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")

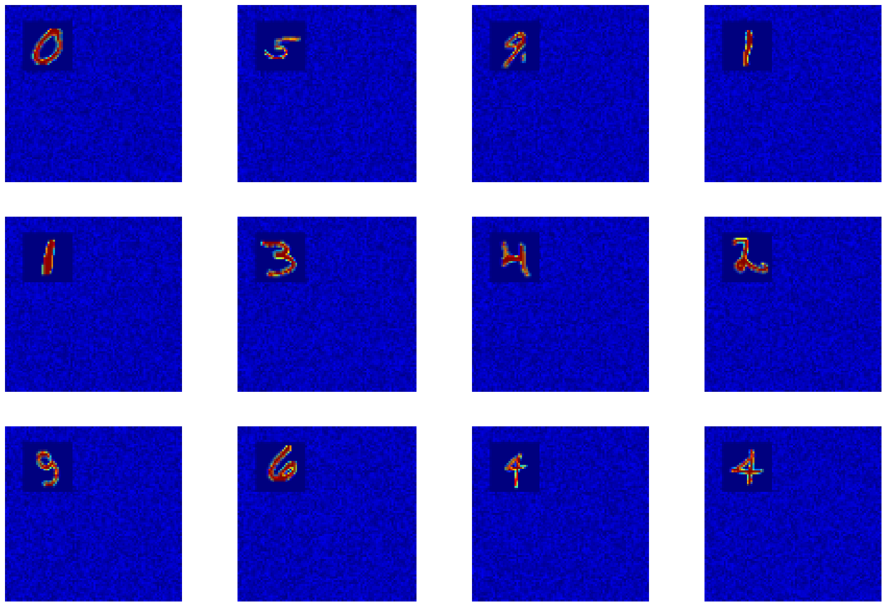
gap = tf.reduce_mean(conv3_2, [1,2]) ccn = tf.reshape(gap,[-1,10,nb_CCN]) ccn = tf.reduce_mean(ccn, 2)

1.2 train params

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

```
In [17]: 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 [18]: """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 [19]: """Perform training"""
         gen = training_generator(lr=lr, lr_decay=lr_decay, back_size=back_size, nc
         for _ in range(25):
             _,accs = gen.next()
             print "max acc so far : "+str(max(accs)*100)
```

```
***** EPOCH 0 ******
1-Loss on testset is 0.677705
1-Accuracy now is 81.10
2-Loss on testset is 0.311588
2-Accuracy now is 91.53
lr now is 0.00045
max acc so far : 81.1

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

```
1-Loss on testset is 0.267934
1-Accuracy now is 92.75
2-Loss on testset is 0.090015
2-Accuracy now is 98.38
lr now is 0.00041
max acc so far : 92.75

***** EPOCH 2 ******
1-Loss on testset is 0.207295
1-Accuracy now is 93.72
2-Loss on testset is 0.075427
2-Accuracy now is 98.81
lr now is 0.00036
max acc so far : 93.72

***** EPOCH 3 ******
1-Loss on testset is 0.124372
1-Accuracy now is 97.41
2-Loss on testset is 0.072415
2-Accuracy now is 98.89
lr now is 0.00033
max acc so far : 97.41

***** EPOCH 4 ******
1-Loss on testset is 0.119845
1-Accuracy now is 97.29
2-Loss on testset is 0.070866
2-Accuracy now is 98.79
lr now is 0.00030
max acc so far : 97.41

***** EPOCH 5 ******
1-Loss on testset is 0.199614
1-Accuracy now is 95.17
2-Loss on testset is 0.079770
2-Accuracy now is 98.66
lr now is 0.00027
max acc so far : 97.41

***** EPOCH 6 ******
1-Loss on testset is 0.092299
1-Accuracy now is 98.16
2-Loss on testset is 0.064588
2-Accuracy now is 99.12
lr now is 0.00024
max acc so far : 98.16

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

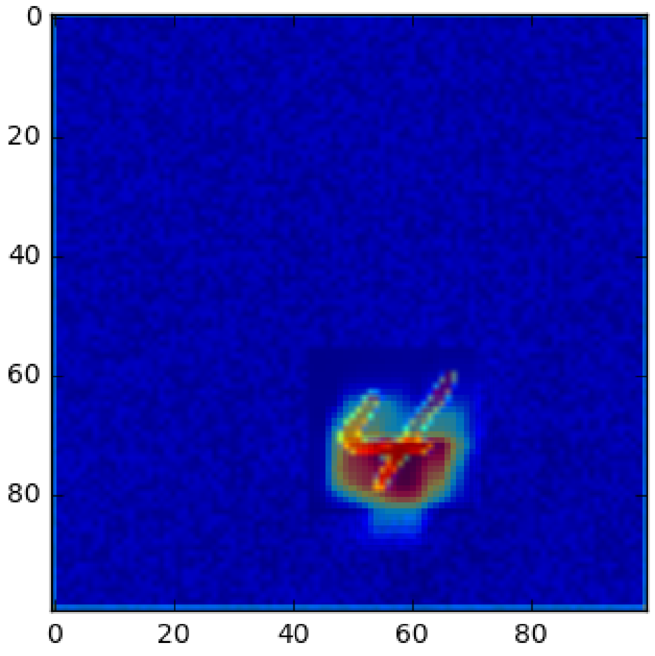
```
1-Loss on testset is 0.140198
1-Accuracy now is 96.22
2-Loss on testset is 0.055726
2-Accuracy now is 99.20
lr now is 0.00022
max acc so far : 98.16

***** EPOCH 8 ******
1-Loss on testset is 0.088816
1-Accuracy now is 97.97
2-Loss on testset is 0.059481
2-Accuracy now is 99.14
lr now is 0.00019
max acc so far : 98.16

***** EPOCH 9 ******
1-Loss on testset is 0.112950
1-Accuracy now is 97.49
2-Loss on testset is 0.057615
2-Accuracy now is 99.10
lr now is 0.00017
max acc so far : 98.16

***** EPOCH 10 ******
1-Loss on testset is 0.133874
1-Accuracy now is 97.07
2-Loss on testset is 0.057338
2-Accuracy now is 99.12
lr now is 0.00016
max acc so far : 98.16

***** EPOCH 11 ******
1-Loss on testset is 0.094960
1-Accuracy now is 98.05
2-Loss on testset is 0.058852
2-Accuracy now is 99.12
lr now is 0.00014
max acc so far : 98.16

***** EPOCH 12 ******
1-Loss on testset is 0.079086
1-Accuracy now is 98.65
2-Loss on testset is 0.065456
2-Accuracy now is 99.04
lr now is 0.00013
max acc so far : 98.65

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

```
1-Loss on testset is 0.122504
1-Accuracy now is 97.29
2-Loss on testset is 0.051278
2-Accuracy now is 99.36
lr now is 0.00011
max acc so far : 98.65

***** EPOCH 14 ******
1-Loss on testset is 0.102289
1-Accuracy now is 97.87
2-Loss on testset is 0.051238
2-Accuracy now is 99.24
lr now is 0.00010
max acc so far : 98.65

***** EPOCH 15 ******
1-Loss on testset is 0.108635
1-Accuracy now is 97.72
2-Loss on testset is 0.049588
2-Accuracy now is 99.29
lr now is 0.00009
max acc so far : 98.65

