# show_results

November 8, 2016

# 1 Load model

## 1.1 Model

conv1_1 = new_conv_layer( image, [9, 9, 1 , 16], “conv1_1” ) conv1_2 = new_conv_layer( conv1_1, [9, 9, 16, 16], “conv1_2” ) pool1 = tf.nn.max_pool( conv1_2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1],padding=‘SAME’, name=‘pool1’)

conv2_1 = new_conv_layer( pool1 , [9, 9, 16, 16], “conv2_1”) conv2_2 = new_conv_layer( conv2_1, [9, 9, 16, 16], “conv2_2”) pool2 = tf.nn.max_pool( conv1_2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1],padding=‘SAME’, name=‘pool2’)

conv3_1 = new_conv_layer( pool2 , [9, 9, 16, 16], “conv3_1”) conv3_2 = new_conv_layer( conv2_1, [9, 9, 16, 16], “conv3_2”)

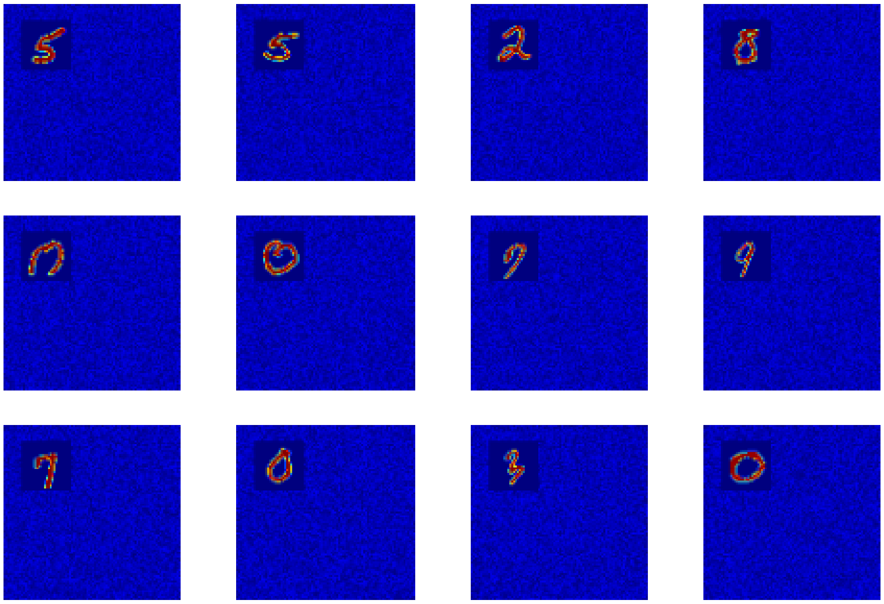
gap = tf.reduce_mean( conv3_2, [1,2] ) ccn = tf.reshape(gap,[-1,10,nb_CCN]) ccn = tf.reduce_mean(ccn, 2)

## 1.2 train params

L2 on weights (5e-5) L1 on GAP Batch normalisation

```
In [9]: import matplotlib.pyplot as plt
        import numpy as np
        import simple_model
        from simple_model import training_generator
        import utils


        simple_model=reload(simple_model)
        lr         = .0005
        lr_decay   = .9
        back_size  = 100
        noise      = .1
        crop_pos   = (10,10)
        n_CCN      = 5
```

```
Exception AssertionError: AssertionError("Nesting violated for default stack of <ty
In [10]: """Plot training samples"""
         batch = utils.get_batch('train', im_size=back_size, noise=noise, crop_pos=

         fig, axs = plt.subplots(3,4)
         for ax,img in zip([b for a in axs for b in a],batch[0]):
             ax.imshow(img.reshape(back_size,back_size), vmin=0, vmax=1)
             ax.set_axis_off()
         plt.show()
```



```
In [11]: """Perform training"""
         gen = training_generator(lr=lr, lr_decay=lr_decay, back_size=back_size, no
         for _ in range(25):
             _,accs = gen.next()
             print "max acc so far : "+str(max(accs)*100)
```

```
***** EPOCH 0 ******
1-Loss on testset is 0.623310
1-Accuracy now is 92.33
2-Loss on testset is 0.482131
2-Accuracy now is 95.79
lr now is 0.00045
max acc so far : 92.33

***** EPOCH 1 ******
```

```
1-Loss on testset is 0.365448
1-Accuracy now is 97.75
2-Loss on testset is 0.315835
2-Accuracy now is 98.47
lr now is 0.00041
max acc so far : 97.75

***** EPOCH 2 ******
1-Loss on testset is 0.365651
1-Accuracy now is 97.42
2-Loss on testset is 0.310321
2-Accuracy now is 98.18
lr now is 0.00036
max acc so far : 97.75

***** EPOCH 3 ******
1-Loss on testset is 0.287191
1-Accuracy now is 98.58
2-Loss on testset is 0.260151
2-Accuracy now is 98.92
lr now is 0.00033
max acc so far : 98.58

***** EPOCH 4 ******
1-Loss on testset is 0.276115
1-Accuracy now is 98.58
2-Loss on testset is 0.243866
2-Accuracy now is 99.02
lr now is 0.00030
max acc so far : 98.58

***** EPOCH 5 ******
1-Loss on testset is 0.285786
1-Accuracy now is 98.35
2-Loss on testset is 0.243088
2-Accuracy now is 99.04
lr now is 0.00027
max acc so far : 98.58

***** EPOCH 6 ******
1-Loss on testset is 0.270696
1-Accuracy now is 98.31
2-Loss on testset is 0.232653
2-Accuracy now is 99.06
lr now is 0.00024
max acc so far : 98.58

***** EPOCH 7 ******
```

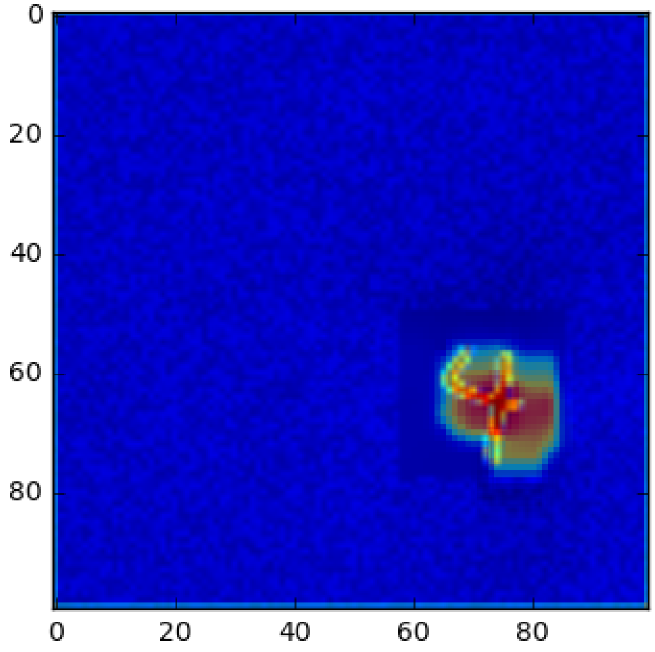
```
1-Loss on testset is 0.282187
1-Accuracy now is 98.48
2-Loss on testset is 0.246294
2-Accuracy now is 98.94
lr now is 0.00022
max acc so far : 98.58

***** EPOCH 8 ******
1-Loss on testset is 0.260417
1-Accuracy now is 98.80
2-Loss on testset is 0.229931
2-Accuracy now is 99.06
lr now is 0.00019
max acc so far : 98.8

***** EPOCH 9 ******
1-Loss on testset is 0.252398
1-Accuracy now is 98.56
2-Loss on testset is 0.211901
2-Accuracy now is 99.35
lr now is 0.00017
max acc so far : 98.8

***** EPOCH 10 ******
1-Loss on testset is 0.234339
1-Accuracy now is 99.19
2-Loss on testset is 0.217477
2-Accuracy now is 99.30
lr now is 0.00016
max acc so far : 99.19

***** EPOCH 11 ******
1-Loss on testset is 0.230298
1-Accuracy now is 99.14
2-Loss on testset is 0.218464
2-Accuracy now is 99.08
lr now is 0.00014
max acc so far : 99.19

***** EPOCH 12 ******
1-Loss on testset is 0.224320
1-Accuracy now is 99.21
2-Loss on testset is 0.209950
2-Accuracy now is 99.31
lr now is 0.00013
max acc so far : 99.21

***** EPOCH 13 ******
```

```
1-Loss on testset is 0.217308
1-Accuracy now is 99.31
2-Loss on testset is 0.201499
2-Accuracy now is 99.45
lr now is 0.00011
max acc so far : 99.31

***** EPOCH 14 ******
1-Loss on testset is 0.234372
1-Accuracy now is 98.82
2-Loss on testset is 0.203177
2-Accuracy now is 99.17
lr now is 0.00010
max acc so far : 99.31

***** EPOCH 15 ******
1-Loss on testset is 0.210033
1-Accuracy now is 99.40
2-Loss on testset is 0.194031
2-Accuracy now is 99.38
lr now is 0.00009
max acc so far : 99.4

***** EPOCH 16 ******
1-Loss on testset is 0.211489
1-Accuracy now is 99.17
2-Loss on testset is 0.191161
2-Accuracy now is 99.38
lr now is 0.00008
max acc so far : 99.4

***** EPOCH 17 ******
1-Loss on testset is 0.220471
1-Accuracy now is 99.15
2-Loss on testset is 0.196733
2-Accuracy now is 99.37
lr now is 0.00008
max acc so far : 99.4

***** EPOCH 18 ******
1-Loss on testset is 0.217570
1-Accuracy now is 99.21
2-Loss on testset is 0.195320
2-Accuracy now is 99.41
lr now is 0.00007
max acc so far : 99.4

***** EPOCH 19 ******
```

