

Model

May 16, 2018

```
In [30]: from keras.layers import Conv2D, Conv2DTranspose, UpSampling2D
         from keras.layers import Activation, Dense, Dropout, Flatten, InputLayer
         from keras.layers.normalization import BatchNormalization
         from keras.callbacks import TensorBoard
         from keras.models import Sequential
         from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array,
         from skimage.color import rgb2lab, lab2rgb, rgb2gray
         from skimage.io import imsave
         import numpy as np
         import os,keras
         import random

In [31]: # get 5000 images from mountain5000
X = []
for filename in os.listdir('../mountain5000/'):
    X.append(img_to_array(load_img('../mountain5000/'+filename)))
#to float array
X = np.array(X, dtype=float)
# Set up train data
Xto1 = 1.0/255*X

In [53]: # Image generator
datagen = ImageDataGenerator(
    shear_range=0.2,
    zoom_range=0.2,
    rotation_range=20,
    horizontal_flip=True)

# training data
batch_size = 10
def dataset(Xto1,batch_size):
    for batch in datagen.flow(Xto1, batch_size=batch_size):
        lab_batch = rgb2lab(batch)
        Xtr_batch = lab_batch[:,:,:,:0]
        Ytr_batch = lab_batch[:,:,:,:1:] / 128
        yield (Xtr_batch.reshape(Xtr_batch.shape+(1,)), Ytr_batch)

In [33]: keras.backend.clear_session()
model = Sequential()
```

```

model.add(InputLayer(input_shape=(256, 256, 1)))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same', strides=2))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same', strides=2))
model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(256, (3, 3), activation='relu', padding='same', strides=2))
model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(2, (3, 3), activation='tanh', padding='same'))
model.add(UpSampling2D((2, 2)))
model.compile(optimizer='rmsprop', loss='mse')

```

In [34]: model.summary()

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 256, 256, 64)	640
conv2d_2 (Conv2D)	(None, 128, 128, 64)	36928
conv2d_3 (Conv2D)	(None, 128, 128, 128)	73856
conv2d_4 (Conv2D)	(None, 64, 64, 128)	147584
conv2d_5 (Conv2D)	(None, 64, 64, 256)	295168
conv2d_6 (Conv2D)	(None, 32, 32, 256)	590080
conv2d_7 (Conv2D)	(None, 32, 32, 512)	1180160
conv2d_8 (Conv2D)	(None, 32, 32, 256)	1179904
conv2d_9 (Conv2D)	(None, 32, 32, 128)	295040
up_sampling2d_1 (UpSampling2D)	(None, 64, 64, 128)	0
conv2d_10 (Conv2D)	(None, 64, 64, 64)	73792
up_sampling2d_2 (UpSampling2D)	(None, 128, 128, 64)	0
conv2d_11 (Conv2D)	(None, 128, 128, 32)	18464

```

-----
conv2d_12 (Conv2D)                (None, 128, 128, 2)        578
-----
up_sampling2d_3 (UpSampling2D)    (None, 256, 256, 2)        0
=====
Total params: 3,892,194
Trainable params: 3,892,194
Non-trainable params: 0
-----

```

In [63]: # *Train model*

```
model.fit_generator(dataset(Xto1,batch_size), epochs=200, steps_per_epoch=10)
```

```

Epoch 1/50
10/10 [=====] - 128s 13s/step - loss: 0.0088
Epoch 2/50
10/10 [=====] - 68s 7s/step - loss: 0.0083
Epoch 3/50
10/10 [=====] - 69s 7s/step - loss: 0.0089
Epoch 4/50
10/10 [=====] - 67s 7s/step - loss: 0.0073
Epoch 5/50
10/10 [=====] - 65s 6s/step - loss: 0.0077
Epoch 6/50
10/10 [=====] - 65s 7s/step - loss: 0.0103
Epoch 7/50
10/10 [=====] - 71s 7s/step - loss: 0.0079
Epoch 8/50
10/10 [=====] - 66s 7s/step - loss: 0.0094
Epoch 9/50
10/10 [=====] - 65s 6s/step - loss: 0.0071
Epoch 10/50
10/10 [=====] - 65s 7s/step - loss: 0.0091
Epoch 11/50
10/10 [=====] - 66s 7s/step - loss: 0.0106
Epoch 12/50
10/10 [=====] - 65s 7s/step - loss: 0.0077
Epoch 13/50
10/10 [=====] - 66s 7s/step - loss: 0.0079
Epoch 14/50
10/10 [=====] - 1493s 149s/step - loss: 0.0095
Epoch 15/50
10/10 [=====] - 69s 7s/step - loss: 0.0077
Epoch 16/50
10/10 [=====] - 67s 7s/step - loss: 0.0072
Epoch 17/50
10/10 [=====] - 66s 7s/step - loss: 0.0093

