Assignment #4: Flower Classification

Name 1: Omar Youssef Abdelsattar

ID 1: 43

Name 2: Islam Mostafa Abdelaziz

ID 2: 13

Name 3: Mohamed Saad

ID 3: 51

Name 4: Kamal Rashid

ID 4: 47

Problem statement:

You are given the petals to metals dataset, which is a dataset for flower classification.

We're classifying 104 types of flowers based on their images drawn from five different public datasets.

Some classes are very narrow, containing only a particular sub-type of flower (e. g. pink primroses), while other classes contain many sub-types (e. g. wild roses).
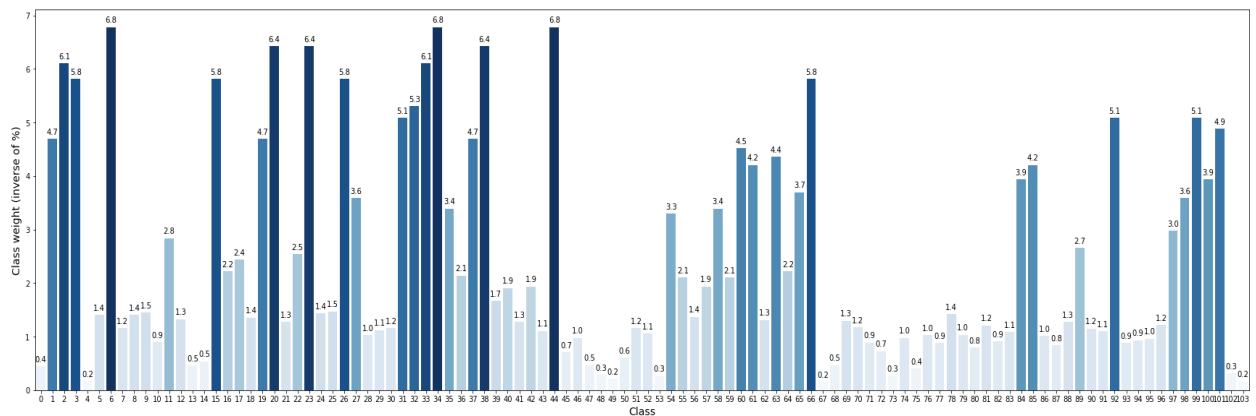
1. Download the Dataset and Understand the Format.
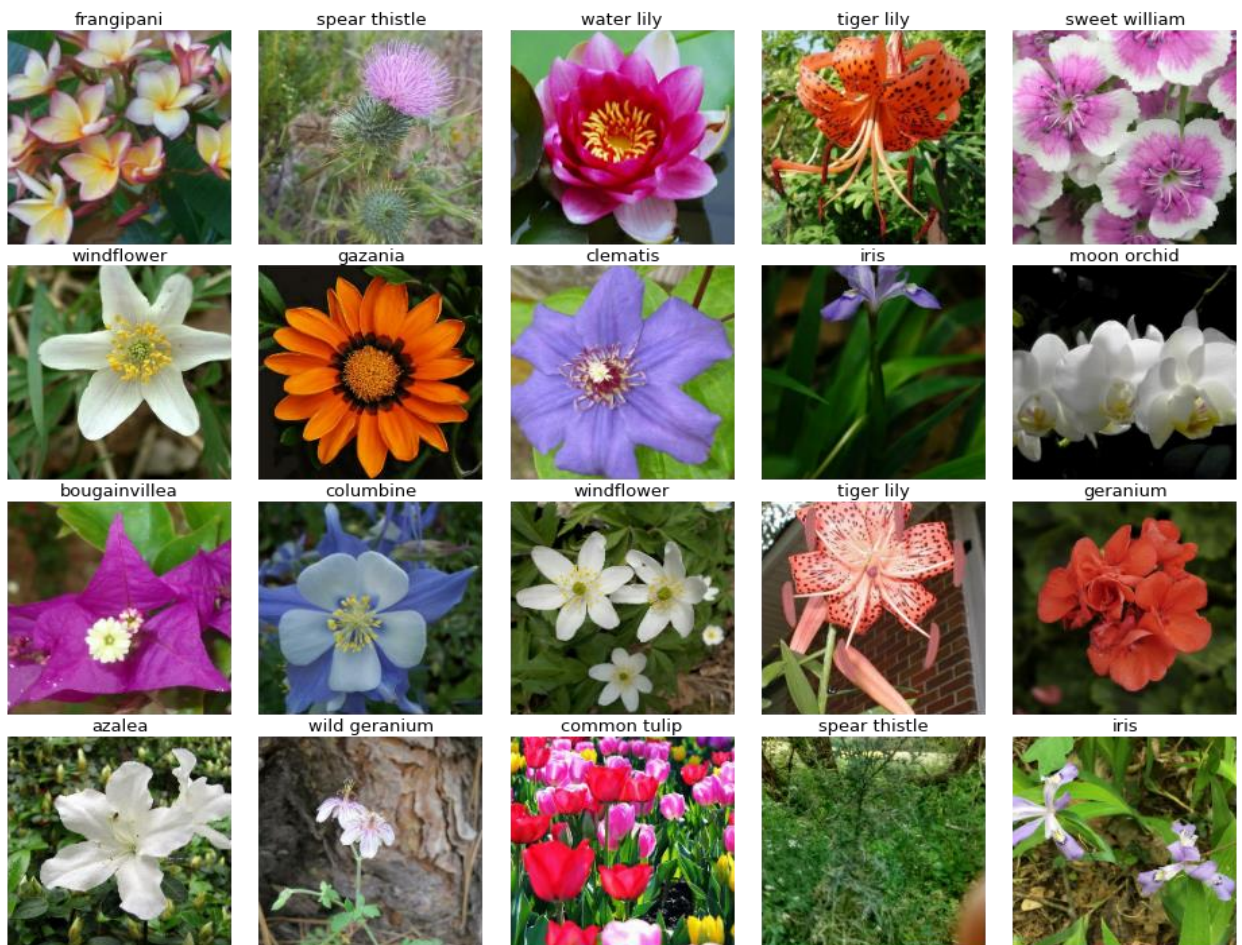
2. Build Model.