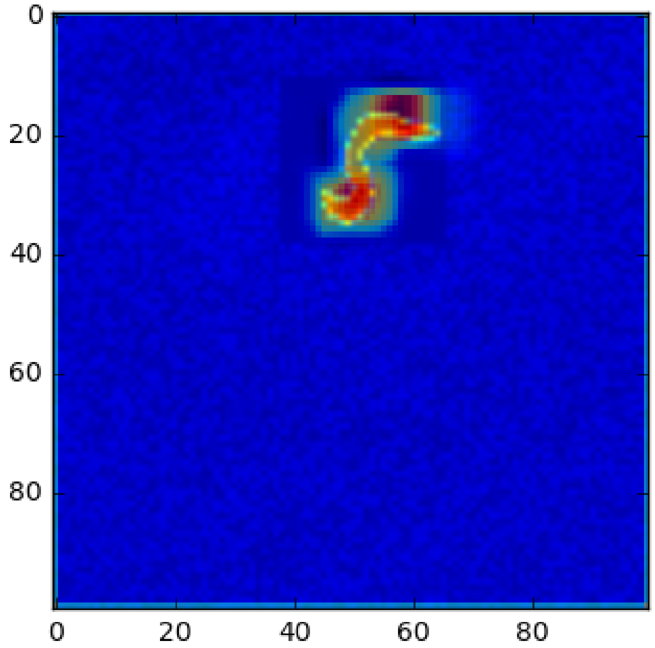
```
1-Loss on testset is 0.242923
1-Accuracy now is 98.79
2-Loss on testset is 0.198314
2-Accuracy now is 99.25
lr now is 0.00006
max acc so far : 99.4

***** EPOCH 20 ******
1-Loss on testset is 0.218623
1-Accuracy now is 99.30
2-Loss on testset is 0.191605
2-Accuracy now is 99.46
lr now is 0.00005
max acc so far : 99.4

***** EPOCH 21 ******
1-Loss on testset is 0.207488
1-Accuracy now is 99.26
2-Loss on testset is 0.191065
2-Accuracy now is 99.38
lr now is 0.00005
max acc so far : 99.4

***** EPOCH 22 ******
1-Loss on testset is 0.205563
1-Accuracy now is 99.35
2-Loss on testset is 0.185548
2-Accuracy now is 99.44
lr now is 0.00004
max acc so far : 99.4

***** EPOCH 23 ******
1-Loss on testset is 0.204758
1-Accuracy now is 99.28
2-Loss on testset is 0.186399
2-Accuracy now is 99.40
lr now is 0.00004
max acc so far : 99.4

***** EPOCH 24 ******
1-Loss on testset is 0.231354
1-Accuracy now is 97.94
2-Loss on testset is 0.183702
2-Accuracy now is 99.46
lr now is 0.00004
max acc so far : 99.4
```

```
In [12]: back_size  = 100

         for _ in range(10):
             print '----------------'*5
             imgs, lbls = utils.get_batch('test', 1, back_size, .1).next()

             simple_model.show_activation(imgs[0])
```

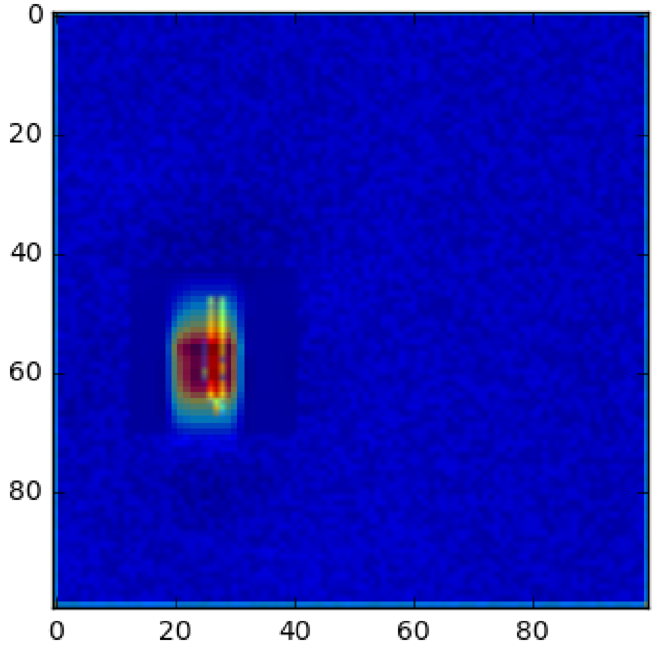
```
--------------------------------------------------------------------------------
prediction is : 4 with 11.781
```



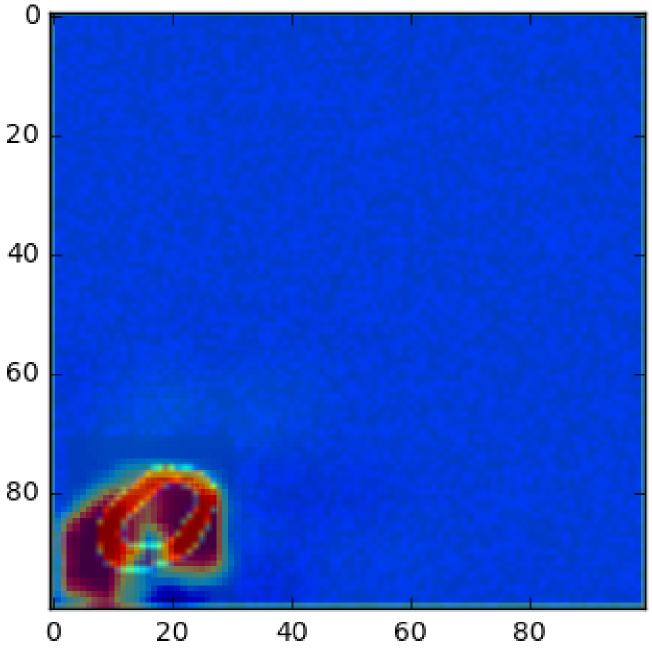
```
--------------------------------------------------------------------------------
prediction is : 5 with 13.845
```



