Image Super-Resolution Using Deep Convolutional Networks

(5) implementation

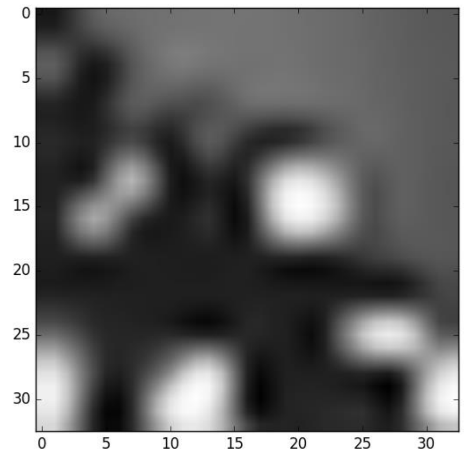
Visual Computing Lab

YoungHoon Kwon

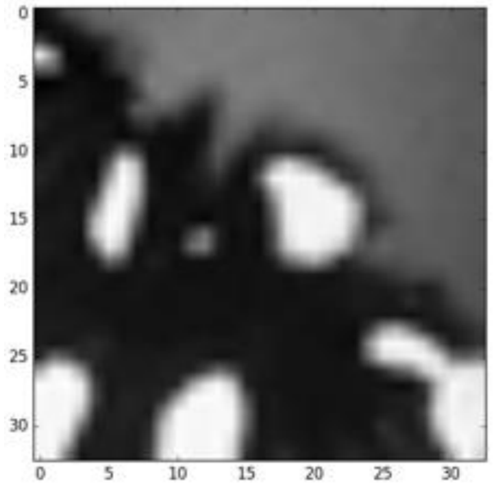
Previous Result

- Training Data 30 set → 91 set
- Gradient Descent → Rmsprop, Adam
- Cost 0.032

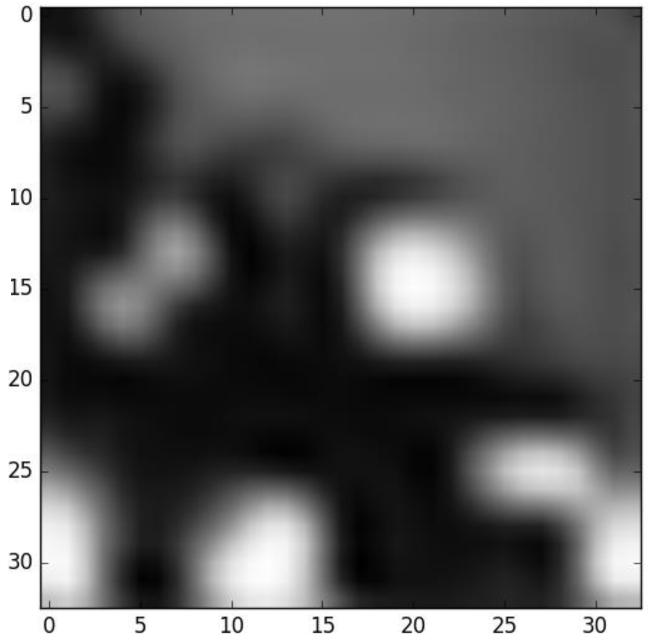
