<Image Classifier Assignment>

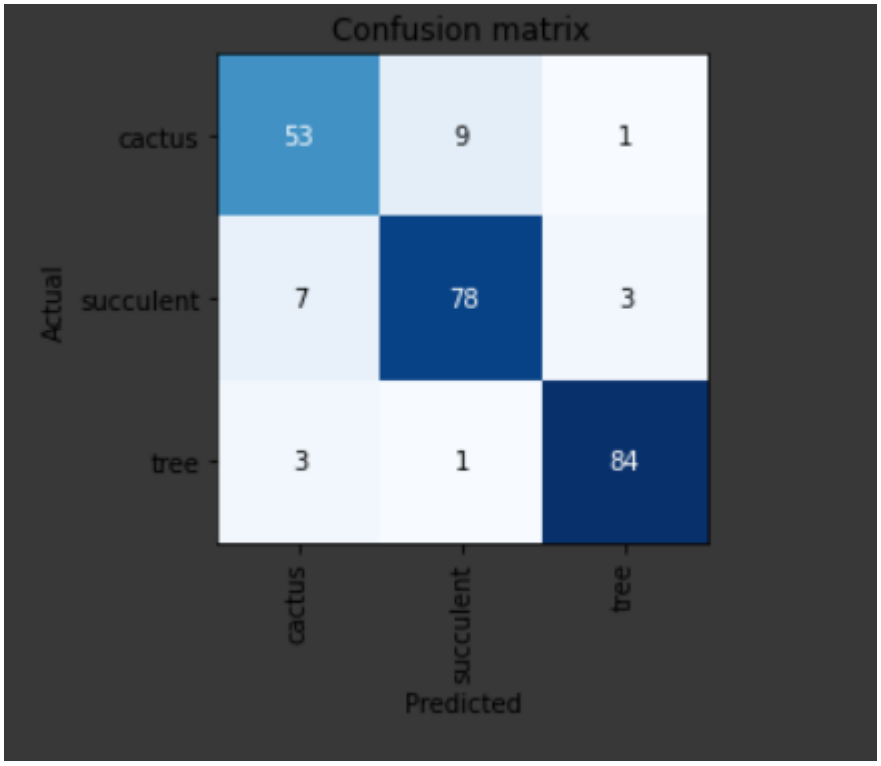
- PeopleSpace Online Internship Program -

Name (ENG) : Lily

Name (KOR) : Hyomin Kim

Category : Nike vs Adidas vs Reebok (Shoes)

1. The confusion matrix for your dataset (2points)



2. A screenshot of the output printed of the 10 images with the highest loss in your dataset (1 point)

3. For each of the 10 images, give an explanation of why this image was included in the top loss plot, and the resulting action you took. Did you remove the image, why? And if not, why did you include it? (2 points)

Image	Prediction	Ground Truth(Label)	Action	Reason
	Succulent	Cactus	Moved to succulent	I think small leaves with no thorn is feature of succulent
	Succulent	Cactus	Moved to succulent	I think small leaves with no thorn is feature of succulent
	Tree	Succulent	Removed	It is not focused on one sort of plants and most part of the image is about house
	Cactus	Succulent	Moved to cactus	I think plant with thorn is feature of cactus

	Cactus	Succulent	Moved to cactus	I think pink-color flower is not feature of succulent but cactus
	Cactus	Succulent	Removed	It is not focused on one sort of plant so it may be hard to get good feature of succulent
	Tree	Succulent	Moved to succulent	It has good features of succulent (It has no trunk so it is not tree and it has no thorn so it is not cactus)
	Succulent	Tree	Removed	It doesn't have good features of tree(No trunk, just leaves)
	Succulent	Cactus	Moved to Cactus	It has good features of cactus(Thorn on leaves)

