

<Image Classifier Assignment>

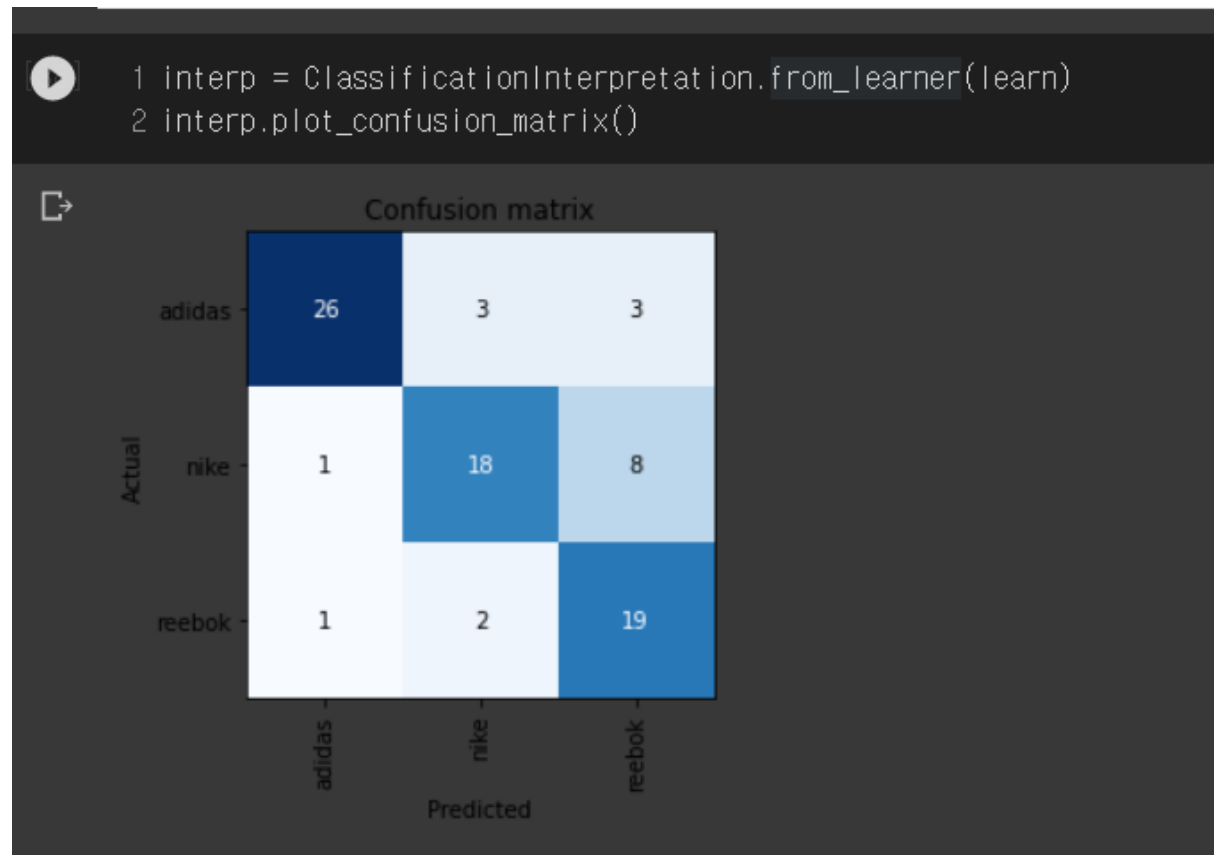
- PeopleSpace Online Internship Program -

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)

