

Where's Waldo?: A Machine Learning Approach to a Classic Childhood Problem



Introduction

Waldo is hard to find. Why not let a computer do it? In this project, I demonstrate that a U-Net neural network can segment and localize Waldo in complex images. Overall, a neural network and simple localization algorithm is able to find Waldo.

Network

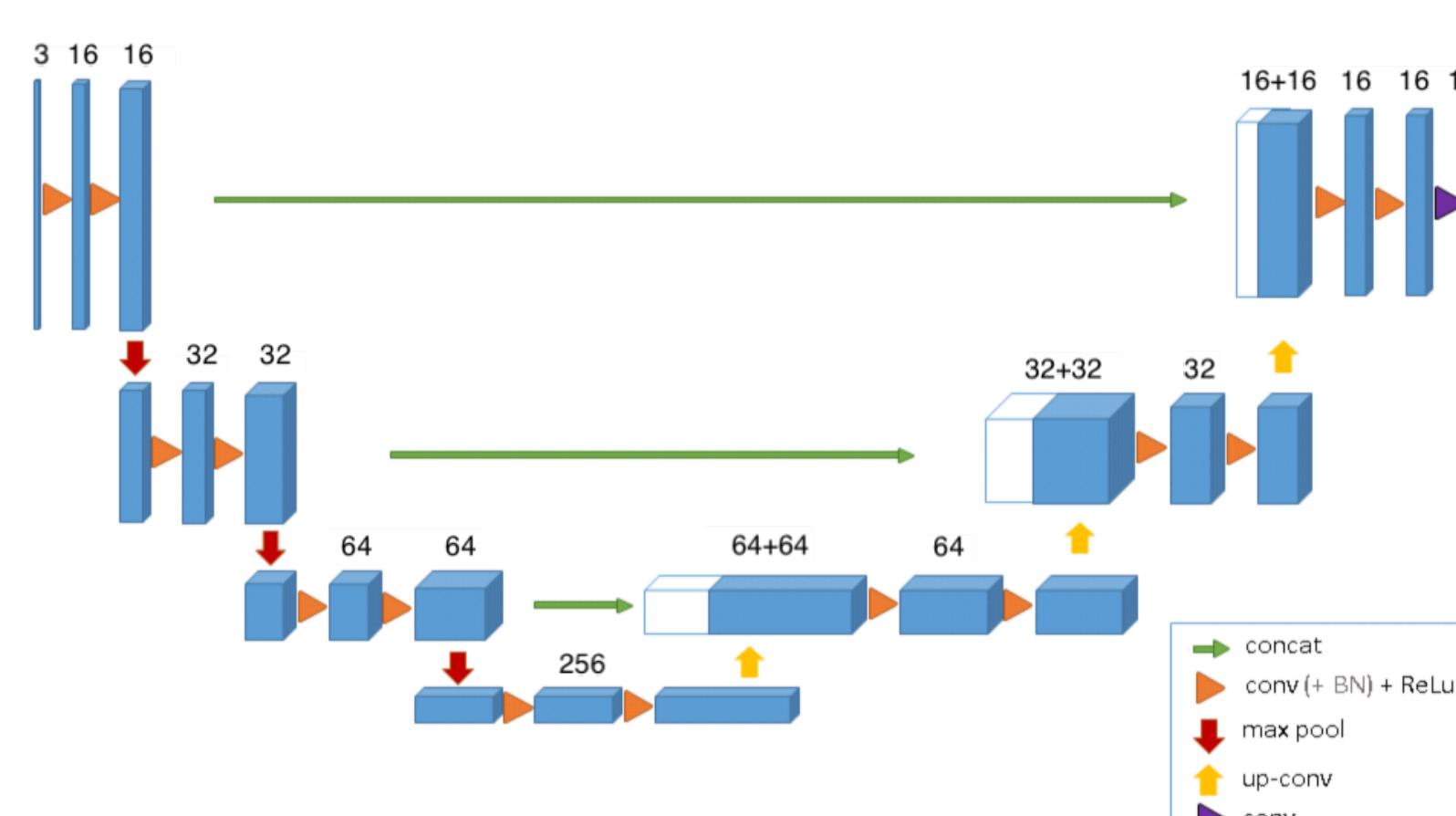


Figure 1: U-Net architecture used for segmentation (<http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a/>)

The network used was based on the U-Net architecture in Ronneberger et al.'s paper [1]. This network was developed for medical image segmentation, and **performs well on small datasets**. Much of the network architecture code was obtained and modified from Sterbak's website [2]. The network consists of a contracting path, which repeats a block of a convolutional layer, followed by a ReLU layer and downsampled with a max pooling layer. **The contracting path is sensitive to image context**. An expansive path uses up-convolutional layers, followed by a concatenation with the symmetric layer and another convolutional and ReLU layer. **The expansive path enables precise localization**, making this network ideal for small dataset image segmentation. Batch normalization was added after each layer.

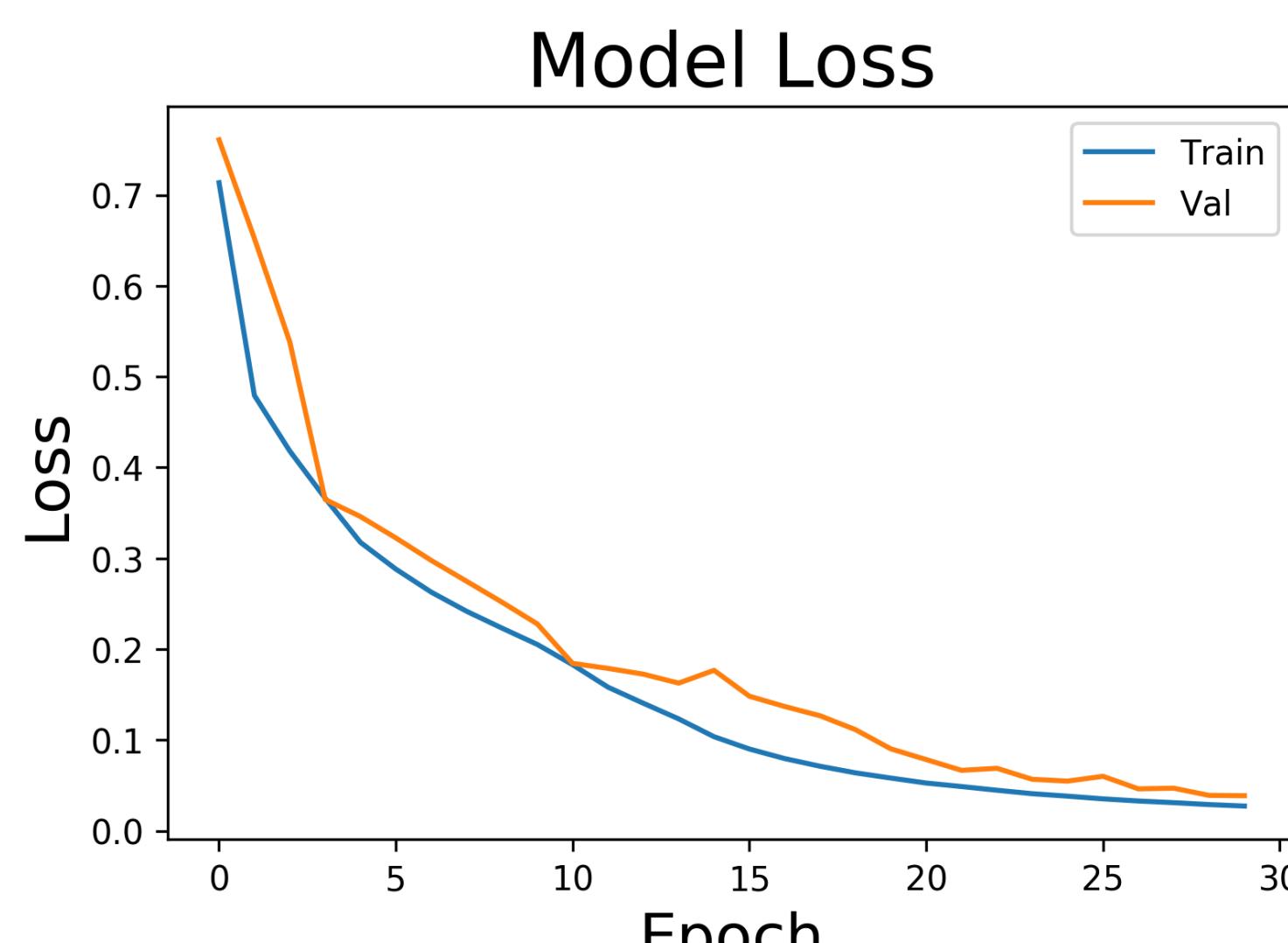


Figure 2: Training and validation loss metrics

Data

The data for this project came from six "Where's Waldo?" books [2]. The books were scanned at 5100 x 3300 pixels. Of the six books, five were included in the dataset and one exclusively for testing. Truth masks for all training images were made using Matlab. An example is shown in Figure 3. The training dataset was split randomly into 65% training, 20% validation, and 15% testing. A data generator was created to randomly flip the training images horizontally, and take a random 256x256 pixel crop that contains Waldo. In addition to preserving image resolution, this step ensures that Waldo's location in the training images is different each time the network sees it, avoiding training location biases. Horizontal flipping effectively doubles the dataset.



Figure 3: Example data generator output used for training network. Both horizontal flipping and cropping are used.

Methods

The network was implemented in Python using the Keras library. The network was trained with a batch size of 16 images for 30 epochs of 20 steps each. Training took about two hours on a CPU. An "Adam" optimizer was used. Loss metrics were calculated as the binary cross-entropy for pixel outputs. The training loss history is shown in Figure 2.

Thomas Rhines

https://github.com/tjrhines/Final_Public

Evaluation

Three separate methods were used to evaluate performance of the model:

- Testing on images originally allocated in training data, cropping them randomly around Waldo. Results can be seen in Figure 4. Using the predicted and truth masks, I calculated the intersection over union (IOU) for all images. The **mean IOU is 0.748** for these seven images, suggesting reasonably good results with a very small dataset and few training epochs.
- I developed a simple algorithm to tile the input of a larger image and input each tile into the network. The largest area object in the prediction was classified as Waldo. This algorithm found Waldo in all seven original testing images.
- The same algorithm was used for the independent, untrained Waldo book. Waldo was found 3/6 times, Wenda was found 2/6 times (a near-success) and Waldo was not found 1/6 times. Examples are shown in Figure 5.

Results

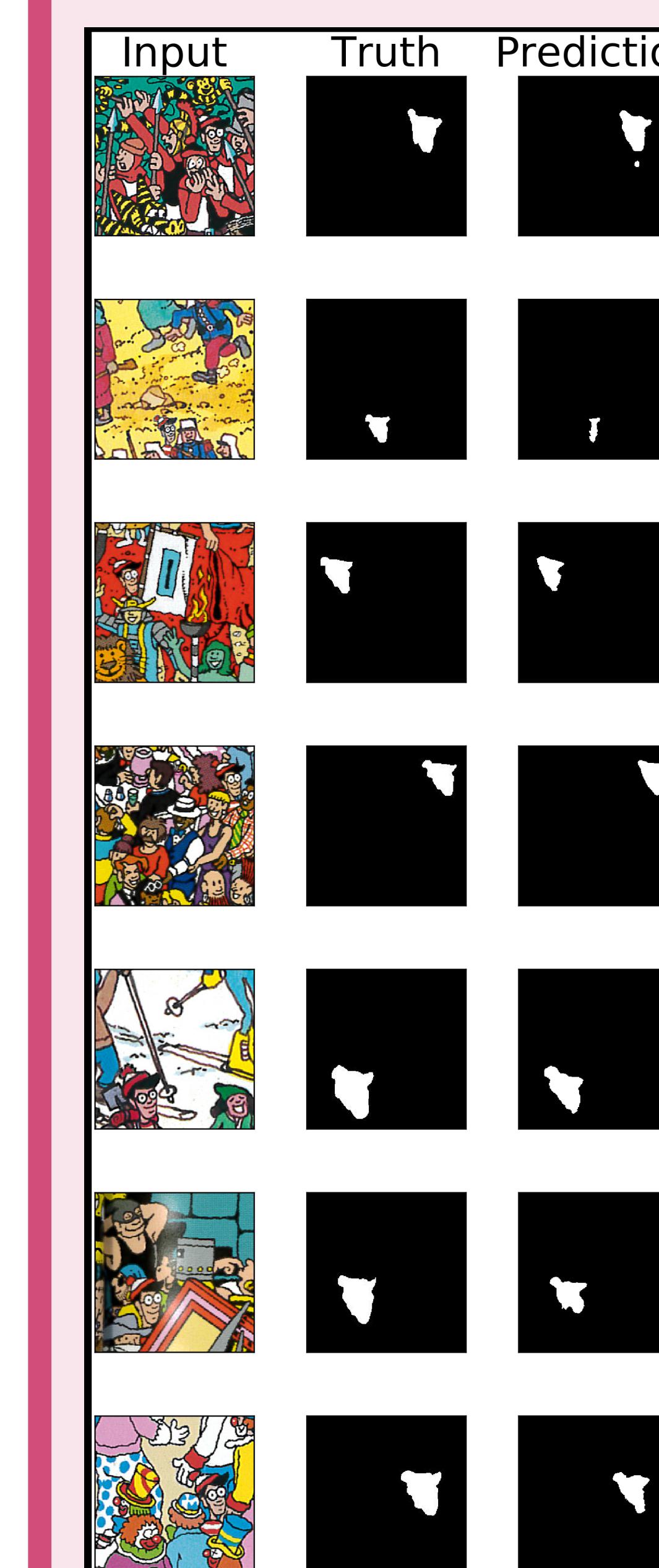


Figure 4: Input images, truth masks, and predictions for original testing set

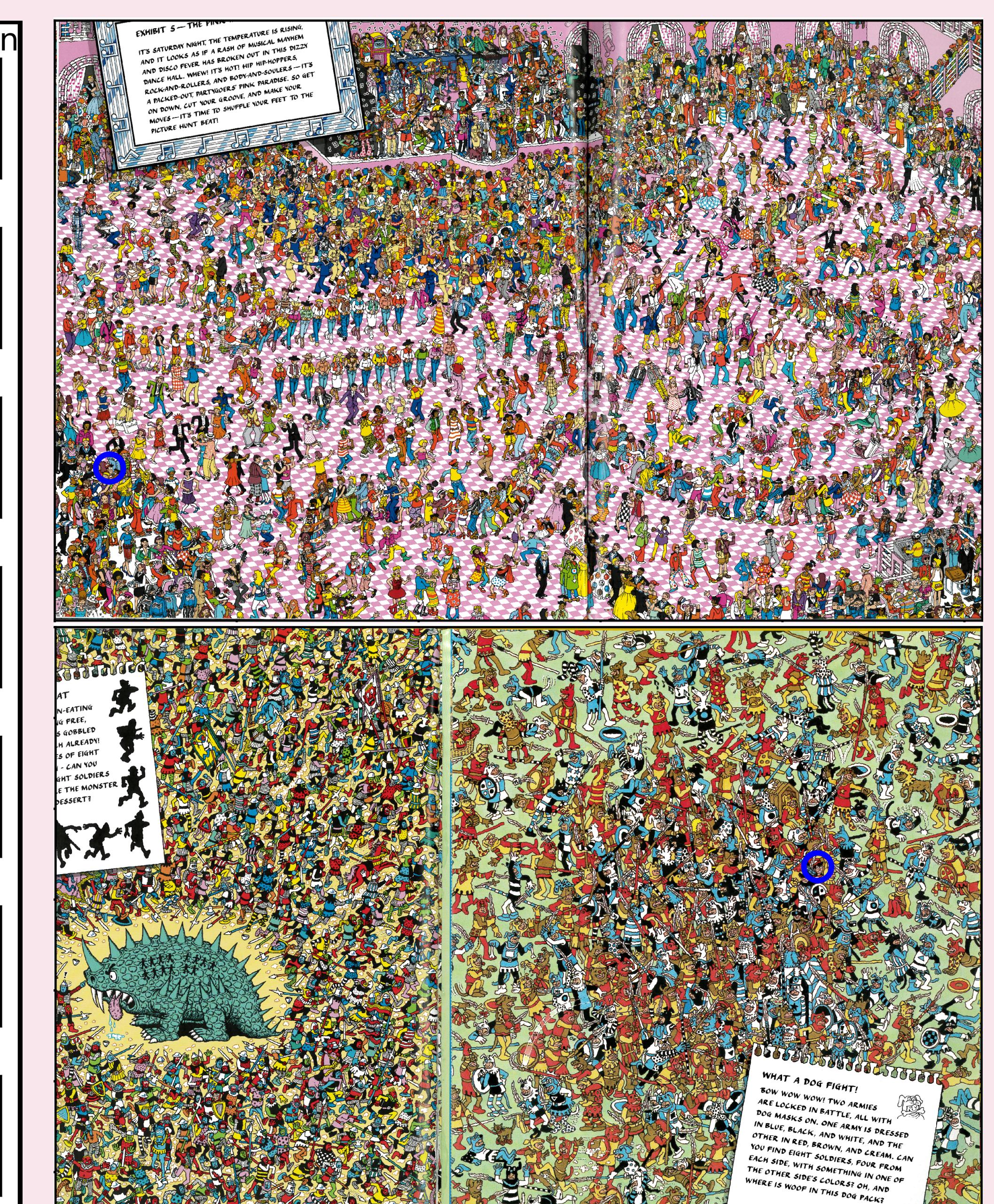


Figure 5: Example results of the tiling algorithm. The first is a correct localization from the original test set. The second image, from the independent testing book, finds Wenda.

Conclusion

This project shows that neural networks are able to identify and localize objects in complicated visual data. Results are primarily qualitative, and successfully identify the target in most tests. Waldo is a 'toy model' for bigger problems, which are relevant to physics. Medical physicists segment diseases in medical images, and astrophysicists automatically probe huge quantities of stellar images. Applications of neural networks are applicable whenever image data is available.

References

- [1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*.
- [2] Tobias Sterbak. *U-Net for Segmenting Seismic Images with Keras*, 2018.
- [3] Martin Handford. *Where's Waldo?*, volume 1-7. Candlewick Press, 2007.

Acknowledgements

Special thanks to my dad for scanning the "dataset"