Previous Result - Batch 적용 후



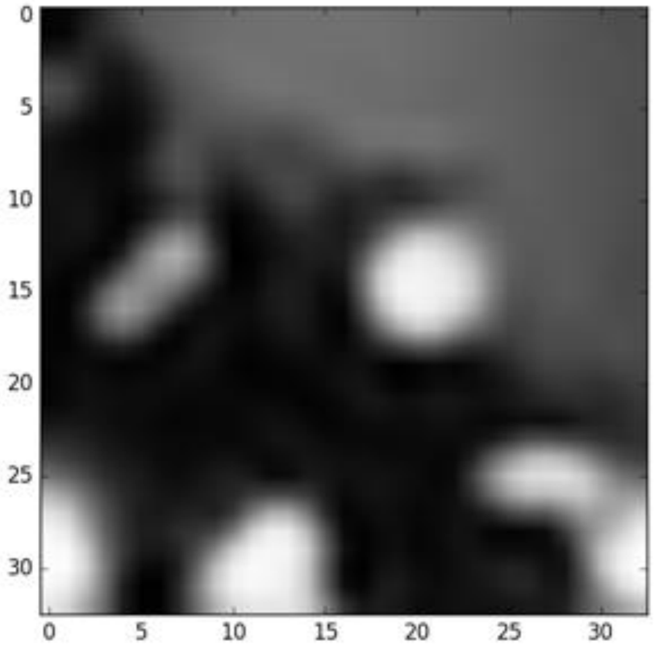
Input



Original



적용 전
50만번 30set
Cost:0.12847



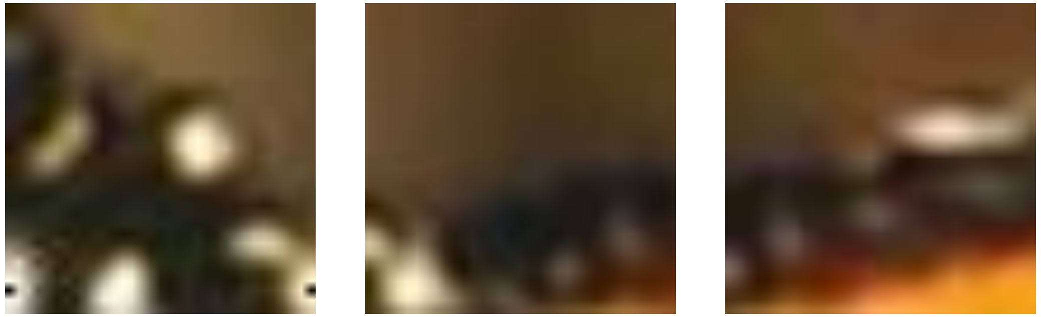
적용 후
300만번 90set
Cost:0.03232

Result

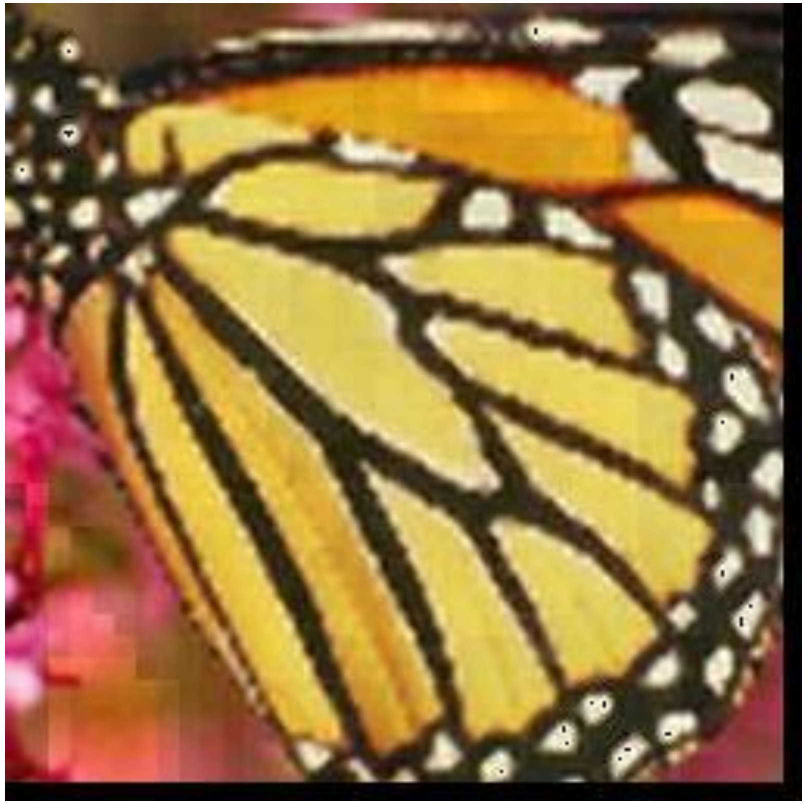
- 1 color → 3 colors
- sub image output→ full image output
- Overfitting test vs Learning test
- Cost 0.0323 → 0.01158

