Image Classification: 6-Label Model Report

1. Introduction

Image classification plays a crucial role in modern AI applications. Using a deep learning approach, we trained a
Convolutional Neural Network (CNN) for high accuracy and efficient performance.

This project was undertaken as part of the DataMites Internship program, focusing on advanced deep learning techniques
for image recognition.

2. Business Case

Develop a robust and scalable Machine Learning (ML) model capable of accurately classifying images into six distinct
categories based on learned visual features.

3. Objectives

- Develop a high-performance image classification model.
- Ensure the model generalizes well on external data.
- Optimize training strategies for enhanced accuracy.

4. Project Goal

1. Data understanding and preprocessing.
2. Implement a CNN model for image classification.

5. Data Overview

- Dataset Source: Provided as a folder with labeled images.
- Number of Classes: 6
- Training Data: Separate directory with labeled subfolders.
- Validation Data: Separate validation directory.

- Test Data: Provided test images for evaluation.
- Image Resolution: Resized to (224, 224).

6. Methodology

Data Extraction: Data extraction is crucial in preparing data for analysis and model training.

Data Preprocessing:

- Data Cleaning: Removed corrupted images.
- Handling Missing Values: Verified dataset integrity.

Data Transformation:

- Normalization & Scaling: Rescaled pixel values to (0,1) for better convergence.

7. Model Details

Chosen Model: Convolutional Neural Network (CNN)

Why CNN?

- Efficient for image-based tasks.
- Strong feature extraction capability.
- Proven performance in classification tasks.

Architecture:

- Input Shape: (224, 224, 3)
- Pretrained Weights: None (Trained from scratch)
- Convolutional Layers: Extract spatial features.
- Pooling Layers: Reduce dimensionality.
- Fully Connected Layers: Classify the extracted features.
- Dense Output Layer: 6 neurons with softmax activation.
- Loss Function: Categorical Crossentropy.
- Optimizer: Adam.
- Evaluation Metric: Accuracy.

8. Accuracy & Performance Evaluation

Accuracy Plots:

- Training and Validation Accuracy: Measures model learning over epochs.
- Training and Validation Loss: Indicates error minimization.

Confusion Matrix: Visualizes predictions versus actual labels.

Classification Report (Heatmap Representation): Displays precision, recall, and F1-score for each class.

9. Challenges

Low Initial Accuracy:

- Solution: Switched to optimized CNN and applied data augmentation.

Class Imbalance:

- Solution: Verified dataset balance and adjusted augmentation strategies.

10. Summary

- Model: CNN for image classification.
- Dataset: 6 classes, resized images to (224, 224).
- Data Preprocessing: Used ImageDataGenerator for augmentation.
- Training: Optimized for better accuracy.
- Performance: Achieved high accuracy and minimized loss.
- Model Saving: Final model stored in Google Drive.

11. Conclusion

The image classification model using CNN demonstrated high training accuracy and robust validation performance,
indicating strong generalization capabilities. The implementation of data augmentation, hyperparameter tuning, and model
optimization contributed significantly to the model's effectiveness.

The deployment approach can be extended to real-world applications, and future enhancements may include transfer learning
techniques to further improve accuracy.

This project demonstrates the effectiveness of deep learning for image classification tasks and provides a foundation for
further advancements in AI-driven computer vision applications.