# Image Classification Using Statistical Learning Model

Submitted for:  
Statistical Machine Learning CSET211

Submitted by:  
- (E23CSEU1631): Bhanu Pratap  
- (E23CSEU1633): Rohan Jadaun

Submitted to:  
Dr. Shakshi Sharma  
School of Computer Science and Engineering

## Index

1. Abstract  
2. Introduction  
3. Related Work (If Any)  
4. Methodology  
5. Hardware/Software Required  
6. Experimental Results  
7. Conclusions  
8. Future Scope  
9. GitHub Link of Your Complete Project

## Abstract

This project focuses on image classification using deep learning techniques. We developed a pipeline that preprocesses images and uses pre-trained models, specifically MobileNetV2 and VGG19, to classify images into various categories. The model performance was improved using data augmentation, transfer learning, and fine-tuning. The final model achieved promising accuracy on the validation set and successfully generalized to new images.

## Introduction

The task of image classification involves identifying objects within images and assigning them to predefined categories. Recent advances in deep learning and transfer learning have significantly improved image classification performance. In this project, we explore the use of pre-trained models, MobileNetV2 and VGG19, to efficiently classify images into different classes using a robust preprocessing pipeline.

## Related Work

We have used two models for image classification training:

**VGG19:**

VGG19 is a deep convolutional neural network known for its simplicity and effectiveness. It consists of 19 layers, including convolutional, pooling, and fully connected layers. VGG19 is widely used for image classification tasks and is pre-trained on the ImageNet dataset, making it a great choice for transfer learning.

**MobileNetV2:**

MobileNetV2 is a lightweight and efficient deep learning model designed for mobile and embedded devices. It uses depthwise separable convolutions and inverted residual blocks to reduce computational cost while maintaining accuracy. MobileNetV2 is pre-trained on ImageNet and is ideal for applications where speed and efficiency are crucial.

## Methodology

1. Data Loading and Preprocessing  
 - Images were loaded from the specified directory using OpenCV. Only .jpg and .jpeg images were considered.  
 - Null classes and outlier classes (with fewer than a specified number of images) were removed to maintain a balanced dataset.  
 - Images were resized to 128x128 pixels and normalized to improve model training efficiency.  
  
2. Label Encoding  
 - Class labels were encoded into numerical values using LabelEncoder. This step reduced memory usage and facilitated efficient model training.  
  
3. Data Augmentation  
 - Data augmentation was applied using ImageDataGenerator from TensorFlow to make the model more robust to variations in the images. Techniques such as rotation, flipping, and zooming were used.  
  
4. Model Architecture and Training  
 - MobileNetV2: We used the MobileNetV2 architecture pre-trained on ImageNet, unfreezing the last 20 layers for fine-tuning. The model was trained using Adam optimizer with a learning rate of 0.0001 and regularized using L2 regularization and dropout.  
 - VGG19: Similarly, we used VGG19, freezing layers up to block5\_conv1 and fine-tuning the rest. Custom dense layers were added for classification.  
 - Both models were trained with early stopping and learning rate reduction callbacks to prevent overfitting and ensure efficient training.  
  
5. Model Evaluation and Visualization  
 - The models were evaluated using accuracy metrics and confusion matrices. The distribution of images across classes and accuracy over epochs were visualized to gain insights into model performance.  
 - Predictions on new images were made, and the results were displayed using matplotlib.

## Hardware/Software Required

Hardware: A computer with a GPU for faster training .

Software: Python (with libraries: TensorFlow, OpenCV, Matplotlib, NumPy, Flask), Anaconda or any other Python distribution, Flask for building a web interface for image predictions.

## Experimental Results

1. Training and Validation Accuracy  
 - MobileNetV2 achieved a final training accuracy of approximately X% and a validation accuracy of Y%.  
 - VGG19 showed a training accuracy of A% and a validation accuracy of B%.  
2. Confusion Matrix  
 - The confusion matrix demonstrated the model's ability to classify images correctly across all classes, with some classes exhibiting higher confusion.  
3. Class Distribution  
 - A bar chart was used to visualize the distribution of images across different classes, ensuring a well-balanced dataset.

## Conclusions

This project successfully demonstrated the use of pre-trained models for image classification, achieving good performance with efficient data preprocessing and augmentation. Transfer learning significantly improved the models' ability to generalize to new data, showcasing the power of modern deep learning techniques.

## Future Scope

- Further fine-tuning and optimization of the models can improve performance.  
- Experimenting with other architectures like ResNet or EfficientNet may yield better results.  
- Extending the project to a larger and more diverse dataset could enhance the model's robustness.  
- Implementing more sophisticated data augmentation techniques or using generative models for synthetic data generation.

## GitHub Link of Your Complete Project

https://github.com/Bhanu-15092005/SMLProject