1 color → 3 colors

Sub image → Full image



33x33 sub Images output
Merge to full image



Bad result

Sub image → Full image

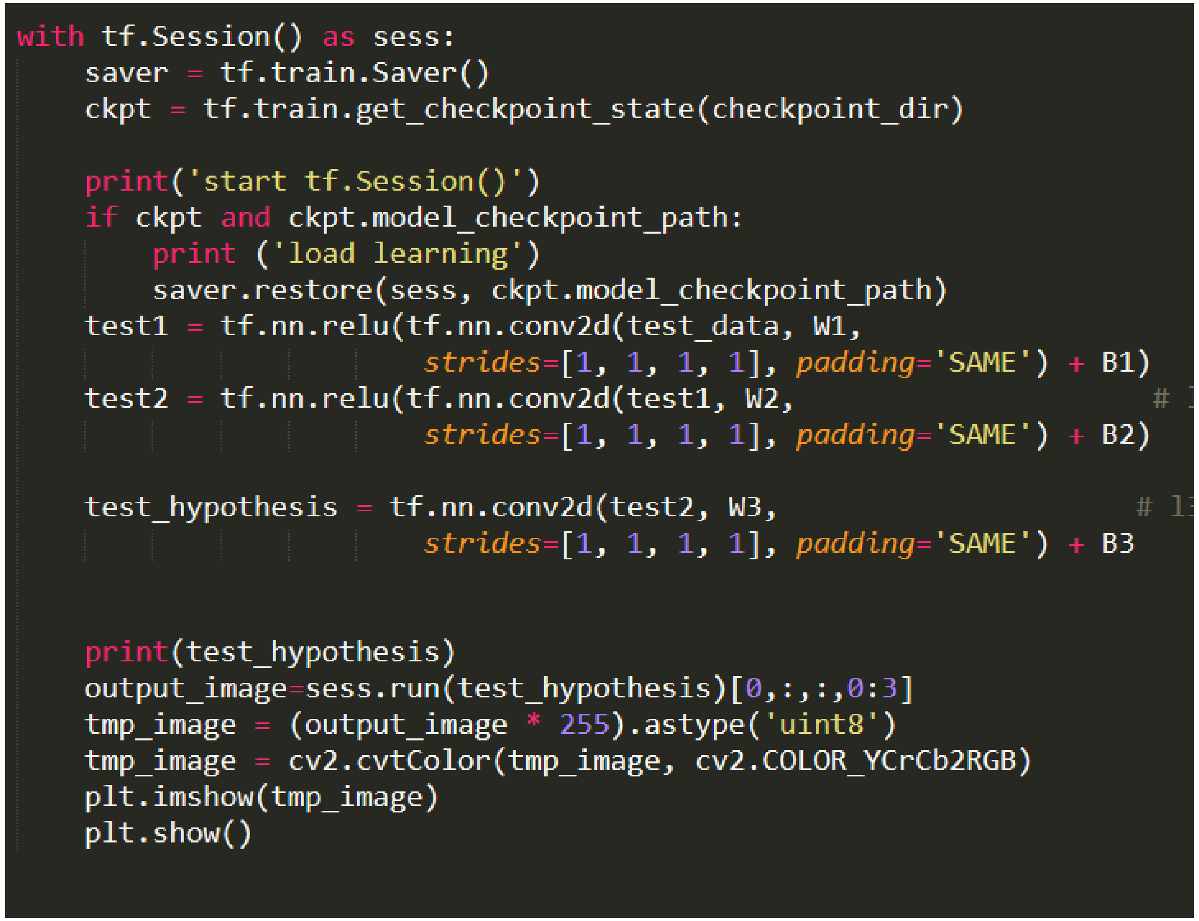
Test data

[subImages,33,33,3]

↓

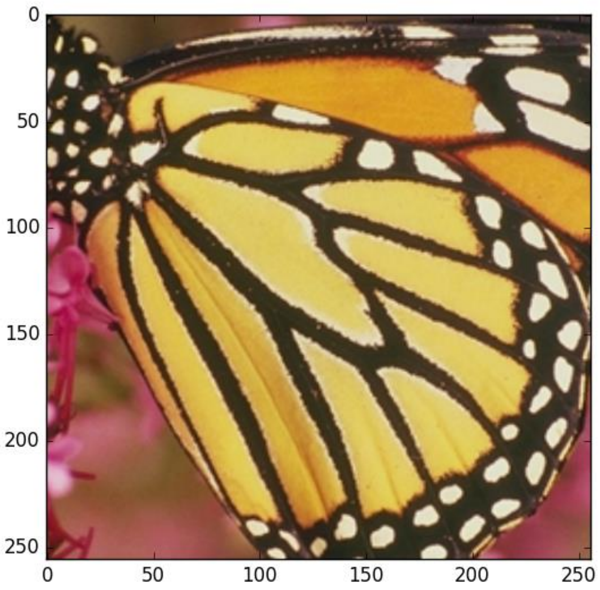
[1,width,height,3]

Test 파일 크기 변환

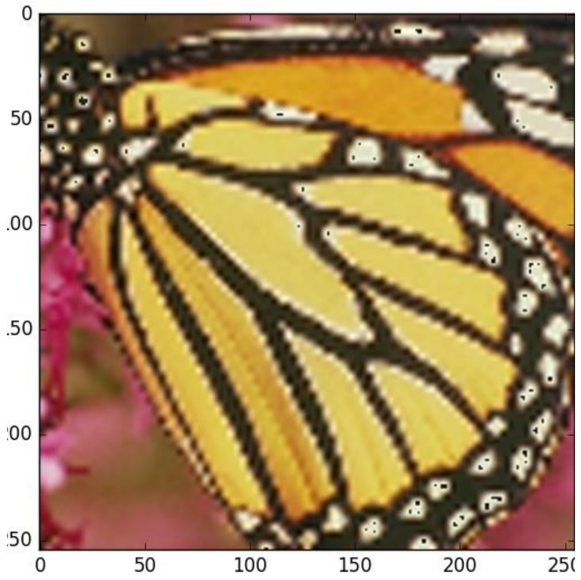


```
with tf.Session() as sess:
    saver = tf.train.Saver()
    ckpt = tf.train.get_checkpoint_state(checkpoint_dir)

    print('start tf.Session()')
    if ckpt and ckpt.model_checkpoint_path:
        print ('load learning')
        saver.restore(sess, ckpt.model_checkpoint_path)
    test1 = tf.nn.relu(tf.nn.conv2d(test_data, W1,
                        strides=[1, 1, 1, 1], padding='SAME') + B1)
    test2 = tf.nn.relu(tf.nn.conv2d(test1, W2,                         # 1
                        strides=[1, 1, 1, 1], padding='SAME') + B2)

    test_hypothesis = tf.nn.conv2d(test2, W3,                         # 1
                        strides=[1, 1, 1, 1], padding='SAME') + B3


    print(test_hypothesis)
    output_image=sess.run(test_hypothesis)[0,:,:,0:3]
    tmp_image = (output_image * 255).astype('uint8')
    tmp_image = cv2.cvtColor(tmp_image, cv2.COLOR_YCrCb2RGB)
    plt.imshow(tmp_image)
    plt.show()
```

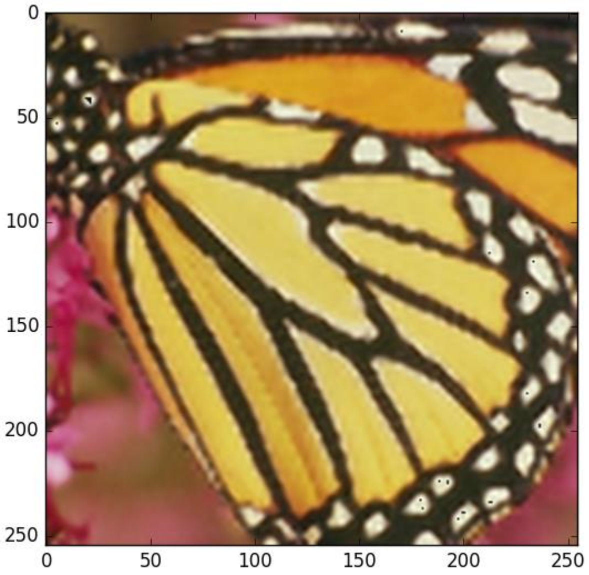
Sub image → Full image



Original



Input

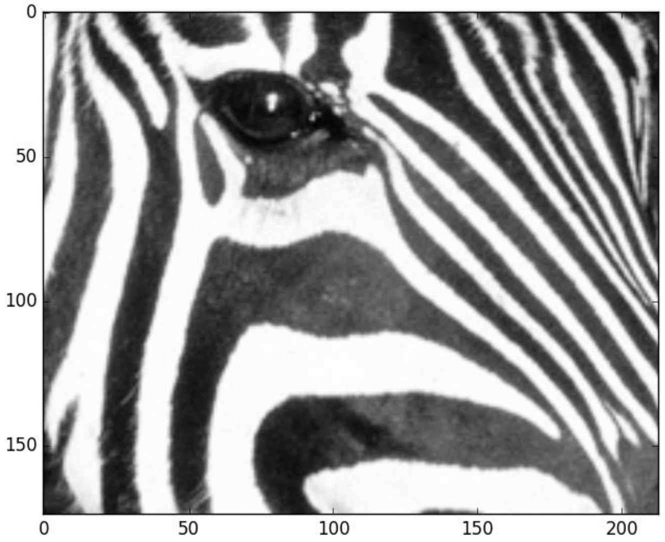


Output

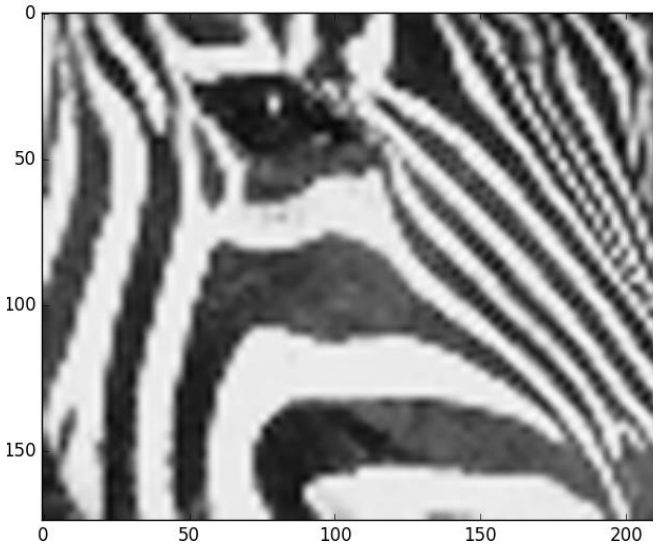
AdamOptimizer
Learning_rate = 1/10^5
Step_count =3*10^6

