

# Implementation of Regional-CNN and SSD Machine Learning Object Detection Architectures for the Real Time Analysis of Blood Borne Pathogens in Dark Field Microscopy

Daniel Fleury<sup>1</sup> and Febi Fleury<sup>2</sup>

<sup>1</sup>John Marshall

<sup>2</sup>University of Minnesota

July 4, 2018

## Abstract

The emerging utilization of visualization techniques in pathology and microbiology has been accelerated by machine learning (ML) approaches towards image preprocessing, classification, and feature extraction in an increasingly complex series of datasets. Modern Convolutional Neural Network (CNN) architectures have developed into an umbrella of vast image reinforcement and recognition methods, including a combined classification-localization of single/multi-object featured images. As a subtype neural network, CNN creates a rapid order of complexity by initially detecting borderlines, edges, and colors in images for dataset construction, eventually capable in mapping intricate objects and conformities. This paper investigates the disparities between Tensorflow object detection APIs, exclusively, Single Shot Detector (SSD) Mobilenet V1 and the Faster RCNN Inception V2 model, to sample computational drawbacks in accuracy-precision vs. real-time visualization capabilities. The situation of rapid ML medical image analysis is theoretically framed in regions with limited access to pathology and disease prevention departments (e.g. 3rd world and impoverished countries). Darkfield microscopy datasets of an initial 62 XML-JPG annotated training files were processed under Malaria and Syphilis classes. Model training were halted as soon as loss values were regularized and converged.

## Introduction

This medical object detection project explores and compares the benefits and costs in utilizing the SSD (Single Shot Detector) MobileNet and the Faster R-CNN Tensorflow Models as it pertains to detecting blood-borne pathogens- specifically Malaria and Syphilis in dark field microscopy images. Further, the paper will analyze the real-world practicality of utilizing both of these models in object detection challenges in addition to both models' capacities in detecting sample slides from Malaria and Syphilis cultures. The authors propose a cost-benefit system to assess the performance of two distinct models in various settings. The Faster R-CNN model's training process requires about half the time as needed to create a checkpoints and detect loss values in the images versus the SSD MobileNet architecture. In addition, Faster-RCNN marks a higher accuracy in detecting a greater number of cells as opposed to the SSD MobileNet. Although an assessment of precision and accuracy is required in both API models, the practical real time nature of the framework must be considered by observing the graphical compactness, efficiency, and lightness of each model in a limited environment (i.e. 3rd world setting). The communicability and variance of infectious diseases, such as Malaria and Syphilis, create a public health problem by making onsite medical intervention difficult. Further, the microheterogeneity of Malaria complicates the process of rapid human-controlled pathology diagnoses in local clinical laboratories.

The infeasibility of “human eye” feature extraction alone is apparent as a full determination and processing of sample information can require up to 7 days in the case of unexpected complications in tissue-blood analysis. Although existing feature segmentation techniques facilitate the visualization of histology samples, there is a growing gap in diagnosis due to the lack of specificity and machine-confirmed confidence/percentile scores. In the scope of a limited 3rd world environment, the ideal of accessibility and functionality is questioned in models such as Faster RCNN Inception v2 and SSD Mobilenet v1 due to their inherent graphical capabilities in FPS rate and real time motion detection. Utilizing a deployable object detection model that can be integrated into common IoT (Internet of Things) or system architectures would minimize the accessibility gap for a multi diagnostic app. However, the concept of a Cognitive Domain in the pathology treatment process must be attained by preserving detection accuracy, specificity, and sensitivity.

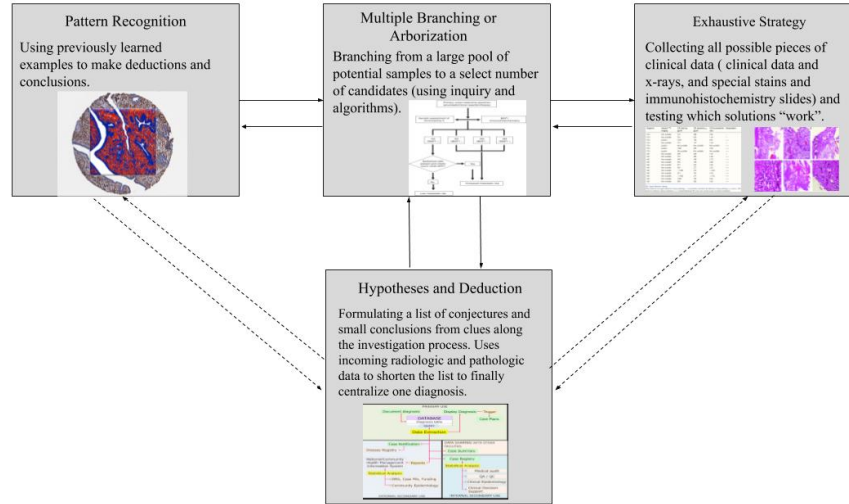
GPU (Graphical Processing Unit) tests on the R-CNN and Mobilenet Tensorflow models reveal distinct parameters and characteristics: Faster R-CNN preserves suitable precision and accuracy, however, demands more processing time whilst operating at a low FPS rate, making real time object detection difficult. The SSD Mobilenet architecture (v1-v2) creates an opportunity for moderate accuracy and average precision coupled with high scalability, FPS rates, real time capabilities, and low processing times. The SSD Mobilenet versions-series offers high deployability on low-CPU/GPU graded devices, including smartphones, Raspberry pies, and other other low-performance motherboards and computers by offering minimal overconsumption in the image processing procedure. Common operating system platforms including Android and IOS function with the Mobilenet architecture due to high scalability and compactness in CPU consumption. Although the pretrained Faster R-CNN Inception model offers critical image detection accuracies, a combination of low scalability, inaccessibility, and graphics limitations creates an unpromising use of the model. The SSD Mobilenet architecture demands additional training to suffice the loss-accuracy values of the R-CNN model, however, offers practicality, scalability, and easy accessibility on smaller devices which reveals the SSD model as a promising candidate for further assessment.

## Pathology Diagnosis: The Cognitive Domain

The cognitive domain establishes the analytical process in a pathology diagnosis through a systematic walk-through of various strategies and tools in the collection of data and the manipulation of clinical-microscopy findings. The strategies allow a deduction of biological and clinical data to determine objective solutions. However, an overarching mechanism, known as metacognition, is exerted over these hypotheses and is checked/evaluated in relation to the data collection. Metacognition is defined as a direct-active control over one’s thought process (Pena & Andrade-Fiho, 2009). Although the Cognitive Domain is a clinical laboratory ideal, human performance complications can cause missed details during analysis and confirmation. More importantly, personal biases during the pathology investigations can cause low specificity and false-positives or false-positives during the diagnostics process. Integrating an ML approach ensures that specificity and biases are continuously regulated in the training process by using constant hyperparameters that prevent anomalies and complications during the machine evaluation. The traditional Cognitive Domain is demonstrated in figure 1.