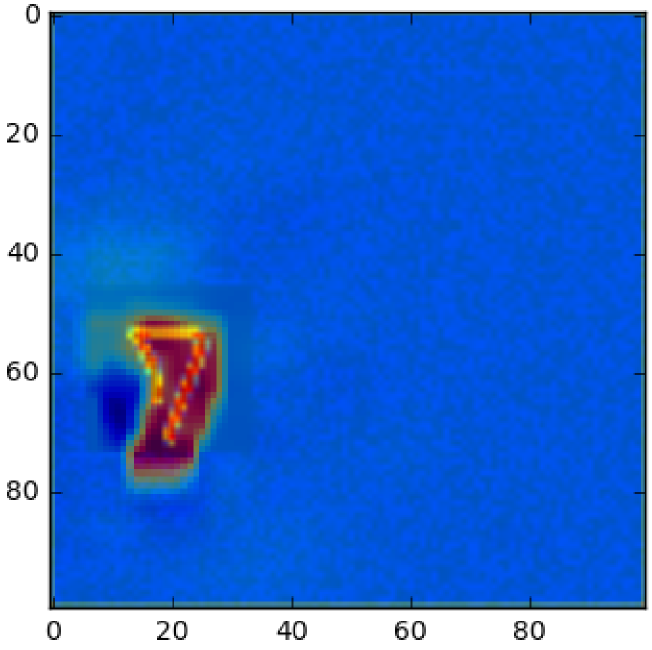
```
--------------------------------------------------------------------------------
prediction is : 1 with 10.484
```



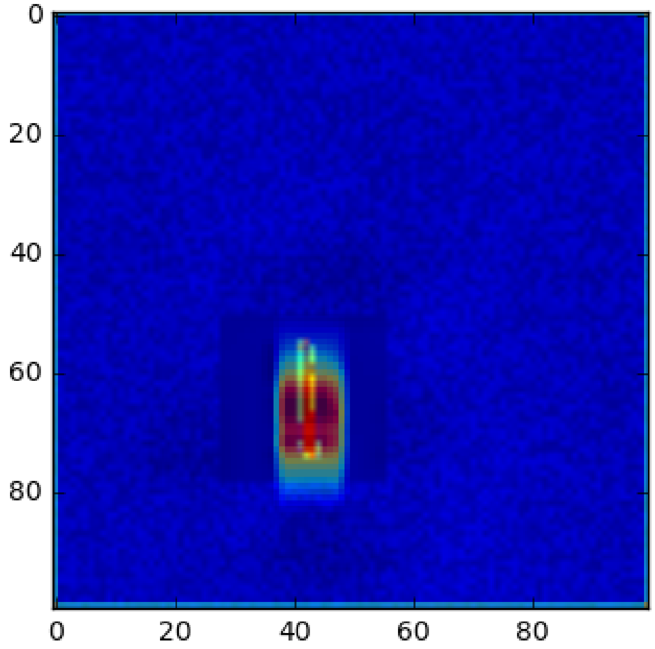
```
--------------------------------------------------------------------------------
prediction is : 0 with 2.679
```



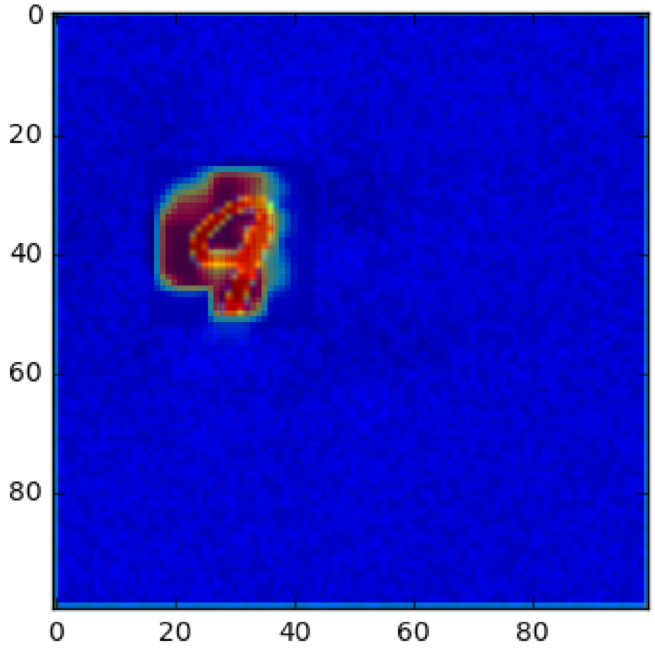
```
--------------------------------------------------------------------------------
prediction is : 7 with 8.640
```



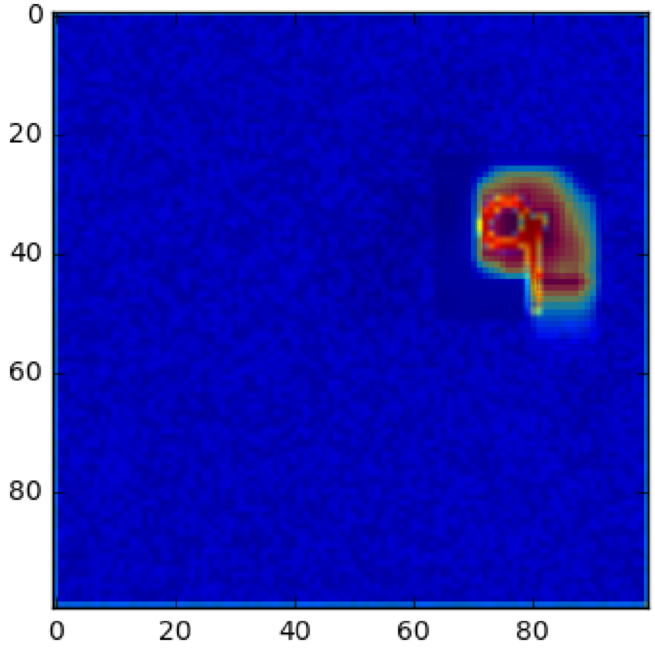
```
--------------------------------------------------------------------------------
prediction is : 1 with 10.407
```



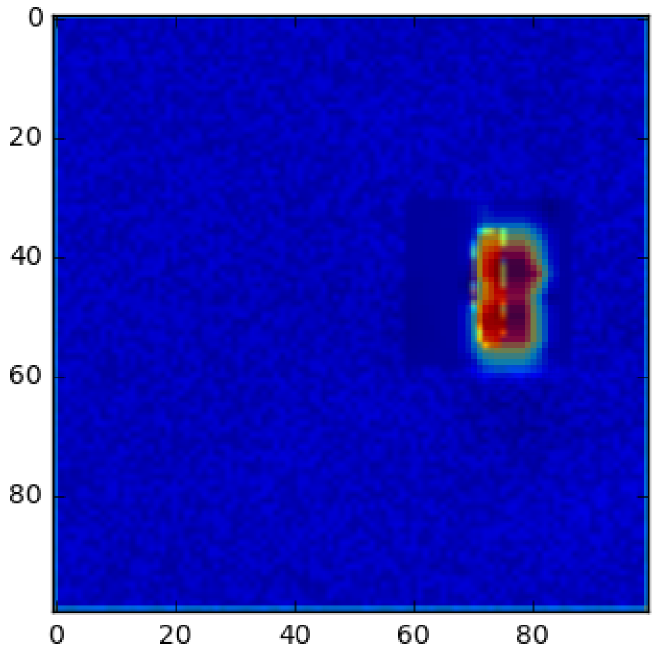
```
--------------------------------------------------------------------------------
prediction is : 9 with 9.370
```



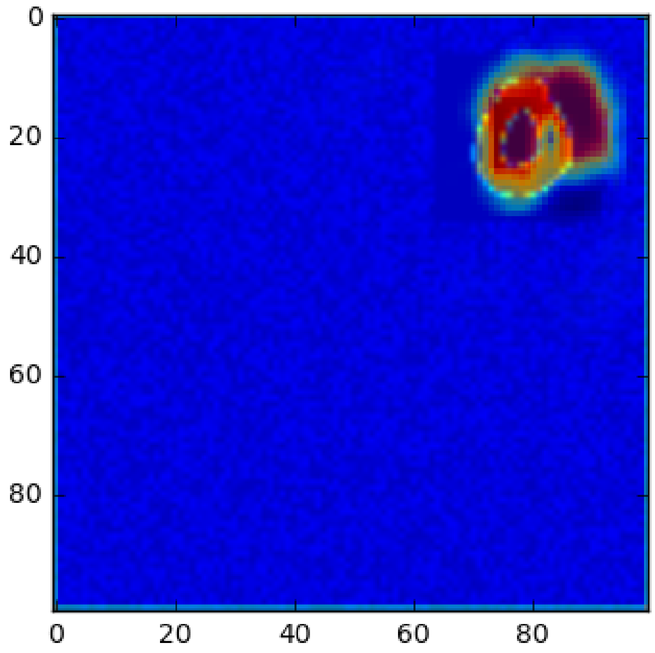
```
--------------------------------------------------------------------------------
prediction is : 9 with 15.312
```



```
--------------------------------------------------------------------------------
prediction is : 1 with 11.298
```



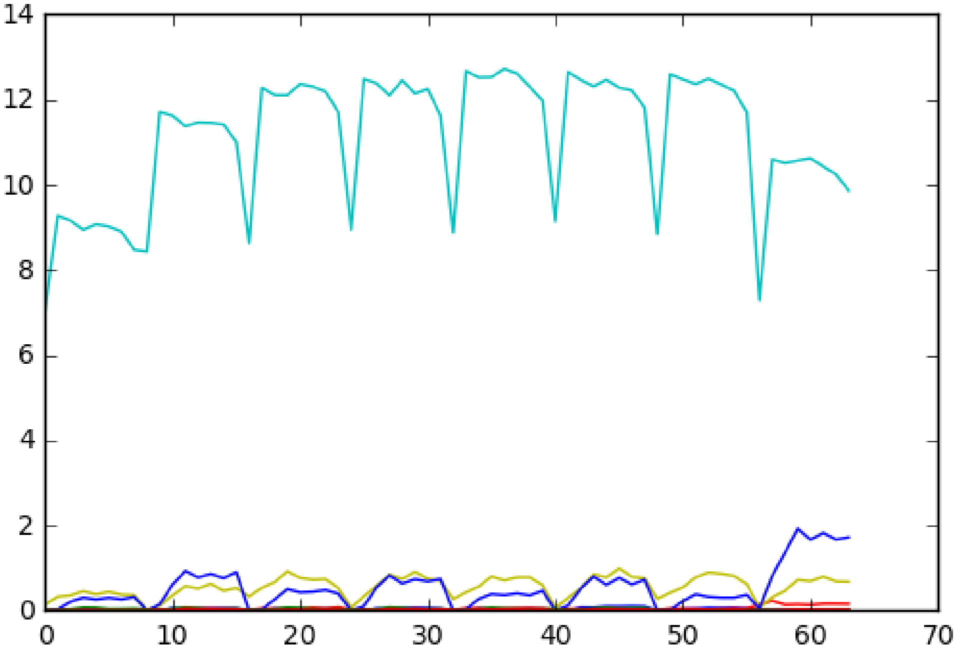
```
--------------------------------------------------------------------------------
prediction is : 0 with 10.595
```

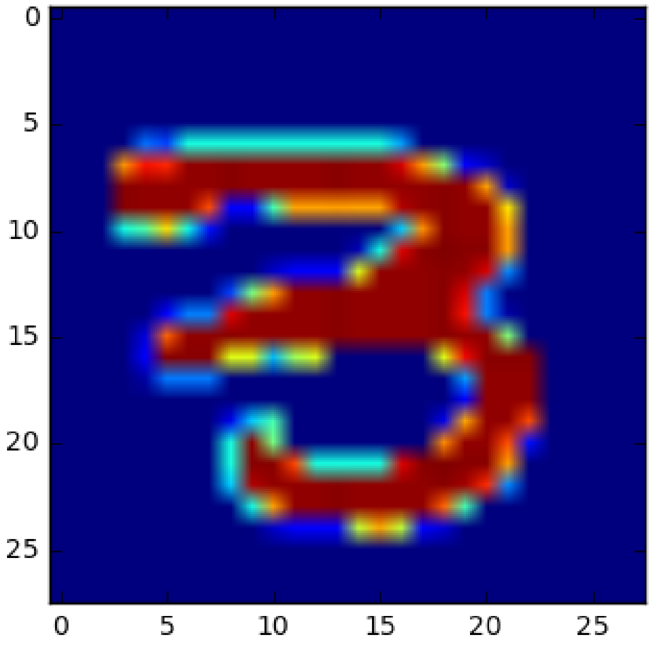


```
In [13]: accuracies = []
         for _ in range(100):
             digit,lbl  = utils.get_batch('test', 1, .1).next()
             digit      = digit[0].reshape(28,28)
             img        = np.random.random((100,100))*.1
             height     = img.shape[0]
             width      = img.shape[1]
             box_size   = 28
             step_size  = 10
             n_x_boxes  = (width -box_size)/step_size +1
             n_y_boxes  = (height-box_size)/step_size +1
             imgs       = np.tile(img, (n_x_boxes*n_y_boxes,1,1))

             for xx in range(0, n_x_boxes ):
                 for yy in range(0, n_y_boxes ):
                     idx = xx*n_x_boxes+yy
                     x   = xx*step_size
                     y   = yy*step_size
                     imgs[idx, x:x+box_size, y:y+box_size ] = digit


             imgs  = imgs.reshape((-1,100,100,1))
             preds = simple_model.sess.run(simple_model.tf_out, feed_dict={simple_m
             pred  = np.argmax(preds[-1])

             accuracy = sum(preds.argmax(axis=1) == lbl) / float(len(preds))
             accuracies.append(accuracy)

         print sum(accuracies)/len(accuracies)

0.99578125
```

```
In [14]: digit,lbl  = utils.get_batch('test', 1, .1).next()
         digit      = digit[0].reshape(28,28)
         img        = np.random.random((100,100))*.1
         height     = img.shape[0]
         width      = img.shape[1]
         box_size   = 28
         step_size  = 10
         n_x_boxes  = (width -box_size)/step_size +1
         n_y_boxes  = (height-box_size)/step_size +1
         imgs       = np.tile(img, (n_x_boxes*n_y_boxes,1,1))

         for xx in range(0, n_x_boxes ):
             for yy in range(0, n_y_boxes ):
                 idx = xx*n_x_boxes+yy
                 x   = xx*step_size
                 y   = yy*step_size
                 imgs[idx, x:x+box_size, y:y+box_size ] = digit
         lr_decay

         imgs  = imgs.reshape((-1,100,100,1))
         preds = simple_model.sess.run(simple_model.tf_out, feed_dict={simple_model
         pred  = np.argmax(preds[-1])

         accuracy = sum(preds.argmax(axis=1) == lbl) / float(len(preds))

         print preds.argmax(axis=1)
         print accuracy*100
         plt.plot(preds)
         plt.show()

         plt.imshow(digit)
         plt.show()
```