```

Epoch 18/50
10/10 [=====] - 65s 6s/step - loss: 0.0086
Epoch 19/50
10/10 [=====] - 66s 7s/step - loss: 0.0076
Epoch 20/50
10/10 [=====] - 65s 7s/step - loss: 0.0092
Epoch 21/50
10/10 [=====] - 65s 7s/step - loss: 0.0081
Epoch 22/50
10/10 [=====] - 65s 6s/step - loss: 0.0089
Epoch 23/50
10/10 [=====] - 65s 6s/step - loss: 0.0086
Epoch 24/50
10/10 [=====] - 65s 7s/step - loss: 0.0083
Epoch 25/50
10/10 [=====] - 66s 7s/step - loss: 0.0082
Epoch 26/50
10/10 [=====] - 66s 7s/step - loss: 0.0073
Epoch 27/50
10/10 [=====] - 65s 7s/step - loss: 0.0074
Epoch 28/50
10/10 [=====] - 65s 7s/step - loss: 0.0072
Epoch 29/50
10/10 [=====] - 66s 7s/step - loss: 0.0083
Epoch 30/50
10/10 [=====] - 66s 7s/step - loss: 0.0075
Epoch 31/50
10/10 [=====] - 65s 7s/step - loss: 0.0075
Epoch 32/50
10/10 [=====] - 66s 7s/step - loss: 0.0079
Epoch 33/50
10/10 [=====] - 65s 7s/step - loss: 0.0077
Epoch 34/50
10/10 [=====] - 66s 7s/step - loss: 0.0082
Epoch 35/50
10/10 [=====] - 66s 7s/step - loss: 0.0076
Epoch 36/50
10/10 [=====] - 66s 7s/step - loss: 0.0078
Epoch 37/50
10/10 [=====] - 65s 7s/step - loss: 0.0102
Epoch 38/50
10/10 [=====] - 81s 8s/step - loss: 0.0091
Epoch 39/50
10/10 [=====] - 75s 7s/step - loss: 0.0077
Epoch 40/50
10/10 [=====] - 70s 7s/step - loss: 0.0078
Epoch 41/50
10/10 [=====] - 68s 7s/step - loss: 0.0083

```

Epoch 42/50
10/10 [=====] - 67s 7s/step - loss: 0.0091
Epoch 43/50
10/10 [=====] - 66s 7s/step - loss: 0.0087
Epoch 44/50
10/10 [=====] - 67s 7s/step - loss: 0.0087
Epoch 45/50
10/10 [=====] - 71s 7s/step - loss: 0.0094
Epoch 46/50
10/10 [=====] - 67s 7s/step - loss: 0.0067
Epoch 47/50
10/10 [=====] - 66s 7s/step - loss: 0.0079
Epoch 48/50
10/10 [=====] - 66s 7s/step - loss: 0.0085
Epoch 49/50
10/10 [=====] - 67s 7s/step - loss: 0.0084
Epoch 50/50
10/10 [=====] - 67s 7s/step - loss: 0.0072

```

```

Out[63]: <keras.callbacks.History at 0x12dfcd0f0>

```

```

In [65]: #evaluation
t = []
keyword='mountain'
for i in range(0,100):
    t.append(img_to_array(load_img("../Mountaintest/{0:s}_{1:04d}.jpg".format(keyword, i))))
t = np.array(t)
t= rgb2lab(1.0/255*t)
Xts=t[:, :, :, 0]
Xts = Xts.reshape(Xts.shape+(1,))
yts=t[:, :, :, 1:]
yts=yts/128

```

```

In [61]: # evaluate
print(model.evaluate(Xts, yts, batch_size=batch_size))

```

```

100/100 [=====] - 23s 227ms/step
0.007777458056807518

```

```

In [66]: # Test model
output = model.predict(Xts)
output = output * 128

# Output
for i in range(len(output)):
    color = np.zeros((256, 256, 3))
    color[:, :, 0] = t[i][:, :, 0]

```

```
color[:, :, 1:] = output[i]
imsave("result/col_"+str(i)+".png", lab2rgb(color))
imsave("gray/gray_"+str(i)+".png", rgb2gray(lab2rgb(color)))

/anaconda/lib/python3.6/site-packages/skimage/util/dtype.py:122: UserWarning: Possible precision loss
.format(dtypeobj_in, dtypeobj_out))
/anaconda/lib/python3.6/site-packages/skimage/util/dtype.py:122: UserWarning: Possible precision loss
.format(dtypeobj_in, dtypeobj_out))
```

In []: