Detecting Endometriosis Within Laparoscopic Imaging

By Allie Wicklund

7/11/2021

Table of Contents

The Main Purpose…………………………………………………………………………………3

What is Endometriosis?...................................................................................................................3

Obtaining the GLENDA Database………………………………………………………………...4

Cleaning and Pre-Processing the Image Data……………………………………………………..5

If At First You Don’t Succeed…………………………………………………………………….7

Modeling in a Black Box………………………………………………………………………….8

The Personality of a Model………………………………………………………………………..9

Conclusions and Recommendations………………………………………………………………9

References………………………………………………………………………………………..11

The Main Purpose

“For a definitive diagnosis of endometriosis, visual inspection of the pelvis at laparoscopy is the gold standard investigation” (1)

-European Society of Human Reproduction and Embryology

In other words, surgery is the main option employed to diagnose endometriosis in persons that have a uterus. Patients rely on the surgeon’s ability to detect endometriosis while the patient lies on the operating table. No surgeon is perfect, but recent advancements in neural networks allow for modeling of image and video data. Before today, no object detection models conducted for endometriosis were available to the public online. Now this model, dubbed The Endo Project, aims to make medical modeling through Tensorflow 2 easier to access and increase public awareness of endometriosis. For readers that desire a less technical description of the project, here is the link to the Endometriosis Modeling 101 documentation (11).

What is Endometriosis?

The Oxford Dictionary defines endometriosis as: a condition resulting from the appearance of endometrial tissue outside the uterus and causing pelvic pain. (2) Endometrial tissue is tissue that typically forms in the uterus to protect the fertilized egg during pregnancy. When the person with a uterus does not have the egg fertilized, the body rejects the tissue and unfertilized egg in a process known as menstruation. If the endometrial tissue forms outside the uterus (usually on the uterus, but can be anywhere in the body), then this tissue is not shed during menstruation. As the quote in the main purpose section stated, the tissue is found (and removed) using surgery. Many patients experience cramping during periods, but endometriosis may cause pain that’s described as “far worse than usual.” (3) Removing the uterus (hysterectomy) will not necessarily cure endometriosis since patients have reported returning pain and symptoms from the disease after hysterectomy. (4) If surgery is the main option, then this model may assist surgeons in finding endometriosis outside the uterus by training with annotated images. To do this, we need a database that contains annotated images from laparoscopy surgeries. The GLENDA (Gynecologic Laparoscopy ENdometriosis DAtaset) version 1.0 has the images needed.

Obtaining the GLENDA Database

GLENDA version 1.0 can be obtained through the Alpen-Adria University in Austria. (5) It contains 25,683 images of laparoscopic surgery and 521 annotations. It is a 1.7 GB file that contains many subfolders. Here is the subfolder structure:

├── DS

│   ├── no\_pathology

│   │   └── frames

│   └── pathology

│   ├── annotations

│   └── frames

├── Readme.txt

└── statistics

   ├── annotations.csv

   └── images.csv

Some of the images contain multiple annotations for one image, and the no\_pathology folder has no endometriosis in it. All annotations are in .png format, while all images are in .jpg format. Some images have the same name, but are located in a differently named folder.

First, the file was imported using Keras and a direct download link. This worked, but the subdirectories proved to be a challenge. There were no images showing up with Keras, even when searching for wildcard .jpg files. Then, the direct download link was used to download the file outside of Jupyter Notebook for some cleaning by hand. After a few more attempts to make Keras recognize the file, it became apparent that the file needed to be completely reorganized into a form that would be more suitable for this project’s needs.