### Figure 1.0: The Cognitive Domain

The Cognitive Domain magnifies strategies that pathologists can use to draw preliminary hypotheses and deduction based on radiologic, microscopic, and clinical data. A default (given) procedure of pattern recognition is followed throughout the tissue-blood analysis by allowing the pathologists to extract features from the image and relate this to other biological samples. Multiple branching or arborization allows a breakdown



of incoming data, while the exhaustive strategy is a rushed and uncoordinated approach towards massive data analysis.

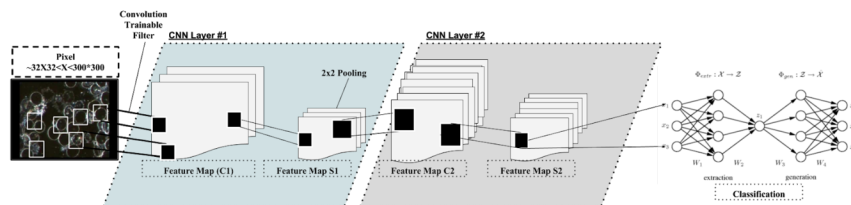
The Cognitive Domain reveals a human oriented investigation in the pathology prognosis process, however, the efficacy of the system is not universal and is sacrificed due to natural human error. Moreover, the backbone of the Cognitive Domain takes advantage of metacognition to create an efficient toolkit and guideline of strategies to be utilized during the investigation.

## Convolution Neural Network (CNN) and Image Processing

Prior to considering Faster R-CNN, it is necessary to briefly introduce the basis of Convolutional Neural Networks since this model uses this neural network as a framework. Convolutional Neural Networks have been effective in image recognition and object detection. There are four essential components of the neural network: Convolutional Step, ReLU, Pooling and Fully Connected Layer. The Convolution Step extracts features from the input image and learns the image features using matrices of input data while allowing to cultivate the spatial relationship of each feature within the image- this will produce a feature map. Non-Linearity(ReLU) is then introduced by replacing all negative pixel values in the feature map by zero in order to insert non-linearity. As a result, the network will learn to process inputs in a non-linear fashion that is similar to how data is conveyed in the real world.

Pooling reduces the dimensionality of the feature map, however it retains the important information of each in addition to minimizing overfitting by reducing the number of parameters and computations in the network. The final stage is the input approaching the Fully Connected Layer, which is a Multi-Layer Perceptron which utilizes a softmax activation function in the output layer to ensure that the outputs are probabilities that add up to 1. The softmax function does this by taking a vector of arbitrarily valued scores and minimizing it to a vector of values explicitly between 0 and 1. As one of the classes approaches a value of 1, this would create a positive inference for the detection class, leaving all other negative values near 0 to indicate a fail. At this stage in the process, the output represents high-level features of the input image. The objective of the Fully Connected Layer is to utilize the features to classify the input image into different classes dependent upon the training data (Britz, 2015).

**Figure 1.1:** Typical feature extraction in a CNN passing through pooling layers, reaching the final neural network.



## Tensorflow Object Detection API

Surfacing as a popular toolkit of machine learning technologies in early-mid 2017, the Tensorflow object detection API, released by Google, is an open source framework for object detection related tasks used for training both Single Shot Detector (SSD) and regional-Convolutional Neural Network (R-CNN) models from their model zoo (Mustamo 2018). The Tensorflow API was essentially purposed to offer scalability and potential for device deployment by Google. More importantly, Google prepared Tensorflow tools with necessary support for leading methods such as Multibox/SSD, and Fast/Faster R-CNN, which will be discussed further in this paper. The object detection API was created with an order/hierarchy of levels ranging from deployment to simple box operations. A low level API generally consists of box operations, Box representations, Target Assignment, and Visualization operations. A high level API is comprised of the heart-core structure of meta-architectures including SSD, Faster R-CNN, etc. Eventually, serving and deployment on technologies such as Jupyter Notebook and android is reached once training and visualization has been finalized. Figure 1.1 reveals an example output from the Jupyter Notebook object detection demo application.

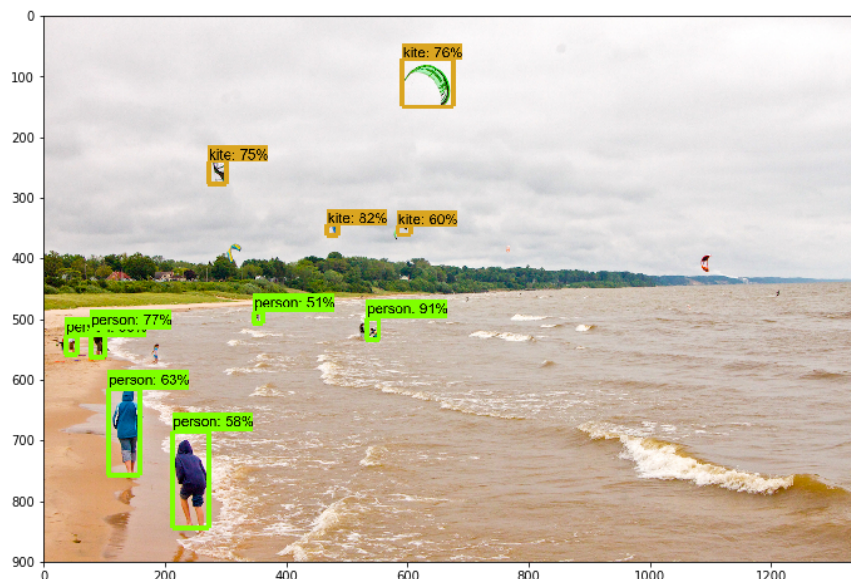


Figure 1: This is a caption

## Faster R-CNN Tensorflow Model

Building on the foundation of Convolutional Neural Networks, the Faster R-CNN essentially utilizes the CNN computed features and the Region Proposal Network half of the model by using the features collected to detect bounding boxes that have a probability of containing the object(s) of interest by obtaining bounding boxes, labels assigned to the boxes and probabilities (objectiveness score) for each label and box. The architecture for the model is as follows: Region Proposal Network, Anchors, Training/Loss, Region of Interest (RoI) Pooling, and Region Based CNN. Region Proposal Networks take an image as input and output rectangular object proposals, which are the regions believed to contain the object, with an objectiveness score. This is done by sliding a small network over the convolutional feature map output created by the last convolutional layer. The network then takes a sliding spatial window of the input convolutional feature map and the sliding window is then mapped to a lower-dimensional feature using ReLU. The newly created feature is then inputted into fully connected layers of a box-regression layer and a box-classification layer. This creates a single, unified network for object detection.