***** EPOCH 16 ******
1-Loss on testset is 0.107123
1-Accuracy now is 97.46
2-Loss on testset is 0.048646
2-Accuracy now is 99.31
lr now is 0.00008
max acc so far : 98.65

***** EPOCH 17 ******
1-Loss on testset is 0.086688
1-Accuracy now is 98.30
2-Loss on testset is 0.049181
2-Accuracy now is 99.34
lr now is 0.00008
max acc so far : 98.65

***** EPOCH 18 ******
1-Loss on testset is 0.121276
1-Accuracy now is 97.03
2-Loss on testset is 0.049449
2-Accuracy now is 99.33
lr now is 0.00007
max acc so far : 98.65

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

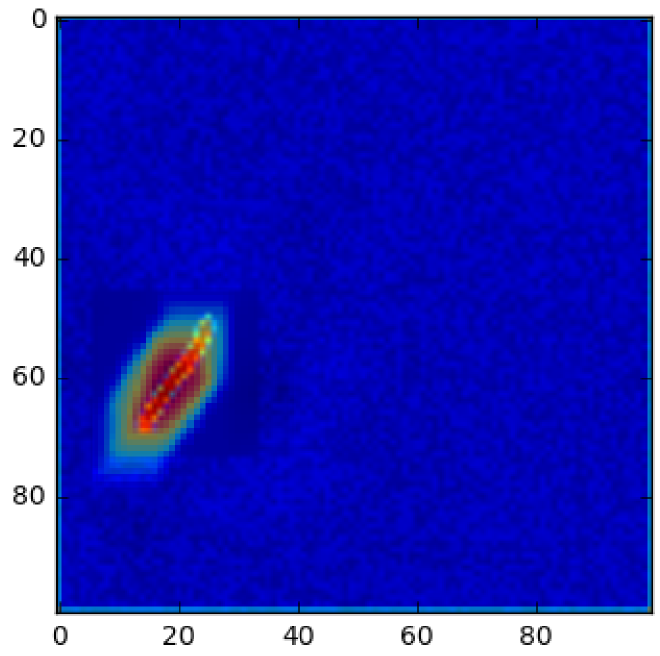
```
1-Loss on testset is 0.067110
1-Accuracy now is 98.90
2-Loss on testset is 0.047295
2-Accuracy now is 99.32
lr now is 0.00006
max acc so far : 98.9

***** EPOCH 20 ******
1-Loss on testset is 0.079866
1-Accuracy now is 98.40
2-Loss on testset is 0.049092
2-Accuracy now is 99.30
lr now is 0.00005
max acc so far : 98.9

***** EPOCH 21 ******
1-Loss on testset is 0.088322
1-Accuracy now is 98.30
2-Loss on testset is 0.046980
2-Accuracy now is 99.34
lr now is 0.00005
max acc so far : 98.9

***** EPOCH 22 ******
1-Loss on testset is 0.087880
1-Accuracy now is 98.16
2-Loss on testset is 0.046492
2-Accuracy now is 99.32
lr now is 0.00004
max acc so far : 98.9

***** EPOCH 23 ******
1-Loss on testset is 0.104565
1-Accuracy now is 97.83
2-Loss on testset is 0.047745
2-Accuracy now is 99.34
lr now is 0.00004
max acc so far : 98.9

***** EPOCH 24 ******
1-Loss on testset is 0.098388
1-Accuracy now is 98.10
2-Loss on testset is 0.050408
2-Accuracy now is 99.31
lr now is 0.00004
max acc so far : 98.9
```

```
In [20]: 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])
```

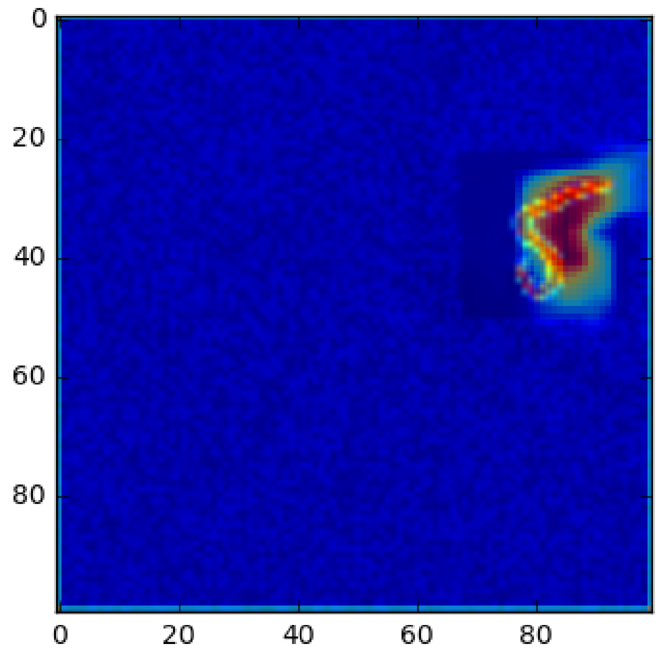
```
--------------------------------------------------------------------------------
prediction is : 4 with 45.299
```



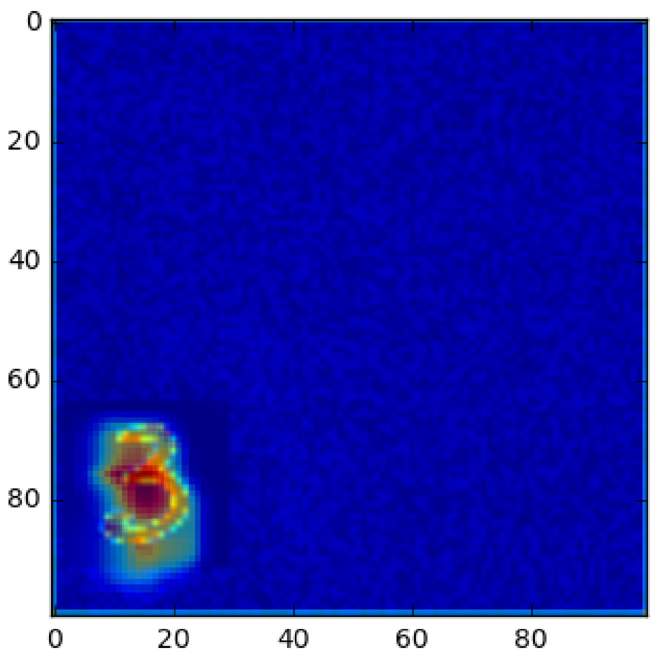
```
--------------------------------------------------------------------------------
prediction is : 1 with 21.400
```



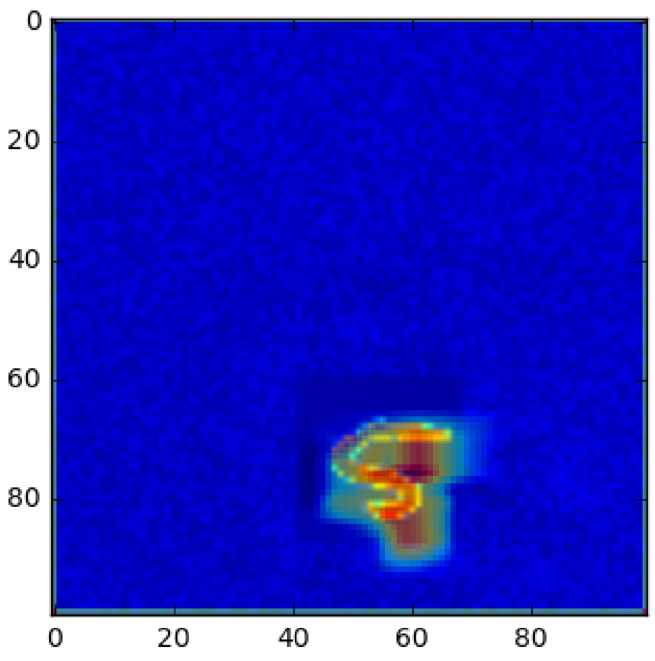
```
--------------------------------------------------------------------------------
prediction is : 5 with 19.076
```



