Proposal

February 25, 2018

1 Machine Learning Engineer Nanodegree

1.1 Capstone Proposal

David Eraso
February 23th, 2018

1.2 Find the nuclei in divergent images to advance medical discovery

Kaggle/Booz Allen Hamilton competition 2018 Data Scince Bowl

1.2.1 Domain Background

Identifying the cells' nuclei allows researchers to identify each individual cell in a sample, and by measuring how cells react to various treatments, the researcher can understand the underlying biological processes at work. By automating nucleus detection, research could unlock cures faster by allowing more efficient drug testing.

I am personally interested in using Machine Learning techniques in topics related to health.

Related Videos:

2018 Data Science Bowl

Technical Overview

1.2.2 Problem Statement

The objective is to "create a computer model that can identify a range of nuclei across varied conditions". Since this is an object detection task, the evaluation metric that will be used is the mean average precision (mAP) at different Intersection over Union (IoU) thresholds. Details will be explored in more detail on the 'Evaluation Metrics' section.

1.2.3 Datasets and Inputs

Datasets are obtained from the Data Web Page of the competition. From this reference:

This dataset contains a large number of segmented nuclei images. The images were acquired under a variety of conditions and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). The dataset is designed to challenge an algorithm's ability to generalize across these variations.

Each image is represented by an associated ImageId. Files belonging to an image are contained in a folder with this ImageId. Within this folder are two subfolders:

- `images` contains the image file.
- `masks` contains the segmented masks of each nucleus. This folder is only included in the training set. Each mask contains one nucleus. Masks are not allowed to overlap (no pixel belongs to two masks).

The second stage dataset will contain images from unseen experimental conditions. To deter hand labeling, it will also contain images that are ignored in scoring. The metric used to score this competition requires that your submissions are in run-length encoded format. Please see the evaluation page for details.

As with any human-annotated dataset, you may find various forms of errors in the data. You may manually correct errors you find in the training set. The dataset will not be updated/re-released unless it is determined that there are a large number of systematic errors. The masks of the stage 1 test set will be released with the release of the stage 2 test set.

File descriptions:

`/stage1_train/*` - training set images (images and annotated masks)
`/stage1_test/*` - stage 1 test set images (images only, you are predicting the masks)
`/stage2_test/*` (released later) - stage 2 test set images (images only, you are predicting the masks)
`stage1_sample_submission.csv` - a submission file containing the ImageIds for which you must predict during stage 1
`stage2_sample_submission.csv` (released later) - a submission file containing the ImageIds for which you must predict during stage 2
`stage1_train_labels.csv` - a file showing the run-length encoded representation of the training images. This is provided as a convenience and is redundant with the mask image files.

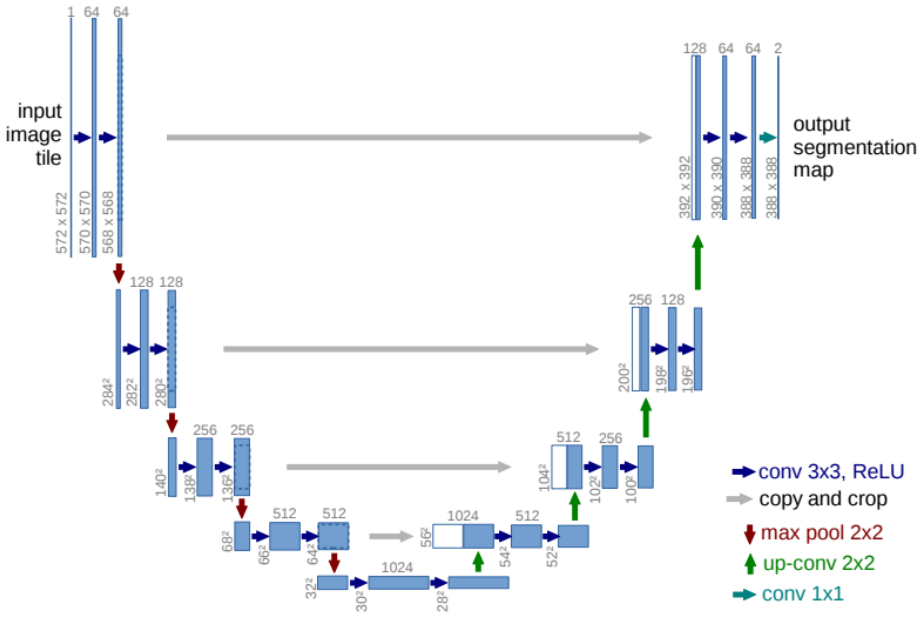
1.2.4 Solution Statement

I'm looking to develop an architecture capable of achieving a score good enough to make it into the first 100 places in the leaderboard. By February 19, the 100th place had a score of 0.395. The first place, not prize eligible, is a score of 0.634.

This is my first Kaggle competition and I don't want to be overambitious. The evaluation metric used in the leaderboard is a mAP described in the 'Evaluation Metric' section.

1.2.5 Benchmark Model

My Benchmark model is the kernel `U-Net starter - LB 0.277`, by Kjetil Åmdal-SævikKeras. Beyond implementing the basic setup of getting the data as well as encoding results for submission, Kjetil builds a U-Net model, *'loosely based on U-Net: Convolutional Networks for Biomedical Image Segmentation and very similar to this repo from the Kaggle Ultrasound Nerve Segmentation competition.'*



This kernel is a good starting point with a score of 0.277.

1.2.6 Evaluation Metrics

From the 'Evaluation' link on the competition webpage:

This competition is evaluated on the mean average precision at different intersection over union (IoU) thresholds. The IoU of a proposed set of object pixels and a set of true object pixels is calculated as:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

The metric sweeps over a range of IoU thresholds, at each point calculating an average precision value. The threshold values range from 0.5 to 0.95 with a step size of 0.05.

In other words, at a threshold of 0.5, a predicted object is considered a "hit" if its intersection over union with a ground truth object is greater than 0.5.

At each threshold value t, a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP) resulting from comparing the predicted object to all ground truth objects:

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

A true positive is counted when a single predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object had no associated ground truth object. A false negative indicates a ground truth object had no associated predicted object. The average precision of a single image is then calculated as the mean of the above precision values at each IoU threshold:

$$\frac{1}{|thresholds|} \sum_t \frac{TP(t)}{TP(t) + FP(t) + FN(t)}$$

Lastly, the score returned by the competition metric is the mean taken over the individual average precisions of each image in the test dataset.

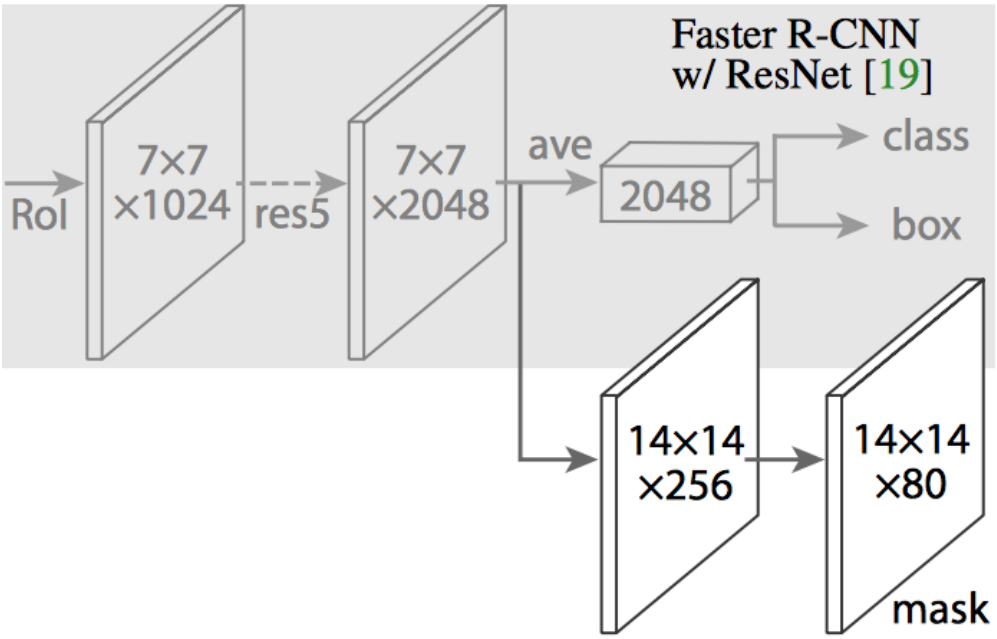
1.2.7 Project Design

References:
A Brief History CNNs in Image Segmentation: From R-CNN to Mask R-CNN
Image Segmentation
A 2017 Guide to Semantic Segmentation with Deep Learning

Convolutional Networks have been used succesfully for both image classification and image recognition. The benchmark model mentioned above is based on a Convolutional Network designed specifically for Biomedical Image Segmentation, the U-Net.

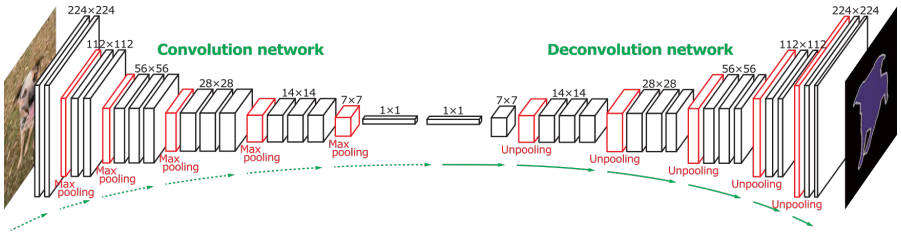
This is clearly an Image Segmentation problem. The images have overlapping objects, nuclei, and we want to correctly identify the object and its boundaries. Since we are interested in masking, we need pixel level segmentation. Pixel segmentation was explored at Facebook AI by Kaiming He and a team of researchers, including Girshick, using an architecture known as Mask R-CNN. Basically they used an existing architecture called Faster R-CNN and replaced the last layers that gave the classification and boundary box with a Fully Convolutional Network (FCN) to generate the mask:

From Mask R-CNN:



The mask is a binary classifier that defines whether or not a given pixel is part of an object. In this case, we are interested in whether the pixel is part of a given nucleus or not (background or other nuclei). The FCN is a normal CNN, where the last fully connected layer is substituted by another convolution layer with a large "receptive field". Here there is an example of how to convert a normal CNN used for classification, ie: Alexnet to a FCN used for segmentation.

The U-Net also uses the Transposed Convoultional Layer idea, also called "Deconvolution", to scale up the scale down effect done on the first layers. This is the "Deconvnet" architecture that according to the reference *it takes 6 days to train on a TitanX* and needs to be trained in 2-stages (Single objects centered, and fine tuning with difficult examples):



There are several architectures that have been developed to tackle this kind of problem. Understanding the image at pixel level is also called Semantic Segmentation. A list of architectures for Semantic Segmentation can be found here. They are:

1. Fully Convolutional Networks for Semantic Segmentation
2. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation
3. Multi-Scale Context Aggregation by Dilated Convolutions
4. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
5. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation
6. Pyramid Scene Parsing Network
7. Large Kernel Matters -- Improve Semantic Segmentation by Global Convolutional Network
8. Rethinking Atrous Convolution for Semantic Image Segmentation

The Steps in This Project: I propose to explain roughly the architectures mentioned above, and why each one is or isn't a relevant candidate for the specific problem at hand. I will then implement three architectures based on the ones that pose the most promise. Every experiment will have a score. The best alternative can then be tuned further.