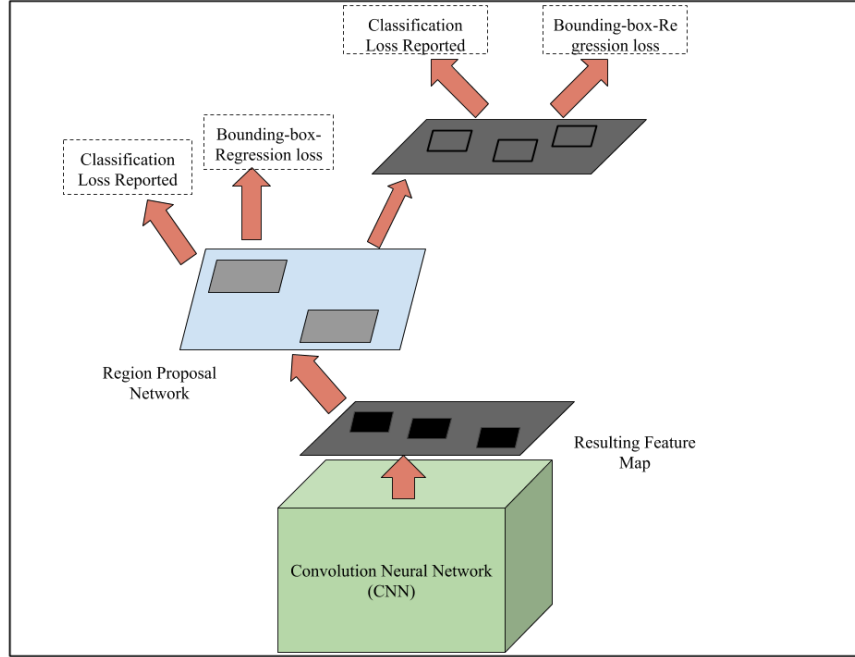
Anchors are fixed sized reference bounding boxes within the window placed uniformly throughout the original image and the anchors are both Translational-Invariant, meaning that the functions and the proposals of that anchor are translatable to varying locations of an object within the image.

Multi-Scale anchors classify and regress bounding boxes with reference to anchor boxes of varying scales and aspect ratios to address multiple scales and sizes of the images.

In training the Region Proposal Network (RPN), a class label of either being an object or not is assigned to each anchor. A positive label is assigned to anchors with the highest Intersection-over-Union overlap with a ground truth box or with an overlap with a value  $>0.7$  with any ground truth box, indicating an object has been detected. A negative label is assigned if the IoU ratio is  $<0.3$  for all ground-truth boxes, indicating there is no object detected. The RPN is able to be trained through backpropagation and stochastic gradient descent. To train the network, each mini-batch derives from a single image that contains many positive and negative example anchors in order to compute the loss function and to obtain a near 1:1 ratio (Ren et. al, 2016).

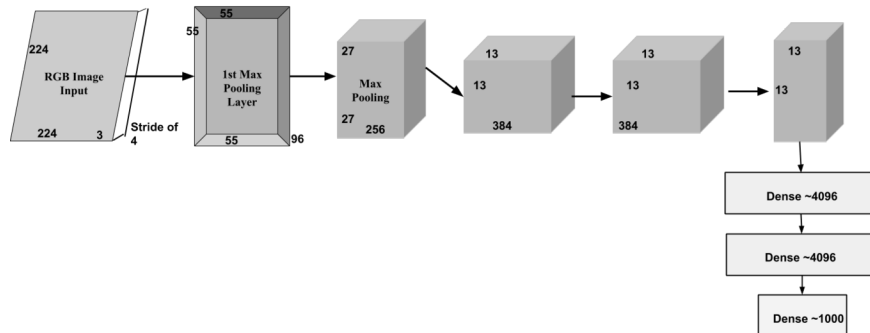
The Region of Interest Pooling (RoI Pooling) uses convolutional neural networks to detect multiple objects in an image by performing max pooling on inputs of varying sizes to obtain a fixed-size feature maps of the image by dividing the region proposal into equal sized sections, finding the largest value of each section and copying the max values into the output buffer. (deepsense ai). This layer takes an input of a fixed-size feature map obtained from a deep convolutional network with several convolutions and max pooling layers. It also takes an input of a Region of Interests  $\times 5$  matrix that represents a list of regions of interests, where the first column represents the image index and the four are coordinates of the corners of the region. Figure 1.2 reveals a simplified CNN-associated process diagram of the Faster R-CNN model:

The benefits of implementing a Faster RCNN Tensorflow model is that the training interval in which it takes in order to create a checkpoint file along with acceptable loss values is decreased by half the time as opposed to using the SSD MobileNet Tensorflow model. In addition, it also detects a higher amount of objects per image as well as increased accuracy for detection in comparison. However, the model is not able to be of use in real time object detection analysis (i.e. mobile smartphones) due to lacking the depthwise and pointwise convolutional layers, which also contributes to a slower detection rate.



## SSD (Single Shot Detector) Mobilenet v1 Architecture

The Single Shot Detector Mobilenet architecture model is derived from depth wise separable convolutions, defined as a form of factorized convolutions which factor a standard convolution into a depthwise convolution and a 1X1 convolution known as a pointwise convolution. The Mobilenet model applies a default single filter to each neural input channel to begin feature extraction. Following a depthwise convolution, a 1X1 pointwise convolution follows to combine the outputs of the depthwise convolution. The depthwise convolution will eventually split the resulting outputs into two layers, a separate layer for filtration and a layer for combination. The mixture of both output filtration and combination minimizes the model size reducing computational power demands (Howard et al, 2017). Moreover, depthwise convolution maximizes model efficiency by preventing GPU overconsumption on less demanding devices (e.g. mobile devices and laptops). However, a reduction in GPU consumption also creates a lack of usage equilibrium, which hinders the training model causing slow progress and intervals. Figure 1.3 below exhibits the consecutive and systematic pooling layers of the SSD Mobilenet model and the compression/condensing of pointwise and depthwise outputs:



The networking integrity of the Mobilenet system uses default mechanisms such as a batchnorm and ReLU

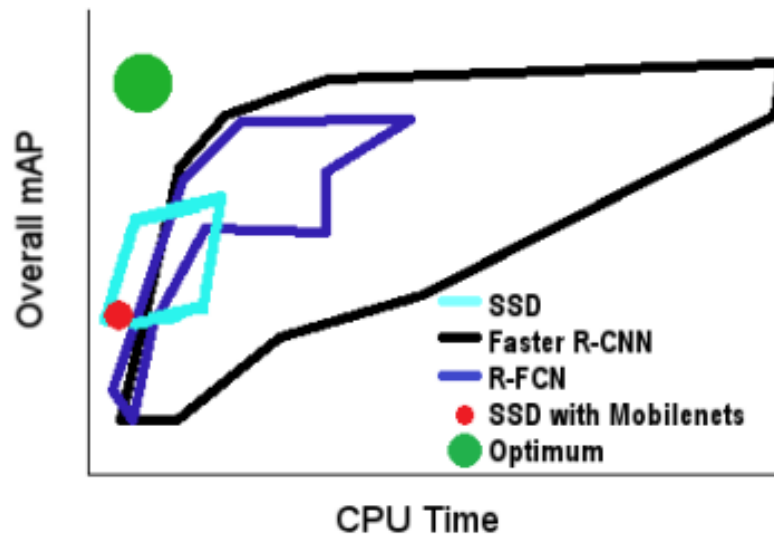
nonlinearity operation (excluding the final fully connected layer possessing no non-linearity and utilizes a softmax layer for classification) for dataset processing. The compact nature of the SSD Mobilenet framework is associated with a reduction in both parameters and Mult-adds. Although the neural network structure of SSD Mobilenet is notorious for minimal accuracy in relation to the Faster R-CNN Inception models, an intensively trained Mobilenet model can equate accuracy and precision on output object detection. Nevertheless, preliminary average precision scores (mAP) on small background objects reveal reduced values in relation to the inception models. Table 1.0 exhibits the pretrained COCO-Tensorflow models and their relationships in speed (ms) and mAP(^1):

Coco Model Name	Speed (ms)	mAP <sup>[1]</sup>
SSD Mobilenet V1	30	21
SSD Mobilenet V2	31	22
SSD Lite mobilenet v2	27	22
SSD inception v2	42	24
Faster RCNN Inception v2	58	28
Faster RCNN ResNet 50	89	30
Faster RCNN ResNet 50 Low	64	30
<b>Proposals</b>		
RFCN-ResNet 101	92	30
Faster RCNN ResNet 101	106	32
Faster RCNN ResNet 101	82	32
<b>Low Proposals</b>		
Faster RCNN Inception	620	37
ResNet Atrous V2		
Faster RCNN Inception	241	37
ResNet Atrous v2 Low		
<b>Proposals</b>		
+	1833	43
Faster RCNN Nas Low	540	43
<b>Proposals</b>		

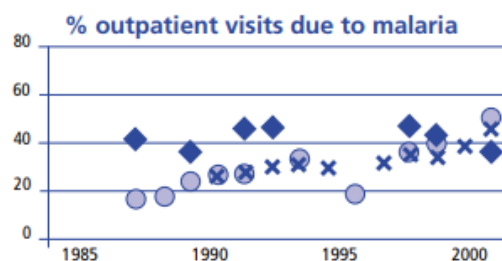
Table 1.0 (above) and Figure 1.4 (below): Relationships between typical object detection API models and their associated speeds and measures of average precision. The models of interest, SSD Mobilenet and Faster RCNN Inception, exhibit the most compatibility for future deployment with a coupling of efficiency and precision.

## Prevalence of *Plasmodium Malariae* and *Treponema Pallidum*

Particularly in 3rd world impoverished countries, the onset generation of devastating diseases has critically impacted communities by establishing communicability into waterways, sewers, and the common plumbing systems of undeveloped neighborhoods and towns. Although mobile medical departments have emerged in regions South of the African Sahara, immediate and handheld attention is scarce in households. Ultimately, individuals with ambiguous symptoms have minimal resource and awareness in detecting parasitic and onset disease types. More importantly, local pathology departments do not ensure full scale access to the broad community which translates into delayed diagnostic times and disruptions in quality medical intervention. Nearly 90% of present-day Malaria cases are rampant in Africa south of the Sahara. Further, in regions of free flowing Malaria transmission, both young children and pregnant women are vulnerable as a potential disease group being at risk of high morbidity and mortality. Children experiencing an initial Malaria infection experience behavioral-physical symptoms in their first year or two of life.



The upsurge of Malaria in the young demographic is partly caused by inherent limitations in regional Obstetric departments and little mechanisms for on-birth diagnosis. Malaria creates cellular dysfunction and results in death by establishing an infection during pregnancy, an Acute Febrile illness, or Chronic repeated infection. An infection during pregnancy results in low birth weight during preterm delivery, whereas an Acute Febrile illness (a rapid onset of fevers, headaches, chills or muscle-joint pain) escalates Cerebral malaria causing a combination of respiratory distress and hypoglycaemia (World Health Organization, 2013). Chronic exposure and repeated infections result in severe anaemia leading to eventual death in children. According to the World Health Organization, outpatient clinical visits in Sub-Saharan-South Africa have been climbing the rungs by nearly 60% in regions including Malawi, UR Tanzania, Uganda, and Zambia since its emergence in 1985-2000. Increasing facilitation and global health initiative has been organized in regions resulting in the inhibited spread of Malaria across generations, however, little disruptive medical-technology implications have arose from 21st century clinical involvement in Africa.



The other hand of disease control is deals with Syphilis, or *Treponema Pallidum*. Syphilis is underlined as an infectious disease transmittable through sex or vertically during a child's pregnancy. The physiologic and psychological map of Syphilis is characterized by periods of activity and latency, erratic systemic involvement, and a progression to acute complications in patients remaining untreated or having been inadequately treated. The diagnostic cycle for Syphilis is simply divided into both a recent case and a late case: A recent case is marked within a one year living period, whereas, following a one year bacterial presence, a late diagnosis is apparent. The current form of on point and personal diagnosis is a lookout for skin lesions, typically



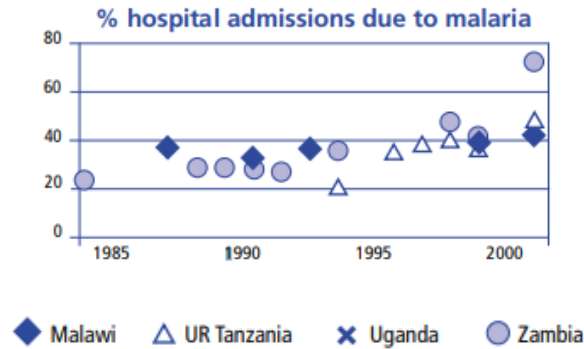


Figure 2: **Figure 1.5- Source:** Routine health information system data, ARFO, for under-fives, Against Malaria Foundation

indicating primary Syphilis. However, skin lesions further delay the 1 year diagnostic stage by allowing the bacteria to continually replicate and eventually revealing skin lesions. Ultimately, superficial imaging techniques do not suffice immediate and real time diagnostic reports personally for both the patient and physicians in the region.