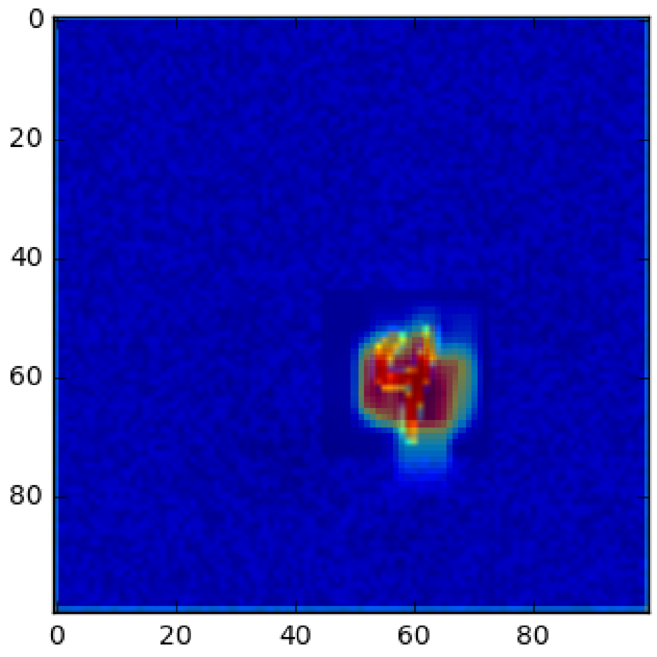
```
--------------------------------------------------------------------------------
prediction is : 3 with 28.649
```



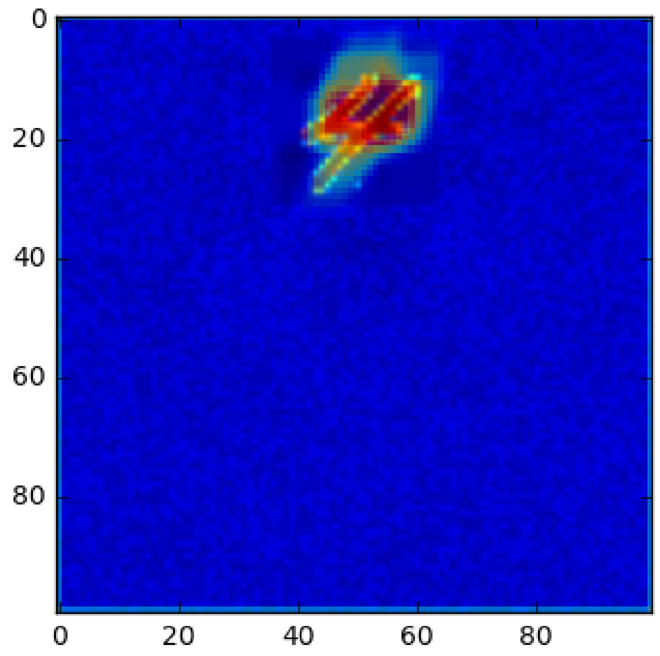
```
--------------------------------------------------------------------------------
prediction is : 5 with 28.657
```



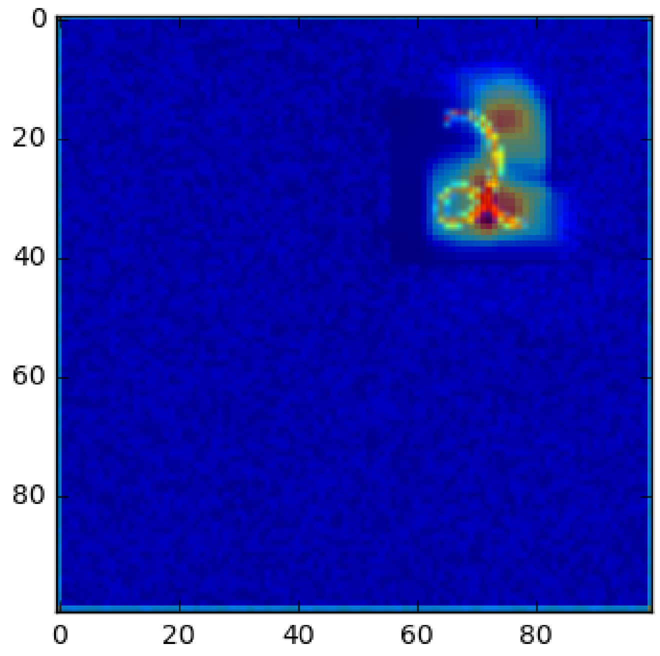
```
--------------------------------------------------------------------------------
prediction is : 4 with 37.223
```



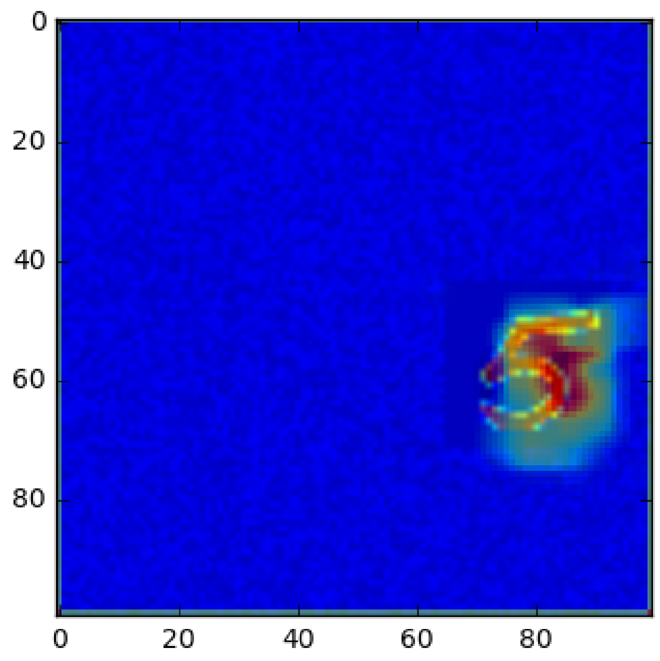
```
--------------------------------------------------------------------------------
prediction is : 4 with 23.949
```