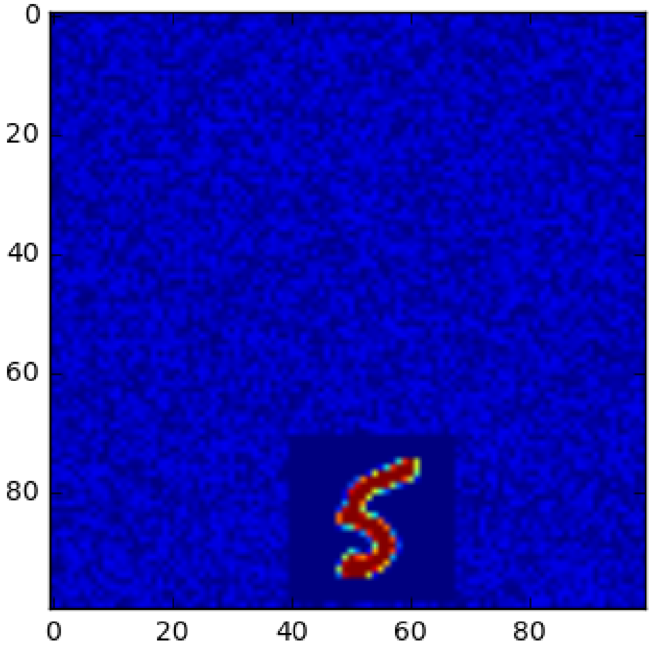
```
[3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]
100.0
```

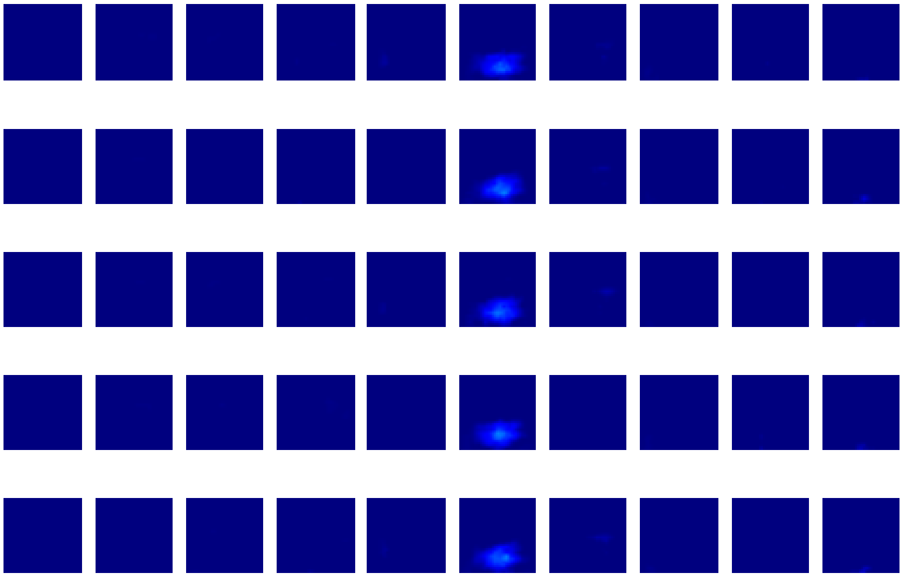




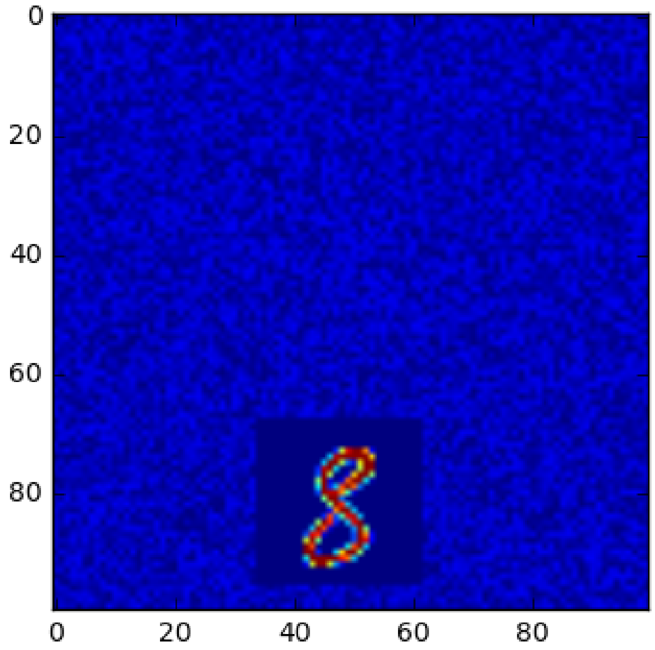
```
In [15]: for _ in range(10):
             simple_model.plot_classes_maps(100, n_CCN)
```

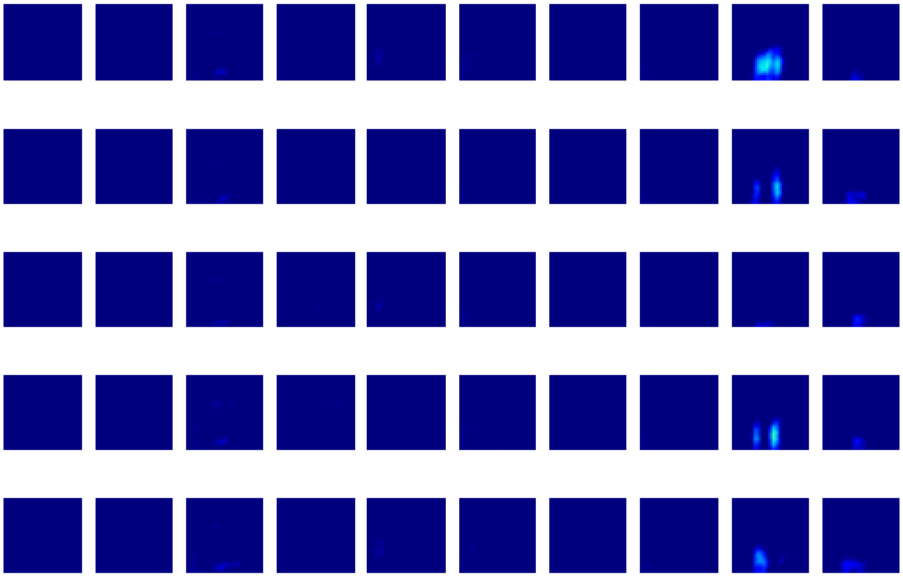
```
5
```



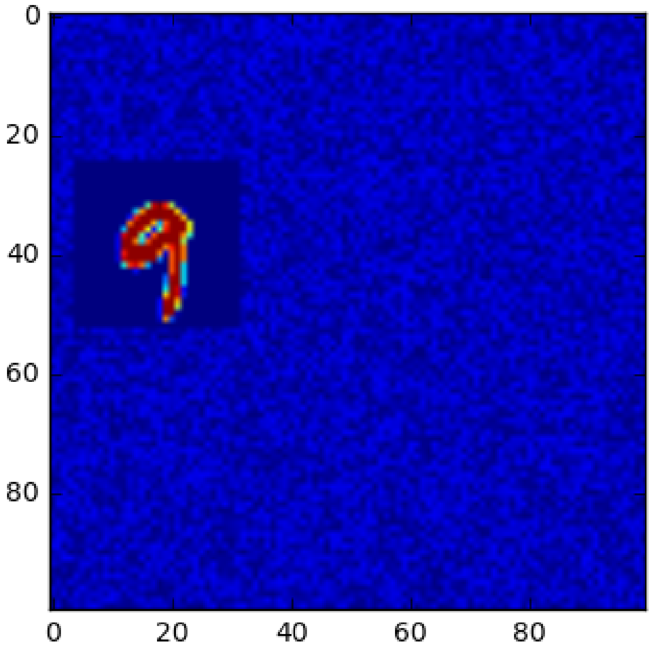


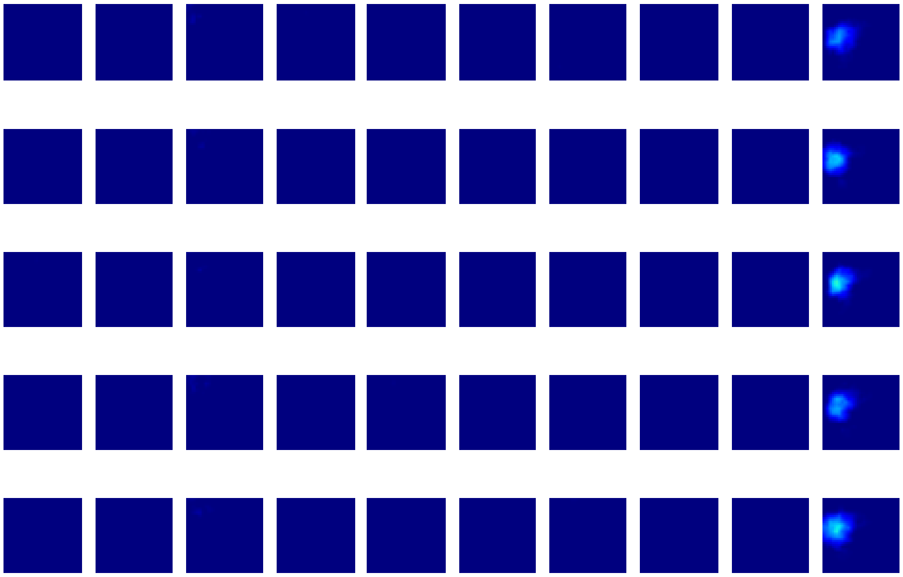
8



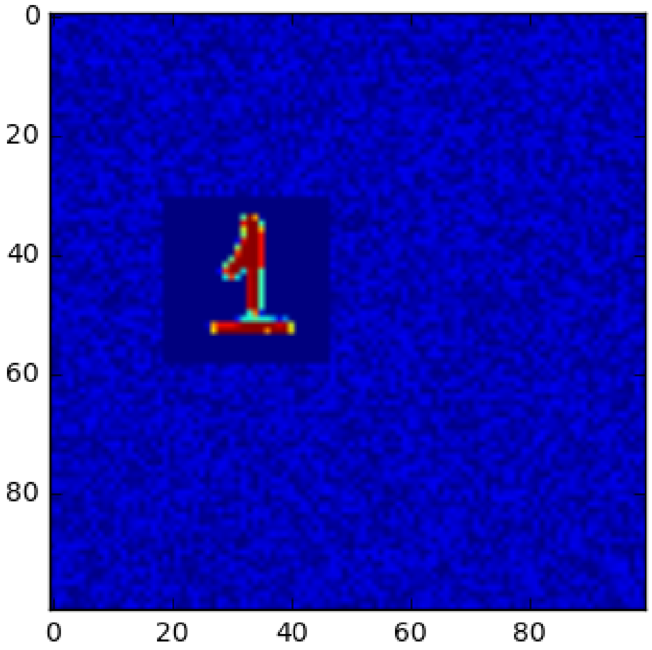


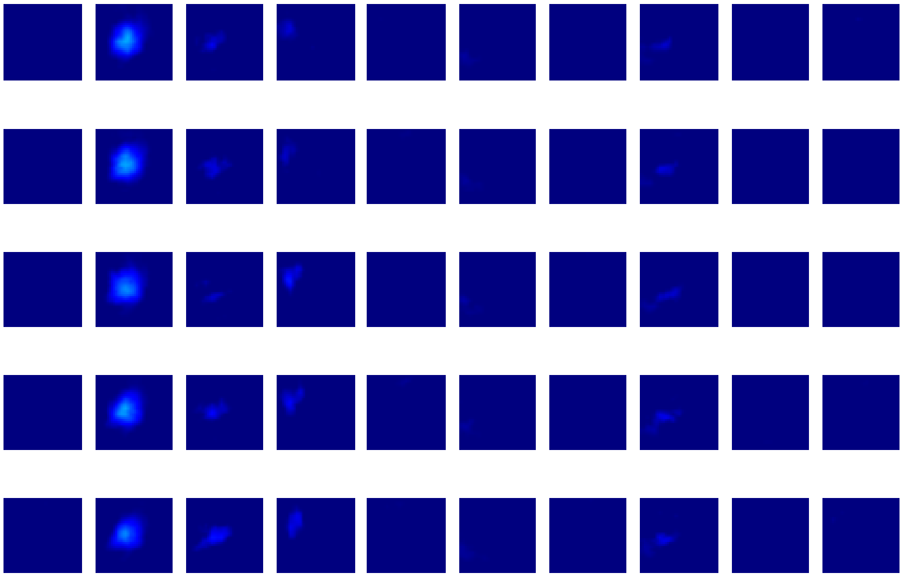
9



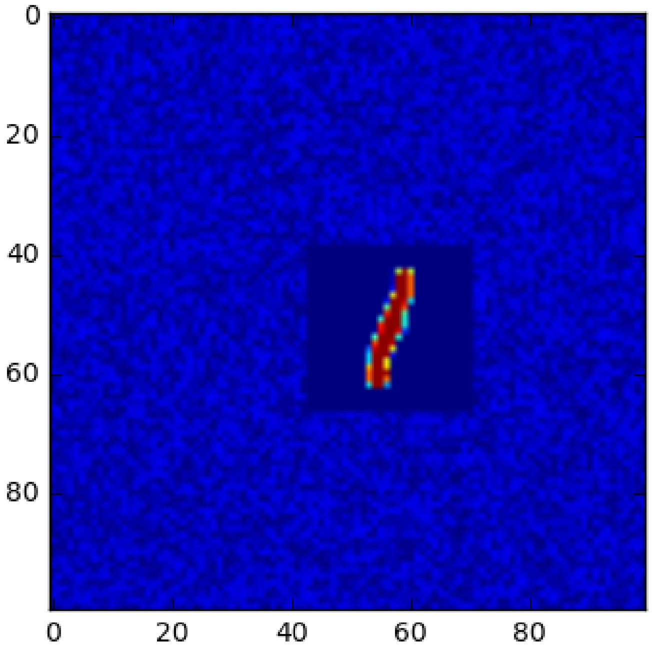


