. We present a comprehensive methodology that involves utilizing a diverse dataset, designing a CNN architecture, and conducting an in-depth analysis of the experimental results. Our findings demonstrate a significant improvement in scene recognition accuracy, proving the potential of CNNs in addressing complex visual recognition tasks. This work contributes to the ongoing efforts to enhance scene understanding in real-world scenarios.**

Keywords—Convolution Neural Network, Scene Recognition, Image Indexing, Classification Accuracy, Multiclass Classification, Deep learning

I. INTRODUCTION

Scene recognition, the process of automatically categorizing the content and context of images based on their visual features, is a fundamental task in computer vision. This capability holds significant importance in various domains, ranging from autonomous vehicles that navigate dynamic environments to surveillance systems [2] that monitor diverse scenes in real-time. The motivation behind this study arises from the need to develop robust and efficient methods for scene recognition, capable of handling the inherent complexities associated with diverse scenes.

Despite advancements in computer vision, challenges persist in accurately classifying scenes characterized by variations in lighting, viewpoint, and object arrangement. Convolutional Neural Networks (CNNs) have emerged as powerful tools for image analysis and pattern recognition, demonstrating notable success in tasks such as image classification and object detection. This study focuses on harnessing the capabilities of CNNs to enhance the accuracy and generalization of scene recognition models.

In this introduction, we provide a brief overview of existing literature on scene recognition, highlighting the evolving landscape and underscoring the need for innovative approaches. We articulate the specific problem statement and objectives of our study, emphasizing the aim to design a CNN architecture tailored for scene recognition. The subsequent sections will detail our methodology, experimental results, and discussions, ultimately contributing to the ongoing research in advancing scene understanding through the application of CNNs. [1]

II. DATA COLLECTION

In the pursuit of data enhancement tailored to image data for our scene recognition project, various online options for scene datasets were explored. Eventually, the Places365 [4] dataset was chosen due to its comprehensive nature, despite its substantial size of 25 gigabytes. Given our limited computation power, we opted to commence the scene recognition efforts with a smaller dataset, specifically a subset of Places365 [4]. For this initial phase, we selected two categories and gathered 500 images for each, aiming to construct a foundational model.

III. DATA PREPROCESSING

A. Image Transformation

Upon delving into the dataset, an essential pre-processing step involved assessing image dimensions. Utilizing the torch-vision library's built-in functionality for converting images into tensors, it was revealed that the original dimensions were 3x256x256. Recognizing the computational challenges posed by these substantial dimensions, a strategic decision was made to resize the images to 3x48x48. [3] This resizing not only alleviated computational burdens but also facilitated expedited image processing, contributing to the efficiency of our scene recognition model. This adjustment was pivotal in adapting the dataset to the constraints of our computational resources while optimizing the overall performance of the image processing pipeline.

B. Batch data processing

Batch processing is vital in data processing, particularly with large datasets. It enhances computational efficiency by dividing data into manageable subsets and optimizing memory usage. This approach facilitates parallelization, speeding up machine learning model training. Additionally, batch processing enables iterative learning, contributing to smoother convergence during training.

For efficient batch data processing in our scene recognition project, where the dataset's size posed a challenge for seamless computation, we implemented the use of the Data Loader module from torch.utils.data. Given the substantial volume of data in the dataset, adopting a batch-processing strategy became

imperative. Specifically, we configured the batch size to 32 within the Data Loader. This deliberate choice ensured that during each iteration within an epoch, the model would process a manageable subset of 32 images, as opposed to attempting to handle the entire dataset at once. The adoption of batch processing, facilitated by the Data Loader module, not only optimized computational efficiency but also enhanced the overall training process by enabling the model to iteratively learn from smaller, more manageable subsets of the dataset.

IV. Model Architecture

Convolutional Neural Networks (CNNs) offer significant advantages in scene recognition, automatically learning intricate features from images. [1] Their adaptability to diverse datasets, incorporation of non-linear activation functions like ReLU, and effective capture of both local and global contextual information make CNNs powerful tools for precise and efficient scene recognition tasks.