3. Big Picture.

Observations:

- We downloaded the data from Kaggle in colab and understanded the format as it has images of different sizes that consists of train , test , validation.
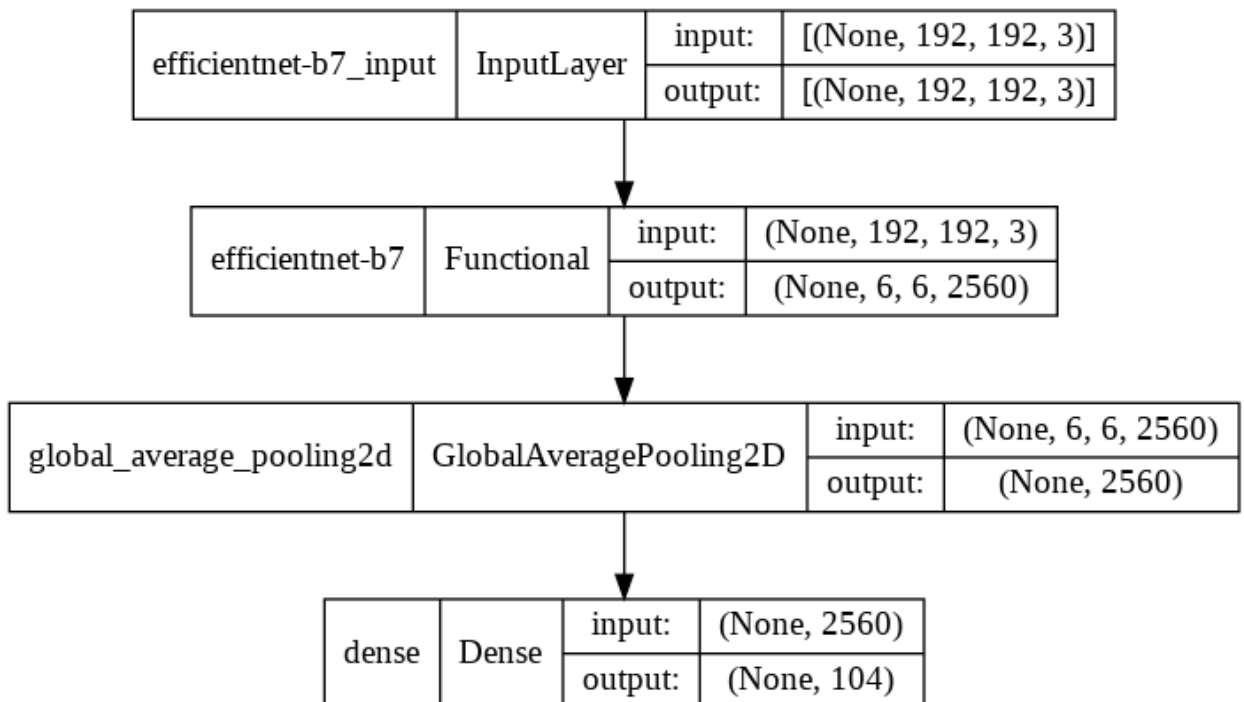- We got the weights of classes.