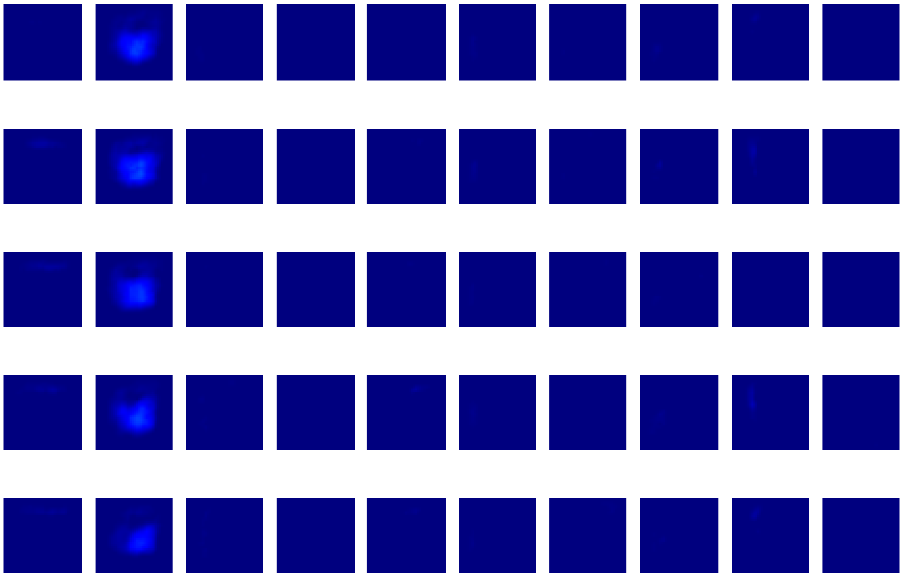
1



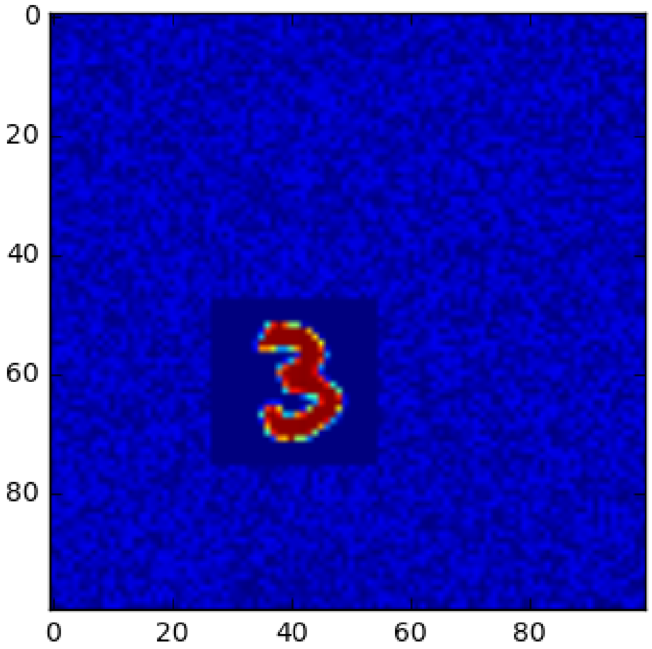


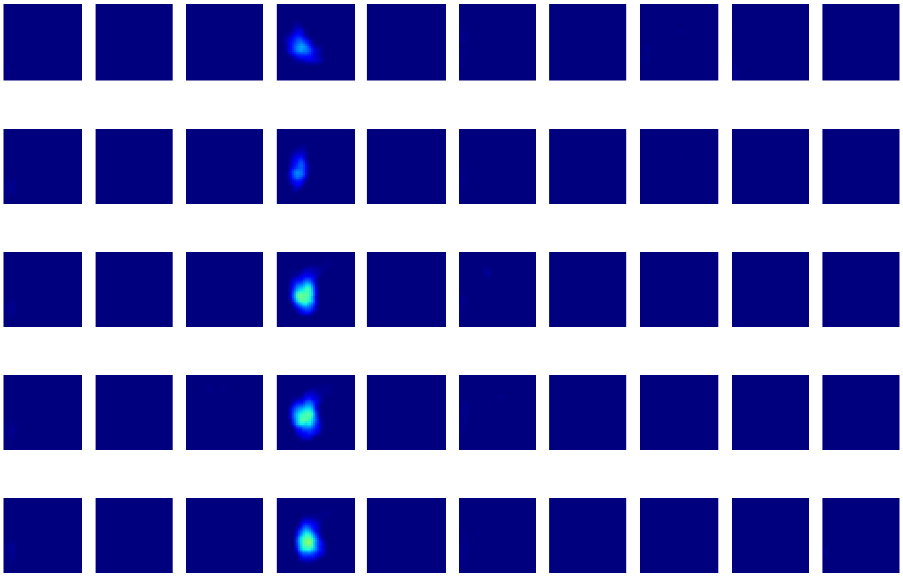
1



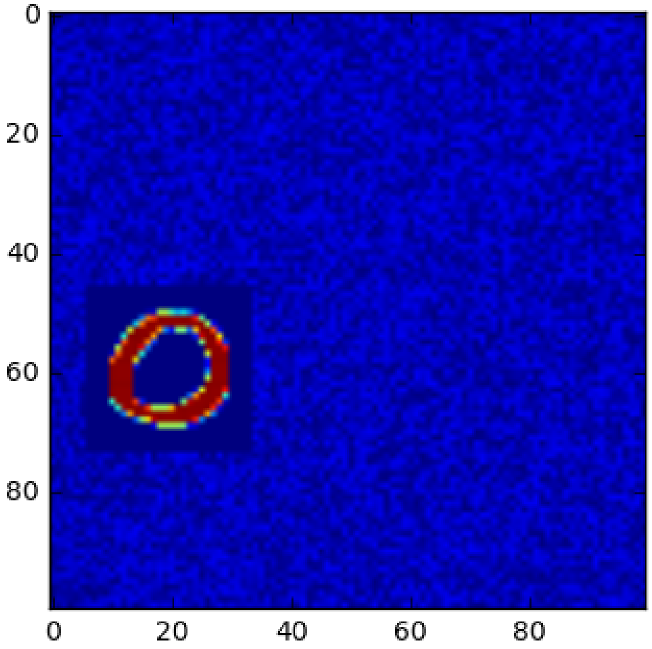


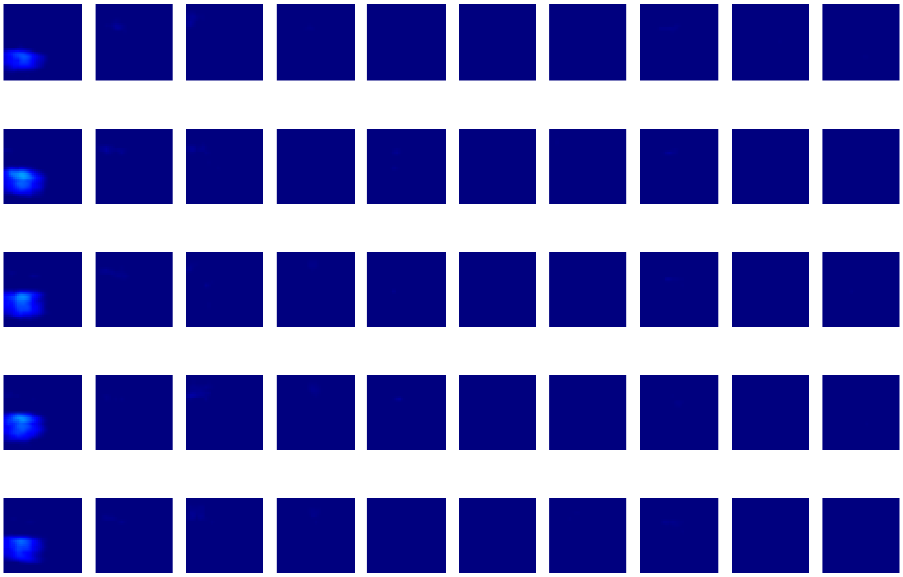
3



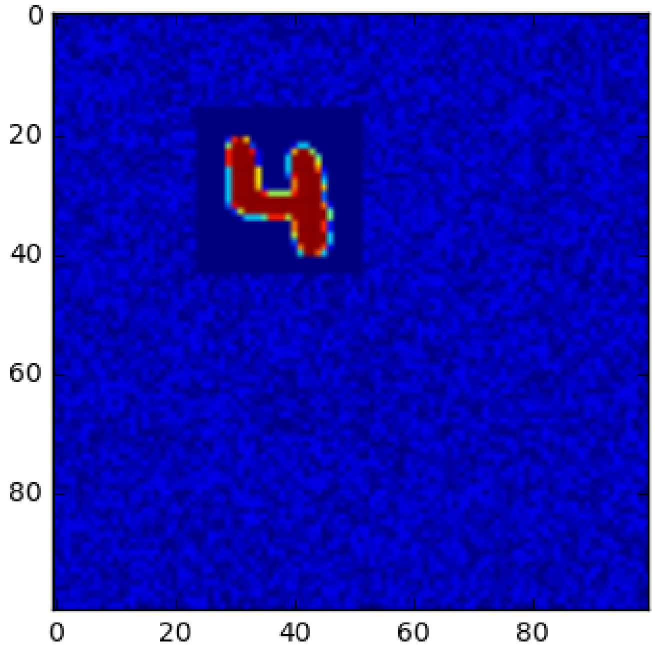


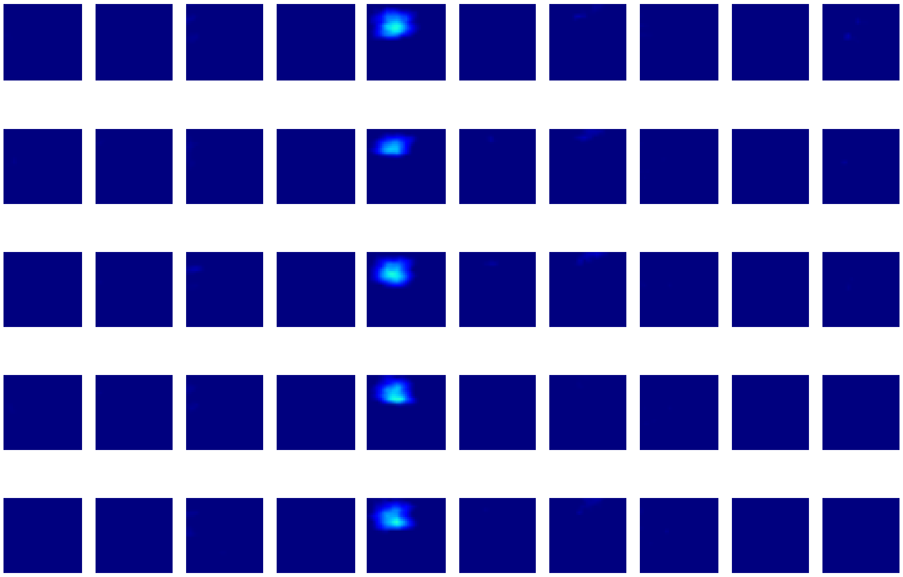