## Dark Field Microscopy: Live Blood Analysis and Accessibility

In broad terms, dark field microscopy is critical in the diagnosis of vital blood which is critical as an important holistic universal medical examination process carried out by clinics around the world. Although dark field microscopy reveals the internal environment (e.g the behavior of blood plasma, blood volume, and content), the macroscopic influences of small protein molecules in the human body from which microorganisms can arise are imaged on the sample. During dark field investigations, a drop of fresh blood is examined under the microscope without staining and the blood taken from the patient is put directly onto the slide and covered with a glass coverslip. Avoiding contamination and maintaining relative sterilization, the slide is examined immediately under 1200x magnification. The blood is ultimately exposed to severe stress in slide preparation through a lack of oxygen, non-circulation and exposure to intense light (Rau 2016). More importantly, dark field microscopy reveals the early and indirect signs and tendencies towards internal body conditions: By revealing the circulatory, and degenerative malign illnesses in the blood, doctors can use this as a precursor in a full case investigation/diagnosis. Due to the compact and non-staining requirement of dark field microscopes, immediate accessibility is maximized without strict laboratory preparation. Recent investigations in 2015 underlined the use of dark microscopy imaging with high power LED illumination, in contrast to traditional image staining techniques and halogen lamps. High power and energy efficient LED light bulbs have continually replaced halogen lamps, enhancing darkfield applications marking darkfield microscopy as an increasingly accessible imaging option. Figure 1.5 below reveals a high contrast imaging of *Treponema Pallidum* under a traditional dark field microscope.

Figure 1.5: High contrast dark field imaging of *Treponema Pallidum* (Syphilis) blood sample. The image exhibits the potential for clear-sharp extraction of the specimen for diagnosis.



## Disease Control in Impoverished 3rd World Countries

Health systems in developing countries typically have two objectives in mind to enhance social, economic, and medical advantages in the area: a.) improving the level and distribution of health outcome in the population and b.) to protect individuals from financial risks that are frequently very substantial and cause impoverished lifestyles (World Health Organization, 2000). Common financial risks are caused by illness-related loss of income and the drastic expenditures on treatment and care. However, much of this loss can be ameliorated by preventing illness or its progression by using new medical-financial architectures in the region. This paper centralizes both Malaria and Syphilis as a disease demographic, with a 90% prevalence in South Africa, however, also takes advantage of visualization techniques in a compact and easy diagnosis of human diseases. According to the Global Burden of Disease Study in 2010, diseases including Malaria and Syphilis, have been surprisingly masqueraded and unmentioned by local medical communities and departments in South African countries. This continued lack of awareness is partly rooted from the dense population pool (South- Sub Saharan Africa) experiencing this disease and the lack of regulation, assessment and modern diagnostic tools.

## Proposal and Rationale: Real time Object Detection in Disease Diagnostics

The objective in deploying use of object detection algorithm models in disease diagnostics is to utilize the Faster R-CNN model in order to detect Malaria and Syphilis in a hospital setting, due to the blood samples via dark field microscopy needing to be analyzed as frozen/still images. In addition, this model provides a significantly higher precision and accuracy rate when being able to detect every instance of the pathogens in each image. As seen on the figures and graphs of the Data Analysis portion

The plan to utilize the SSD MobileNet model is for the model to be used in an mobile Android application as to detect pathogens in real time. Theoretically, the rationale is to create a Raspberry Pi device connected to a microscope that is able to gather blood samples and for the SSD MobileNet model to accurately detect instances of the pathogens in these samples. This would be of benefit as it Unfortunately, the accuracy in which the model detects each individual instance of the objects is significantly less than the Faster R-CNN

model as shown in the figures and graphs on the Data Analysis portion of this paper. Therefore, it is ideal to utilize the SSD MobileNet model in order to accomplish this goal.

## Methods and Protocols for Faster R-CNN Inception V2 COCO Model and SSD MobileNet V1

Before considering the methodology, a brief overview of the Tensorflow Object Detection API and data collection methods are overviewed:

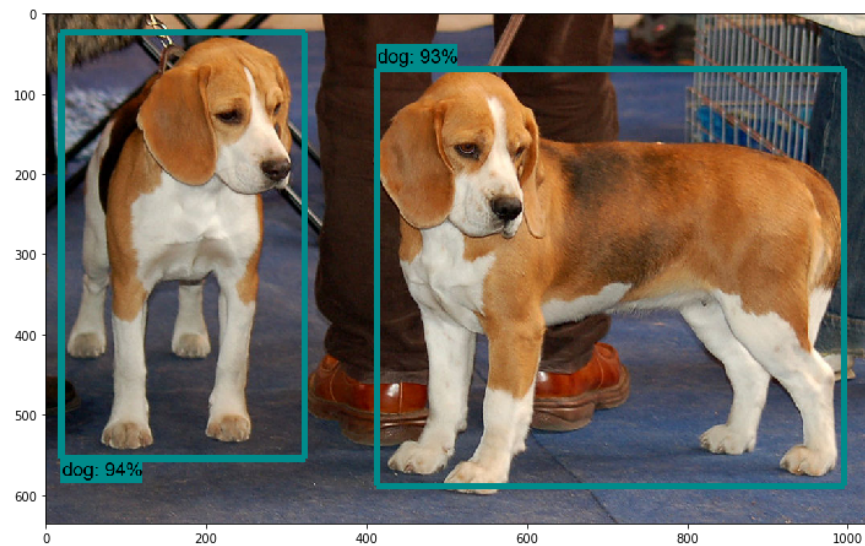
1. Install TensorFlow-GPU v1.5, CUDA v9.0 and cuDNN v7
2. Download official Tensorflow Object Detection API repository from GitHub and setup the Tensorflow directory in the Windows C:/ drive
3. Download the Faster-RCNN-Inception-V2-COCO and SSD Mobilenet V1 models from the official Tensorflow model zoo and place inside of the \object\_detection folder
4. Set up the Anaconda virtual environment and install packages: anaconda, protobuf, pillow, lxml, Cython, jupyter, matplotlib, pandas, opencv-python
5. Configure PYTHONPATH environment variable to point to the correct directories in \models-masters
6. Compile Protobufs and run setup.py in the Anaconda Command Prompt in order to create a name \_pb2.py file from every name.proto file in the \protos folder
7. Test the Tensorflow Object Detection API to verify it has been set up successfully by launching the object\_detection\_tutorial.ipynb script with Jupyter
8. Gathered 31 images of Syphilis and 31 images of Malaria and saved into the images folder of the object detection directory
9. Open folder of images in the LabelImg software and create a new edited folder in LabelImg in order to hold the new .xml files once all of the verification boxes were saved in each image containing the every instance of the object of interest
10. Generate the training data by converting the .xml to .csv files in order to be read by the object detection classifier
11. Create a label map that tells the trainer what each object is defined by mapping class names (Malaria and Syphilis) to class ID numbers (Malaria:1, Syphilis: 2) and save the map as a .pbt.txt file
12. Configure the object detection training pipeline. This is done by changing the number of classes to 2, changing the fine tune checkpoint to contain the Faster RCNN Inception V2 COCO model, changing the number of examples to 62 images and updating the eval\_input\_reader section, specifically the input\_path and label\_map\_path to contain the directory of the test.record input path and the labelmap.pbt.txt in the label map path.
13. Initiate training
14. Export the inference graph and modify the last saved checkpoint value post-training
15. Modify image path in python config file via idle to a newly saved image
16. Run command from Anaconda Prompt

