Submitted by -:

Kushagra Pandey (22070127033)

Vivek Jadhav (22070127073)

Varad Desai (22070127074)

Submitted to-:

Dr. Sameer Sayad

Robotics and Automation

SIT Pune

Explainable AI: Scene Classification and Grad Cam Visualization

- **Literature Review**

Sr.No	Author	Year	Title of the paper	Methodology	Dataset used	Model used	Accuracy	Research Gaps
1	Dr. Annu Sharma & Nagendra Kumar K. S.	2024	Explainable AI: Scene Classification and Grad Cam Visualization	CNN-based model with residual blocks, Grad-CAM for visualization	Satellite images dataset	CNN with Grad-CAM	80%	Emphasizes the need for transparency in AI models; gaps in addressing computational complexity with larger datasets
2	Adadi, A., & Berrada, M.	2018	Peeking Inside the Black-Box: A Survey on Explainable AI	Survey of XAI methods, including Grad-CAM	N/A	Local Interpretable Model-Agnostic, ANN(Rule Extraction & Model Distillation)	survey	Focuses on medical data; does not deeply explore scene classification.
3	Zhao, Li, & Dai	2018	Application of Grad-CAM for Railway Scene Classification	Grad-CAM for visualizing model decisions	Railway scene images	CNN	from 88.7% to 92.5%	Application-specific; limited to railway scene datasets
4	He et al.	2016	Deep Residual Learning for Image Recognition	Deep Residual Learning, Residual Networks (ResNet)	ImageNet	ResNet	96.4%	Significant improvement in image recognition; high computational cost
5	Selvaraju et al.	2017	Grad-CAM: Visual Explanations from Deep Networks	Grad-CAM, gradient-based localization method	ImageNet, CUB-200-2011, PASCAL VOC	Stacked Neural Networks	Top1 Loc Error(57.20%) Top5 Loc Error(45.10%) Top1 CLS Error(33.40%) Top5 CLS Error(12.20%)	Introduced Grad-CAM; generalizable but requires high computation

- ## Research Gap

1. Limited Application of Grad-CAM in Scene Classification: Although Grad-CAM is a well-established XAI technique, its application in scene classification tasks remains underexplored, particularly in scenarios involving complex scenes with multiple elements.

2. Lack of Comprehensive Studies on Diverse Scene Types: Existing literature focuses on specific types of data (e.g., medical images), with little attention given to diverse scene types such as urban (buildings, streets) and natural (forests, glaciers) environments.

3. Need for Improved Interpretability in Complex Environments: There is a gap in the literature regarding the effectiveness of Grad-CAM in providing clear and meaningful visual explanations for classifications made in environments with varying levels of complexity.

- ## Problem Statement

In recent years, convolutional neural networks (CNNs) have demonstrated remarkable success in scene classification tasks, effectively distinguishing between various environments such as urban settings (buildings, streets) and natural landscapes (forests, glaciers). However, the interpretability of these models remains a significant challenge. As these models are often viewed as "black boxes," understanding the rationale behind their predictions is crucial, especially in applications where transparency and trust are paramount.

Despite the development of explainable AI (XAI) techniques like Grad-CAM, their application to complex scenes involving multiple elements is limited. Current literature largely focuses on simple, single-object images or specific domains such as medical imaging, leaving a gap in the understanding of how Grad-CAM can be used to interpret CNN decisions in more intricate and diverse environments. Moreover, existing studies do not provide a comprehensive evaluation of Grad-CAM’s effectiveness in enhancing the interpretability of scene classification models across various scene types.

