2025

Deep Learning Project Report

UNIVERSITY SCENE CLASSIFICATION

SAUS Dataset

Presented BY:

KHADIM ALI & HAMAD AHMED

23-BS(CS)-30 23-BS(CS)-10

Submitted To:

DR. ATTAULLAH SAHITO

Introduction

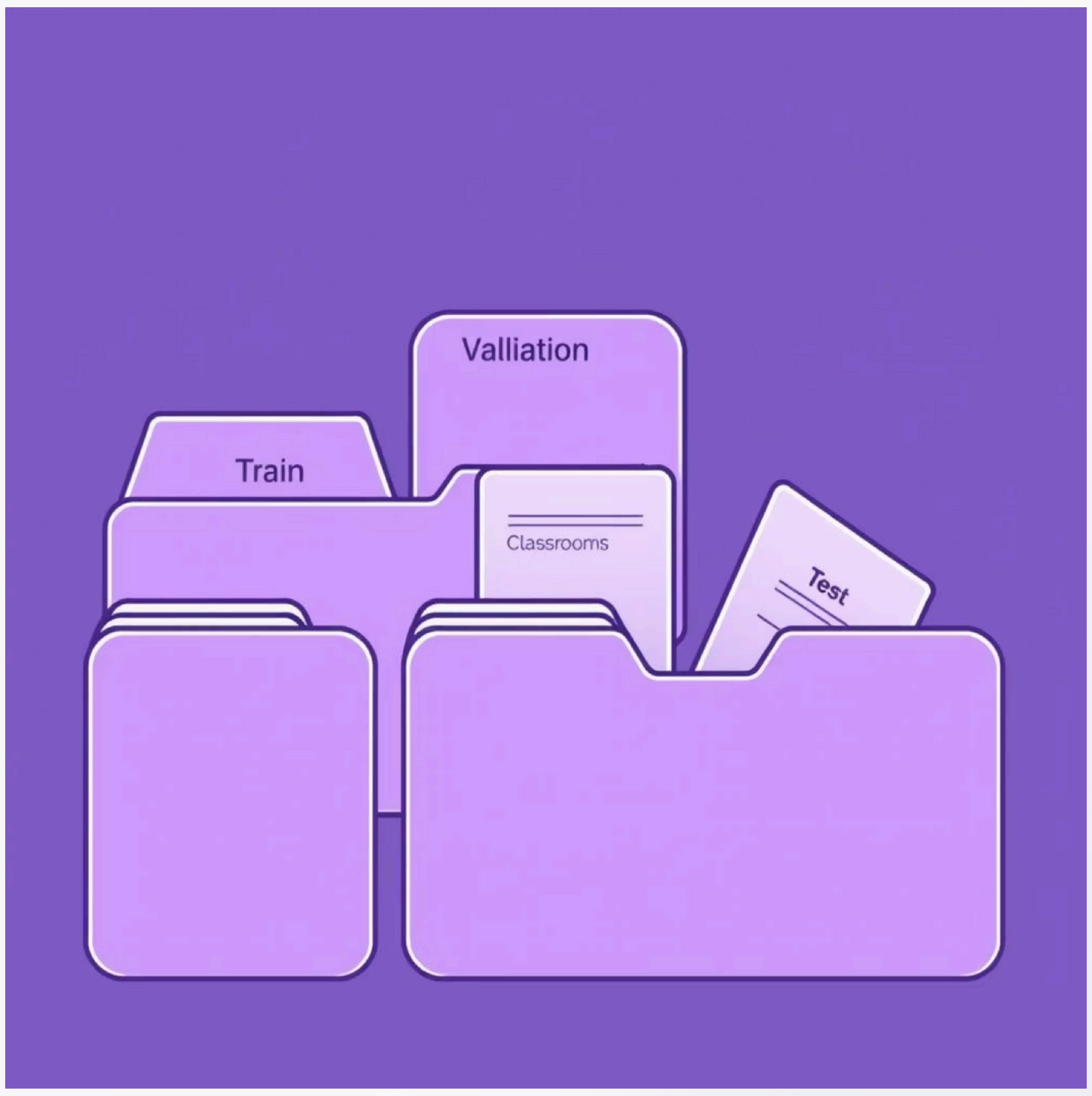
This project focuses on scene classification using a Deep Learning model. We used a custom dataset SAUS, enhanced for better quality and variety. The goal is to accurately classify images using a CNN-based Transfer Learning approach, showing how Deep Learning can be applied to real-world scene recognition.