Cost:0.01158

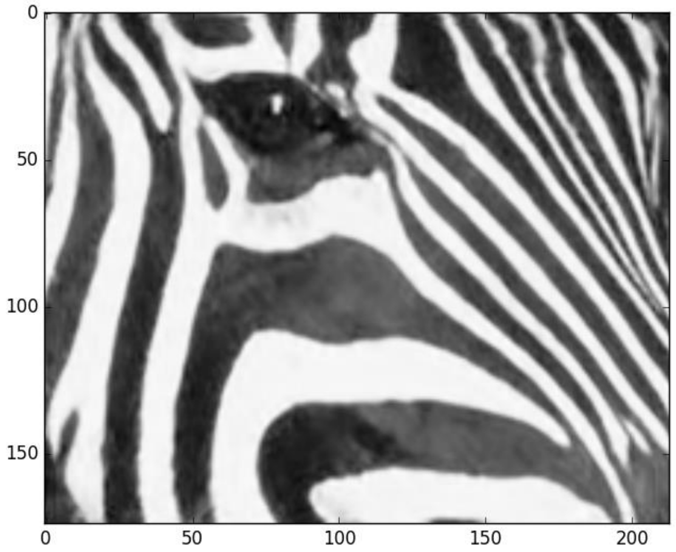
Overfitting test



Original

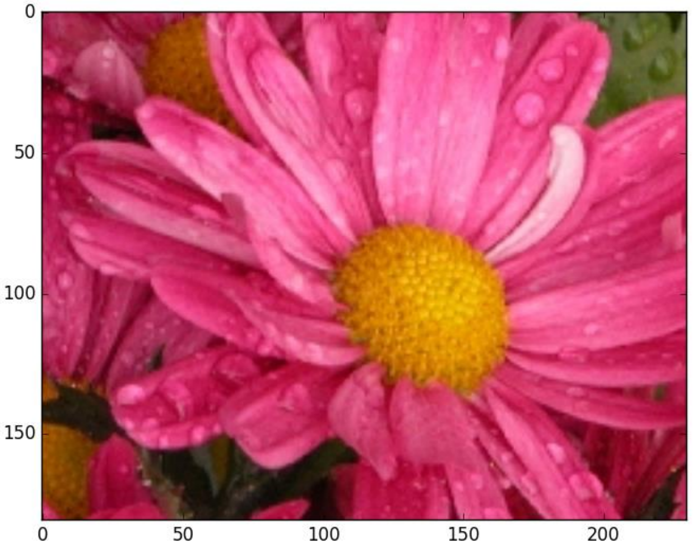


Input

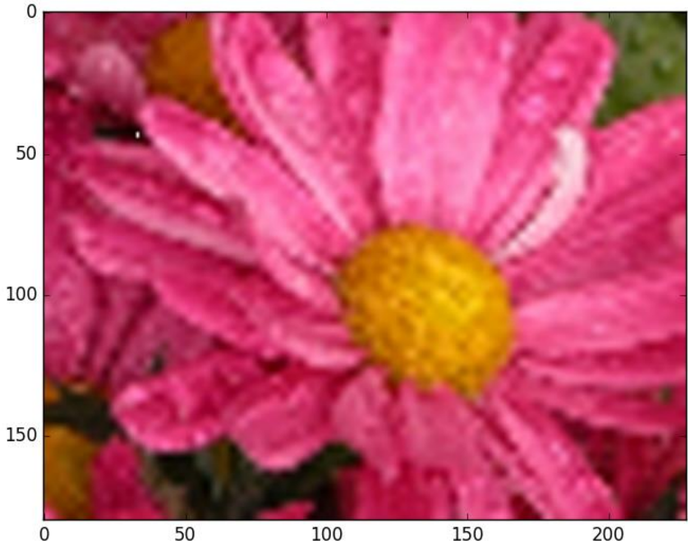


Overfitting

