

Classifying Natural Environments: Using Deep Learning in Image Classification

Code: [Github Link](#)

Machine Learning, Spring 2023

Ankita Nambiar

ankita7@uchicago.edu

Agenda



1. Problem Statement

2. Data Profile

3. Exploratory Data Analysis

4. Feature Engineering & Transformations

5. Modeling

6. Conclusions & Recommendations

1. Problem Statement

Problem Statement

This project's focus is a **multiclass image classification task** involving six distinct categories: buildings, forests, mountains, seas, streets, and glaciers.

The primary aim is to **develop a robust deep learning model capable of accurately identifying and classifying images** of landscapes and urban environments.

Applications

The importance of this problem lies in its applicability and relevance in various domains. Accurate image classification has practical applications:

1. **Urban planning:** By accurately identifying buildings and streets from aerial or satellite images, city planners can analyze urban infrastructure, identify areas for improvement, and make informed decisions regarding city development and expansion.
2. **Environmental monitoring:** Classifying images of forests, mountains, and glaciers can aid in monitoring ecosystem health, studying deforestation patterns, assessing climate change impacts, and supporting conservation efforts.
3. **Tourism and travel:** Efficiently categorizing images of seas, beaches, mountains, and buildings can enhance travel experiences by providing accurate and visually appealing recommendations for tourists.
4. **Remote sensing:** Interpreting satellite imagery, enabling analysis of large-scale landscapes, and assisting in disaster management, agriculture, and forestry.

Assumptions and Hypotheses about the Data and Model

Assumptions

Assumption 1: Sufficient and Representative Data

The dataset contains a **large and diverse collection of images** representing various lightings and environmental settings for each class, ensuring the model learns robust features and can generalize well to unseen images.

Assumption 2: Class Separability

The classes in the dataset are **reasonably separable based on visual features**, allowing our model to learn distinctive patterns and accurately classify images into the correct categories.

Hypotheses

Hypothesis 1: CNNs for Image Feature Extraction

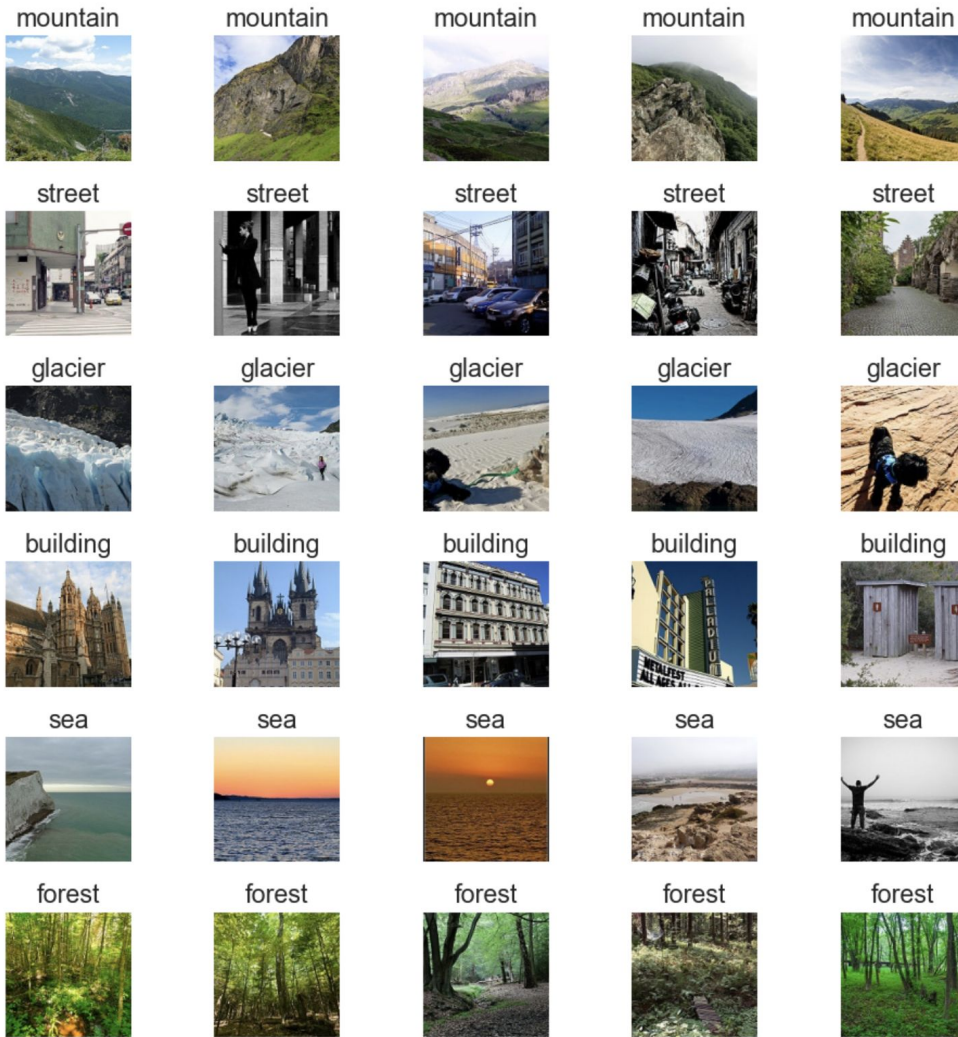
Convolutional neural networks (**CNNs**) **will be effective in extracting relevant features** from images for multiclass image classification. CNNs have proven to be highly successful in image-related tasks due to their ability to capture spatial hierarchies of features.

