

캡스톤 디자인 1

딥러닝을 통한 얼굴 합성



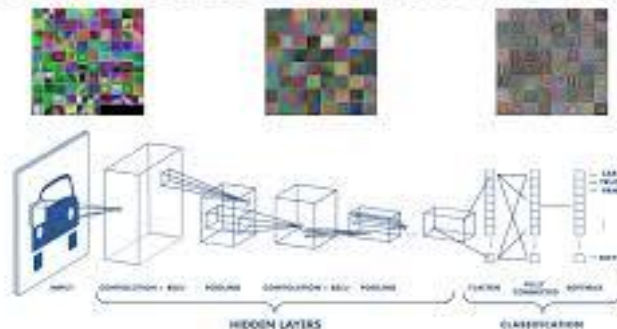
강민석
김성준

동물분류 - CNN 이용

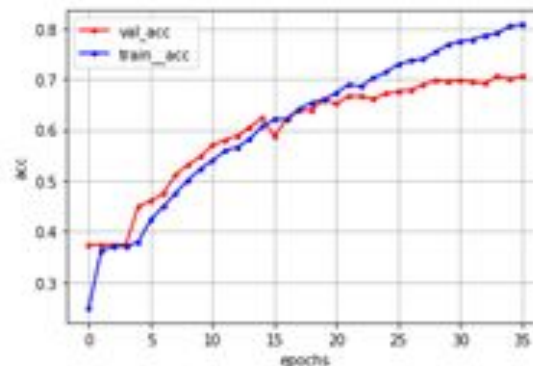
Layer (type)	Output Shape	Param #
conv2d_22 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_22 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout_30 (Dropout)	(None, 32, 32, 32)	0
conv2d_23 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_23 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_31 (Dropout)	(None, 16, 16, 64)	0
flatten_9 (Flatten)	(None, 16384)	0
dense_17 (Dense)	(None, 256)	4194560
dropout_32 (Dropout)	(None, 256)	0
dense_18 (Dense)	(None, 14)	3598

Total params: 4,217,550
Trainable params: 4,217,550
Non-trainable params: 0

Convolutional Neural Network



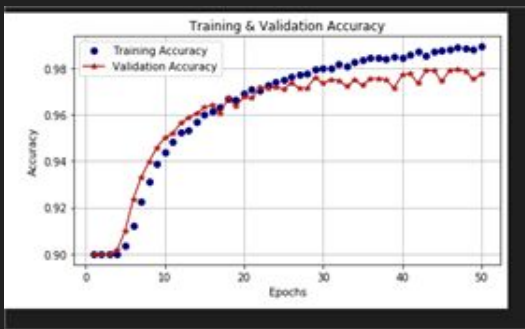
3313/3313 [=====] - 1s 231us/step
정확도 : 0.6837



동물분류 - pretrained network(VGG16)

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
sequential_1 (Sequential)	(None, 10)	6425354

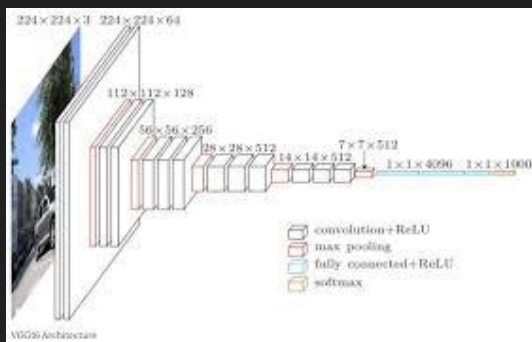
Total params: 21,140,042
Trainable params: 21,140,042
Non-trainable params: 0