- Then we loads the images and visualize some of different classes.

- Then we split the training data into 90% training and the rest for validation.

- We build our simple CNN model from scratch and famous architectures as ENet.

```
Layer (type)                    Output Shape              Param #
=================================================================
efficientnet-b7 (Functional  (None, 6, 6, 2560)          64097680
)

global_average_pooling2d (G  (None, 2560)                0
lobalAveragePooling2D)

dense (Dense)                (None, 104)                 266344

=================================================================
Total params: 64,364,024
Trainable params: 64,053,304
Non-trainable params: 310,720
```

efficientnet-b7_input	InputLayer	input:	[(None, 192, 192, 3)]
		output:	[(None, 192, 192, 3)]

efficientnet-b7	Functional	input:	(None, 192, 192, 3)
		output:	(None, 6, 6, 2560)

global_average_pooling2d	GlobalAveragePooling2D	input:	(None, 6, 6, 2560)
		output:	(None, 2560)

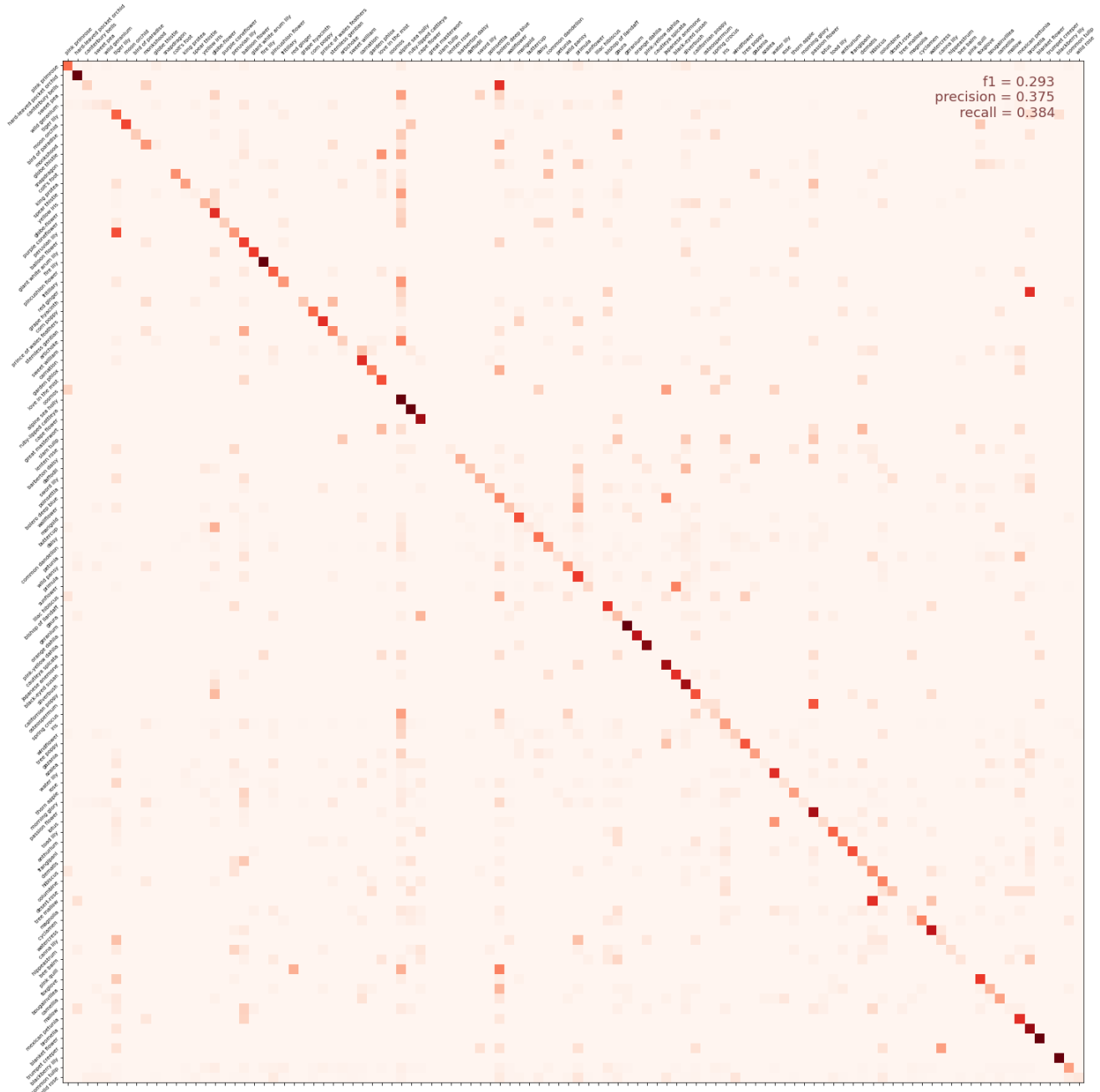| dense	Dense	input:	(None, 2560)
		output:	(None, 104)