## Methodology

### Implementing the Faster RCNN Inception V2 2018 and the SSD Mobilenet V1 2017 Models

Prior to initiating data collection and training, establishing a Tensorflow and Anaconda environment with default Python package imports (e.g. Numpy, Anaconda's Protobuf, lxml, Cython, and Opencv) was neces-

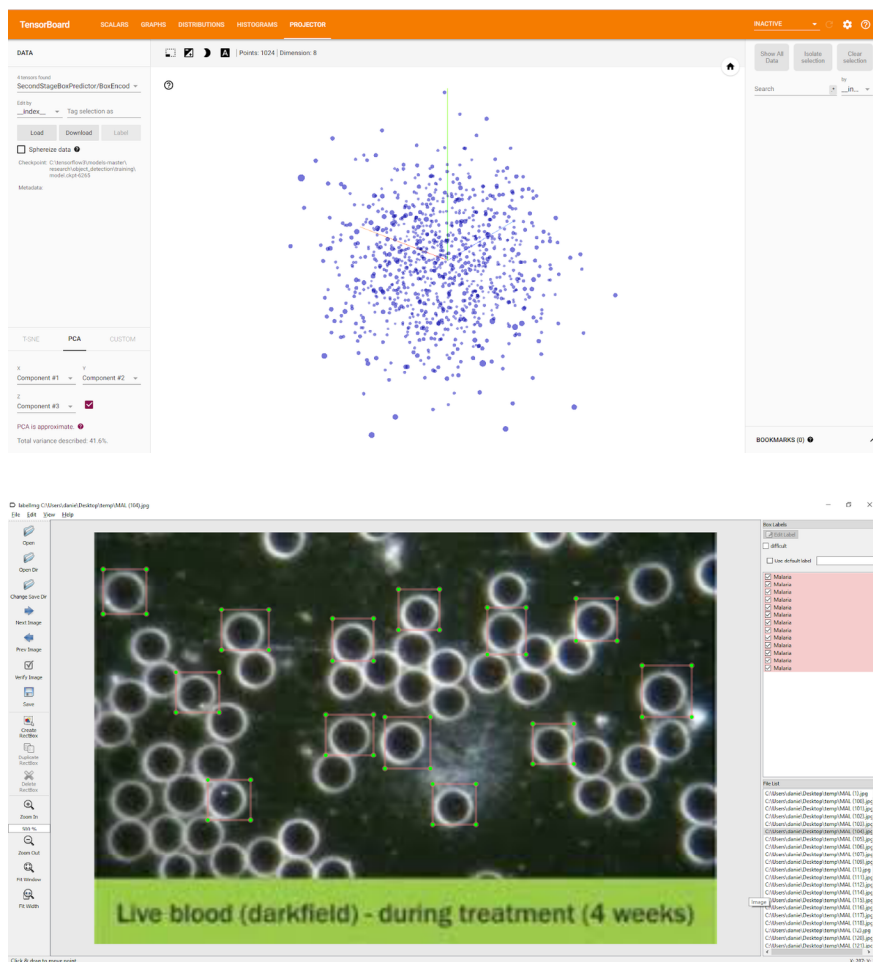
```
C:\WINDOWS\system32\cmd.exe - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.c...
device.cc:1195] Creating TensorFlow device (/device:GPU:0) -> (device: 0, name: GeForce GTX 1060 6GB, pci bus id: 0000:01:00:00, compute capability: 6.1)
INFO:tensorflow:Restoring parameters from C:/tensorflow1/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
INFO:tensorflow:Starting Queues.
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Recording summary at step 0.
INFO:tensorflow:global step 1: loss = 2.6708 (5.383 sec/step)
INFO:tensorflow:global step 2: loss = 3.0352 (0.251 sec/step)
INFO:tensorflow:global step 3: loss = 3.4884 (0.204 sec/step)
INFO:tensorflow:global step 4: loss = 2.9733 (0.193 sec/step)
INFO:tensorflow:global step 5: loss = 2.2184 (0.191 sec/step)
INFO:tensorflow:global step 6: loss = 2.0321 (0.554 sec/step)
INFO:tensorflow:global step 7: loss = 2.0424 (0.211 sec/step)
INFO:tensorflow:global step 8: loss = 2.0252 (0.208 sec/step)
INFO:tensorflow:global step 9: loss = 2.0053 (0.194 sec/step)
INFO:tensorflow:global step 10: loss = 1.3622 (0.193 sec/step)
INFO:tensorflow:global step 11: loss = 1.8027 (0.197 sec/step)
INFO:tensorflow:global step 12: loss = 1.2485 (0.196 sec/step)
INFO:tensorflow:global step 13: loss = 1.0712 (0.193 sec/step)
INFO:tensorflow:global step 14: loss = 1.6604 (0.189 sec/step)
INFO:tensorflow:global step 15: loss = 1.2657 (0.192 sec/step)
INFO:tensorflow:global step 16: loss = 1.4351 (0.193 sec/step)
INFO:tensorflow:global step 17: loss = 1.2152 (0.192 sec/step)
INFO:tensorflow:global step 18: loss = 1.1165 (0.197 sec/step)
INFO:tensorflow:global step 19: loss = 1.6557 (0.192 sec/step)
INFO:tensorflow:global step 20: loss = 1.7777 (0.200 sec/step)
```



sary. Following package construction, the object detection model was then downloaded from the Tensorflow Model Zoo, a holding repository for all current models in both object detection and image classification. In this case, both the Faster-RCNN-Inception-V2-COCO and SSD Mobilenet V1 2017 models were packaged and installed into the directories for testing.