Hypothesis 2: Model Generalization

The trained model will **generalize well to unseen images** from the same classes. By leveraging techniques such as regularization and data augmentation during training, the model will learn meaningful and robust features that can accurately classify images.

2. Data Profile

Data: Classes of Natural Scene Images



Description:

- This is image data of Natural Scenes around the world.
- In this project, the data used was 7k images of size 150x150 pixels.
- Data distributed under 6 category labels:
 - 'mountain', 'street', 'glacier', 'building', 'sea', 'forest'

Source:

- Initially published on [the Datahack Analytics Vidhya Website](#) by Intel to host a Image classification Challenge.

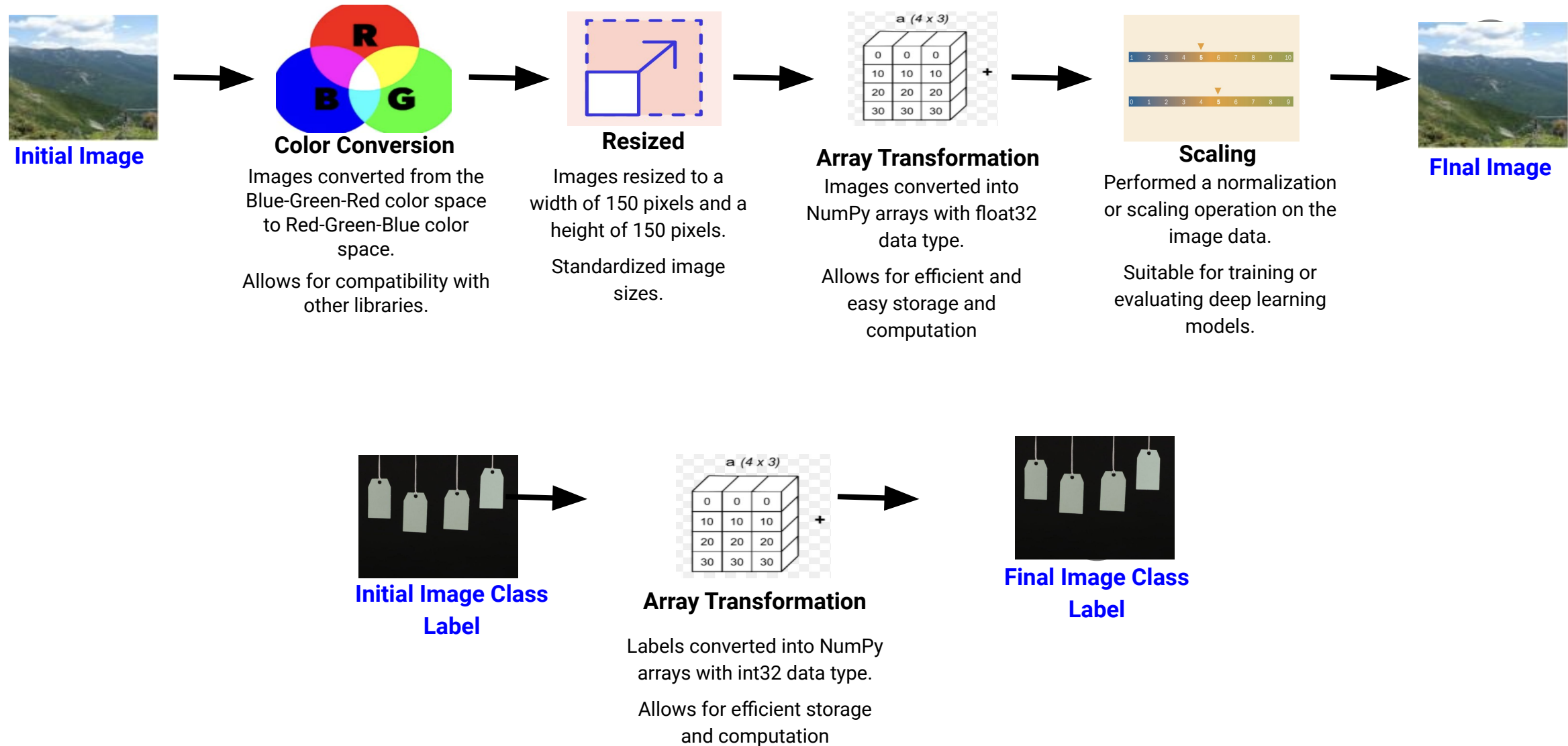
3. Exploratory Data Analysis

Balanced Class Distribution



4. Feature Engineering & Transformations

Feature Engineering and Transformations



5. Modeling

Model Comparisons and Selecting Inception-ResNet-V2

Model	Layers	Results
Simple CNN	Conv2D layer, MaxPooling2D, Flatten, Dense (256, 'Relu'), Dropout (0.5), Dense (num_class, 'softmax')	Train Accuracy: 0.7160091400146484 Test Accuracy: 0.6462638974189758 Underfitting
VGG16	VGG16, Flatten, Dense (256, 'Relu'), Dropout (0.5), Dense (num_class, 'softmax')	Train Accuracy: 0.6886208057403564 Test Accuracy: 0.6732909083366394 Underfitting
Inception ResNet V2	InceptionResNetV2, BatchNormalization, GlobalAveragePooling2D, Dense (256, 'Relu'), Dropout (0.5), Dense (num_class, 'softmax')	Train Accuracy: 0.9468535780906677 Test Accuracy: 0.9141494631767273 Best Model
