# Project 2: Classify a Collection of Aerial Scenes Using a CNN (21 classes)

**Signal and Image Acquisition and Modeling in Environment**

**Student: Kerem Erciyes**
**No: 910560**

**Lecturers:**

**Assoc. Prof. Federico Guio**

**Asst. Prof. Matteo Fossati**

# Report:

---

## Abstract

This report explores the application of Convolutional Neural Networks (CNNs) for classifying aerial images from the UC Merced Land Use Dataset. ResNet50 model is used which is already pre-trained on the ImageNet dataset and fine-tuned for the specific classification task. The process involves data augmentation, fine-tuning, and comprehensive evaluation of the model's performance through various metrics and visualizations.

With my findings I accomplish significant accuracy improvements, demonstrating the effectiveness of CNNs in handling complex aerial image classification problems.

## Introduction

The project aims to adapt a pre-trained ResNet50 model for a customized classification task, for The UC Merced Land Use Dataset which comprises high-resolution aerial images. The advantages of using CNNs include their robustness to variations in the dataset and their ability to learn complex patterns directly from the data. The project can be examined in 4 basic steps: data preprocessing, creating classification model, evaluating the model, visualizing the results. Firstly, data is preprocessed and augmented. Secondly, data is splitted to train, validation and test subsets. Then, the model is defined. The model is tuned by transfer learning to optimize performance and computational efficiency. Apart from this, robust steps are added to evaluate the model more accurately. Finally, model’s visual results close to expectations are obtained.

## Data and Methodology

### Data Description

The dataset consists of images categorized into 21 distinct classes, with each class containing 100 images.

### Data Description

The images are resized to 256x256 pixels for uniformity.

Data augmentation techniques such as color jittering (changing brightness, contrast and saturation $\mp$ 0.2) are applied to increase dataset diversity and improve model generalization. Random horizontal flips and random rotations techniques are not used for our dataset since the aerial images are already in random rotations.

Pixels are normalized with specific criteria for ResNet50.

Data are labeled.

First data is shuffled then divided to train, validation and test subsets but when the test data is checked the unbalanced number of classes are noticed. Then, first each class is divided to its subsets as train, validation and test (as 0.7, 0.15, 0.15 ratio respectively). Later, general train, validation and test subsets of data are created by gathering each classes subset's and shuffling them.

## Methodology

**Model Choice and Justification**: ResNet50 is selected due to its depth and proven performance in image classification tasks. Transfer learning from the ImageNet dataset enables the model to leverage pre-trained weights and improves convergence speed and performance.

**Training and Fine-tuning**:

- Last twenty layers of the model are unfreezed to allow fine-tuning, ensuring the model adapts to the specific features of aerial images. (Because first layers learn more fundamental features and last layers learn high level features, unfreezing the only the last layers were a considerable option) Training with transfer learning reduced computational costs compared to training from scratch.
- First layer of ResNet is adjusted according to image size and last layer is adjusted according to fully connected network.
- Modifications included adding batch normalization and dropout layers to the fully connected layers to improve generalization and prevent overfitting.
- The model is trained using Stochastic Gradient Descent (SGD) with momentum and weight decay, which helps in stable and efficient convergence. The use of SGD with momentum facilitated efficient and stable convergence, making the training process more efficient.
- It is trained with 50 epochs and early stopping is added to avoid overfitting and stop the training step when loss doesn't change or start increasing.
- Best model is saved according to validation accuracy.

## Model Architecture and Layer Details

## Layer Details and Parameters:

- The first convolutional layer has 64 filters with a 7x7 kernel, stride of 2, and padding of 3, followed by batch normalization and ReLU activation.
- The initial layers reduce the input size from 256x256 to 128x128, using max pooling.
- Residual blocks throughout the network, including the Bottleneck layers, have 3 convolutional layers each with varying kernel sizes (1x1, 3x3, 1x1).
- The layer sizes and parameter counts are detailed as follows:
  - Example Layer: `Conv2d-1` produces an output of shape [-1, 64, 128, 128] with 9,408 parameters.
  - Last Layer before Fully Connected: `Conv2d-171` produces an output of shape [-1, 2048, 8, 8] with 1,048,576 parameters.

## Fully Connected Layer:

- After the global average pooling, a linear layer reduces the 2048 features to 512, followed by batch normalization, ReLU activation, and dropout (p=0.5).
- Another linear layer reduces the features from 512 to 256, again followed by batch normalization, ReLU activation, and dropout (p=0.5).
- The final linear layer outputs the class scores for the 21 classes.

**Evaluation Metrics**:

- The model's performance is assessed using accuracy, confusion matrices, and ROC-AUC curves. These metrics provide a comprehensive view of the model's classification capabilities.
- Visualization techniques, feature maps and filter visualizations, are examined to interpret the model's internal representations and understand its decision-making process.

## Alternative Approaches

Other potential algorithms are considered before ResNet:

- **VGG16/VGG19**: Simpler architectures but less efficient in capturing complex patterns compared to ResNet.
- **MobileNet**: Optimized for mobile applications, balancing latency and accuracy, but may not perform as well on high-resolution images.
- **Defining a CNN based model from scratch**: Cannot generalize the data good due to lack of data and high number of class compared to number of samples.

## Results and Discussion

**Model Performance**: The fine-tuned ResNet50 achieved a test accuracy of 92.06%. The confusion matrix and ROC-AUC curves indicate high precision and recall for all of the classes.

**Filter Visualization**: Example filter’s weights are shown as colored map.

**Feature Map Visualization**: Visualizing feature maps at different layers showed how the model abstracts the input image, capturing edges, textures, and complex patterns. This shows that the model learns the representations truly.

**Data Augmentation Impact**:

- Data augmentation significantly improved the model's performance by reducing overfitting and enhancing generalization.
- There is no significant bias introduced, indicating that the augmentations are appropriate for the dataset.

**Computational Efficiency**:

- Training with transfer learning reduced computational costs compared to training from scratch.
- The use of SGD with momentum facilitated efficient and stable convergence, making the training process more efficient.

## Challenges and Adjustments

- Initial attempts at visualizing deeper layers' filters resulted in zero-variance filters, highlighting the need for enhanced normalization techniques.
- Fine-tuning only the last few layers, as opposed to the entire network, provided a balance between computational efficiency and model performance.
- Adjustments in data augmentation strategies are necessary to avoid overfitting and ensure the model's robustness to diverse image conditions.

## References


- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.
- Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.
- UC Merced Land Use Dataset. Retrieved from: /kaggle/input/uc-merced-land-use-dataset