Cleaning and Pre-Processing the Image Data

Cleaning the image data was mostly dependent on the tutorial used, rather than specifically which model was chosen. The first tutorial that was followed explained how the author used Tensorflow 2 to model raccoon detection in images and video. (6) Whether the model is detecting raccoons or endometriosis, the process of training is very similar. Both models are single-class object detection models that rely on a database of images to train the model. Both models use rectangular box coordinates to identify where the object of importance is located. Also, both models use mostly default parameters for modeling for simplicity. The dataset for endometriosis is cleaned by:

1. Searching for a folder in annotations.
2. Searching for the matching folder name in frames.
3. Deleting all but the first image in the frames folder (Letting the program augment the data for us is more helpful than self-augmentation in this case. The annotations were based on the first image, not the rest of the images.)
4. Naming the picture in frames to X.jpg, where X is the number photo in frames looked at.
5. Naming the picture in annotations XY.png, where X is the number photo in frames and Y is missing if there is only one annotation for the frame, and a, b, c, d… for if there is more than one annotation.
6. Creating a .csv file in Excel with the first column being the frame photo name (1.jpg), second and third columns being the height and width of the frame, and the fourth column being endometriosis as the class. 1.jpg would be listed under column one for the number of annotations it had.
7. Repeating steps 1-6 about 300 times for the dataset. Some frames did not have annotations and vice-versa, so these were removed from the dataset.

After these steps, the folders could be rearranged, merged, and removed to further simplify the file directory. Windows search function for files was helpful in this task, as all frames ended in .jpg and all annotations ended in .png. This is how the file directory looked at this point:

── pathology (contains temporary.csv)

── annotations

── frames

The last four columns were created in the temporary.csv by using a program called LabelImg, which allows the user to draw rectangles around the object of interest (in this case, the annotation images were placed into LabelImg) and save the rectangles as .xml files. These xml files were to the annotations, but the rectangles directly apply to the corresponding image in frames. This also was a hand-done process. Later on, it was discovered that if xml files are created first, the csv can be created with a few lines of code.

Then, the data in the csv is imported into a Pandas dataframe in a Jupyter Notebook. There, test\_train\_split was used to separate the test from the training data, grouped by filename. This code created about a 75-25 split between training and test data, but unfortunately this code was lost while rearranging files later on. Two csv files were created to represent this split: test\_labels.csv and train\_labels.csv. To be able to use this data in Tensorflow 2, these csv files were converted to record files: test.record and train.record.

Lastly, the model had to be downloaded. Faster R-CNN Inception ResNet V2 1024 x 1024 was chosen for the large frame size and speed in computation. Once the gz file is downloaded from the object detection zoo repository (7), then the file was extracted twice and put in its own folder outside of pathology. The files were ready to be uploaded to Google Drive for use in Google Colaboratory.

If At First You Don’t Succeed

“404 Page not found”, the Tensorflow documentation page reads. This raccoon tutorial was a dead end in Google Colab. Various other tutorials were looked at, each of them leading to the 404 page in the end. Then, the sixth option was the one that finally worked. This option was a Jupyter Notebook that could be imported into Google Colab directly and explained what files needed editing before training the model (8). Editing the documents and working through the bugs was not an easy task, but eventually the program trained the model. On Google Colab Pro with High RAM and GPU accelerators, the training on 204 images took about 5 hours. A few short lines of code later, the graphs and images were created. This project, spanning over 100 hours, was spent mostly cleaning and debugging code in training. The model was successful, but how successful at its task? There’s an important aspect to look at before continuing to explain the strengths and weaknesses of the model: the nature of black box modeling.

Modeling in a Black Box

Remember the test.record and train.record files from earlier? If a programmer were to open those files in Notepad, the file would contain gibberish. A black box model is one that the programmer knows the inputs and outputs to, but the inner workings are a mystery (9). It is known the basic idea of these models is to apply filters to image data, thus changing the values of the data, but it is unknown how this is implemented exactly. Faster R-CNN is a black box that takes all of these files as input (but primarily the record files, and the config files downloaded with the model), and outputs the detection box coordinates for all the images given. It also computes classes and scores for the object detections. There are some adjustments that can be made in the config files, but the actual process for the rest is hidden. This makes hyperparameter tuning difficult, given the long wait time and black box process. The model’s detection score is also calculated in the black box and that represents the confidence level of the model. If the model wasn’t created using a black box model, it would be easier to tune hyperparameters. However, given the limited time on this project, it was decided that publishing after one successful training would be sufficient. This gives the model a certain “personality” that could be edited using hyperparameter tuning.

