**OPEN ENDED LAB**



**Course:** Deep Learning

**Name:** Zeeshan Saeed

**Registration:** L1F21BSEE0033

**Submitted to:** Engr. Muhammad Umar

**Department of Electrical Engineering**

**University of Central Punjab,**

**Lahore,**

**Image Classification Using CNN with Xception Architecture on Kaggle Dataset**

**Problem Statement**

The task is to develop an image classification system using a **Convolutional Neural Network (CNN)** model from the **Keras library**. The system should be capable of automatically importing a labeled image dataset from an external source (Kaggle), preprocessing the data, training a CNN model, and evaluating its performance.

The objective is to:

* Import and organize the dataset from the provided Kaggle URL.
* Build and train a deep learning model using a suitable CNN architecture from Keras.
* Achieve an image classification **accuracy of at least 95%** on the validation data.
* Visualize training and validation performance using accuracy and loss plots.

The end goal is to implement a complete image classification pipeline with high accuracy using modern deep learning techniques available in Keras.

Certainly! Here's a concise and professional description **"About the Dataset"** based on the information you provided, suitable for reports, documentation, or presentations:

**About the Dataset**

The dataset used in this project is the **RSI-CB256 Satellite Image Classification Dataset**, sourced from Kaggle. It contains high-resolution remote sensing images gathered from a combination of sensors and Google Maps snapshots. The dataset is organized into **four distinct land cover classes**: **desert, green area, cloudy region, and water bodies**, making it well-suited for multi-class image classification tasks.

This dataset plays a crucial role in advancing the field of **remote sensing (RS) image interpretation**, which is foundational for numerous applications in environmental monitoring, urban planning, agriculture, and disaster management. As the accessibility of RS imagery has increased, the demand for automated interpretation using intelligent algorithms has grown accordingly.

The dataset was prepared following modern standards for benchmark dataset construction, ensuring high-quality, annotated data for training and testing deep learning models. It serves as an essential resource for developing, evaluating, and comparing deep learning approaches, especially in the context of **image scene classification**.

By utilizing this dataset, researchers and practitioners can explore various deep learning techniques in **artificial intelligence**, **computer vision**, and **image processing**, and contribute to the growing body of knowledge in **satellite image analysis** and **remote sensing automation**.

**Introduction– Xception**

Convolutional Neural Networks (CNNs) have become the cornerstone of modern computer vision due to their ability to automatically learn spatial hierarchies of features from images. In this project, we have employed **Xception**, a state-of-the-art CNN architecture from Keras, to perform high-accuracy image classification.

The **Xception model**—short for **Extreme Inception**—was introduced by François Chollet as an extension of the Inception architecture. The key innovation in Xception is the use of **depthwise separable convolutions**, which decompose a standard convolution into two simpler operations: depthwise convolution and pointwise convolution. This significantly reduces the number of parameters and computational cost, while often improving performance.

Xception is designed with three primary sections:

1. **Entry Flow** – which handles initial feature extraction.
2. **Middle Flow** – a deep stack of repeated layers that model complex patterns.
3. **Exit Flow** – where feature maps are condensed into a final output representation.

In this project, we used the Xception model with include\_top=False, meaning we excluded the default ImageNet classifier layer and replaced it with custom dense layers suitable for our 4-class classification task. The base model was initially frozen to utilize transfer learning, leveraging the pretrained weights from ImageNet, and later fine-tuned for improved performance.

**Xception Convolutional Layer Architecture**

The following table outlines the major layers in the Xception architecture when used without the top classifier layer:

| **Block** | **Type** | **Filters / Units** | **Stride** | **Details** |
| --- | --- | --- | --- | --- |
| Input | Input Layer | 224 × 224 × 3 | – | RGB image |
| Conv1 | Conv2D | 32 filters, 3×3 | 2×2 | Standard convolution |
| Conv2 | Conv2D | 64 filters, 3×3 | 1×1 | Standard convolution |
| Entry Flow Block 1 | SeparableConv2D ×2 + MaxPool | 128 filters | 2×2 | With skip connection |
| Entry Flow Block 2 | SeparableConv2D ×2 + MaxPool | 256 filters | 2×2 | With skip connection |
| Entry Flow Block 3 | SeparableConv2D ×2 + MaxPool | 728 filters | 2×2 | With skip connection |
| Middle Flow ×8 | SeparableConv2D ×3 | 728 filters | 1×1 | Repeated 8 times with residual connections |
| Exit Flow Block 1 | SeparableConv2D ×2 + MaxPool | 1024 filters | 2×2 | With skip connection |
| Exit Flow Block 2 | SeparableConv2D | 1536 filters | 1×1 | Deeper feature representation |
| Exit Flow Block 3 | SeparableConv2D | 2048 filters | 1×1 | Final feature extractor |
| Global Pooling | GlobalAveragePooling2D | – | – | Output shape: (None, 2048) |

**Why Xception Was Chosen**

* **Efficiency:** Depthwise separable convolutions drastically reduce computation.
* **Performance:** Xception has demonstrated superior results on standard benchmarks compared to older architectures like VGG or Inception-v3.
* **Transfer Learning Ready:** It is pretrained on ImageNet, making it ideal for fine-tuning on custom datasets.
* **Scalability:** Its modular structure allows freezing or unfreezing layers for staged training.

In our project, the Xception model delivered excellent performance with a final validation accuracy of **97.2%** after just 3 epochs of training—well above our target threshold of 95%. This confirms its suitability for real-world classification tasks.

**Conclusion**

In this project, we successfully developed a deep learning-based image classification model using the **Xception Convolutional Neural Network (CNN)** architecture from Keras. The dataset, sourced from Kaggle, consisted of four image categories. To enhance the model's ability to generalize, data augmentation techniques and transfer learning were employed.

The model was trained over **3 epochs**. During training:

* **Training Accuracy** steadily improved from **75.3% in Epoch 1** to **96.8% in Epoch 3**.
* **Validation Accuracy** started at **95.5%**, increased to **96.7%**, and finally reached an impressive **97.2%**.
* Similarly, **Validation Loss** dropped consistently from **0.30** to **0.13**, indicating improved generalization and minimal overfitting.

The achieved validation accuracy of **97.2%** not only meets but exceeds the target threshold of **95%**, validating the effectiveness of the chosen model architecture and training approach.

The results confirm that with appropriate data preprocessing, transfer learning, and regularization techniques, high-performance image classification is achievable even with a relatively small number of training epochs. This approach can now be extended or fine-tuned for other image classification problems or datasets.