Additionally, the official Tensorflow/models repository was extracted in order to obtain the correct files needed to work with the model. After importing packages and setting up the environment for training, we had tested the setup in order to verify that it was successful by running an object detection tutorial in the Jupyter notebook. In order to train our object detector, we had created our custom file of dark field microscopy images of Syphilis and Malaria. In addition, we had created verification boxes of every instance of the targeted objects in each image to be created into a .xml file for training, using LabelImg. We had focused on a small number of images, resulting in a total of 31 for Syphilis and 31 for Malaria. The .xml files were then converted to .csv files in order to be read by Tensorflow. We then made our own label map with two classes: class 1 for Malaria and class 2 for Syphilis in order to train the classifier to detect these objects. The last method was to configure the object detection training pipeline. Since the pipeline defines which model and what parameters will be used for training. In the configuration file, the num\_classes, fine\_tune\_checkpoints, num\_examples were altered in addition to the train\_input\_reader and eval\_input\_reader sections to fit our

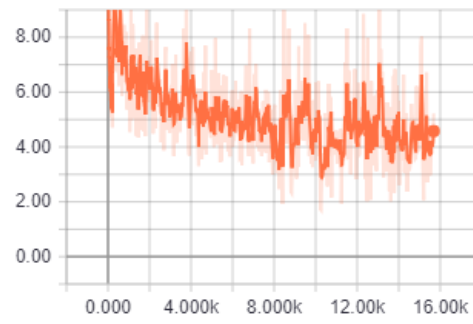
custom path directories. After this has been completed, the model was then trained through the Anaconda command prompt. The live data indicating loss values was then recorded via Tensorboard.



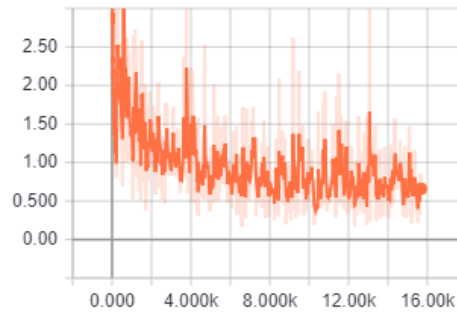
## Data Analysis: SSD Mobilenet V1 Training Loss Scalars

**Figures 1.6-1.9:** Automated Tensorflow Object Detection API data was collected throughout the global steps as seen in the command prompt. Total loss parameters, as acknowledged in the CNN background, are ultimately summation functions of the biases and weights of the trained neural network (which result from the Softmax function and its counterparts). Although exponential loss decay is not comparable to the drastic decrease in the Faster RCNN Inception model, the SSD Mobilenet model maintained a high speed of ~30 ms for full image analysis. Based on the Single Shot approach of the neural mobile network, training and loss values did not converge at low values immediately. Loss value convergence occurred around 8,000 steps near a value of 4.00. Nevertheless, the SSD architecture allowed realtime analysis of images (e.g. an OpenCV webcam window which detected live images). Thus, although loss value, in proportion to precision, was lower for the SSD Mobilenet model, immediate image visualization was easier.

