Flower recognition using convolutional neural networks

Google Colab Notebook Link :
https://colab.research.google.com/drive/1bYDh4OHhzdQ2ywwQT-ujdM7XI9R8ozcT?usp=sharing

Google Drive Link of the pre-processed dataset and the trained model :
https://drive.google.com/drive/folders/1UGSKOiDd-WwfKmtVmkX9B6THEsBniUfj?usp=sharing

The approach followed for the given problem statement :

- **Dataset preprocessing:** The dataset was divided into 5 categories of flowers. Therefore the first step was to combine all the images and labels together so that training and testing become easier.

- **Model architecture:**
  - The input layer consists of the neurons which decide the number of features used to make predictions. Here I have given it the dimensions (224,224,3) keeping in mind the dimensions of the input image.
  - The hidden layers used in the architecture were four as it would be sufficient to train such models. I progressively kept on increasing the number of neurons in the next hidden layer and kept track of the loss and accuracy performance of the model so as the model doesn't overfit. I avoided choosing a small number of layers/neurons, as it might result in the model not learning the underlying patterns in the dataset. I tried some combinations of weights and biases and found this to be the perfect fit for my model.
  - As ours was a classification problem, the output layer consisted of five neurons as we had five categories of flowers in our dataset.
  -
- **Model parameters :**
  - **Loss function:** In the case of our model, we have used cross_entropy(sparse_categorical_crossentropy) as it was the best fit for classification problems.
  - **Epochs:** According to me, 20 epochs were enough for the model to reach maximum accuracy and minimum loss. Also, we could trace, if the model's progress is improving when we have more epochs.
  - **Batch Size:** Batch size was taken to be 100 for training the model.
  - **Feature scaling:** I had ensured that all the features are normalized so that the training and optimization process is fast.

    - **Activation functions:** For the hidden layers ‘relu’ activation function was used and for the output layer ‘softmax’ was used, as it was a multi-class classification and also to ensure the output probabilities added up to 1.

- **Evaluation metrics :**
    - **Confusion Matrix:** The confusion matrix is used to differentiate the true positives, true negatives, false positives and false negatives for all five categories in the model.
    - **Classification report:** The classification report tells us about the precision, recall and f1 score for the model.