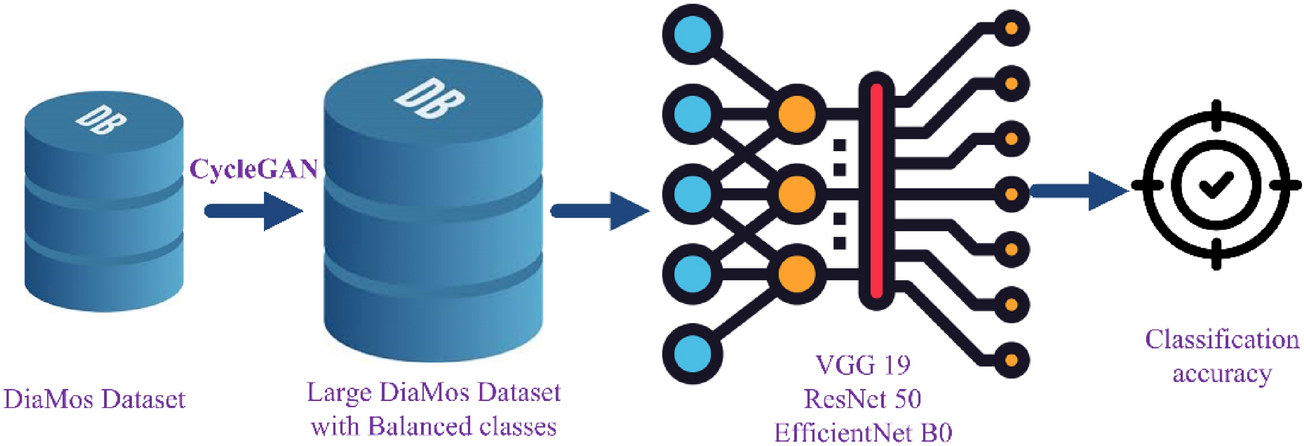
Plan for Data Augmentation

The training process of deep learning model requires large number of records of paired examples. However, in most cases, preparing large datasets are not valid, as with the employed DiaMOS dataset. The **CycleGAN** is a method that involves an autonomous training of image-to-image translation models without paired examples using the GAN architecture, where these models are trained in an unsupervised approach, using a collection of images from the source and target domain that do not require to be related in any way.

In general, the CycleGAN architecture consists of two models: the generator model and a discriminator model. The generator receives a point from a latent space as an input and produces new plausible images from the domain, whereas the discriminator takes an image as an input and predicts whether the taken image is a real one (from the DiaMOS dataset) or fake (from the generated dataset).

Our approach involves generating new labelled images through training the CycleGAN, for two main reasons: balancing the classes in the DiaMOS dataset, and increase the size of the DiaMOS dataset in order to obtain better classification accuracy, where the new generated images will be added to the original image dataset in order to enhance the initial dataset, through incorporating more images. Next, the designed CNN model is employed to train using the new generated dataset.

DiaMOS dataset contains a small number of records, in addition, DiaMOS includes imbalanced data classes, and hence this leads to poor plant disease classification performance. Therefore, one of our main goals was to increase the size of the DiaMOS dataset and balance the records in each class.



Source: [An improved pear disease classification approach using cycle generative adversarial network](https://www.nature.com/articles/s41598-024-57143-6)

Preprocessing includes:

1. **Image Normalization**:

* Normalizing image pixel values helps in faster convergence of the neural network training process. It ensures that the input data has a consistent scale and prevents issues related to varying magnitudes of pixel values.
* **Technique**: Common normalization techniques involve scaling pixel values to a specific range, such as [0, 1] or [-1, 1]. For example, normalizing to [-1, 1] can be done using the formula: normalized\_pixel=(pixel−127.5)/127.5\text{normalized\\_pixel} = (\text{pixel} - 127.5) / 127.5normalized\_pixel=(pixel−127.5)/127.5.

1. **Resizing and Cropping**:

* Standardizing the size of images is crucial because neural networks typically require fixed-size inputs. Resizing ensures uniformity, while random cropping can increase the model's robustness by introducing slight variations in the input data.
* **Technique**: Resize images to the target dimensions (e.g., 256x256 pixels). Random cropping involves selecting random regions of the resized images to use as input during training.

1. **Histogram Equalization**:

* Histogram equalization improves the contrast of an image by spreading out the most frequent intensity values. This technique can be particularly useful in images with poor lighting or low contrast.
* **Technique**: Convert the image to grayscale, compute the histogram, and redistribute pixel values to achieve a uniform histogram.

1. **Noise Reduction**:

* **Theory**: Reducing noise in images can help the model focus on the relevant features rather than the noise. This is especially important in real-world images where noise can be prevalent.
* **Technique**: Apply denoising filters like Gaussian blur, median filtering, or more advanced methods like Non-Local Means Denoising.

1. **Color Space Conversion**:

* **Theory**: Different color spaces can highlight different features. For instance, converting to the Lab color space separates lightness (L) from color information (a and b), which can be useful for certain image processing tasks.
* **Technique**: Convert images from RGB to another color space (e.g., YUV, HSV, Lab) using appropriate transformation functions.