A. Convolution Neural Network

In our CNN classifier, comprising three convolutional layers alternated with three max-pooling layers, the inclusion of Rectified Linear Unit (ReLU) activation functions enhances feature extraction by introducing non-linearity. The subsequent use of a single fully connected neural network layer, also employing ReLU activation, refines learned features for precise classification. The final output layer is equipped with a SoftMax activation, providing probability scores across multiple categories. Initially designed for two categories—amusement park and greenhouse—the classifier's scope expanded to encompass four categories: amusement park, mountains, bridge, and greenhouse. Implemented in an object-oriented paradigm, the classifier's modular design, coupled with ReLU and SoftMax activations, enhances code clarity, promoting an intuitive understanding and adaptability for future modifications.

B. Training Process

In our training process, we conducted thorough experiments with diverse hyperparameter configurations to enhance the performance of our scene recognition classifier. We explored epochs ranging from 50 to 100, strategically seeking an optimal balance between model convergence and computational efficiency. The stochastic gradient descent (SGD) optimization algorithm, complemented by a learning rate of 0.1, facilitated effective weight manipulation, ensuring the iterative refinement of model parameters. Our criterion for model optimization was the cross-entropy loss function, chosen for its ability to measure dissimilarity between predicted and actual scene categories accurately. Additionally, we adopted an 80-20 split, allocating 80% of the total dataset for training and reserving 20% for testing. This partitioning strategy adheres to best practices, enabling robust training and comprehensive evaluation of our Convolutional Neural Network (CNN) for scene recognition.