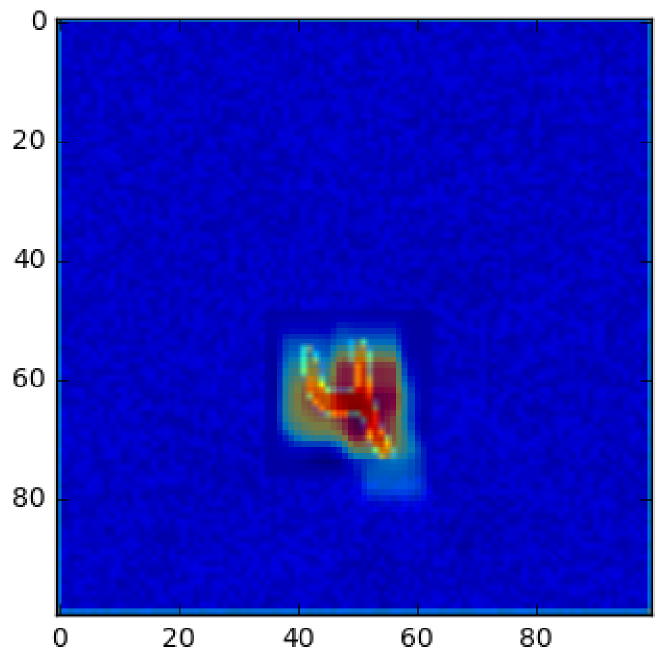
```
--------------------------------------------------------------------------------
prediction is : 2 with 19.709
```



```
--------------------------------------------------------------------------------
prediction is : 5 with 27.889
```



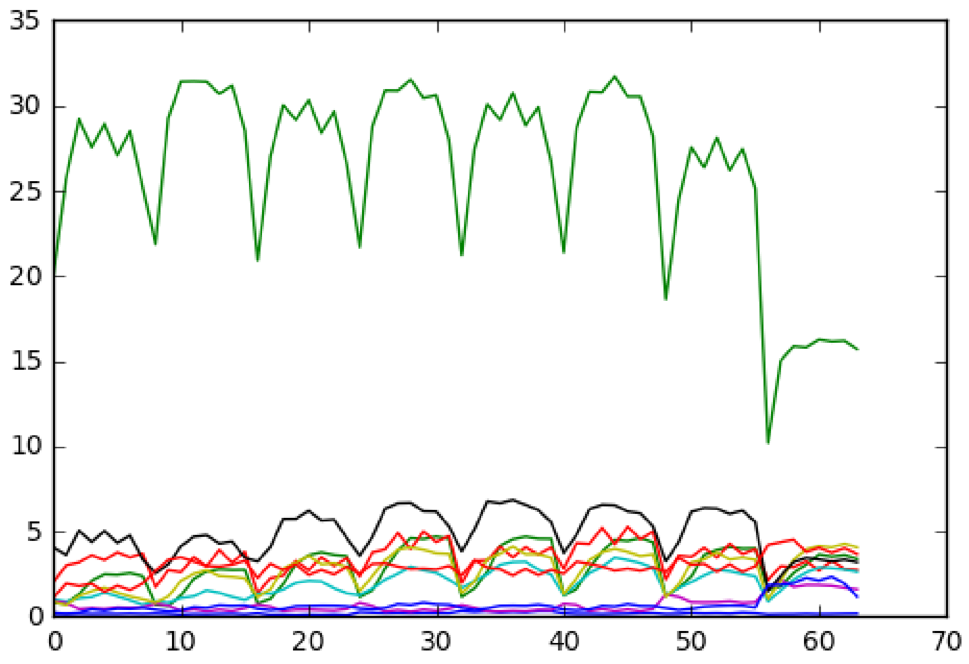
```
--------------------------------------------------------------------------------
prediction is : 4 with 33.453
```

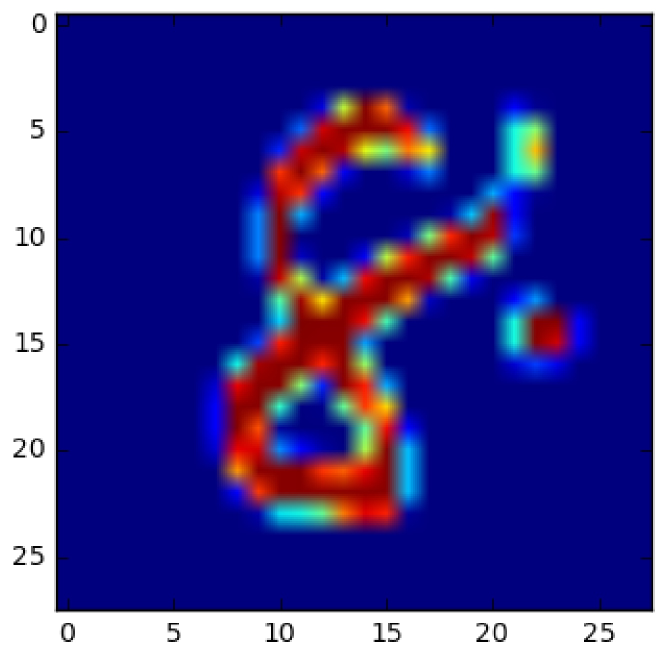


```
In [21]: accuracies = []
         for _ in range(100):
             digit,lbl   = utils.get_batch('test', 1, .1).next()
             digit       = digit[0].reshape(28,28)
             img         = np.random.random((100,100))*.1
             height      = img.shape[0]
             width       = img.shape[1]
             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)

0.983125
```

```
In [22]: 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()
```

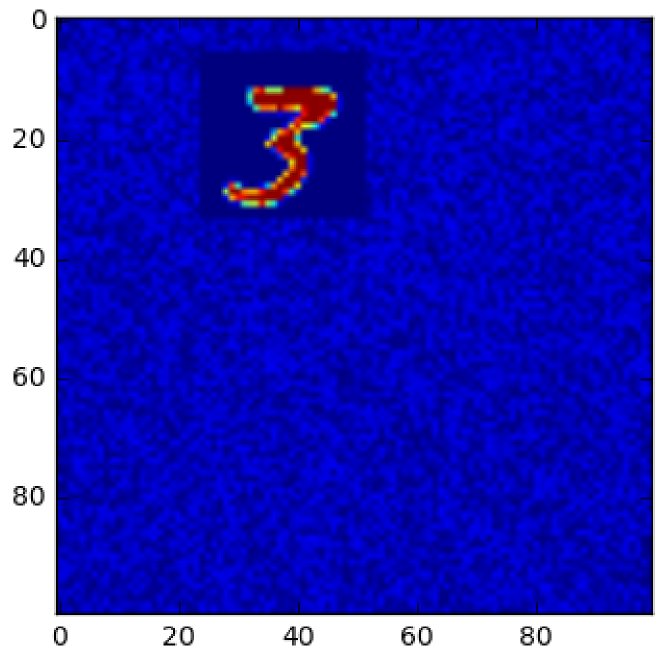
```
[8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8]
100.0
```

