14 - Visualizing CNN

Ludwig Krippahl

Visualizing CNN

Summary

- How to understand convolutional networks:
  - Occlusion maps
  - Saliency maps
- Note: simplified versions; there are many variants
- Next sessions and test (May 10)

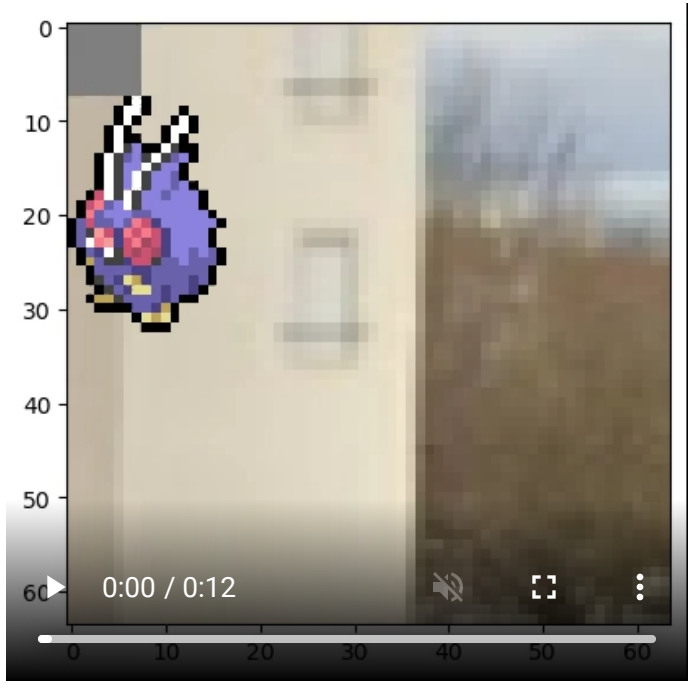
Occlusion maps

Occlusion maps

- How important is a part of the image for classification?
- We can check by covering it with a mask, and then measuring the effect on classification

Occlusion maps

- How important is a part of the image for classification?
- We can check by covering it with a mask, and then measuring the effect on classification



Occlusion maps

- Iterating occlusions over the image:

```
def iterate_occlusion(image, size=8):
    for row in range(0, image.shape[1]-size+1, size):
        for col in range(0, image.shape[2]-size+1, size):
            tmp = image.copy()
            tmp[0,row:row+size,col:col+size,:]=(0.5,0.5,0.5)
            yield row,col, tmp
```

- Computing the occlusion map:

```
def occlusion_map(image,model,true_class,size = 8):
   occ_map = np.zeros(image.shape[1:-1])
   for row,col, occluded in iterate_occlusion(image,size):
       pred = model.predict(occluded)
       print(row,col, pred[0,true_class])
       occ_map[row:row+size,col:col+size] = pred[0,true_class]
   return occ_map
```

Occlusion maps

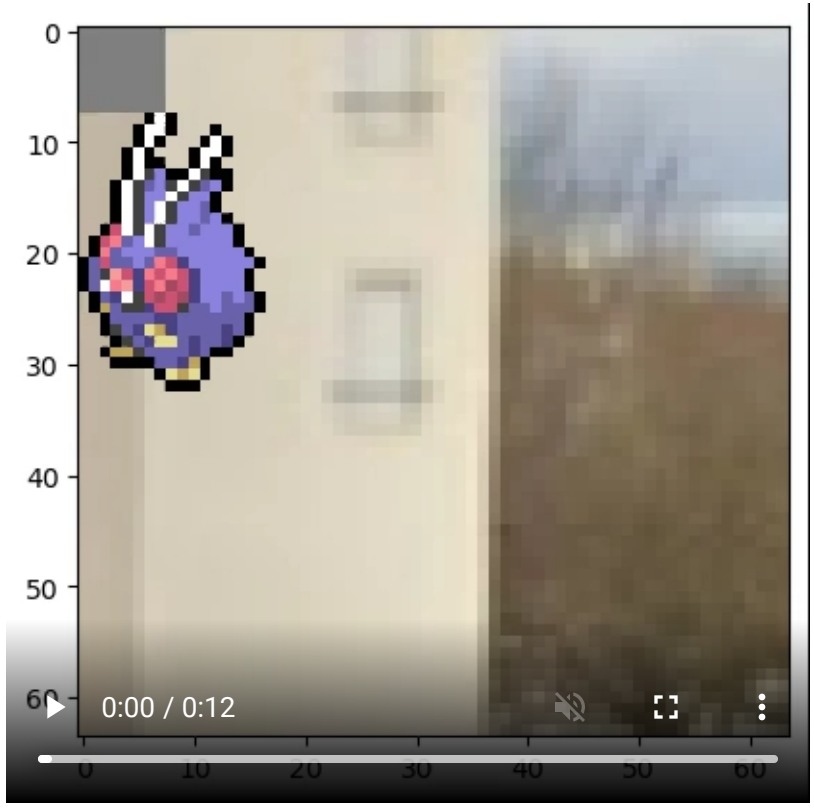
- Representing the map:

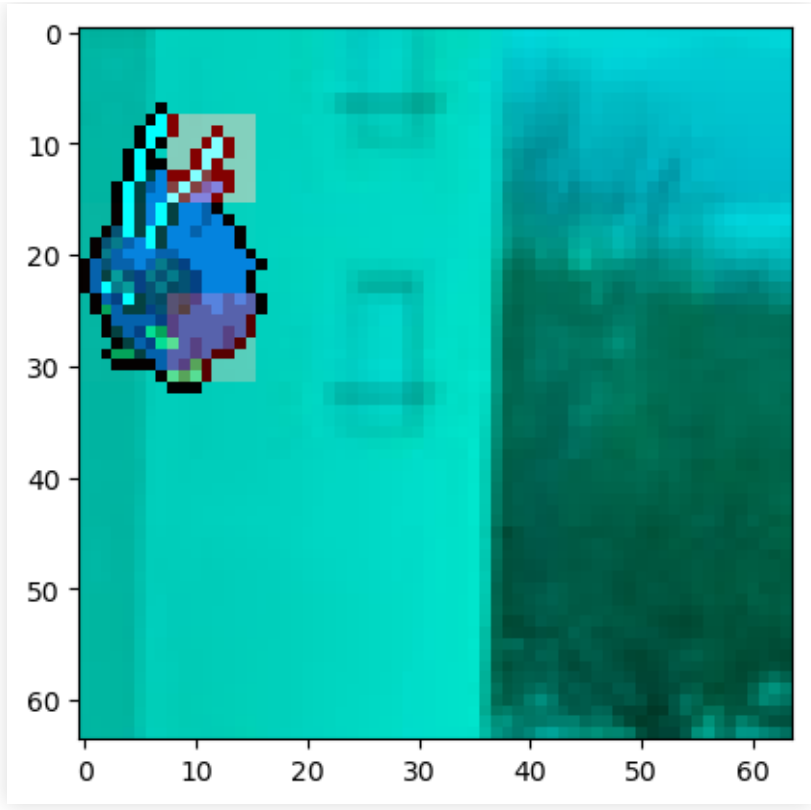
```
model = keras.models.load_model('multiclass.model')
ds = load_data()

img_ix = 2
img = ds['train_X'][img_ix:img_ix+1]
class_ix = np.argmax(ds['train_classes'][img_ix])

occ_map = occlusion_map(img,model,class_ix,size=8)
tmp = img[0].copy()
tmp[:,:,0] = 1-occ_map
plt.imshow(tmp)
```

Occlusion maps





Saliency maps

Saliency maps

Saliency Map

- The derivative of the classification w.r.t each pixel in the input
- The larger the derivative the greater the dependency of the output on this input value
- How to comput:

- Use the `GradientTape` class

Saliency maps

(Based on Stackoverflow answer by Daniel Lang)

```
def saliency_map(model, image, class_idx):
    image_tensor = tf.constant(image)
    with tf.GradientTape() as tape:
        tape.watch(image_tensor)
        predictions = model(image_tensor)
        loss = predictions[:, class_idx]
    gradient = tape.gradient(loss, image_tensor)
    gradient = tf.reduce_max(tf.math.abs(gradient), axis=-1)
    gradient = gradient.numpy()
    min_val, max_val = np.min(gradient), np.max(gradient)
    smap = (gradient - min_val) / (max_val - min_val + keras.backend.epsilon())
    return smap
```