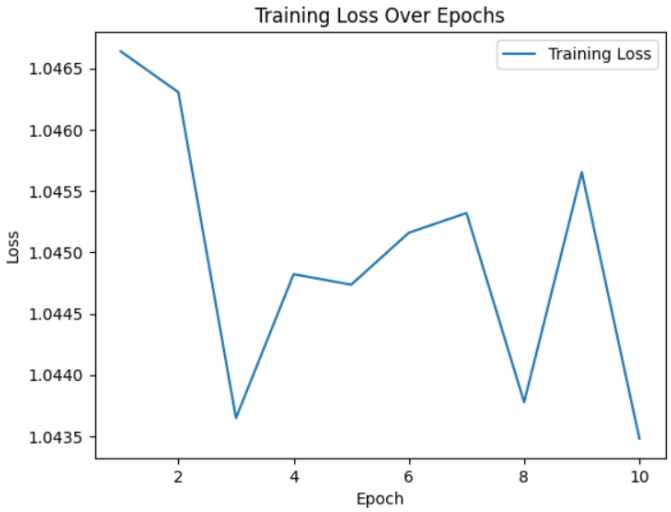

Fig. 1. Loss (Avg. loss of 51 iterations) vs Epoch Graph for 10 Epochs

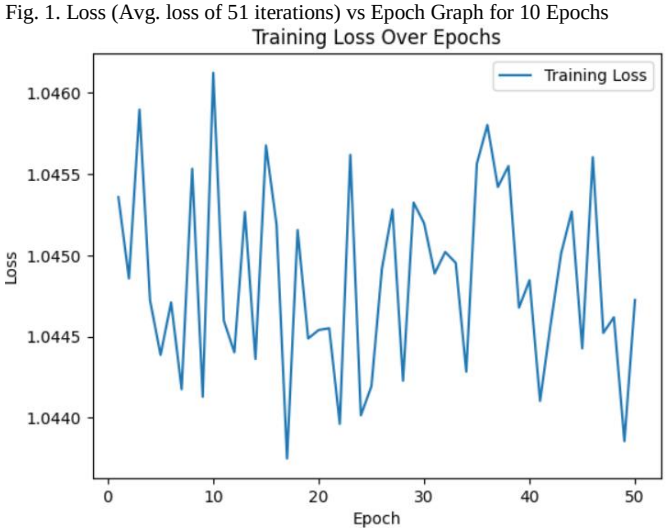

Fig. 2. Loss (Avg. loss of 51 iterations) vs Epoch Graph for 50 Epochs

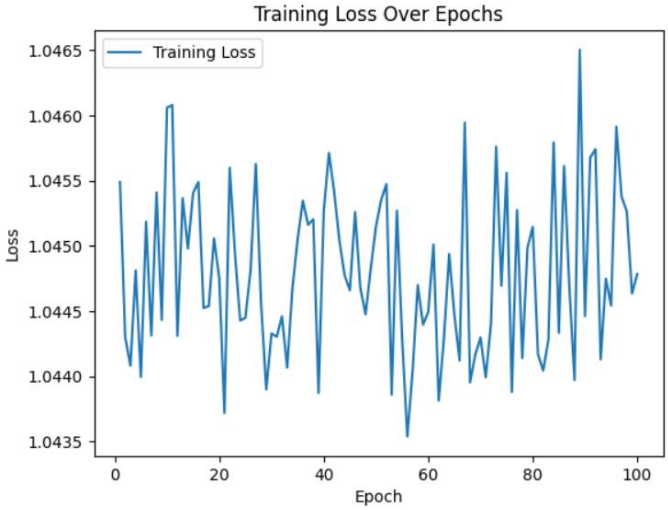

Fig. 3. Loss (Avg. loss of 51 iterations) vs Epoch Graph for 100 Epochs

C. Evaluation Metrics

For the assessment of our scene recognition Convolutional Neural Network (CNN) model, a comprehensive evaluation matrix was employed. The model was set to evaluation mode,

and its performance was rigorously assessed using various metrics. The confusion matrix provided a detailed breakdown of the model's predictions, revealing the distribution of true positive, true negative, false positive, and false negative classifications. The overall accuracy of the model was quantified using the accuracy score, demonstrating the proportion of correctly classified instances. Precision, recall, and F1-score, computed through the precision score, recall score, and f1_score functions respectively, further gauged the model's ability to correctly classify instances across different categories. These metrics were essential in providing a nuanced understanding of the model's performance, shedding light on its strengths and areas for potential improvement.

```
evaluate_model(cnn_classifier, train_loader, device)

Confusion Matrix:
[[339   0   2  58]
 [321   1   3  79]
 [229   4  30 142]
 [229   4   1 178]]
Accuracy: 0.34
Precision: 0.41
Recall: 0.34
F1-score: 0.25

evaluate_model(cnn_classifier, test_loader, device)

Confusion Matrix:
[[85  0  0 22]
 [80  1  1 21]
 [72  1  1 27]
 [47  0  2 45]]
Accuracy: 0.33
Precision: 0.36
Recall: 0.33
F1-score: 0.22
```

Fig. 4. Accuracy and Confusion matrix for Training and Testing with 10 Epochs

```
evaluate_model(cnn_classifier, train_loader, device)

Confusion Matrix:
[[265  70  67   4]
 [ 38 295  47  26]
 [ 11  23 378   4]
 [ 35 123  45 189]]
Accuracy: 0.70
Precision: 0.72
Recall: 0.70
F1-score: 0.69

evaluate_model(cnn_classifier, test_loader, device)

Confusion Matrix:
[[49 33 18  0]
 [22 56 16  7]
 [ 6  5 78  1]
 [11 46 15 42]]
Accuracy: 0.56
Precision: 0.61
Recall: 0.56
F1-score: 0.55
```

Fig. 5. Accuracy and Confusion matrix for Training and Testing with 50 Epochs

```
evaluate_model(cnn_classifier, train_loader, device)

Confusion Matrix:
[[368  10   7  14]
 [ 13 366   8  24]
 [  3   5 399   2]
 [ 11  20   6 364]]
Accuracy: 0.92
Precision: 0.92
Recall: 0.92
F1-score: 0.92

evaluate_model(cnn_classifier, test_loader, device)

Confusion Matrix:
[[55 29 12 11]
 [17 46  7 26]
 [10  8 74  5]
 [13 20  4 68]]
Accuracy: 0.60
Precision: 0.60
Recall: 0.60
F1-score: 0.60
```

Fig. 6. Accuracy and Confusion matrix for Training and Testing with 100 Epochs

D. Results and Discussion

We have selected a set of random images from the test dataset to showcase the model's ability to predict the corresponding scene categories. The following images exemplify the effectiveness of the model in accurately determining the nature of each scene.