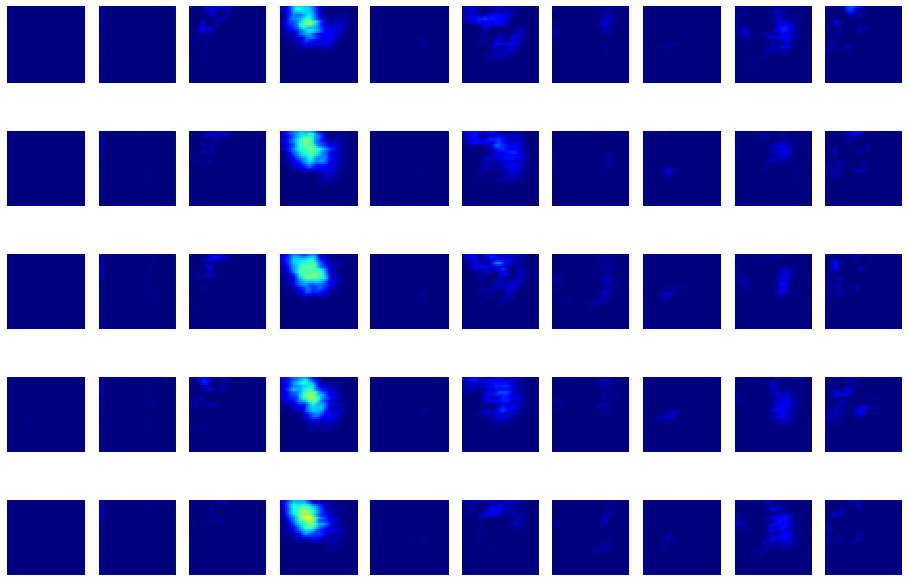




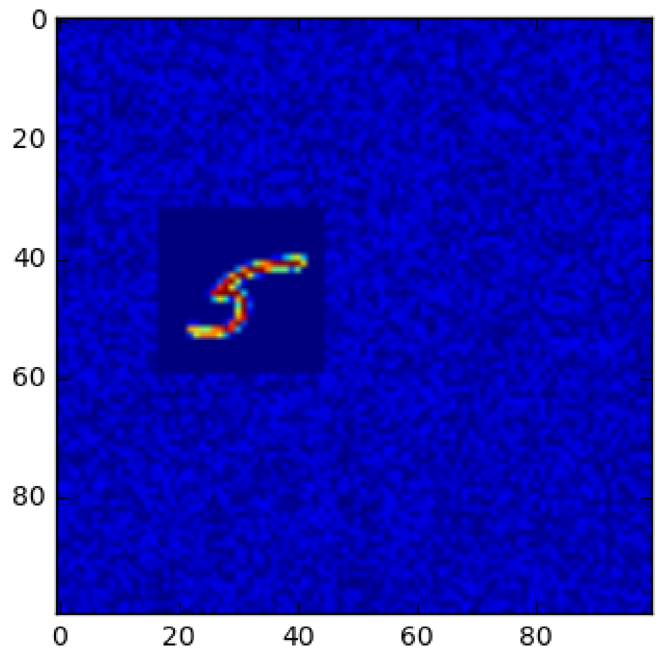
```
In [23]: for _ in range(10):
             simple_model.plot_classes_maps(100, n_CCN)
```

