Where’s Waldo with TensorFlow

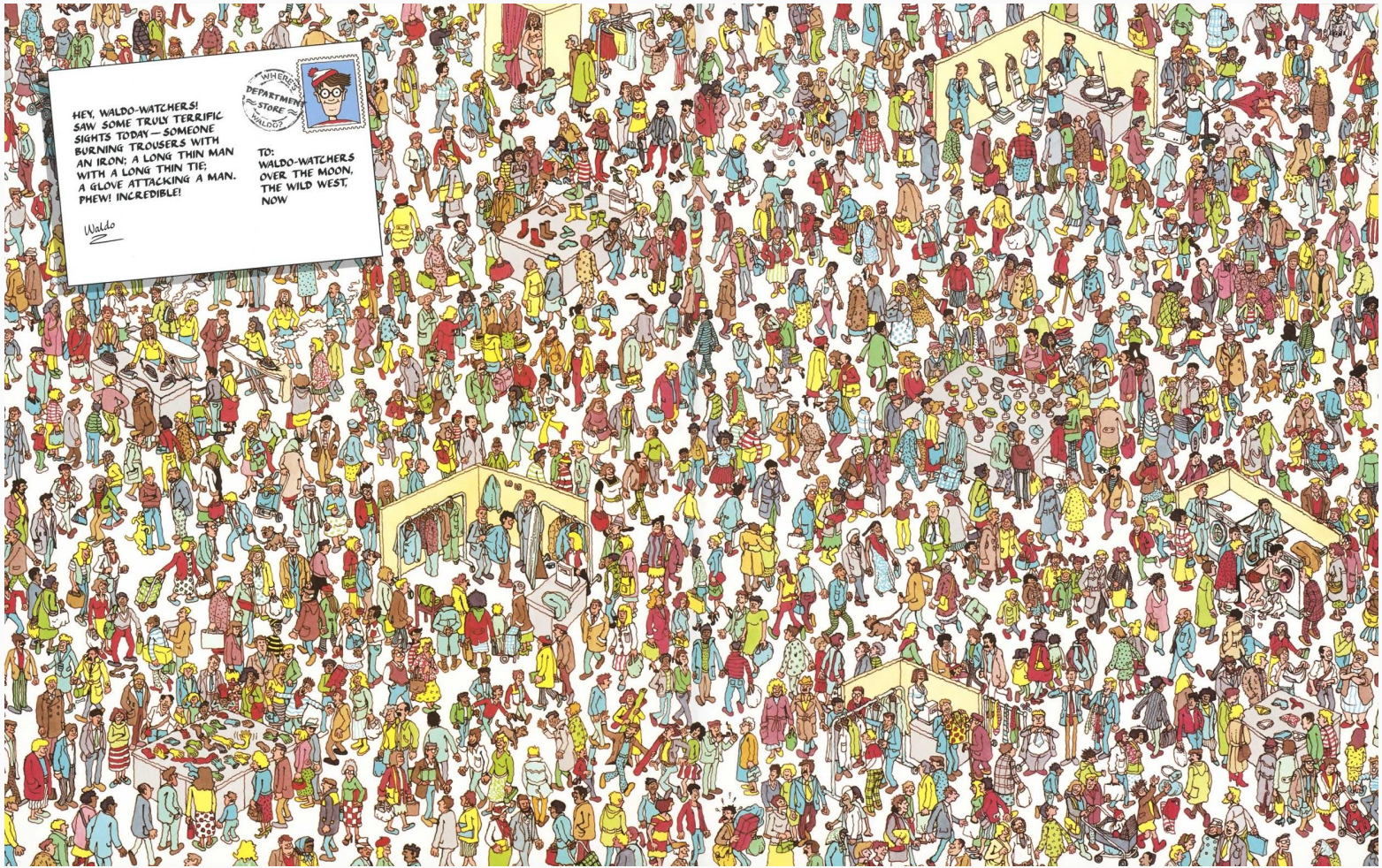
Team

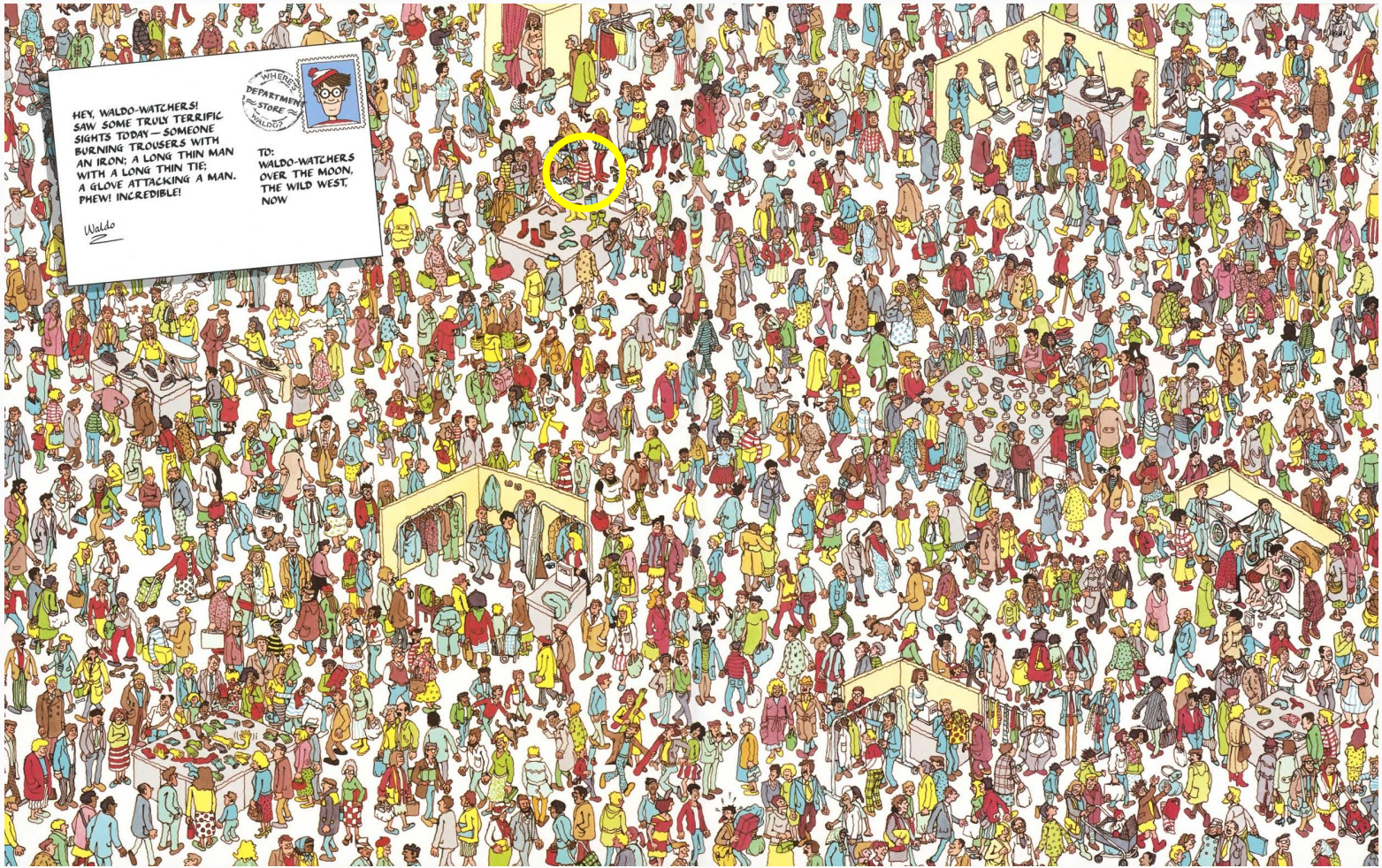
Natasa Lazetic

- Software Engineer, Mom, Student, Interviewing for Jobs
- Bellingham

Lia Gaetano

- Olympic Tryouts, Full time work
- Seattle


HEY, WALDO-WATCHERS! SAW SOME TRULY TERRIFIC SIGHTS TODAY—SOMEONE BURNING TROUSERS WITH AN IRON; A LONG THIN MAN WITH A LONG THIN TIE; A GLOVE ATTACKING A MAN. PHEW! INCREDIBLE!
Waldo
WHERE'S DEPARTMENT STORE WALDO?
TO: WALDO-WATCHERS OVER THE MOON, THE WILD WEST, NOW