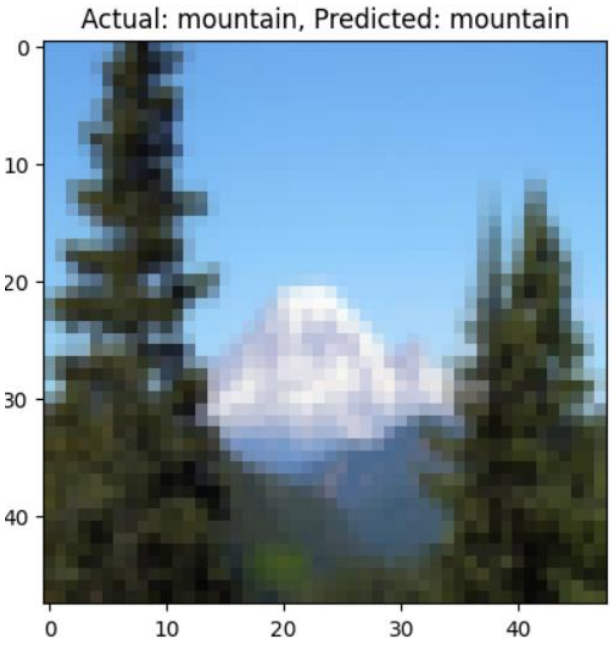

Fig. 7. Output Result 1.

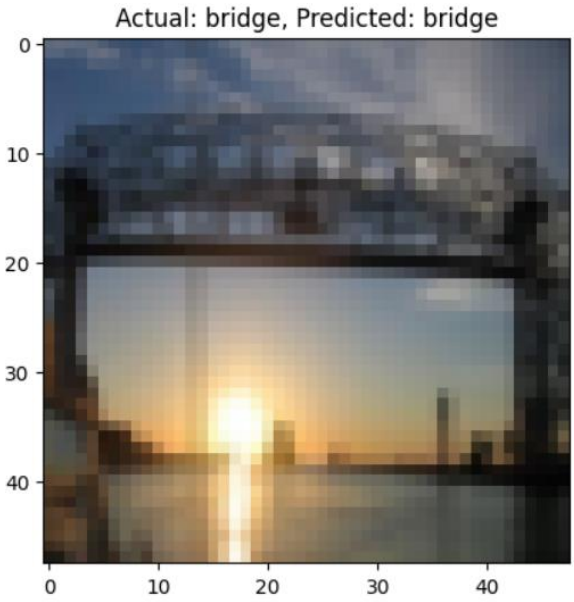

Fig. 8. Output Result 2.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Professor Prof. Rehan Ahmed for his invaluable contribution to this scene recognition project. His guidance and expertise were instrumental in the selection of this project, and his support throughout the research process has been truly invaluable. We are particularly grateful for his assistance with the dataset, providing essential resources and insights that significantly enriched the project. Prof. Ahmed's encouragement and thoughtful guidance during the data analysis phase were pivotal, enabling me to explore various parameters and obtain meaningful results. His mentorship has been a source of inspiration, and I am sincerely thankful for his unwavering support and dedication to the success of this project.

REFERENCES

[1] R. L. Galvez, A. A. Bandala, and E. P. Dadios, “Object Detection Using Convolution Neural Network,” TENCON 2018 - 2018 IEEE Region 10 Conference, October 2018.

[2] V. Gajjar, A. Gurnani and Y. Khandhediya, "Human Detection and Tracking for Video Surveillance: A Cognitive Science Approach," in 2017 IEEE International Conference on Computer Vision Workshops, 2017.

[3] A. Karpathy, October 2017. [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[4] ‘https://www.kaggle.com/datasets/benjaminkz/places365/data’.

[5] "Deep learning to frame objects for visual target tracking," Engineering Applications of Artificial Intelligence, vol. 65, pp. 406- 420, October 2017

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in neural information processing systems, 2012.

[7] R. L. Galvez, N. N. F. Giron, M. K. Cabatuan and E. P. Dadios, "Vehicle Classification Using Transfer Learning in Convolutional Neural Networks," in 2017 2nd Advanced Research in Electrical and Electronic Engineering Technology (ARIEET), Jakarta, Indonesia, 2017.