- Then we get the learning rate; when we use 1 epoch:
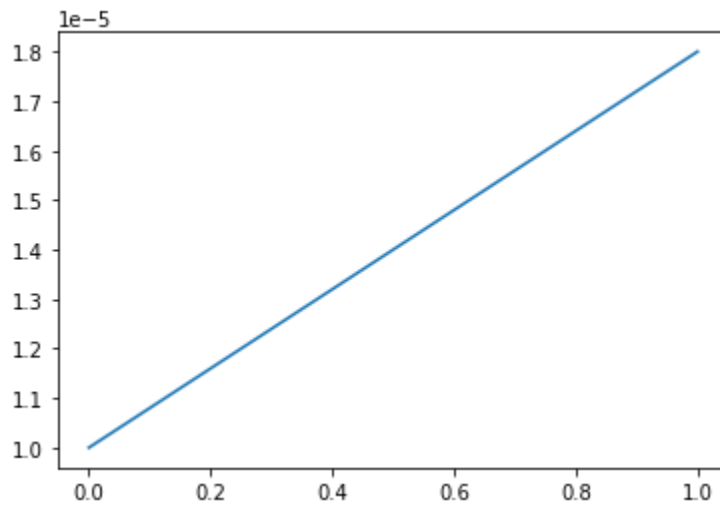
- Then we use the validation data as testing, as we plot the accuracy and loss corresponding to the number of epochs , it doesn't make the plots in the case of 1 epoch:
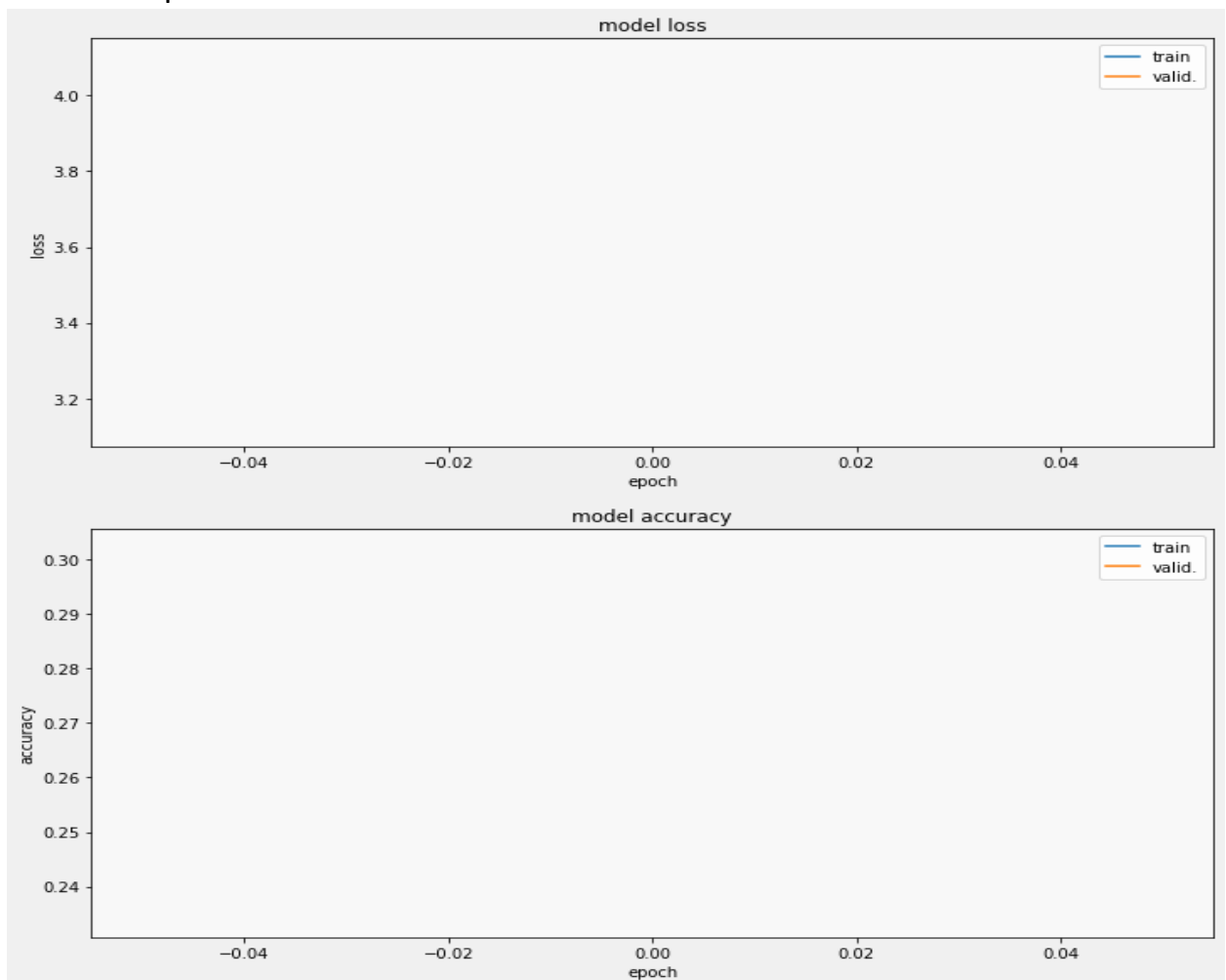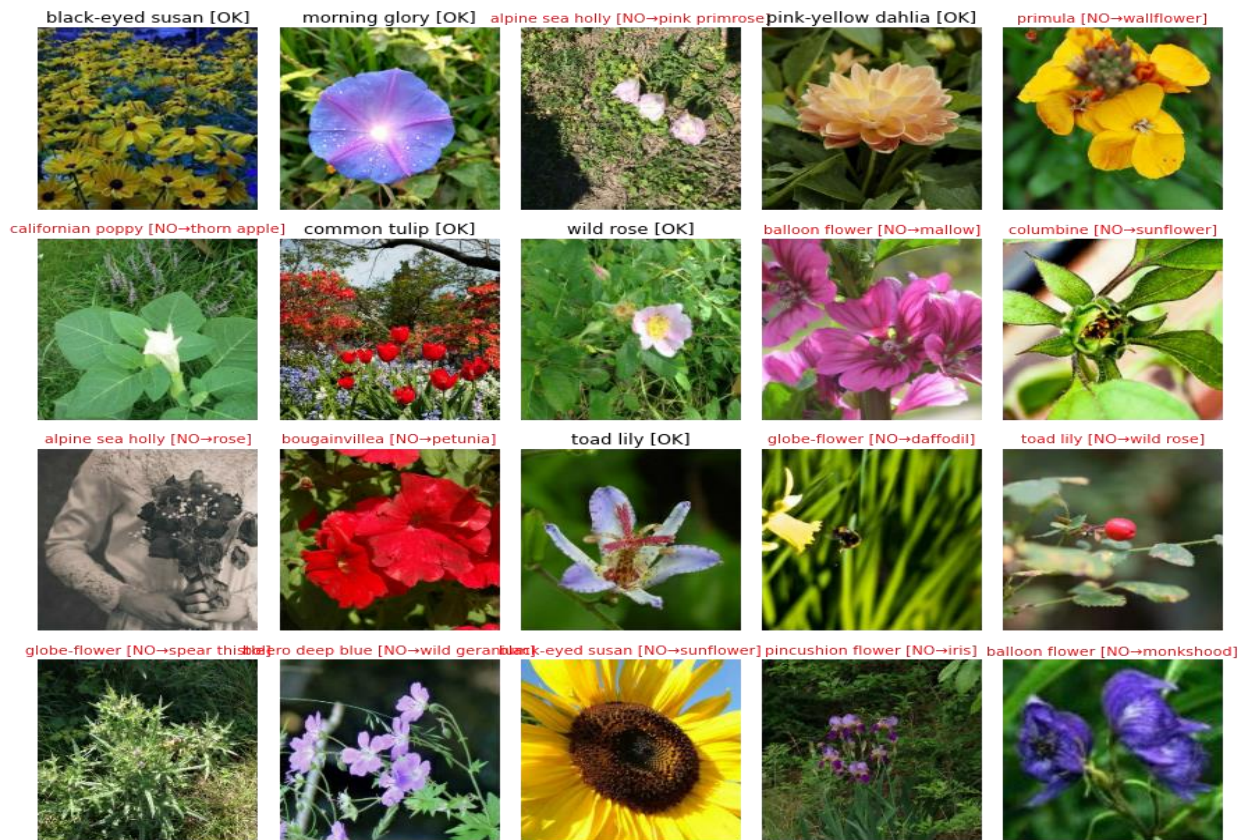
- We made the ensemble model that integrates the models.
- Then we compare between the performance of the learned models by realizing the accuracy as we showed and then the confusion matrix and macro F-score.