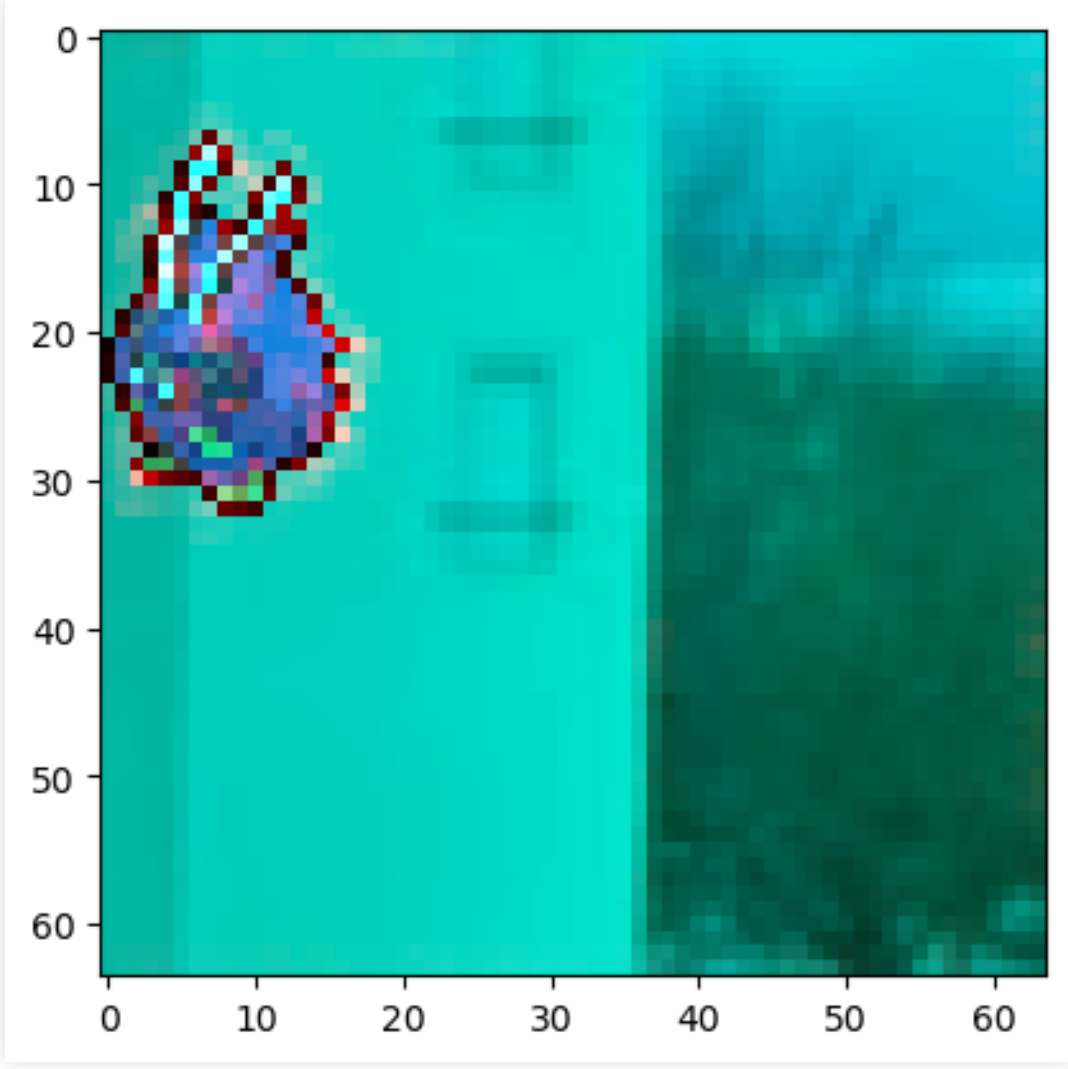
- Options to consider:

- Do we want to rescale or compare different images?
- Do we want the absolute value of the gradient?

Saliency maps

```
model = keras.models.load_model('multiclass.model')
ds = load_data()
img_ix = 2
img = ds['train_X'][img_ix:img_ix+1]
class_ix = np.argmax(ds['train_classes'][img_ix])
smap = saliency_map(model,img,class_ix)
tmp = img[0].copy()
tmp[:,:,0] = smap
plt.imshow(tmp)
```

Saliency maps



First test

First test

This week:

- Tutorials are for questions about the assignment
- Submit the assignment by May 2

- (plus 48h, but should be for solving submission problems)

Next week:

- I will update the lecture notes and include revision exercises
- Next week is for revisions and questions

First test

First test:

- 5 questions, 4/20 each
- 10 minutes per question, 5 minutes break
- Questions are for explaining things

- Structured and consise answers
- Answer with your own words. Do not copy text from other sources
- Note which topics are more important for each question

- Subjects covered until today

Summary

Visualizing CNN

Summary

- Simple examples of two visualization methods for image classification:
  - Occlusion maps
  - Saliency maps
- First test and assignment
- Next week: revisions