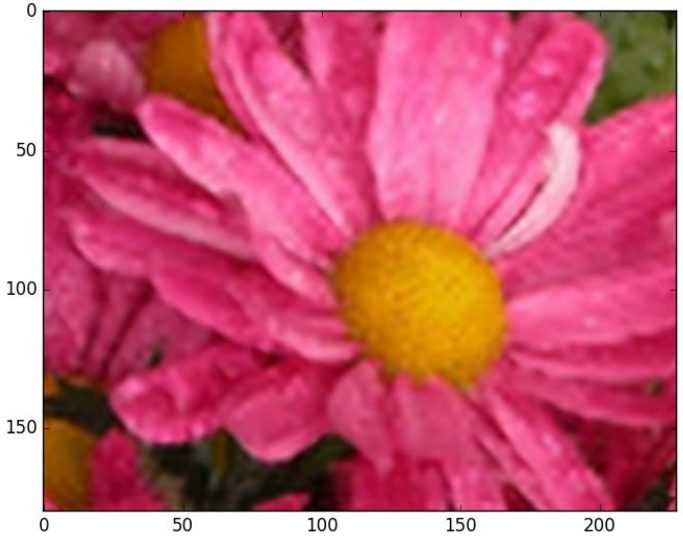
Overfitting test



Original

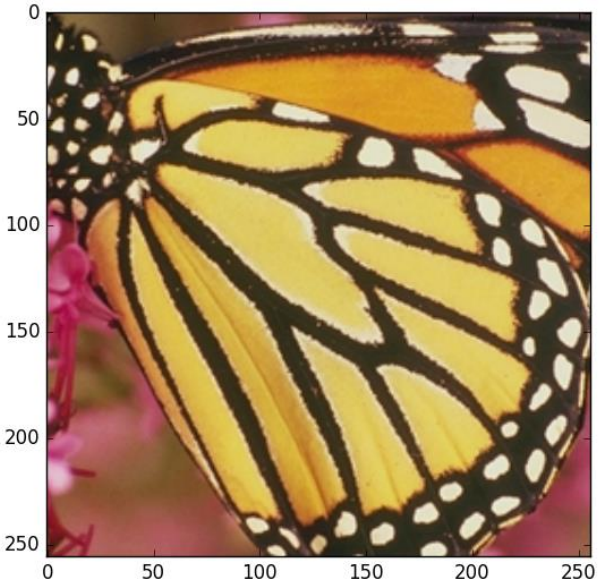


Input

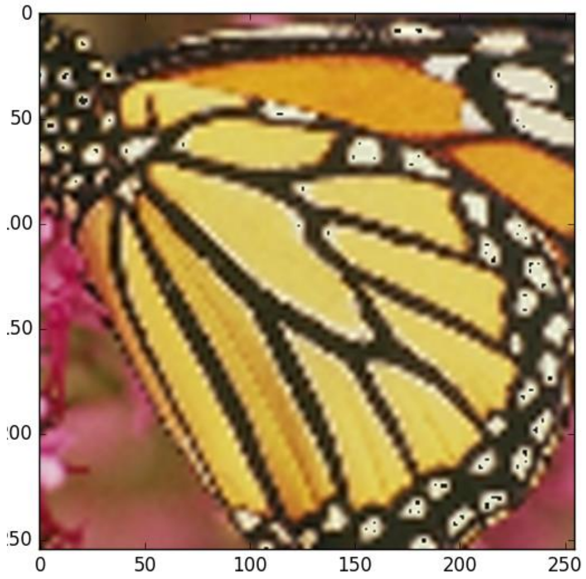


Overfitting