HEY, WALDO-WATCHERS! SAW SOME TRULY TERRIFIC SIGHTS TODAY — SOMEONE BURNING TROUSERS WITH AN IRON; A LONG THIN MAN WITH A LONG THIN TIE; A GLOVE ATTACKING A MAN. PHEW! INCREDIBLE!
Waldo
WHERE'S WALDO'S DEPARTMENT STORE
TO: WALDO-WATCHERS OVER THE MOON, THE WILD WEST, NOW

How to go about finding Waldo?
Our Plan

Image Classification

- Gather Images/preprocess
- Using Transfer Learning with Inception Neural Network to classify images as Waldo/Not Waldo
- Find Waldo in an image
- (optional) Make an Android app that finds Waldo

Our Data

- Used data from https://github.com/vc1492a/Hey-Waldo
- Image Formats
  - 256 x 256 pixels (317 images)
  - 128 x 128 pixels (1344 images)
  - 64 x 64 pixels (5376 images)
-



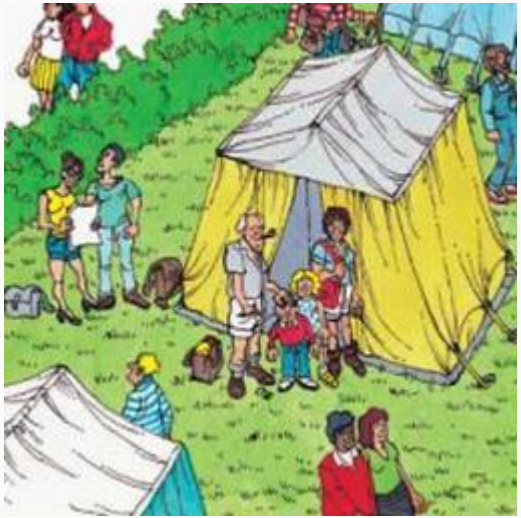




64 x 64




128 x 128

256 x 256

(+ original large images)



Waldo




Waldo




Not Waldo




Not Waldo

Using Transfer Learning with Inception NN

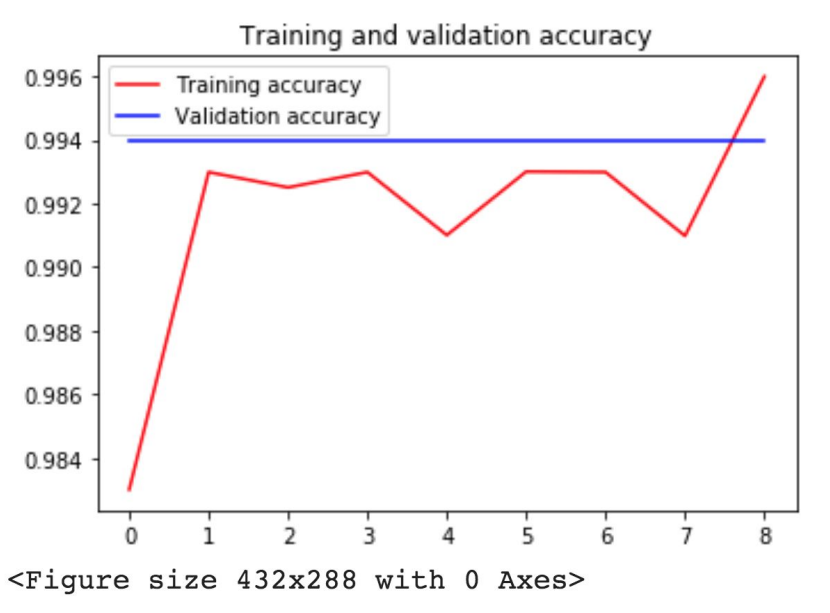
```
from tensorflow.keras.applications.inception_v3 import InceptionV3
local_weights_file = '/tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'
# Instantiate Inception with desired input shape for Where's Waldo data
pre_trained_model = InceptionV3(input_shape = (75, 75, 3),
                                include_top = False,
                                weights = None)
pre_trained_model.load_weights(local_weights_file)

# Lock Inception's pre-trained layers
for layer in pre_trained_model.layers:
  layer.trainable = False

# pre_trained_model.summary()

last_layer = pre_trained_model.get_layer('mixed7')
last_output = last_layer.output
```