Dataset Info

Dataset Name: SAUS

Folder Structure:

- Train
- Validation (Val)
- Test



Classes Used (From Code):

- Auditorium
- Classrooms
- Ground
- Indoor_Places
- Labs
- Office
- Outdoor_Places

This includes all classes found within the SAUS folders.

1

Total Images

The notebook automatically loads images from structured directories. The dataset initially had 400 images, which we enhanced by adding 465 more images, making it larger and more diverse, with a total of 865 images.

2

Image Size Used

128 × 126 pixels, a standard requirement for MobileNetV2.

Preprocessing Applied:

- Rescaling (1/255)
- Data augmentation:
  - Rotation
  - Zoom
  - Width/height shift
  - Horizontal flip

This significantly enhances dataset variety and mitigates overfitting, leading to more robust models.

Data Loading Method

We utilized TensorFlow's ImageDataGenerator for efficient data handling.

Reads images from folders

Automatically scans and loads images based on directory structure.

Labels them using folder names

It gives each image a class label based on its folder name.

Loads them in batches

Optimizes memory usage and training speed by processing images in defined batches.

Splits data cleanly

It keeps separate sets for training, validation, and testing to avoid data mixing.

Model Architecture

Base Model: MobileNetV2 (Pre-trained on ImageNet)

- Used as a **feature extractor**, leveraging its deep understanding of visual patterns.
- **Layers frozen** to preserve learned weights and prevent extensive retraining.
- Known for being **lightweight and fast**, ideal for deployment in resource-constrained environments.

Added Custom Layers:

→ **Global Average Pooling**
Reduces spatial dimensions while retaining important information.

→ **Dense (fully connected) layers**
Interprets the high-level features for classification.

→ **Dropout for regularization**
Prevents overfitting by randomly omitting units during training.

→ **Final Softmax layer**
Outputs probabilities for each class in multi-class classification.

This hybrid architecture ensures high accuracy even with limited training data, effectively balancing complexity and performance.

Training Details

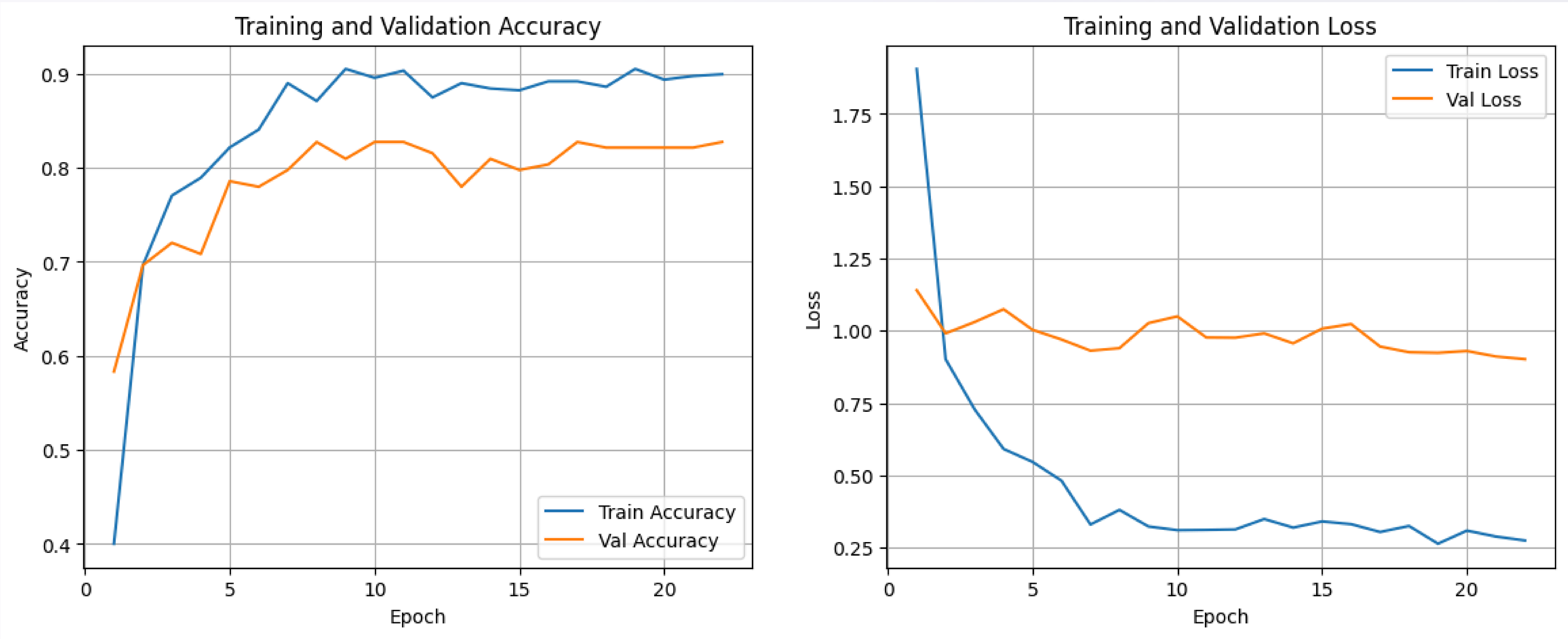
The model training was meticulously configured to ensure optimal performance and generalization.