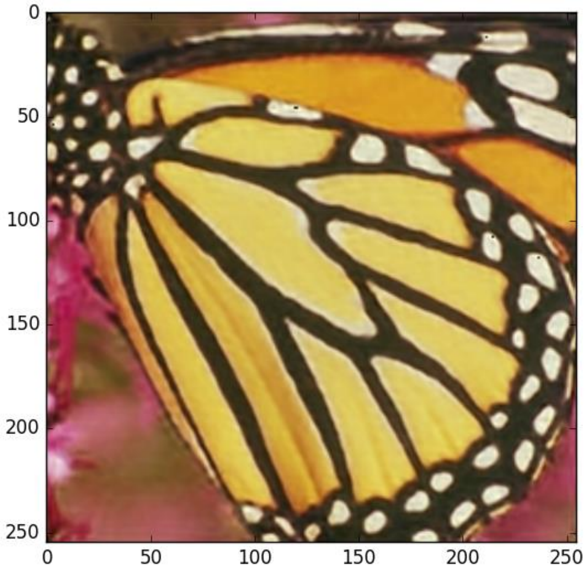
Overfitting test



Original

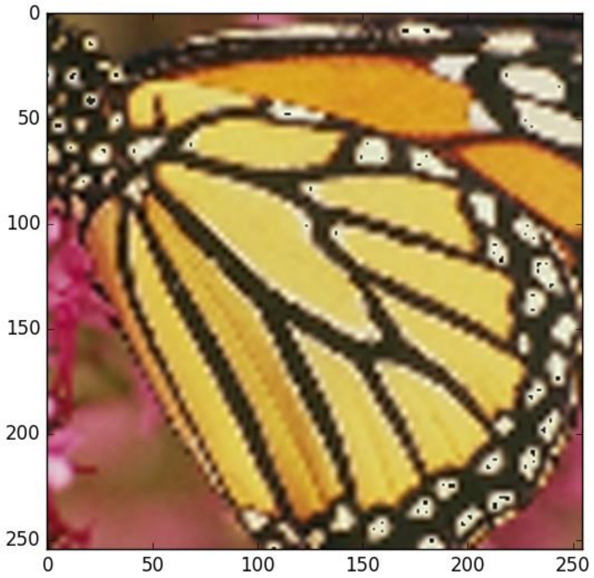


Input

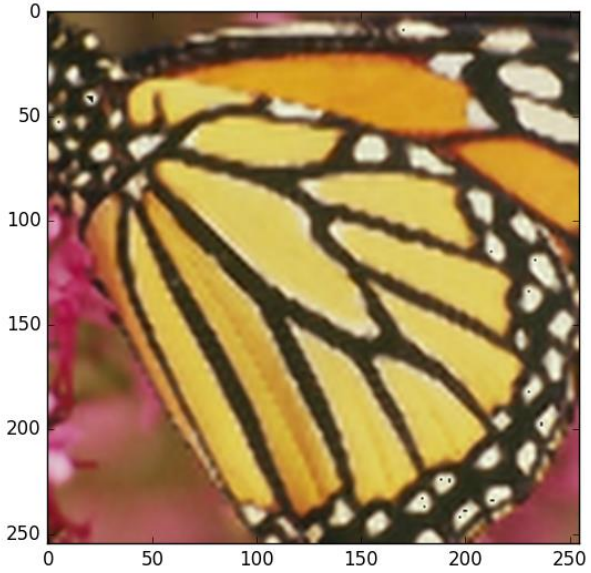


Overfitting

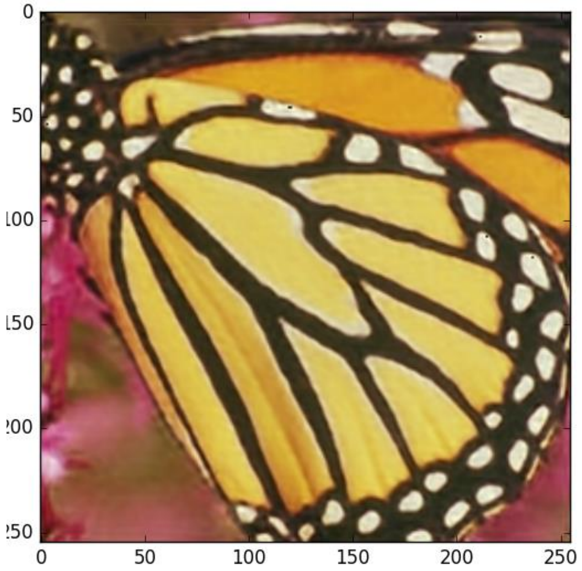
