**Tool and Techniques Name**

**1) Python Libraries**

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Sklearn.metrics
* Sklearn.preprocessing
* Tqdm
* Tensorflow
* Shutil
* Cv2

**2) Dataset Link**

* [**https://www.kaggle.com/code/victorbnnt/data-augmentation-flowers-classification/data**](https://www.kaggle.com/code/victorbnnt/data-augmentation-flowers-classification/data)

**3) Preprocessing Techniques**

* Image resizing
* Changing the colour of background
* Changing its colour mode to RGB
* Applying Normalization

**4) Data Augmentation**

* No Augmentation technique is applied

**5) Feature Extraction/ Selection Technique**

* No feature selection or reduction technique is applied

**6) Classification Technique**

* We have 5 classes in our base paper which are different categories of flower.
* i.e., Daisy, Dandelion, Rose, Sunflower and Tulip

**7) Data Split Ratio**

* We have split the data into training, testing and Validation in the ratio of 68:20:12.
* So, there is 3019 images for training, 865 images for testing and 433 images for validation.

**8) Base Model**

* EfficientNetB4
* Xception
* InceptionV3

**Screenshots of base and proposed results**

**1) Dataset visualization screenshots**



A group of white flowers

Description automatically generated with medium confidence

A group of white flowers

Description automatically generated with medium confidence

A group of white flowers

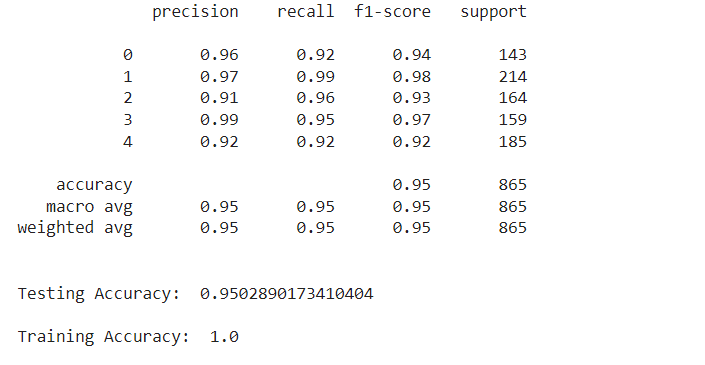
Description automatically generated



**5) Base model results---like confusion matrix, ROC curve, classification**

**Report, and Heat Map and comparison Curve etc.**

1. **EfficientNetB4**



A picture containing diagram

Description automatically generated

Chart, line chart

Description automatically generated

1. **Xception**

Table

Description automatically generated

A picture containing diagram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

1. **InceptionV3**

Table

Description automatically generated

Chart, treemap chart

Description automatically generated

Chart, line chart

Description automatically generated

**Propose Result table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| EfficientNetB4 | **95%** | **0.95** | **0.95** | **0.95** |
| Xception | **93%** | **0.93** | **0.93** | **0.93** |
| InceptionV3 | **92%** | **0.92** | **0.92** | **0.92** |

**Base and proposed results comparison table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| **Base** | **91%** | **0.91** | **0.91** | **0.91** |
| **Propose** | **95%** | **0.95** | **0.95** | **0.95** |