The Personality of a Model

This endometriosis model is conservative in its predictions, which can be correlated to precision. If there is endometriosis in the body, the model is more likely to skip over endometrial tissue that it is unsure of rather than make a false diagnosis. This is determined by the 10 finished images compared to overlay annotations. The endometriosis model found less endometrial tissue overall than how much was actually there. There are a few type I errors, despite the precision of the model. Some spots are determined to be endometriosis that the doctors did not annotate as endometriosis. It would be interesting to survey these doctors after showing them what looks to be a type I error to see if they made an error or if the model did. However, it was likely an error on the model’s part, as will be explained in the last section.

Conclusions and Recommendations

The model was most likely making a type I error because of the small data amount. Endometriosis is a complex disease with many different forms of tissue and can be located almost anywhere in the body. These images were collected primarily near the perimetrium, or uterus walls, as shown in the original GLENDA database naming conventions (10). There were some ovary images, as well as some outside the uterus. For each of these different types of endometriosis, it would be ideal to have a model for each with as many images as possible. However, the more images there are, the more cleaning for the data must be done. The first recommendations are to increase the number of annotated images and to separate the model.

The model also feels a bit “weak” due to its conservative nature. If this experiment was repeated, it would be beneficial to research each hyperparameter to find out what changes from the defaults would need to be done. Considering most models were conducted on the COCO image dataset, a multiclass object detection dataset, it didn’t make much sense to keep the defaults. Tuning hyperparameters takes a considerable amount of time on image data, but even one more run with researched hyperparameters would have been beneficial. The second recommendation is to tune hyperparameters, if able.

If in the future the training time of models decreases significantly, it could be possible to implement real-time video object detection for use on laparoscopy scopes. This is currently not possible due to processing time per frame as getting those 10 images was not that fast. The final recommendation is to apply something similar to this model for live video and develop medical equipment to handle the imaging, but not until processing time decreases significantly. May a cure be found someday for those ailing from endometriosis.

Note 7/25/2021: Version 2.0 of the model is entering deployment. This model improves upon the original model by adding data augmentation and dropout. As a result, there is significantly decreased loss (from 15% to 4%) and significantly increased precision (from 80% to 93%).

References:

1. ESHRE Guidelines for Concise Diagnosis:  
   <https://guidelines.endometriosis.org/concise-diagnosis.html>
2. Oxford Definition of Endometriosis:

<https://www.lexico.com/definition/endometriosis>

1. Mayo clinic – Endometriosis symptoms

<https://www.mayoclinic.org/diseases-conditions/endometriosis/symptoms-causes/syc-20354656>

1. Endometriosis after Hysterectomy

<https://www.newhealthadvisor.org/Endometriosis-After-Hysterectomy.html>

1. GLENDA (site and publication)

<http://ftp.itec.aau.at/datasets/GLENDA/v1_0/index.html>

A. Leibetseder, S. Kletz, K. Schoeffmann, S. Keckstein and J. Keckstein. 2020. GLENDA: Gynecologic Laparoscopy Endometriosis Dataset. In Proceedings of the 26th International Conference on Multimedia Modeling, MMM 2020. Springer, Cham.

1. Modeling Raccoon Detection

<https://towardsdatascience.com/how-to-train-your-own-object-detector-with-tensorflows-object-detector-api-bec72ecfe1d9>

1. Tensorflow 2 Object Detection Zoo

<https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md>

1. The Object Detection Notebook chosen (copy)

<https://github.com/awicklund/EndoObjDetect/blob/main/Google%20Colab/Copy_of_Roboflow_TensorFlow2_Object_Detection.ipynb>

1. Black Box Definition

<https://www.investopedia.com/terms/b/blackbox.asp>

1. Perimetrium Definition

<https://www.merriam-webster.com/medical/perimetrium>

1. Endometriosis Modeling 101

<https://github.com/awicklund/EndoObjDetect/blob/main/Endometriosis%20Modeling%20101.docx>