- ## Objectives

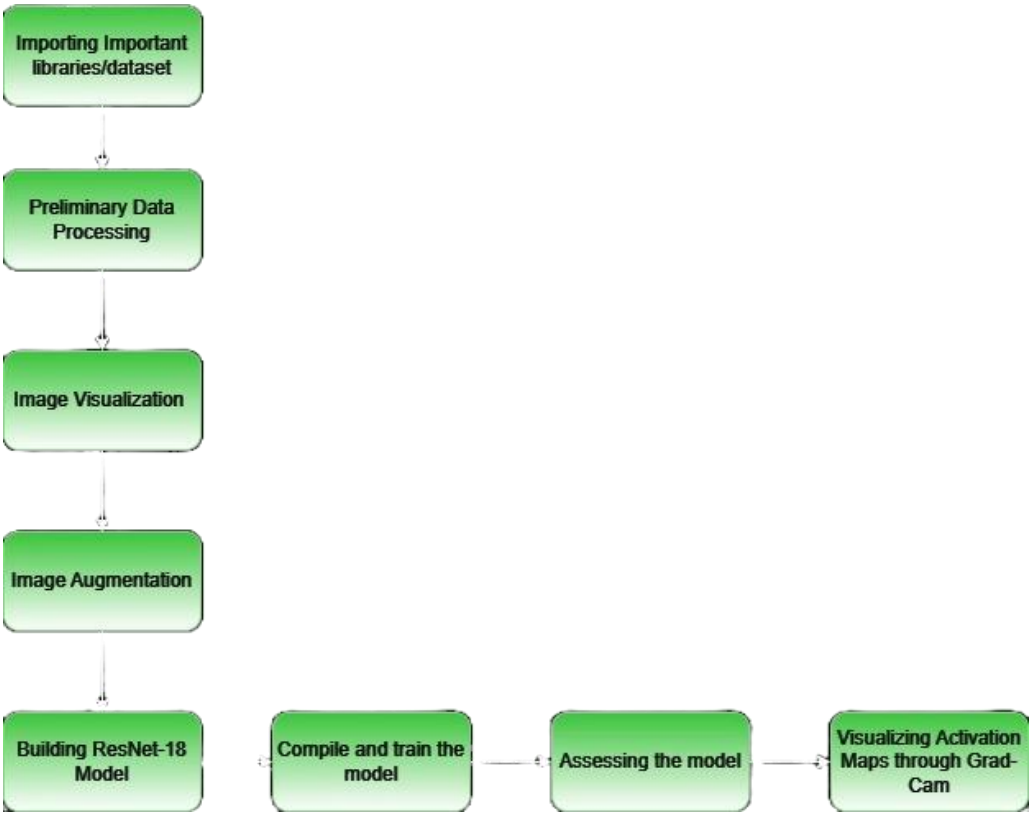
1. To explore the application of Grad-CAM in improving the interpretability of scene classification models, focusing on diverse environments such as buildings, forests, glaciers, mountains, seas, and streets.

2. To develop a comprehensive evaluation framework that measures the effectiveness of Grad-CAM visualizations in explaining CNN decisions across different scene types.

3. To enhance the transparency and trustworthiness of scene classification models by refining Grad-CAM visualizations to handle complex scenes with multiple elements.

- **Refined Methodology**

  - **Flowchart**



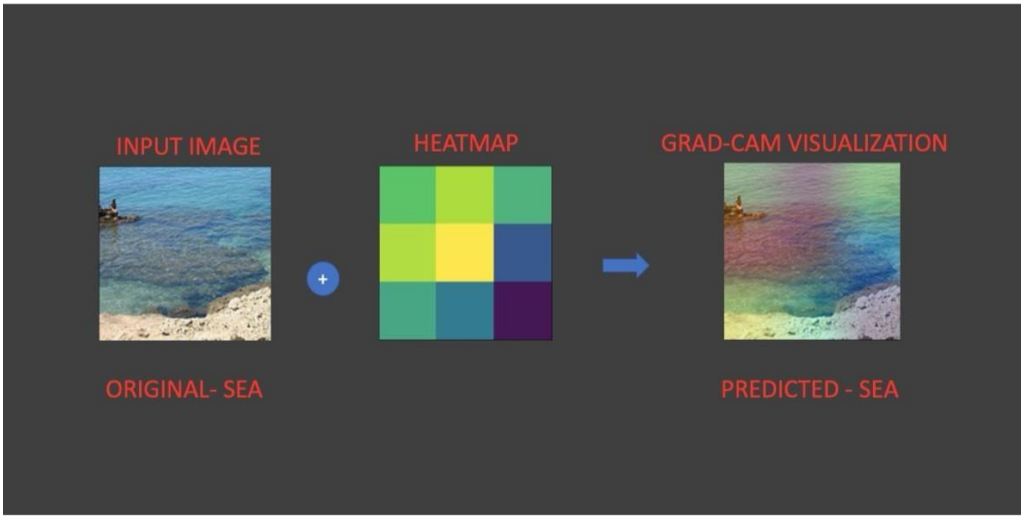
  - **Explanation**

In this project, we will train deep Convolutional Neural Networks(CNNs) and Residual Blocks to detect the type of scenery in an image. This project could be practically used for detecting the type of scenery from satellite image.