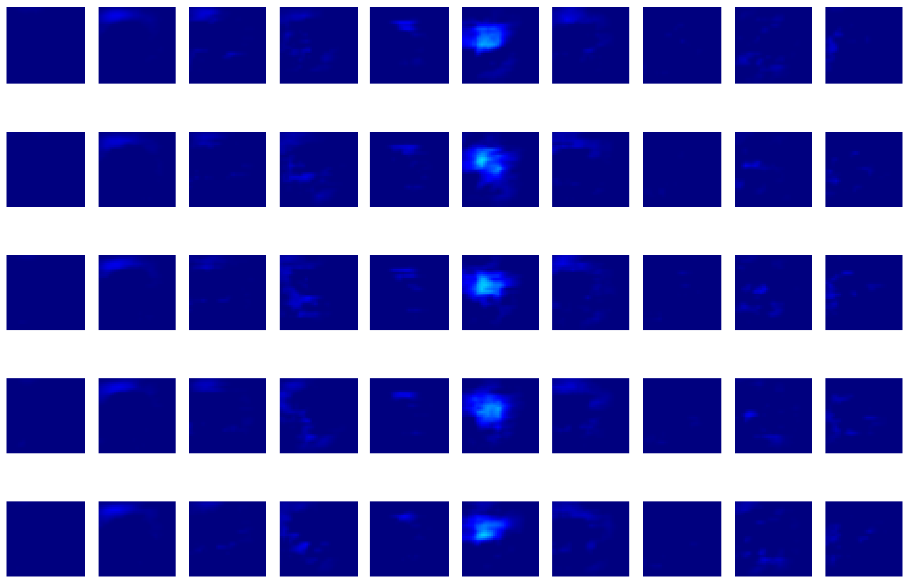
```
3
```



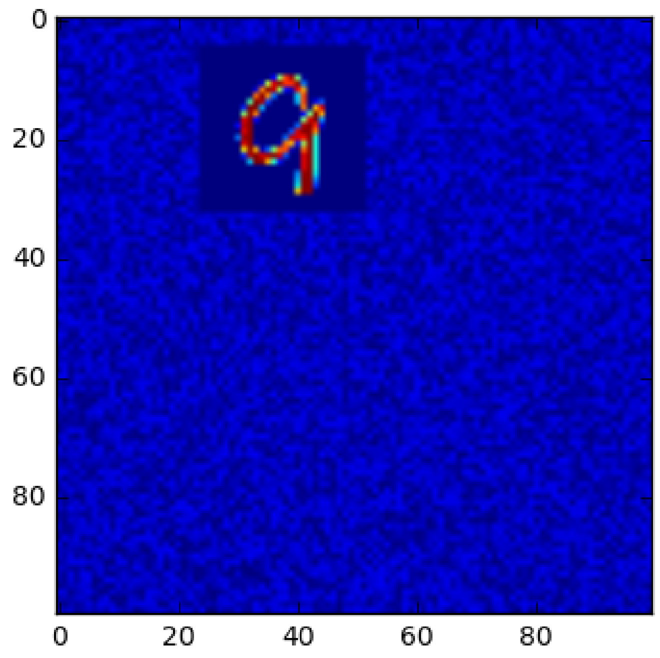


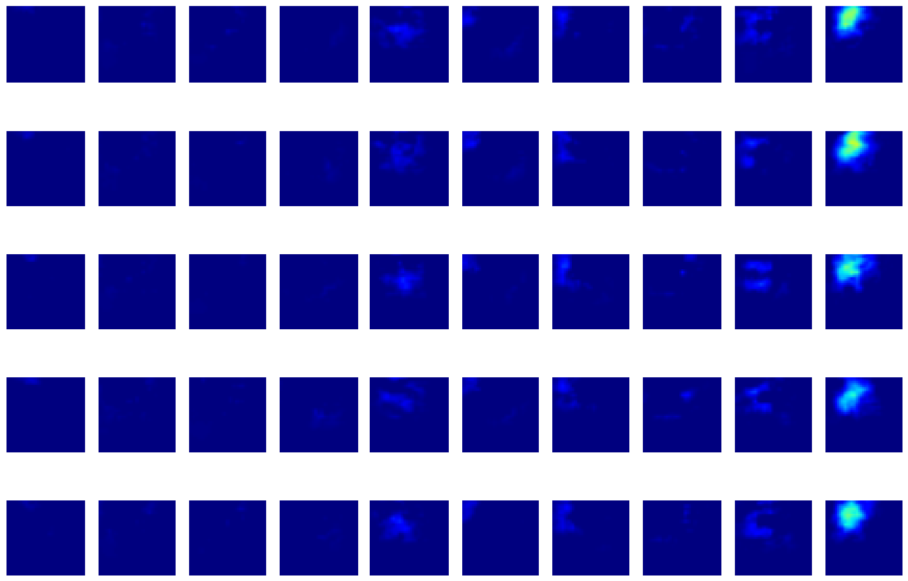
5



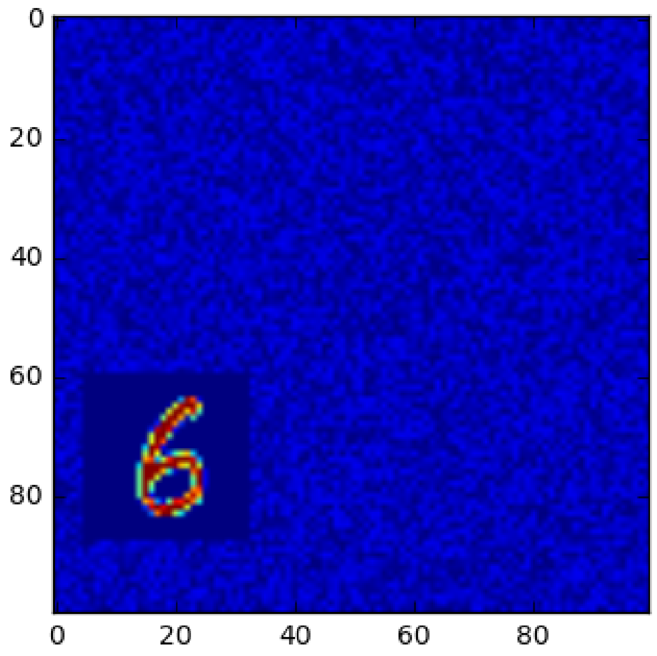


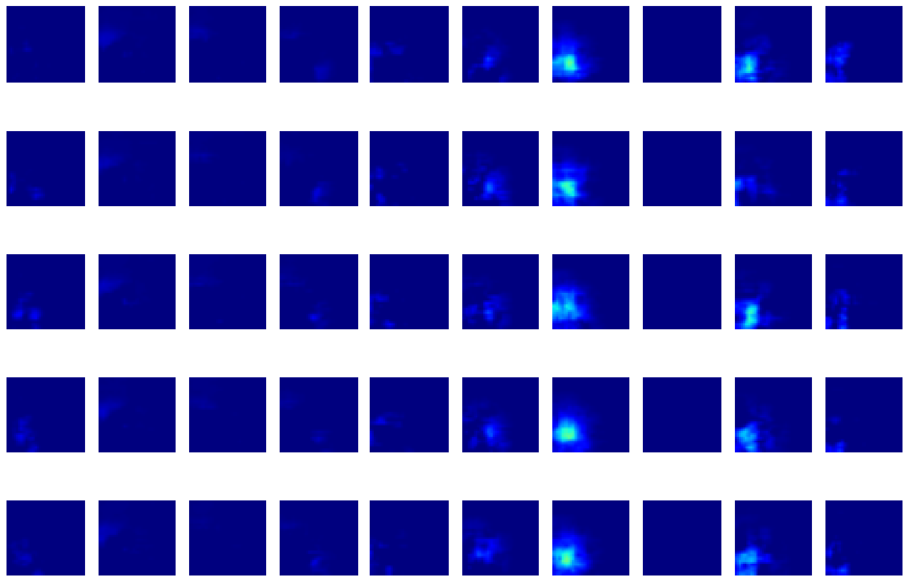
9



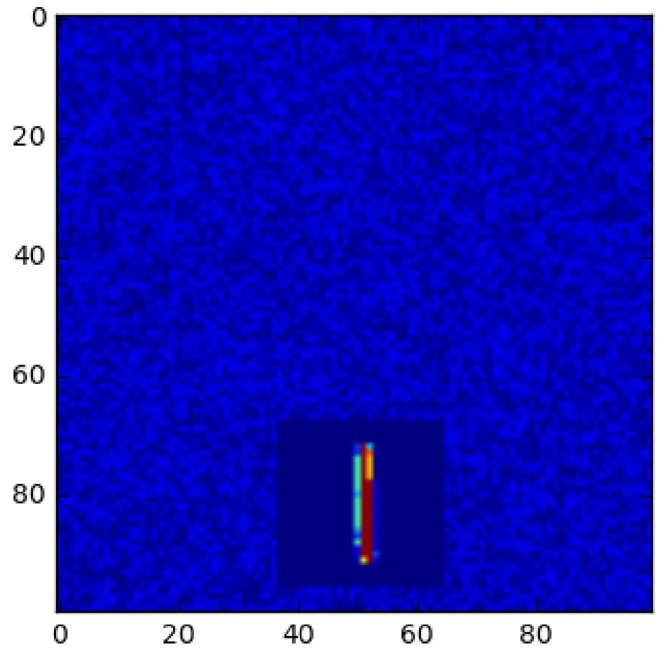