ResNet50	ResNet50, Flatten, Dense (256, 'Relu'), Dropout (0.5), Dense (num_class, 'softmax')	Train Accuracy: 0.579230546951294 Test Accuracy: 0.5588235259056091 Underfitting

Hyperparameter Tuning Inception-ResNet-V2 with Random Search

dense_units:

- Determines the number of units/neurons in the dense (fully connected) layer of the model.
- It is an integer value chosen from a range defined by min_value=128, max_value=512, and step=64.

dropout_rate:

- Controls the dropout rate, which is the fraction of input units to drop during training.
- It is a float value chosen from a range defined by min_value=0.2, max_value=0.5, and step=0.1.

learning_rate:

- Represents the learning rate of the optimizer used for training the model.
- It is a choice value selected from the options [0.001, 0.01, 0.1].

Best Hyperparameters:

- **dense_units:** 128
- **dropout_rate:** 0.30000000000000004
- **learning_rate:** 0.001

Final Model: Regularization and Parameters for Better Results

Regularization:

- Weight Regularization (L2 Regularization):
 - Set the `base_model.trainable` attribute to `False`, which freezes the weights of the pre-trained InceptionResNetV2 model. Prevent the weight from being updated during training.
- Batch Normalization:
 - Used Batch normalization to normalize the activations of the previous layer, stabilizing the training process. It reduces the effect of internal covariate shift.
- Dropout Regularization:
 - Applied Dropout regularization to the dense layer in the model. Dropout randomly sets a fraction of the input units to 0 during training, which helps in preventing overfitting, which helps in preventing overfitting.