Loss Function:	Optimizer:	Metrics:
`categorical_crossentropy`	**Adam**	**Accuracy**
Ideal for multi-class classification, measuring the difference between predicted and true probability distributions.	An adaptive learning rate optimization algorithm known for its efficiency and effectiveness.	The primary metric used to evaluate the model's performance during training and validation.

Callbacks Used:

- **EarlyStopping**: Monitors a specified metric (e.g., validation loss) and stops training when improvement ceases, preventing overfitting.
- **ModelCheckpoint**: Automatically saves the model weights (or the entire model) at regular intervals, particularly when validation performance improves, ensuring the best model is preserved.

Training was exclusively conducted on augmented images to enhance robustness, while validation was performed on pristine, untouched data to provide an unbiased evaluation of generalization capabilities.



Model Output

The trained model demonstrates compelling performance metrics:

~90%

High Accuracy

Achieved on both training and validation datasets, reflecting strong learning capabilities. (Specific values are dependent on the last training run.)

Stable

Stable Learning

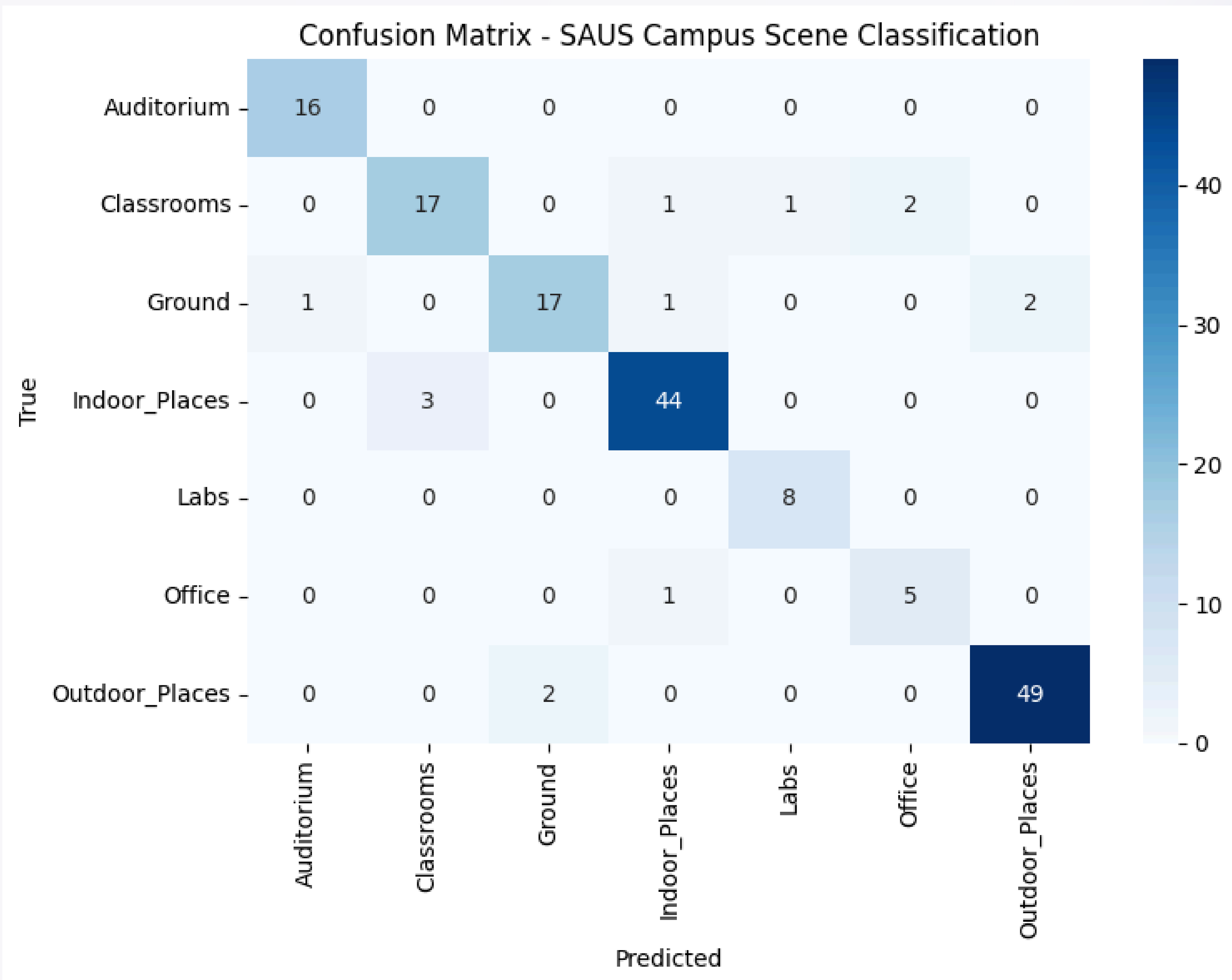
Ensured by comprehensive data augmentation techniques, preventing erratic performance swings.