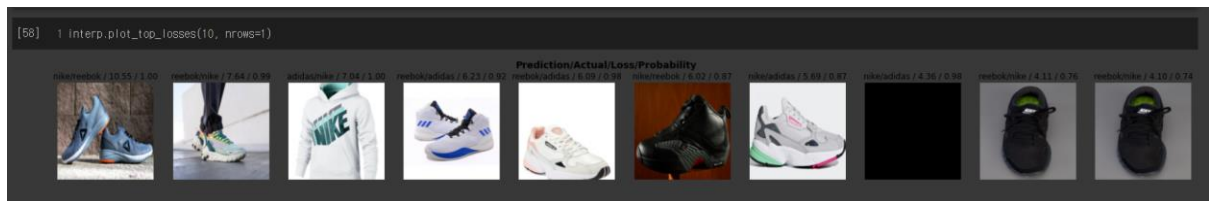


Image	Prediction	Ground Truth(Label)	Action	Reason
	Nike	Reebok	Moved to Reebok	Has good features of Reebok shoes (triangle logo in shoes)
	Reebok	Nike	Moved to Nike	Has good features of Nike(Nike logo on shoes and color combination)
	Adidas	Nike	Removed	It's not image of shoes. It's hoodie and doesn't have any relationship with shoes data.
	Reebok	Adidas	Moved to Adidas	Has good features of adidas shoes(Logo of adidas and design)
	reebok	Adidas	Moved to Adidas	Has good features of adidas shoes(Clean background, logo of shoes, and design)
	Nike	Reebok	Moved to Reebok	Has good features(Design and logo of Reebok)
	Nike	Adidas	Moved to adidas	Has good features(Adidas logo, design, clean background)

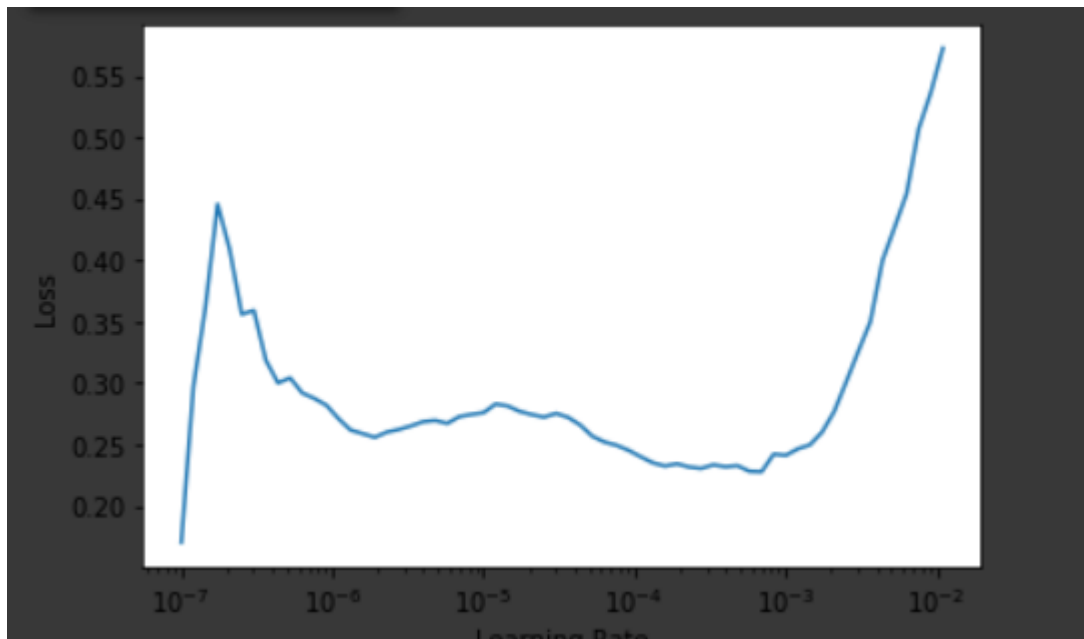
			Nike	Adidas	Removed	Doesn't have any information. Just a black background.
			Reebok	Nike	Moved to Nike	Has good features(Nike logo and design)
			Reebok	Nike	Moved to Nike	Has good features(Nike logo and design)

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 training. To get more image data I used modified Bing API function (search_images_bing_many) and gathered 449 images for each category(nike/adidas/reebok). However accuracy became much lower than before. Therefore I decided to use model which got more higher accuracy but smaller dataset(150 images for each category).

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".

I tried to get optimized learning rate by drawing graph of loss-learning rate graph.



I tried to train again with learning rate about 1e-3. However, after training with modified learning rate, I got lower accuracy. Therefore, I used model which I used before.

Finally, I can get the accuracy with about 81%.

epoch	train_loss	valid_loss	error_rate	time
0	1.842211	2.671608	0.580247	02:01
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:932: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images				
epoch	train_loss	valid_loss	error_rate	time
0	1.153562	1.902706	0.469136	02:50
1	1.025188	1.229774	0.308642	02:32
2	0.891434	0.989559	0.333333	02:42
3	0.779697	0.977535	0.308642	02:45
4	0.691033	0.898852	0.234568	02:57
5	0.621642	0.855881	0.271605	02:42
6	0.561179	0.842546	0.197531	02:58
7	0.518351	0.832619	0.185185	02:30
8	0.476852	0.798015	0.197531	02:32
9	0.440538	0.815685	0.197531	02:53