0



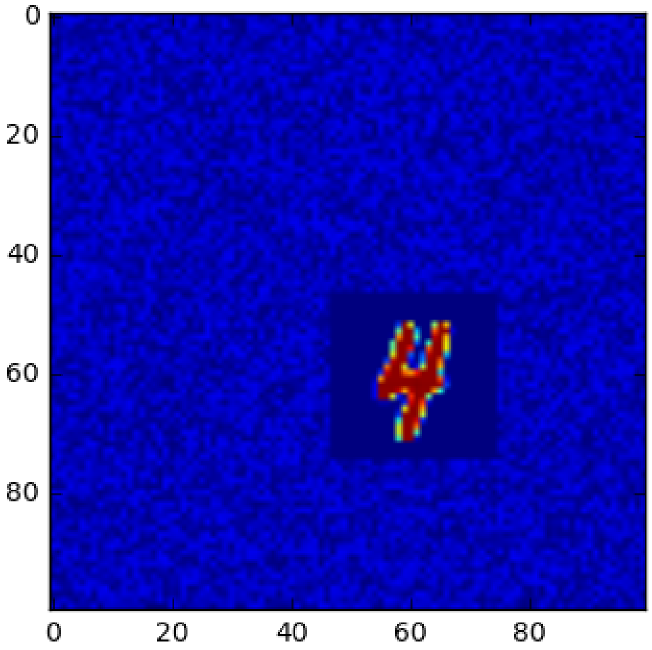


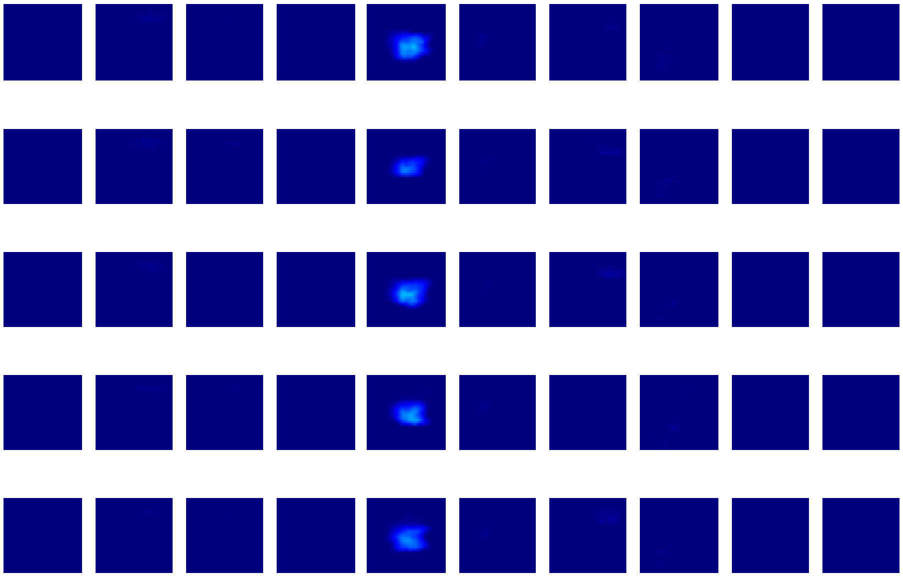
4



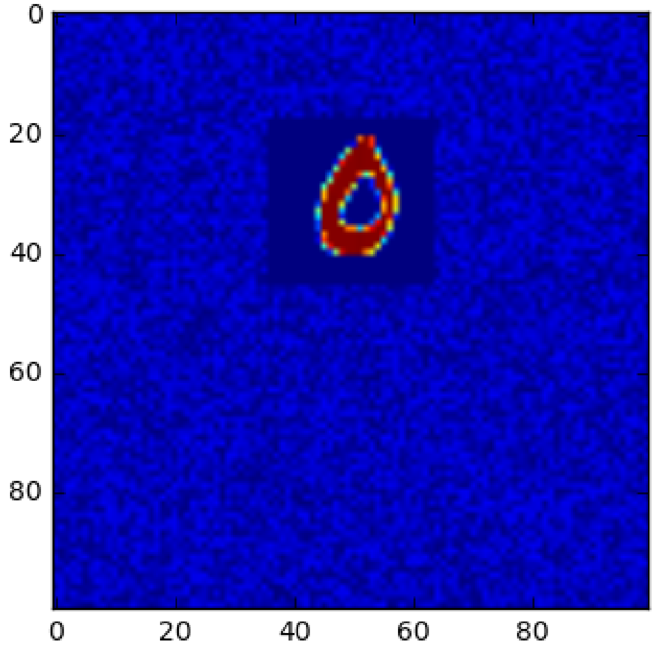


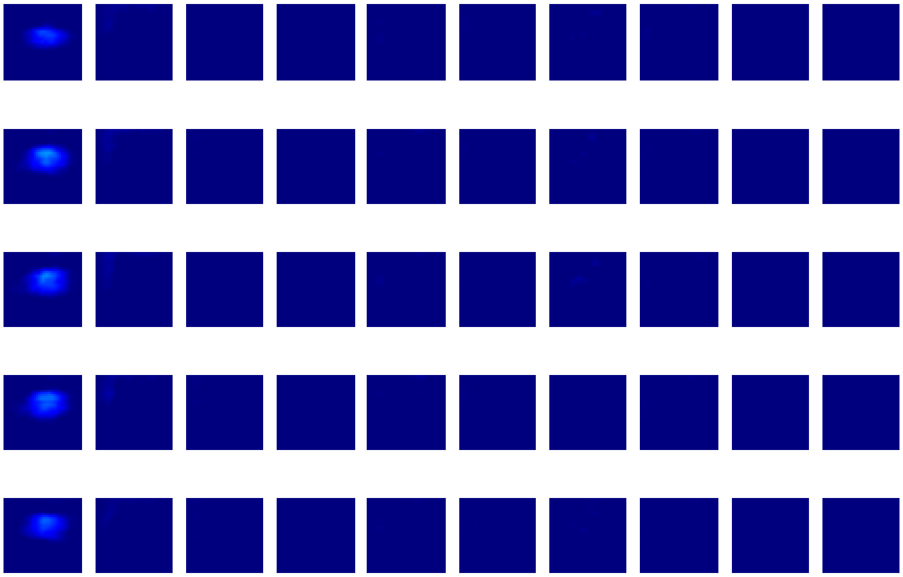
4





0





In [16]:

```
Exception AssertionError: AssertionError("Nesting violated for default stack of <ty
```