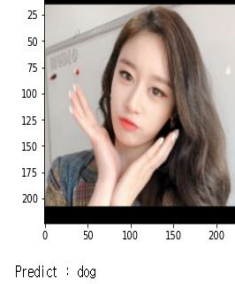
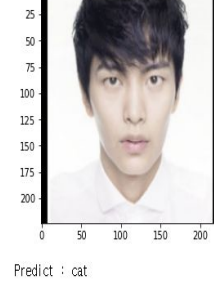
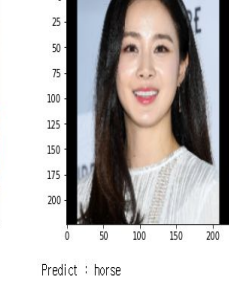
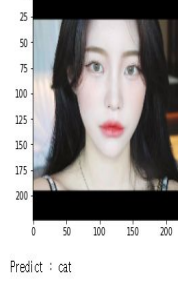
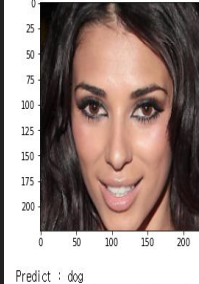
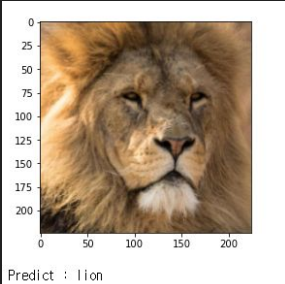
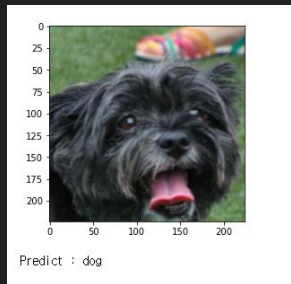
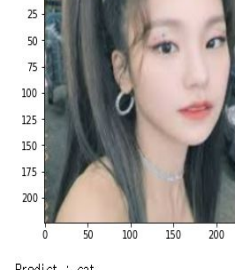
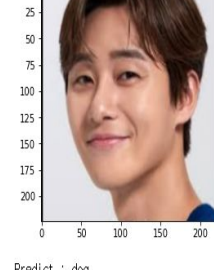
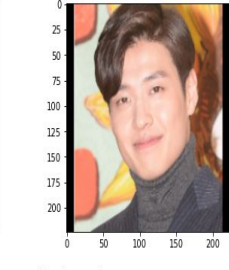
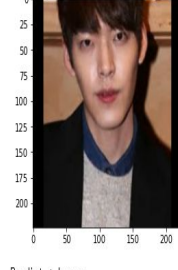
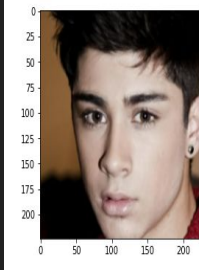
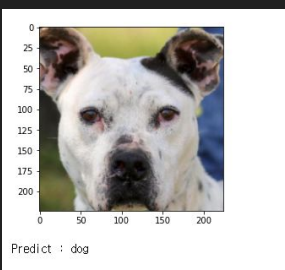
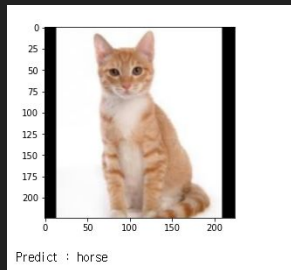
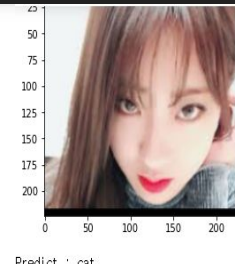
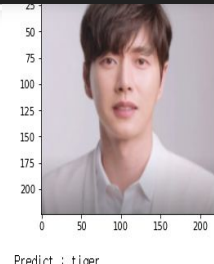
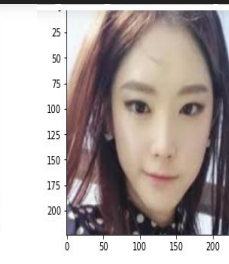
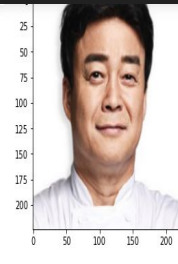
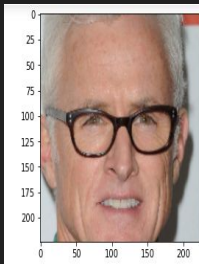
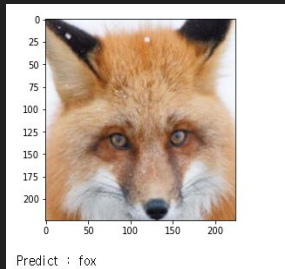
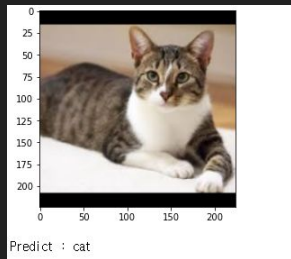
585MB (614,352,331 바이트)