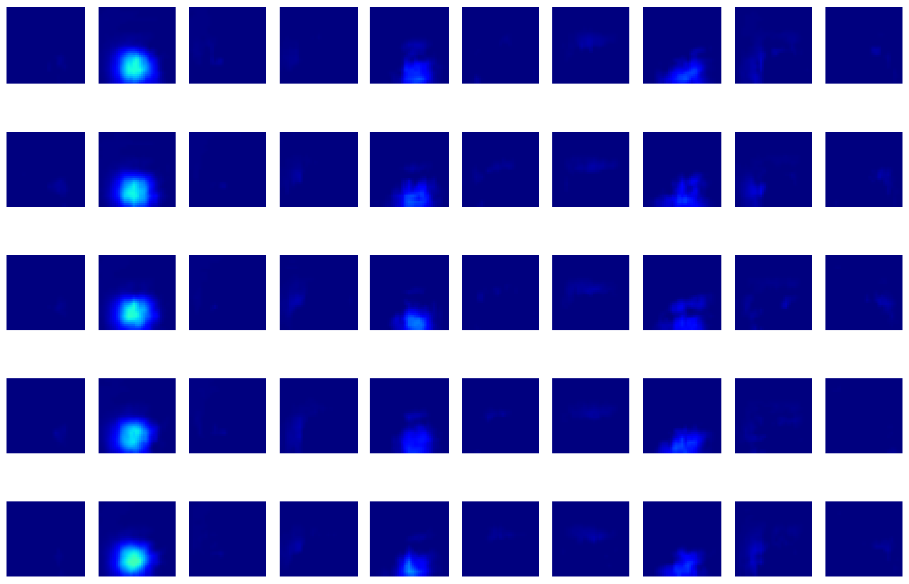
6



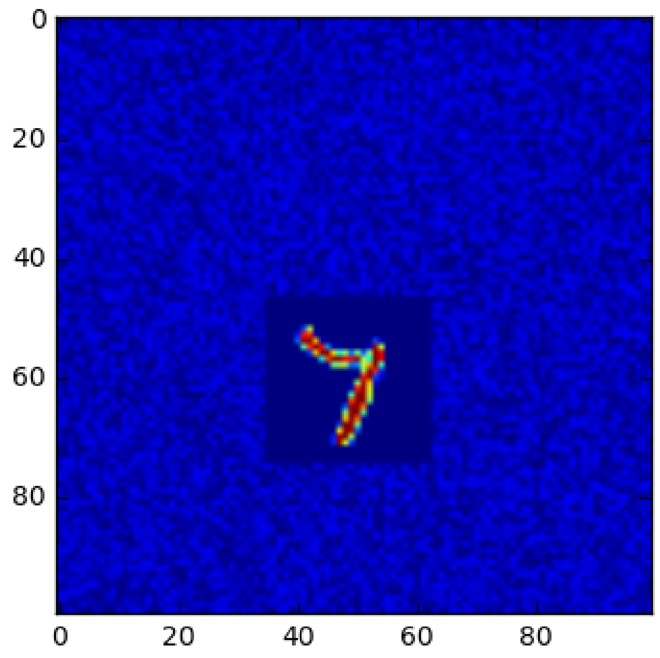


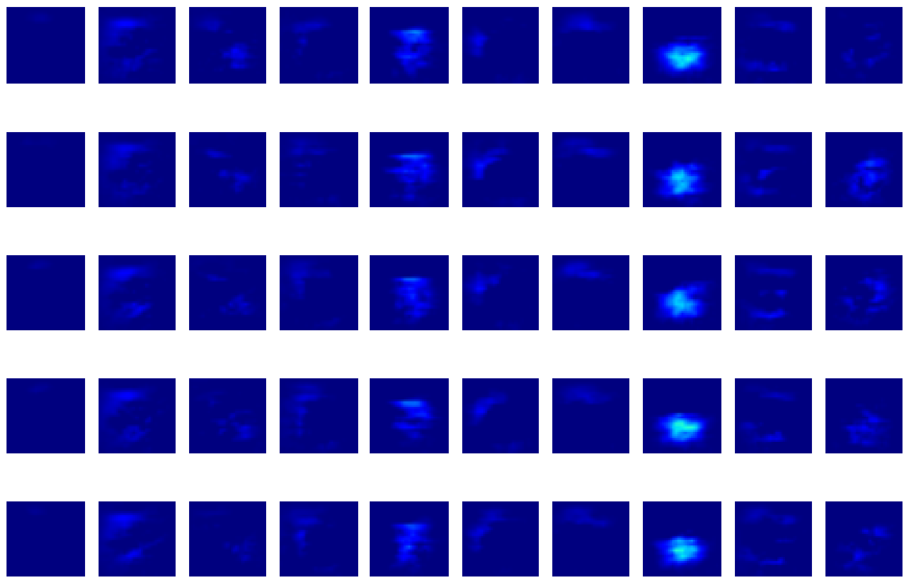
1



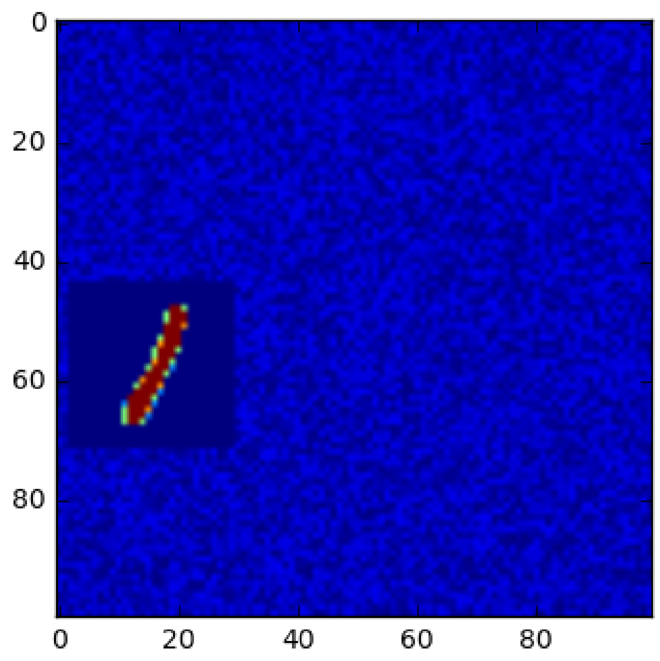


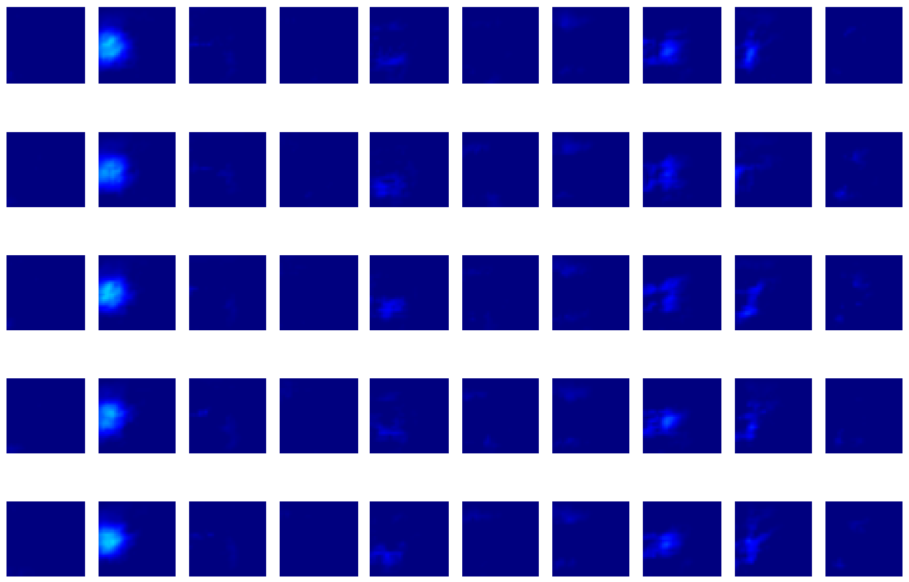
7



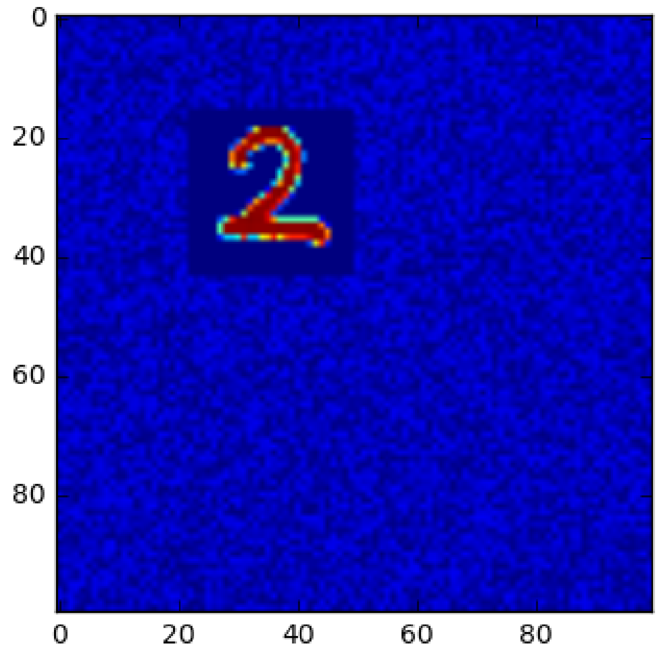


