greatlearning

# Project Description – Image classification using CNNs in Keras

## Data Description:

You are provided with a training set and a test set of images of plant seedlings at various stages of grown. Each image has a filename that is its unique id. The dataset comprises 12 plant species. The goal of the competition is to create a classifier capable of determining a plant's species from a photo.

### Dataset:

The project is from a dataset from Kaggle.

Link to the Kaggle project site: https://www.kaggle.com/c/plant-seedlings-classification/data

Note: For project purposes, download only train images folder from Kaggle. (See note at the end of problem statement.)

The dataset has to be downloaded from the above Kaggle website.

### Context:

Can you differentiate a weed from a crop seedling?

The ability to do so effectively can mean better crop yields and better stewardship of the environment.

The Aarhus University Signal Processing group, in collaboration with University of Southern Denmark, has

recently released a dataset containing images of unique plants belonging to 12 species at several growth stages

## Objective:

To implement the techniques learnt as a part of the course.

## Learning Outcomes:

- Pre-processing of image data.
- Visualization of images.
- Building CNN.
- Evaluate the Model.

## Steps and tasks:

1. Import the libraries, load dataset, print shape of data, visualize the images in dataset. (5 Marks)
2. Data Pre-processing: (15 Marks)
    a. Normalization.
    b. Gaussian Blurring.
    c. Visualize data after pre-processing.
3. Make data compatible: (10 Marks)
    a. Split the dataset into training, testing, and validation set.
    (Hint: First split train images and train labels into training and testing set with test_size = 0.3. Then further split test data into test and validation set with test_size = 0.5)
    [Read the note at the end of the problem statement for the reason behind using the train images for using for training and testing set.]
    b. Reshape data into shapes compatible with Keras models.
    c. Convert labels from digits to one hot vectors.
    d. Print the label for y_train[0].
4. Building CNN: (15 Marks)
    a. Define layers.
    b. Set optimizer and loss function. (Use Adam optimizer and categorical crossentropy.)
5. Fit and evaluate model and print confusion matrix. (10 Marks)
6. Visualize predictions for x_test[2], x_test[3], x_test[33], x_test[36], x_test[59]. (5 Marks)

**Note:**

- **Download the train images from the Kaggle dataset, don't download the test images.**
- **As the test dataset is not labeled, so you won't be able to calculate the accuracy score. So use train images and train labels only to split further into training and testing set during your model building.**

**Happy Learning!**