636MB (667,881,472 바이트)

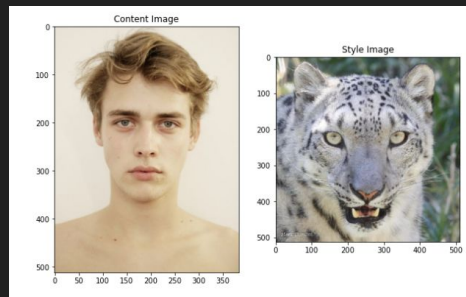
파일 26,179, 폴더 10



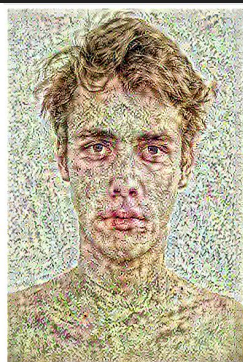
동물분류 - pretrained network(VGG16)



Style Transfer



$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$



Epoch: 495
Total loss: 2.6232e+09, style loss: 5.6187e+05, content loss: 2.6226e+09.



Epoch: 495
Total loss: 2.3141e+06, style loss: 7.8712e+05, content loss: 1.5270e+06.



Epoch: 495
Total loss: 3.7878e+05, style loss: 1.3928e+05, content loss: 2.3950e+05.



Epoch: 495
Total loss: 3.6008e+05, style loss: 1.5010e+05, content loss: 2.0999e+05.



Epoch: 495
Total loss: 6.3004e+05, style loss: 6.0451e+05, content loss: 2.5532e+04.