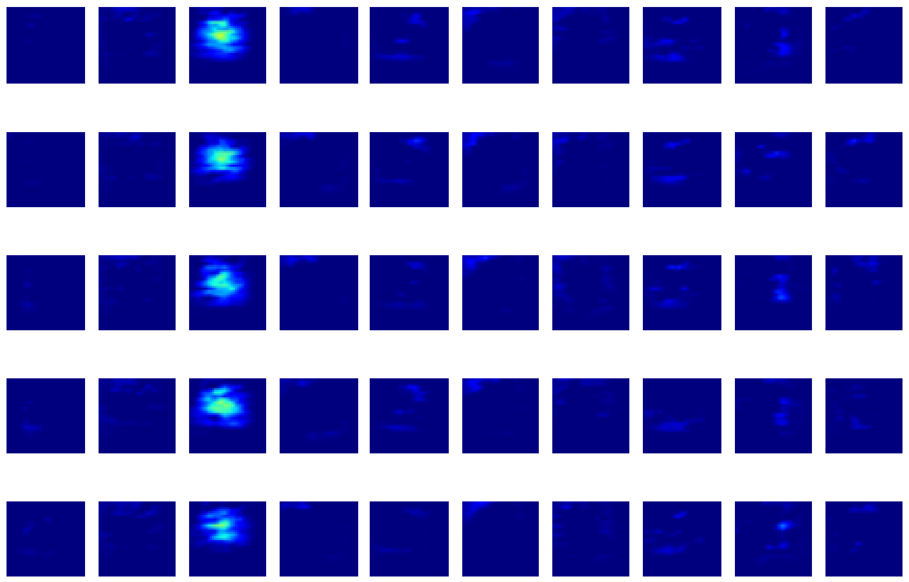
1



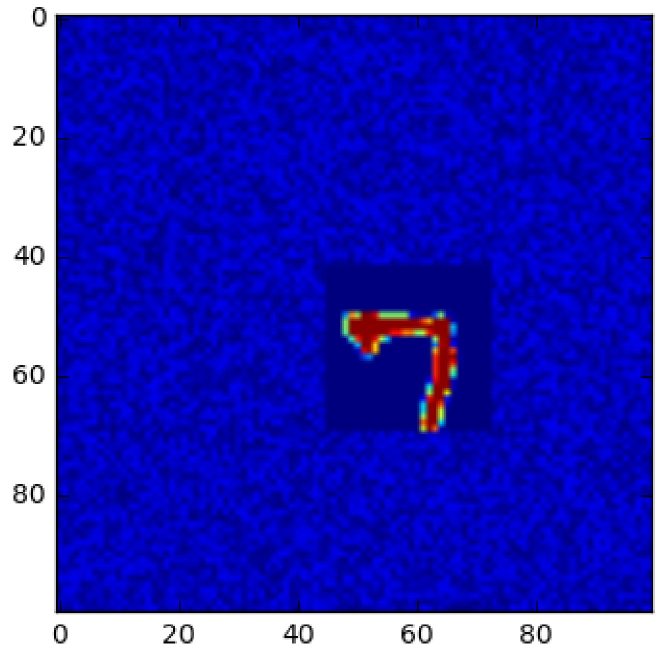


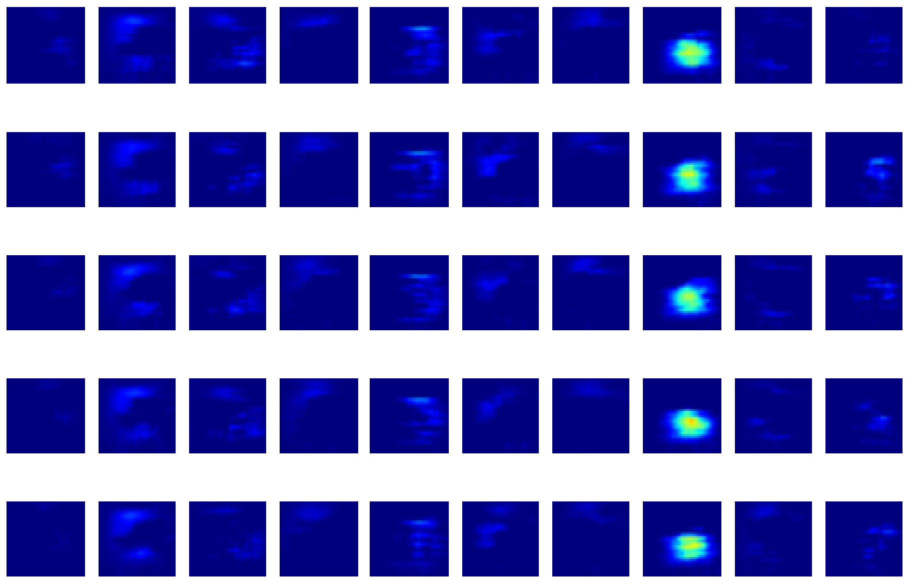
2



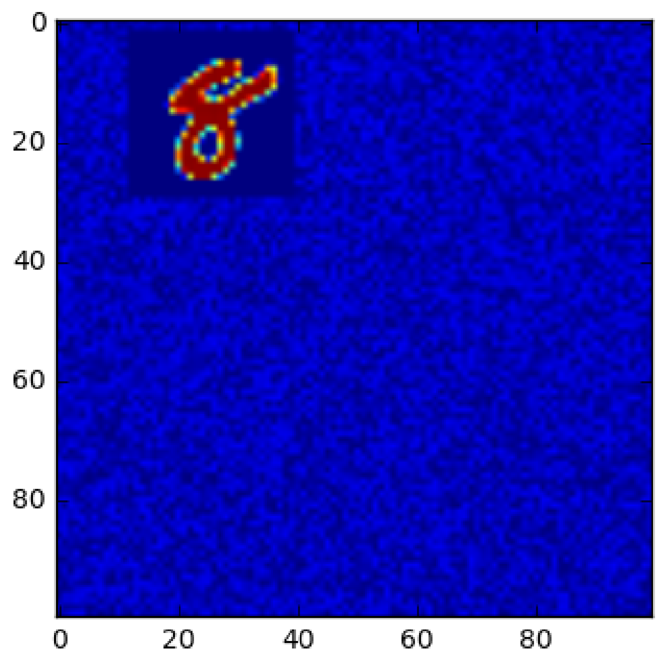


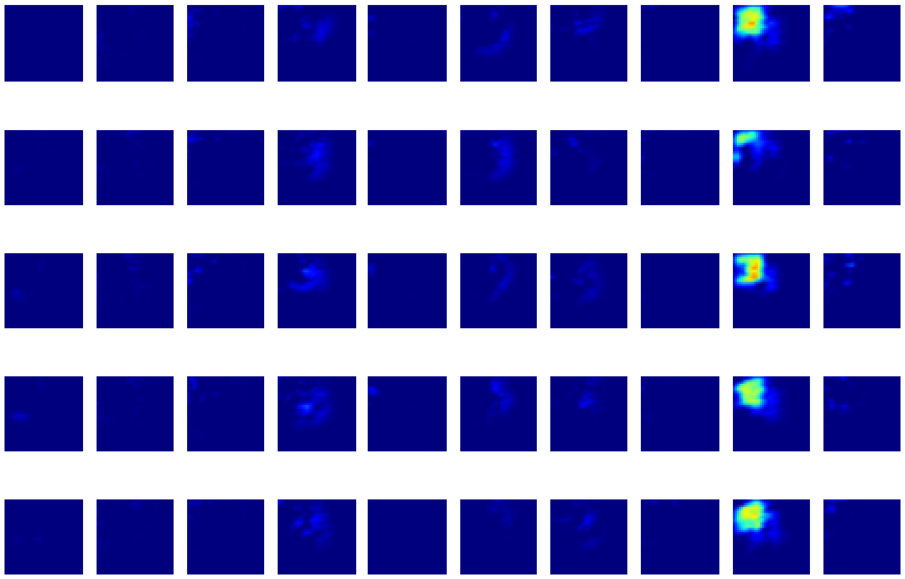
7





8





```
In [ ]:
```