Imagedata Augmentation and Image Classification

Rohit Antony,

Section 1
IIIT-Bangalore

Period of Internship: 21ST January 2026 – 17TH February 2026

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This project focuses on image preprocessing and augmentation techniques for image classification using Python-based computer vision and deep learning libraries. The initial phase involved performing fundamental image processing operations such as grayscale conversion, resizing, shifting, and rotation using OpenCV. Following this, image augmentation techniques were implemented using PyTorch and Torchvision to enhance dataset variability and improve model generalization. A structured Cat vs Dog dataset was used to build an image classification pipeline using the ImageFolder class. Data augmentation techniques such as resizing and horizontal flipping were applied to the training dataset. A DataLoader was created to efficiently load images in batches for model training. The project demonstrates the complete workflow of preparing image data for deep learning applications, from raw image manipulation to batch-wise data loading for classification tasks.

2.Introduction

Computer vision is a key domain of artificial intelligence that enables machines to interpret and process visual data. Image classification is one of the most fundamental tasks in computer vision, where an image is assigned to a predefined category.

Deep learning models require large and diverse datasets to perform effectively. Image augmentation techniques artificially increase dataset size and variability by applying transformations such as flipping and resizing. This helps improve model robustness and prevent overfitting.

During the Spring Internship Program (60 hours), the following topics were covered:

Python Foundations

- Introduction and Internship Orientation
- Python Basics (Data Types, Variables, Lists, Loops)
- Data Structures
- Classes and Functions
- Object-Oriented Programming (OOPS)
- NumPy
- Pandas

Machine Learning & Advanced Topics

- Machine Learning Overview
- Regression
- Classification
- LLM Fundamentals
- Communication Skills

The purpose of this project was to apply the knowledge gained during training to build a complete image data preparation pipeline for classification tasks.

3.Project Objective

- To perform basic image preprocessing using OpenCV.
- To implement geometric transformations such as shifting, resizing, and rotation.
- To apply image augmentation techniques using Torchvision transforms.
- To create a structured dataset using the ImageFolder class.
- To implement a DataLoader for efficient batch-wise data handling.

4.Methodology

4.1 Data Collection

Two types of image data were used:

- A sample moon image for demonstrating basic image processing operations.
- A Cat vs Dog dataset organized into training and testing directories.

Dataset structure:

Cat_Dog_data/
    train/
        cat/
        dog/
    test/
        cat/
        dog/

Images were automatically labelled based on folder names using the ImageFolder class.

4.2 Image Preprocessing Using OpenCV

The following operations were performed:

- Image loading using `cv2.imread()`
- Grayscale conversion using `cv2.cvtColor()`
- Image saving using `cv2.imwrite()`
- Affine transformation (shifting)
- Image resizing using `cv2.resize()`
- Image rotation using `cv2.getRotationMatrix2D()` and `cv2.warpAffine()`

These preprocessing steps are fundamental in computer vision and help standardize images before deep learning training.

4.3 Image Augmentation Using Torchvision

To increase dataset variability, the following transformation pipeline was used:

- Resize to (255, 255)
- Random Horizontal Flip
- Conversion to Tensor

The augmentation was applied only to the training dataset to improve generalization.

4.4 Dataset Creation Using ImageFolder

The ImageFolder class was used to automatically:

- Load images from directories
- Assign numeric class labels
- Map class names
- Prepare data for training

This simplifies dataset handling for supervised classification tasks.

4.5 DataLoader Implementation

A DataLoader was created with:

- Batch size = 64
- Shuffle = True

One batch was retrieved using iteration over the DataLoader.

Observed Output Shapes:

- Images Tensor Shape: (64, 3, 255, 255)
- Labels Tensor Shape: (64,)

This confirms:

- 64 images per batch
- 3 color channels (RGB)
- 255 × 255 resolution
- 64 corresponding labels

- The successful output verifies correct implementation of the data pipeline.

4.6 Tools and Technologies Used

- Python
- NumPy
- OpenCV
- PyTorch
- Torchvision
- Jupyter Notebook

5. Data Analysis and Results

The preprocessing operations successfully transformed images into grayscale, resized formats, and rotated versions. Affine transformations were correctly applied and verified using image shape inspection.

The ImageFolder class correctly detected two classes: cat and dog. The DataLoader returned correctly structured batches of images and labels.

The final data pipeline achieved the following workflow:

Raw Image
→ Preprocessing
→ Augmentation
→ Dataset Structuring
→ DataLoader
→ Batch Output

The implementation successfully demonstrated how image data is prepared before feeding into deep learning models.

6. Conclusion

The project successfully demonstrated image preprocessing and augmentation techniques for image classification tasks. OpenCV was used to implement geometric transformations, strengthening understanding of basic computer vision concepts. Torchvision transforms enhanced dataset diversity and prepared images for model input.

The ImageFolder and DataLoader modules enabled efficient dataset structuring and batch processing. This project provided a complete understanding of image data preparation for deep learning applications.