프로젝트 구조



Input image

→
Classification



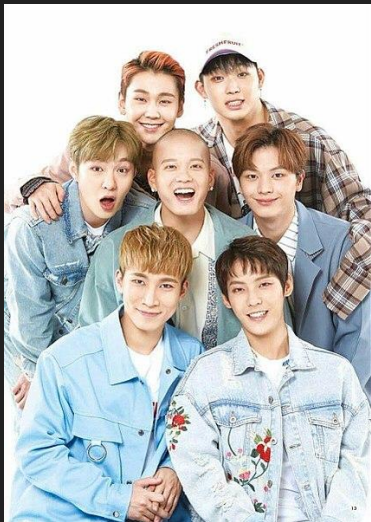
Content image

→
Synthesis



Result

그 외의 시도 - opencv



result0



result1



result2



result3



result4



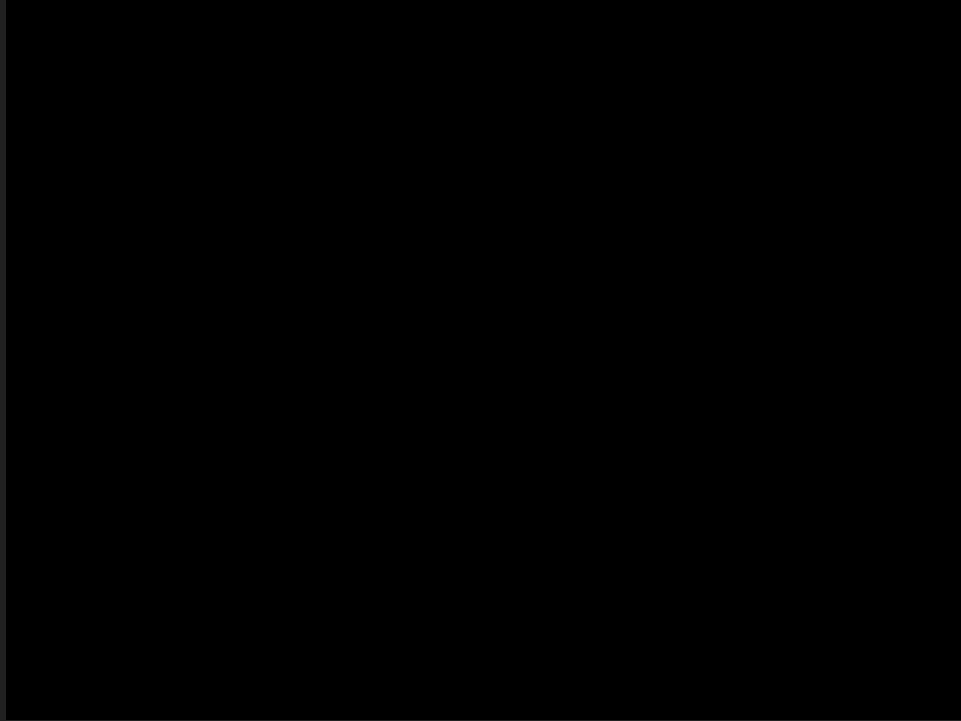
result5

추후 연구 - vae

conv2d_87 (Conv2D)	(None, 112, 112, 64)	73792	conv2d_86[0][0]
conv2d_88 (Conv2D)	(None, 56, 56, 32)	18464	conv2d_87[0][0]
conv2d_89 (Conv2D)	(None, 28, 28, 32)	9248	conv2d_88[0][0]
flatten_17 (Flatten)	(None, 25088)	0	conv2d_89[0][0]
dense_64 (Dense)	(None, 32)	802848	flatten_17[0][0]
dense_65 (Dense)	(None, 2)	66	dense_64[0][0]
dense_66 (Dense)	(None, 2)	66	dense_64[0][0]
lambda_17 (Lambda)	(None, 2)	0	dense_65[0][0] dense_66[0][0]
model_31 (Model)	(None, 224, 224, 3)	1049667	lambda_17[0][0]
custom_variational_layer_16 (Cu	[(None, 224, 224, 3) 0		input_33[0][0] model_31[2][0]
=====			
Total params: 1,957,735			

=====			
Epoch 1/10	4000/4000	[=====]	- 105s 26ms/step - loss: 0.6332 - val_loss: 0.6206
Epoch 2/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6288 - val_loss: 0.6211
Epoch 3/10	4000/4000	[=====]	- 96s 24ms/step - loss: 0.6278 - val_loss: 0.6214
Epoch 4/10	4000/4000	[=====]	- 96s 24ms/step - loss: 0.6267 - val_loss: 0.6203
Epoch 5/10	4000/4000	[=====]	- 96s 24ms/step - loss: 0.6263 - val_loss: 0.6210
Epoch 6/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6265 - val_loss: 0.6196
Epoch 7/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6261 - val_loss: 0.6204
Epoch 8/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6264 - val_loss: 0.6196
Epoch 9/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6260 - val_loss: 0.6237
Epoch 10/10	4000/4000	[=====]	- 97s 24ms/step - loss: 0.6261 - val_loss: 0.6211

시연 영상



감사합니다.

개발 환경

conda : 4.8.3

python : 3.67

tensorflow : 1.15-rc0

keras : keras 2.2.5

GPU : GTX - 1070Ti