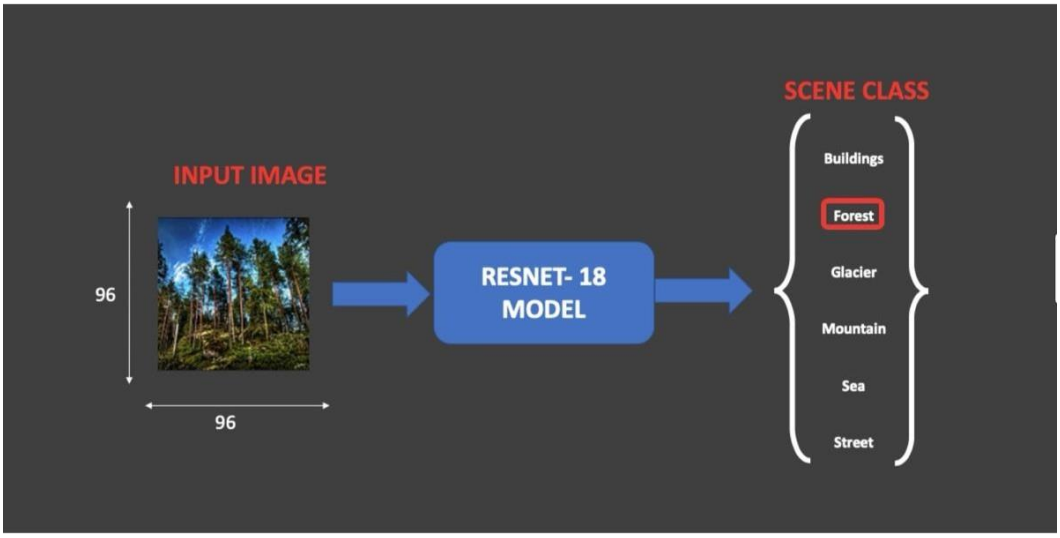
MICROSOFT AI FOR EARTH, has created the most detailed United States forest map using satellite imagery and AI which would essentially be a game changer in reducing deforestation, pests and wildfires.

https://www.microsoft.com/en-us/ai/ai-for-earth

Explainable AI: Gradient-Weighted Class Activation Map (Grad-CAM) helps visualize the regions of the input that contributed towards making prediction by the model.



The dataset consists of **17034 images** belonging to 6 categories. Categories that are present in the data are **buildings, forest, glacier, mountain, sea and street**.



Step 1 : Importing Important libraries and dataset

In this step, we will start by importing important libraries and dependencies which are required for this project which includes *pandas, NumPy, matplotlib, seaborn, TensorFlow, OS, PIL, ImageDataGenerator, Keras, ResNet50, InceptionResNetV2, SGD optimizer, LearningRateScheduler etc.*

After Importing our dependencies, we will check the number of images in our training and testing dataset using the **os.listdir** function.

Step 2: Preliminary Data Processing

In this step we will be exploring our dataset through different ways like plotting graphs, charts etc. We will also check the number of images in each class in the training and testing dataset.

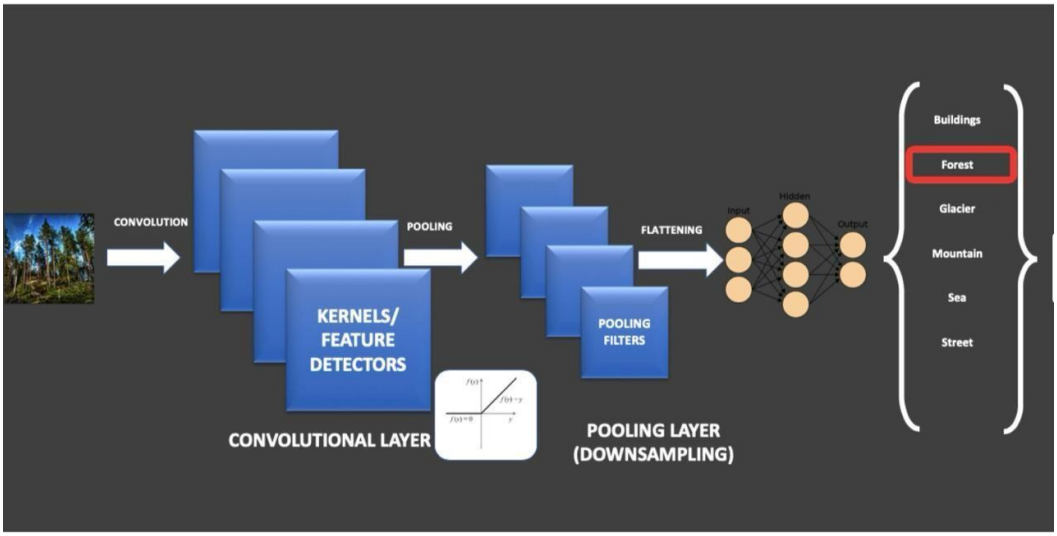
Step 3: Image Visualization

We will be visualizing the images in the dataset using the subplot function, we will get the list of images in the particular class and plot 5-10 images per class for both the testing and training dataset.

