

Data Collection and Preprocessing Phase

| | |
|---------------|---|
| Date | 6 Feb 2026 |
| Student Name | Sushant Dattatray Kadam |
| Project Title | greenclassify: deep learning-based approach for vegetable image |

Preprocessing Template

The images will be preprocessed by resizing and batch normalizing. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

| Section | Description |
|---------------------|---|
| Data Overview | The Greenclassify project utilizes a dataset of vegetable images sourced primarily from a Kaggle dataset. The original dataset contains approximately 21,000 images of 15 different vegetable types, divided into training, validation, and testing sets (approximately 70%, 15%, and 15% respectively). Images are in JPG format but were initially of varying sizes. The goal of preprocessing is to create a standardized, high-quality dataset suitable for training a deep learning model. (See Data Quality Report for details on initial data quality issues and resolutions). |
| Resizing | Resize images to a specified target size. |
| Batch Normalization | Apply batch normalization to the input of each layer in the neural network. |
| | |

| Data Preprocessing Code Screenshots | |
|-------------------------------------|--|
| Loading Data | <pre>train_ds = tf.keras.utils.image_dataset_from_directory(train_path, label_mode='categorical', image_size=(256, 256), batch_size=batch_size, shuffle=True, seed=123)</pre> |
| Resizing | <pre>def resize_images(dataset): return dataset.map(lambda x, y: (tf.image.resize(x, (150,150)), y)) # Resize images in train_ds, validation_ds, and test_ds train_ds = resize_images(train_ds) validation_ds = resize_images(validation_ds) test_ds = resize_images(test_ds)</pre> |
| Batch Normalization | <pre># First Conv Block model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=input_shape)) model.add(BatchNormalization()) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.3))</pre> |