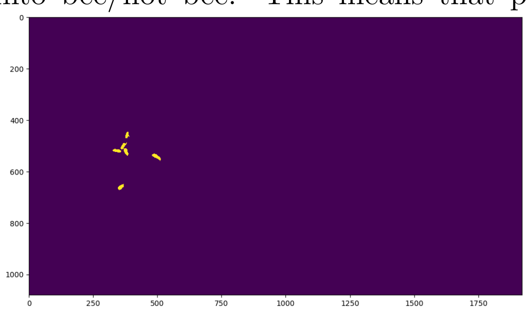
Notes on performance of various networks on bee problem

Tommy

February 8, 2019

1 Introduction

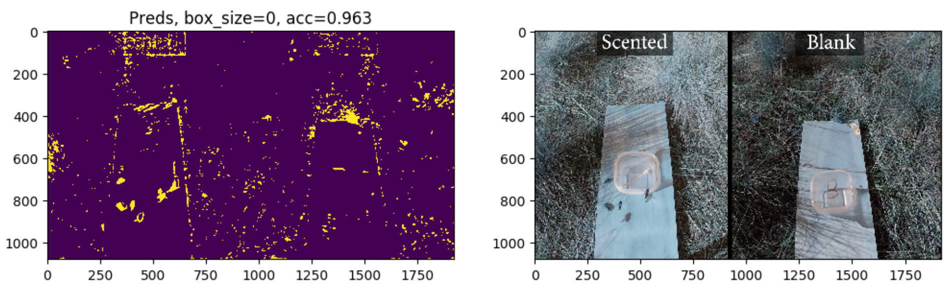
Not ensuring class balance. Now there are no masked “no data” pixels - the image is fully segmented into bee/not bee. This means that pixel-wise class balance is nonexistent.



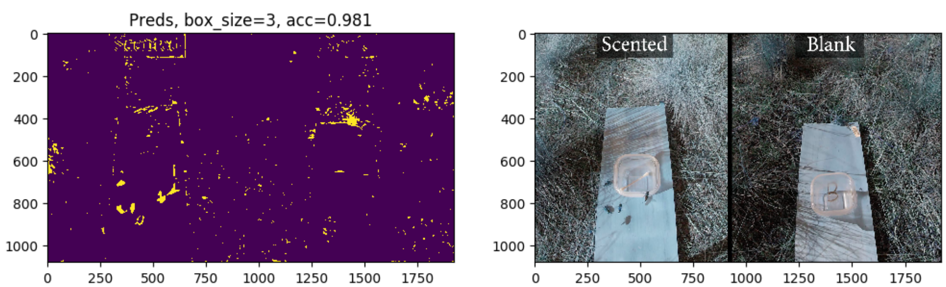
The output of the neural network is all not bee.

Output probabilities, and negative training example size. The larger boxes result in the network not outputting a p ¿ 0.5 for any pixel in the bee category. Using a negative size of 1 results in probabilities in the bee category of 90%. To examine this issue, I trained different networks using different box sizes (6 of them). The results are in the figure below. The larger the box size, the higher probability that the negative points contain a bee. This could be dealt with by segmenting not-bee parts of an image, or choosing not-bee samples more intelligently.

I’m pretty sure the bad results with a large box size are because the random boxes are overlapping bees in some cases. All of the results above were obtained with a simple architecture:

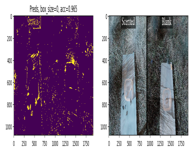


(a) first subfigure

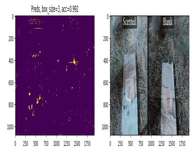


(b) second subfigure

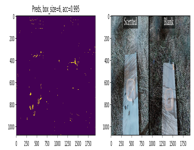
Figure 1: Preliminary results.



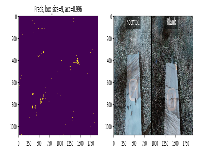
(a) first subfigure



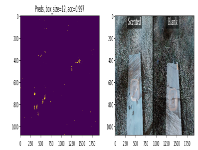
(b) second subfigure



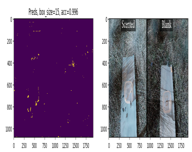
(c) third subfigure



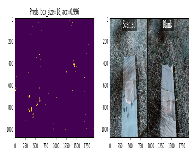
(d) fourth subfigure



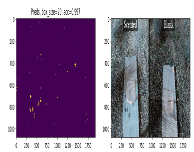
(e) fifth subfigure



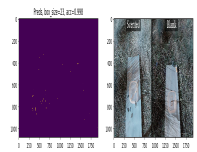
(f) sixth subfigure



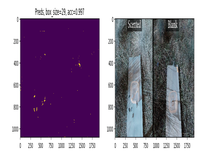
(g) seventh subfigure



(h) eighth subfigure



(i) ninth subfigure



(j) tenth subfigure

Figure 2: Preliminary results.

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=8,
    padding='same', activation='relu',
    input_shape=image_shape, data_format='channels_last'))
model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=4,
    padding='same', activation='relu'))
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=4,
    padding='same', activation='relu'))
model.add(tf.keras.layers.Conv2D(filters=16, kernel_size=2,
    padding='same', activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Conv2D(filters=n_classes,
    kernel_size=2, padding='same',
    activation='softmax'))
```

The results from generating negative samples where the is no bee in the negative sample are below.

2 Model experimentation