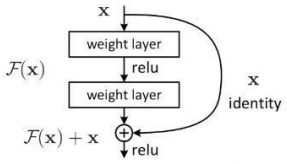
Step 4: Image Augmentation

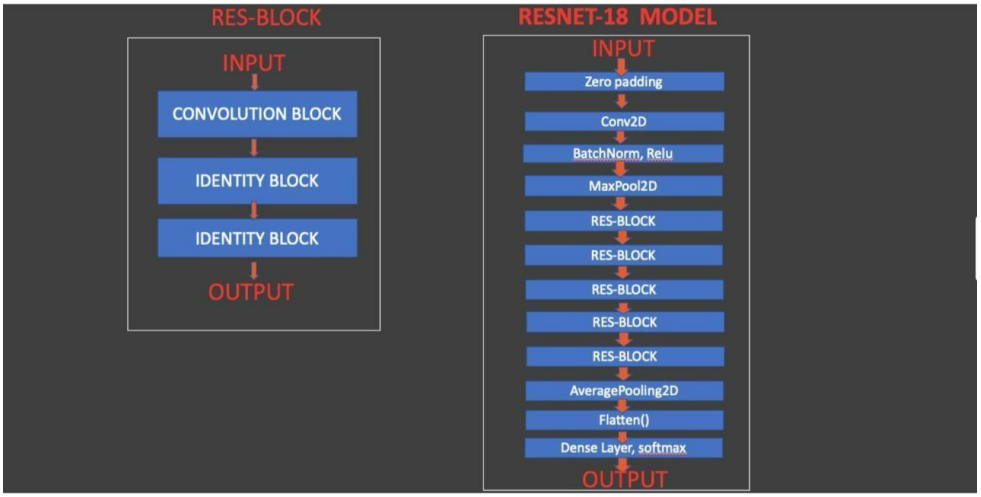
In this step we will perform data augmentation and create data generators. We will start by creating a run-time augmentation on training and test dataset.
For training data generator, we will add normalization, shear angle, zooming range and horizontal flip.
For testing data generator, we will only normalize the data.

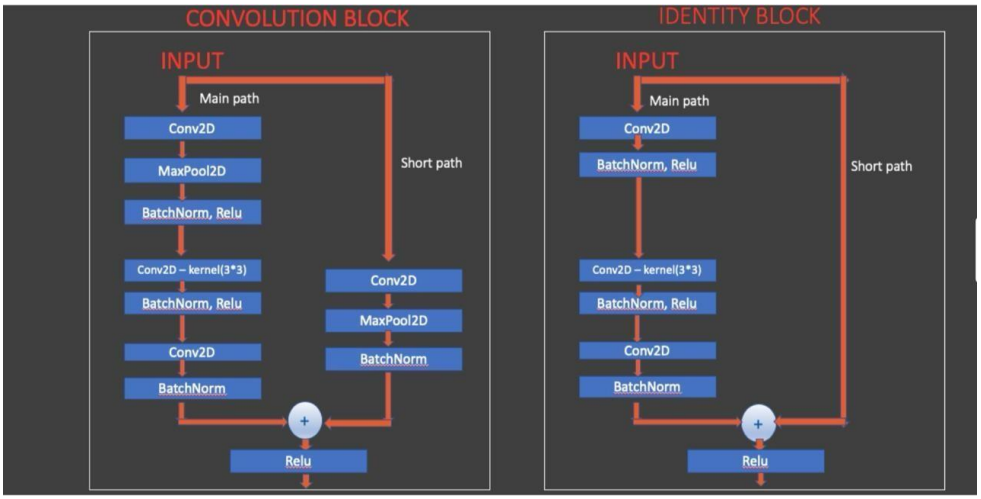
Step 5: Building the ResNet-18 Model



As CNNs grow deeper, vanishing gradient tend to occur which negatively impact network performance.
Vanishing gradient problem occurs when the gradient is back propagated to earlier layers which results in a very small gradient.
Residual Neural Network includes “skip connection” feature which enables training of 152 layers without vanishing gradient issues.
ResNet works by adding “identity mappings” on top of the CNN.







Step 6: Compile and train the model

We will be using early stopping to exit training if validation loss is not decreasing even after certain epochs. Then we will save the best model with lower validation loss.

Step 7: Assessing the model prediction

We will evaluate the performance of the model by using the model.evaluate() function and we will generate test accuracy, and print out the classification report and confusion matrix and comment on the model performance on the basis of that.

Step 8: Visualizing Activation Maps through Grad-CAM

Gradient-Weighted Class Activation Mapping (Grad-CAM) helps visualize the regions of the input that contributed towards making prediction by the model.

It does so by using the class specific gradient information flowing into the final convolutional layer of CNN to localize the important regions in the image that resulted in predicting that particular class.

To visualize the activation maps, we need to pass the image through the model to make predictions.

We will use argmax to find the index corresponding to the maximum value in the prediction, which gives us the predicted class. Now, we take the predicted value for that class by the model.