Where's Waldo 1.0 (only 64x64 images) . . . overfitting

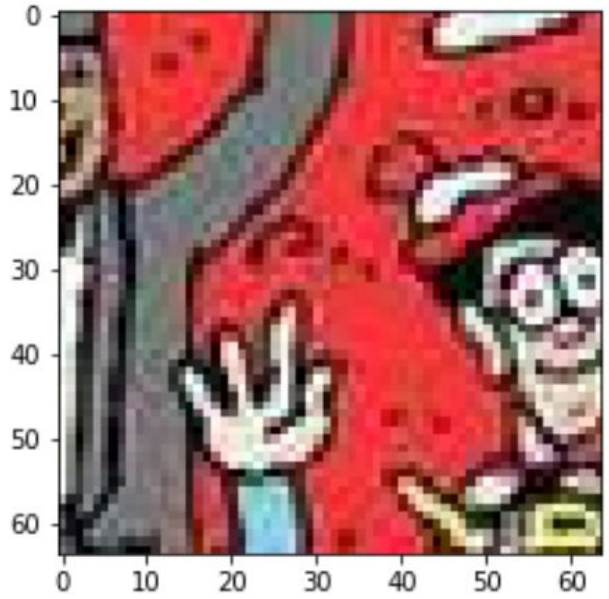


- 4,315 64 x 64 training images
- 1,077 64 x 64 validation images

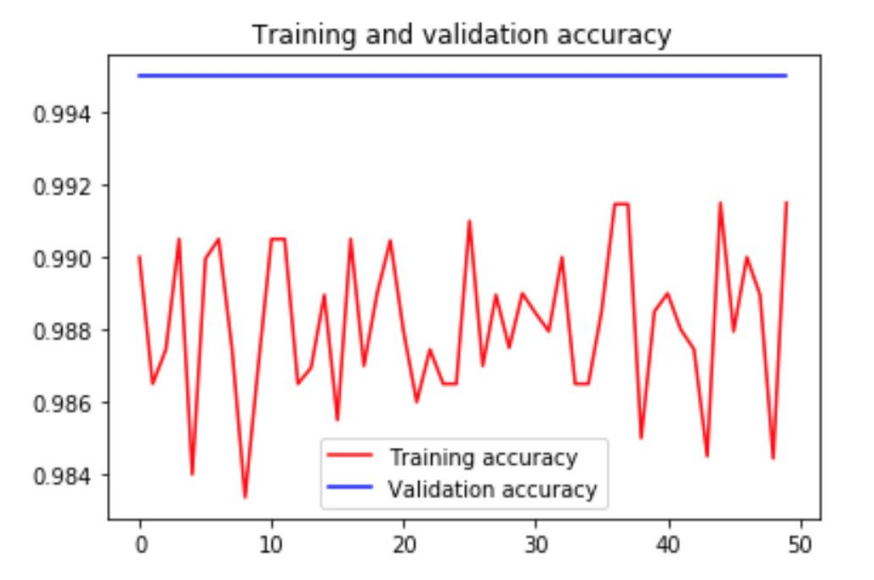
- 9/50 epochs
- 99.6% accuracy
- Evaluation: **0 images are classified as "Waldo"**

Failing to correctly recognize training images

```
Saving 3_15_1.jpg to 3_15_1 (2).jpg
[0.]
3_15_1.jpg is not a picture of Waldo...
Out[22]: <matplotlib.image.AxesImage at 0x7f29b48c9908>
```



Where's Waldo 1.1 use all images (64x64, 128x128, 256x256) . . . still overfitting



- 5,652 64 x 64
  128 x 128
  256 x 256 training images
- 1,075 64 x 64
  128 x 128
  256 x 256 validation images

- **50/50 epochs**
- 99.15%



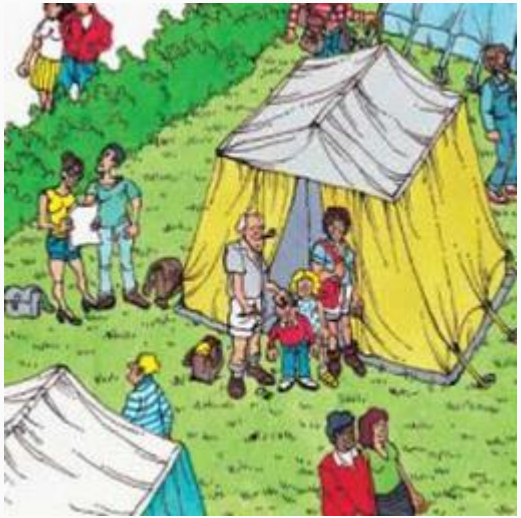












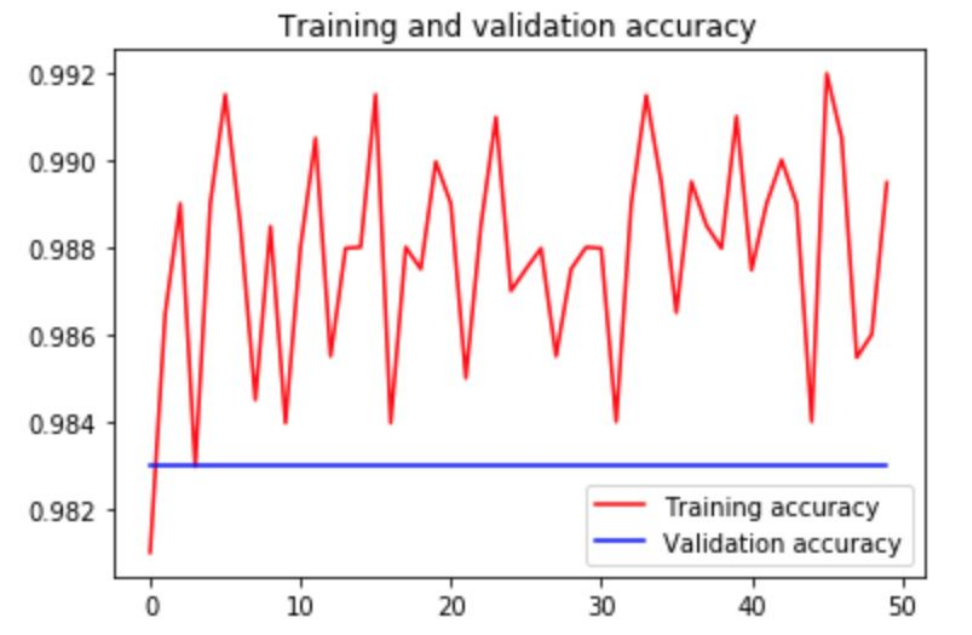
Perhaps the data is the problem

What does our data look like?

- Not specific enough?
- Where **is** Waldo in the 64 x 64?

More specific data, cropped to Waldo the individual

Where's Waldo 1.2 - "More Specific" training . .



- 5,656 64 x 64
  128 x 128
  256 x 256
  "**More specific**" training images
- 1,090 64 x 64
  128 x 128
  256 x 256 validation images

- 50/50 epochs
- **98.95**%
- **Evaluation: 0% success recognizing Waldo images**

Where’s Waldo 1.3 - 1:10 ratio - waldo:notWaldo Images

- Using only 64x64 images

Training:

- Waldo: 34 (was 34)
- Not Waldo: 340 (was 4284)

Validation:

- Waldo: 6 (was 7)
- NotWaldo: 60 (was 1071)

- 50 epochs, training accuracy: around 96%, validation accuracy” about 91%
- **Evaluation: 0% success recognizing Waldo images**

Found existing apps for Finding Waldo

- Web App - http://findingwally.pythonanywhere.com/
- Android - How to build a mobile app to find Waldo using an Artificial Neural Network (with Tensorflow Lite), https://medium.com/@victorbonnet/how-to-build-a-mobile-app-to-find-waldo-using-an-artificial-neural-network-with-tensorflow-lite-afd2a1a14f58

Our next steps

- Find Waldo using Object Detection
- Make our own app

Reflections

- Fun together
- Budding Data Scientist
- Good practice working as a distributed team

Thank you

Many Thanks to

- Leaders, volunteers, and members of Women in Data Science group
- Laurence Moroney
- Google volunteers
- Google