Overfitting vs Output



Input

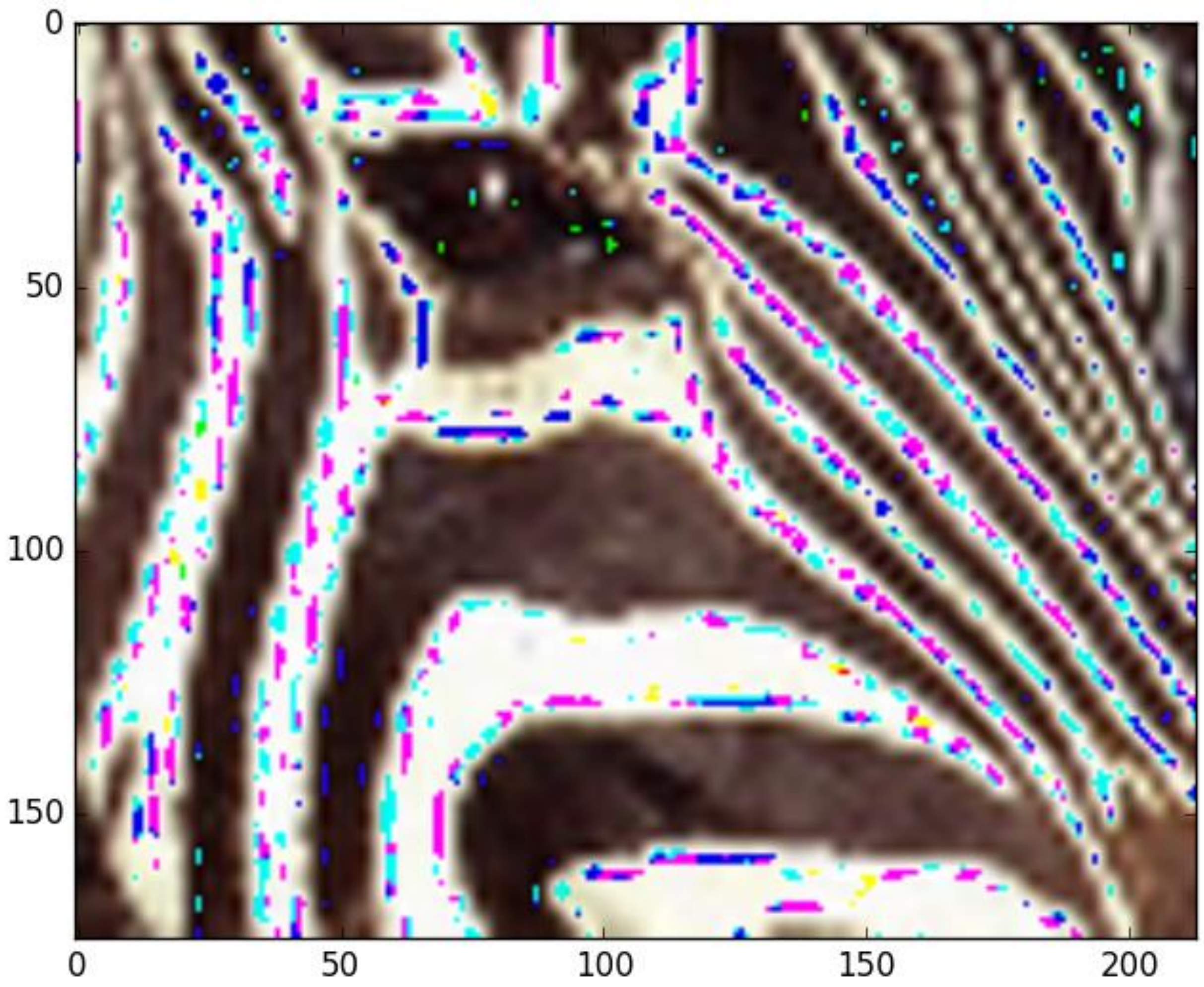


Output

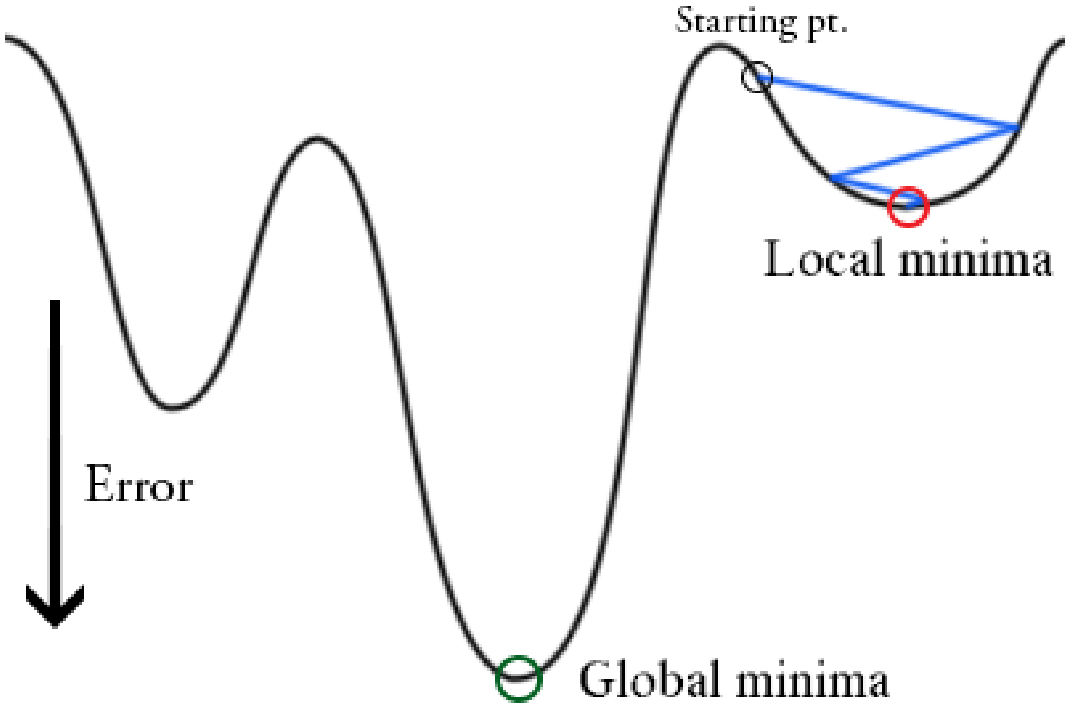


Overfitting

Future works



Future works



- Local Minimum 탈출하기