f1 = 0.293
precision = 0.375
recall = 0.384

- And we this is the result after classification:

black-eyed susan [OK]   morning glory [OK]   alpine sea holly [NO→pink primrose]   pink-yellow dahlia [OK]   primula [NO→wallflower]

californian poppy [NO→thorn apple]   common tulip [OK]   wild rose [OK]   balloon flower [NO→mallow]   columbine [NO→sunflower]

alpine sea holly [NO→rose]   bougainvillea [NO→petunia]   toad lily [OK]   globe-flower [NO→daffodil]   toad lily [NO→wild rose]

globe-flower [NO→spear thistle]   bolero deep blue [NO→wild geranium]   black-eyed susan [NO→sunflower]   pincushion flower [NO→iris]   balloon flower [NO→monkshood]

- Then we made the submission file:

```
# Look at the first few predictions
!head submission.csv

Predictions...
[ 83  57  67 ... 102  94  44]
using_ensemble_models: False
Generating submission.csv file...
id,label
c37a6f3e9,83
630ba700c,57
22149c3fd,67
07c0d22bb,102
b4663783a,102
4acd9d511,62
6066bf3b3,27
584b3a955,67
94d32655a,93

[ ]
```

The failure cases:

- When we use more than 1 epoch, we can't run in colab as the ram crashes.
- So, we use 1 epoch , but it's can't show us the accuracy.