Epochs:

- Specifies the number of times the entire training dataset will be passed through the model during training.
- Set it to 10 since, at 10, it was providing optimal results.

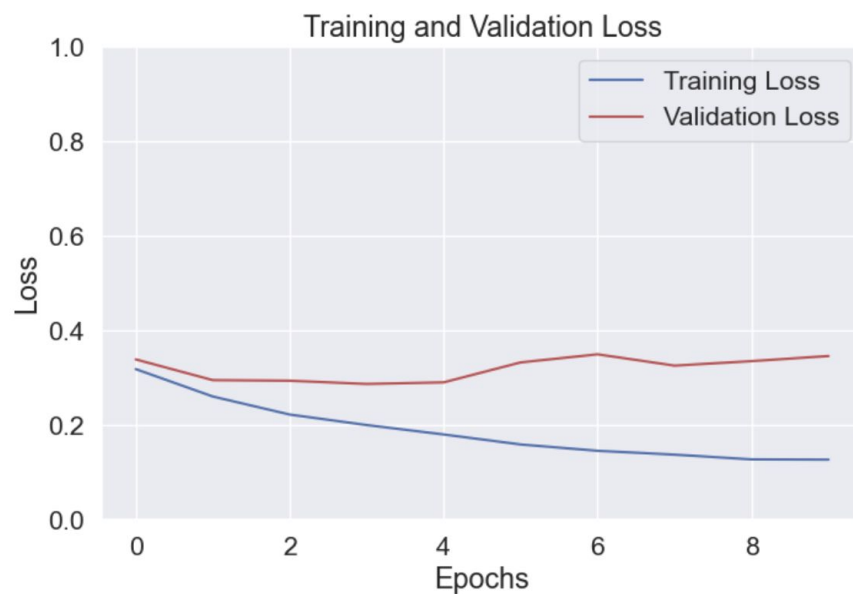
Final Model Results: High Accuracy

Train Accuracy: 0.956

Test Accuracy: 0.903

No Overfitting or Underfitting

High Accuracy and Low Loss. Similar Values for Train and Test Sets.



6. Conclusions & Recommendations

Learnings from the Methodology



Key Findings

- The model with the Inception ResNet V2 base was the best model based on accuracy and loss, making it the best model for the multi-class image classification task.
- Image Feature Transformations of the data are crucial for model building, enhancing the model's ability to capture relevant patterns of the images.
- Transfer learning enhances image classification by leveraging pre-trained models' knowledge. It can reduce training time, improve performance, and facilitate effective feature extraction for a specific domain, in my case multi-class image classification.



Limitations

- Resource Requirements and Computational Speed
 - Loading images, as well as training and testing image classification models demanded significant computational resources. With limited computational resources, the models and images took significant time to run and load, respectively.
- Domain-Specific Generalization
 - For transfer learning, specific pre-trained models are needed for a particular domain. For example, models trained on medical images might struggle to classify natural scenes accurately, requiring specific domain adaptation or additional training on domain-specific data.

Future Work and Applications



Future Work

- Experiment with more model advancement tools:
 - Incorporate Data Augmentation
 - Incorporate other fine-tuning methods
- Exploration of Advanced Models:
 - Investigate the potential of more advanced deep learning architectures. For example, architectures like EfficientNet, ResNeXt, or DenseNet.
- Evaluation on External Datasets:
 - Validate the model's performance and generalization capabilities on external datasets related to landscape and urban image classification.



Applications

- Urban planning: Identify buildings and streets from aerial or satellite images to analyze urban infrastructure.
- Environmental monitoring: Classifying images of forests, mountains, and glacier to aid in monitoring ecosystem health.
- Tourism and travel: Categorizing images of seas, beaches, mountains, and buildings can enhance travel experiences by providing accurate and visually appealing recommendations for tourists.
- Remote sensing: Interpreting satellite imagery and enabling analysis of large-scale landscapes can assist in disaster management.

THANK YOU

For code reference: <https://github.com/AnkitaNambiar/deepLearningInImageClassification-ClassifyingNaturalEnvironments>