Excellent

Good Generalization

The model performs effectively on unseen test data, indicating its ability to classify novel images reliably.

If further details are required, the training accuracy, validation accuracy, and loss graphs can be extracted directly from the notebook to provide a visual representation of the learning process.



Classification Report:

	precision	recall	f1-score	support
Auditorium	0.94	1.00	0.97	16
Classrooms	0.85	0.81	0.83	21
Ground	0.89	0.81	0.85	21
Indoor_Places	0.94	0.94	0.94	47
Labs	0.89	1.00	0.94	8
Office	0.71	0.83	0.77	6
Outdoor_Places	0.96	0.96	0.96	51
accuracy			0.92	170
macro avg	0.88	0.91	0.89	170
weighted avg	0.92	0.92	0.92	170

Testing

The final evaluation of the model included a careful testing phase to check how well it works in real situations.

> Testing was primarily executed using the model.predict() function, which generated class probabilities for the unseen test dataset.

Predicted classes were meticulously mapped back to their original labels using the train_generator.class_indices attribute, ensuring accurate interpretation of the model's output.

Sample Visualizations

A selection of test images were visualized alongside their respective predictions. This qualitative assessment provided crucial insights and confirmed the model's correctness, demonstrating its ability to accurately classify diverse university scenes.

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
Predicted class index: 1
Predicted class name : Classrooms
Confidence           : 99.62%
```

Conclusion

This project shows that transfer learning works very well for scene classification. Using a pre-trained CNN model, we were able to classify images quickly and accurately, even with a smaller dataset. It proves that Deep Learning can solve real-world image recognition tasks effectively.

Improved Performance

Transfer Learning significantly boosted model accuracy and efficiency, leveraging pre-trained knowledge.

Fast & Accurate

MobileNetV2 proved to be an excellent choice, enabling rapid training while maintaining high classification accuracy.

Successful Classification

The model accurately classified various scene types within the SAUS dataset.

Practical Application

This project effectively showcased the practical utility of Convolutional Neural Networks (CNNs) in real-world image categorization tasks.

Future Enhancements

To further advance this project, several exciting avenues can be explored:

TFLite Conversion

Converting the model to **TFLite** for optimized performance on mobile and edge devices.

Other Models

Experimenting with alternative state-of-the-art models like **ResNet** or **EfficientNet** for potential performance gains.

Model Decisions

Visualizing model decisions using techniques like **Grad-CAM** to gain deeper insights into its reasoning.

More Images

Expanding the dataset with more images to enhance the model's robustness and generalization capabilities.

Thank You