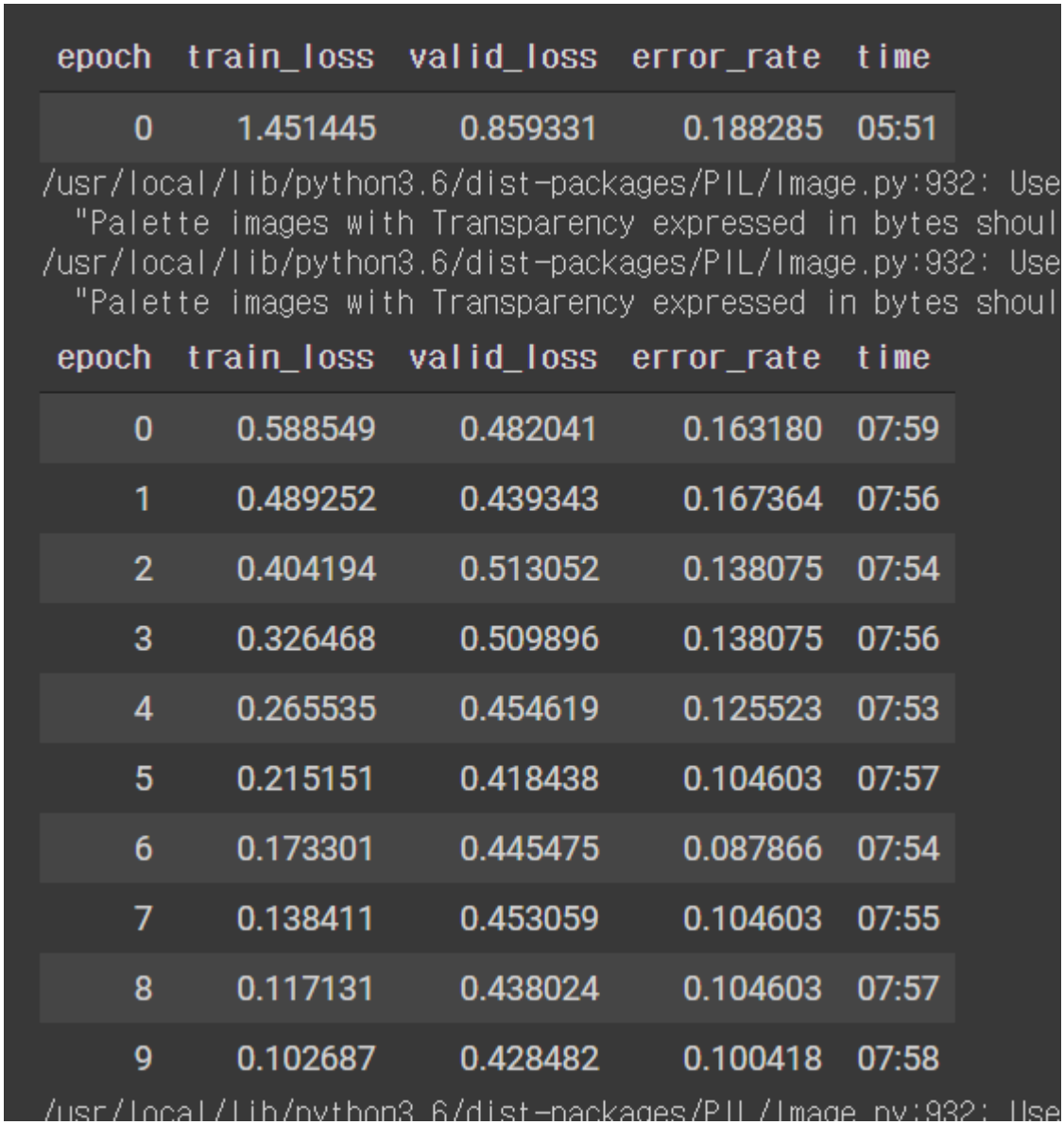
		Succulent	Cactus	Removed	It has both cactus and succulent on images so it's hard to decide what it is.

4. A summary of your approach to acquiring the data. (1 point)

At first I used original Bing API function (search_images_bing). However this function can get up to 150 images and I guessed it is too small for good training. To get more image data I used modified Bing API function (search_images_bing_many) and gathered 449 images for each category(Succulent/Tree/Cactus).

In original code, epoch was set to 5 and I increased it to 10, and I changed model from "Resnet34" to "Resnet50", which has more hidden layer. However "Resnet34" got better result so I used "Resnet34".

Finally, I can get the accuracy with about 90% and the highest accuracy was about 92%..



epoch	train_loss	valid_loss	error_rate	time
0	1.451445	0.859331	0.188285	05:51

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:932: Use
  "Palette images with Transparency expressed in bytes shoul
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:932: Use
  "Palette images with Transparency expressed in bytes shoul
```

epoch	train_loss	valid_loss	error_rate	time
0	0.588549	0.482041	0.163180	07:59
1	0.489252	0.439343	0.167364	07:56
2	0.404194	0.513052	0.138075	07:54
3	0.326468	0.509896	0.138075	07:56
4	0.265535	0.454619	0.125523	07:53
5	0.215151	0.418438	0.104603	07:57
6	0.173301	0.445475	0.087866	07:54
7	0.138411	0.453059	0.104603	07:55
8	0.117131	0.438024	0.104603	07:57
9	0.102687	0.428482	0.100418	07:58

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:932: Use
```