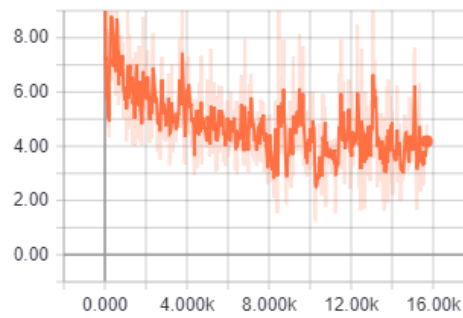
Losses/TotalLoss



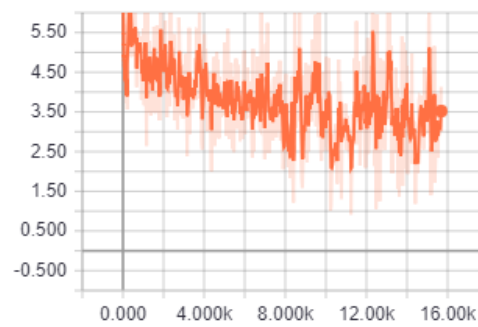
Losses/Loss/localization\_loss



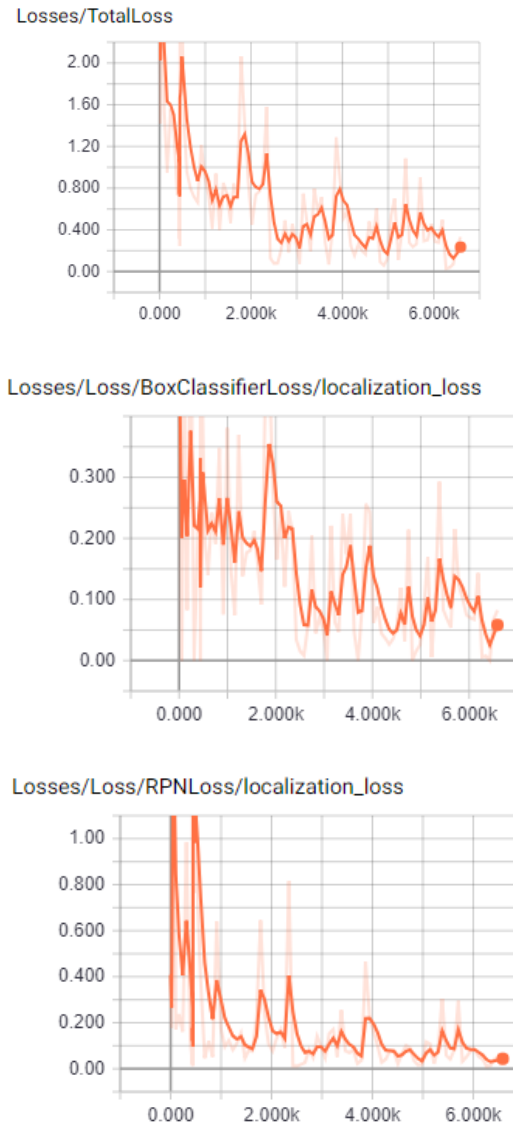
Losses/clone\_loss



Losses/Loss/classification\_loss

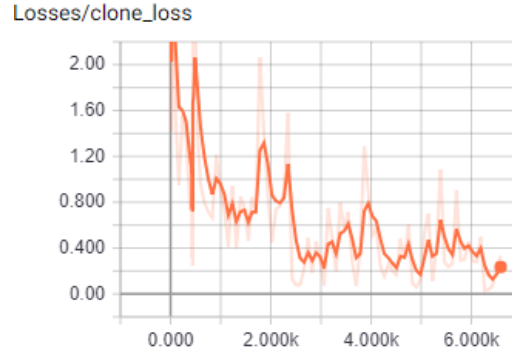


## Data Analysis: Faster RCNN Inception V2 Training Loss Scalars



**Figures 2.0-2.3:** The Faster RCNN Inception V2 model reveals exceptional loss values converging near minimum values of  $\sim 0.05$  during the R-CNN training. Total loss values converged near steps ranging in 4,500-6,000, although, experienced natural fluctuations. Evidently, the Faster RCNN model reveals higher scalability in terms of training parameters. In other words, loss values decreased drastically while precision inversely increases within a range of 10,000 global steps. However, the Faster RCNN model is notorious for overconsumption of GPU contained RAM. On NVIDIA Cuda cores, nearly all accelerators are utilized during the training process which minimizes versatility of the model and the potential for deployment. The training model accurately localized smaller cells and borderline features on the image (i.e. not general-apparent details). Following the Python shell testing, it was apparent that the Faster RCNN demanded more time for image analysis at  $\sim 58$  ms. Due to the high-scale nature of the R-CNN model, real time analysis is limited to  $\sim 2$ -4 frames per second (FPS), which eliminates potential for deployability on small device operating systems such as Android and IOS.





## Object Detection API Deployment: Android OS Platforms

The rationale of this project-investigation is grounded in versatility and the universal nature of an pretrained object API neural network model. More importantly, after a review of disease burdens in regions and countries of South Africa, a standard of accessibility must be attained to allow rapid handheld usage. The levels of Tensorflow API usage are defined by the interactivity of the application itself. The project ultimately moved from a stage of moderate-low complexity (a Python shell console that required necessary path directory modifications) to a fully automated application that used a preloaded frozen\_inference\_graph.py to recognize novel images. The SSD Mobilenet V1 model was the sole candidate in performing real time 30 FPS rate object detection due to its compressed size and capacity. Figure 2.4 reveals the Android application detecting two Syphilis cells in a blood-specimen culture:

The android application was initially configured by utilizing Tensorflow's open source android directory which enables Machine Learning deployment tools on mobile devices. Path definitions in the camera detection activity JAR files of the directory were modified to ultimately match the location of the SSD Mobilenet Frozen Inference graph.

## Object Detection Analysis: Faster RCNN v2 vs SSD Mobilenet v1

### Data Interpretation and Conclusions: SSD Mobilenet and Faster R-CNN Comparisons

The combined approach of a Faster Regional Based Convolutional Neural Network and a Single Shot Detection Mobilenet architecture are apparent as they offer unique dimensions of accuracy, efficiency, and compatibility in immediate blood-data analysis. Although the Faster R-CNN model revealed minimal converging loss values, a standard of speed and immediate detection was sacrificed. In other words, detection time was extended and required demanding GPU performance to prevent crashing and processing overload. The SSD Mobilenet architecture provided moderate precision-accuracy while providing high speed, less demanding CPU-GPU consumption. With a modified batch size of ~5, the SSD Mobilenet model rapidly passed through global steps reporting decreasing loss values. More importantly SSD Mobilenet created an opportunity for scalability and accessible deployment by adapting to the architectures of small operating systems such as Android and IOS.





## Future Applications

The broad constraints of a darkfield-based object detection application were sufficed in terms of moderate accuracy and compactibility in the SSD Mobilenet model. More importantly, the potential of Android-OS device deployment of the SSD Mobilenet architecture enables high accessibility and versatility on other cross-platforms. Moreover, the compressed nature of the SSD Mobilenet architecture allows precise object tracking and more rapid image analysis/processing time. In contrast to the Faster R-CNN model, single shot detection offers high general scalability. Primary future developments suggest that the SSD Mobilenet model may be integrated into a full scale application for multiple diagnostic disease detection. Additionally, to equate precision and accuracy readings with the Faster R-CNN model, continued GPU training is necessary. This project was completed on a local network with insufficient graphics support, however, future work in a high-performance research setting with necessary computational-graphics resources is required for considerable improvement in detection efficacy. Considering the potential of medical visualization in the process of diagnostics reports, cell segmentation techniques plan on being applied using an extension of the Faster R-CNN architecture known as “Mask-R-CNN” which applies regional “masks” and highlights critical areas of the image based on borderline features.

Opencv Image + Analysis	Step Count	Type	Object API Model
	15410	Syphilis	SSD Mobilenet v1
	15410	Malaria	SSD Mobilenet v1
	15410	Syphilis	SSD Mobilenet v1
	6265	Malaria	Faster R-CNN Inception v2
	6265	Syphilis	Faster R-CNN Inception v2
	6265	Syphilis	Faster R-CNN Inception v2

## Bibliography

An Intuitive Explanation of Convolutional Neural Networks. (2016, August 11). Retrieved July 3, 2018, from <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

American, C. (2015, March 30). Reasons for delays in getting your biopsy and cytology test results. Retrieved July 3, 2018, from <https://www.cancer.org/treatment/understanding-your-diagnosis/tests/testing-biopsy-and-cytology-specimens-for-cancer/how-long-does-testing-take.html>

Dark field microscopy. (2011). Paracelsus Clinica,1-2. Retrieved July 3, 2018.

Foundation, A. M. (2008). The burden of malaria in Africa. 1-7. Retrieved July 3, 2018.

Grel, T. (2017, February 28). Region of interest pooling explained. Retrieved July 3, 2018, from <https://deepsense.ai/region-of-interest-pooling-explained/>

Howard, A. G. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Arxiv: Computer Vision and Pattern Recognition,1-9. Retrieved July 3, 2018.

J. (2018, January 18). Faster R-CNN: Down the rabbit hole of modern object detection. Retrieved July 3, 2018, from <https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>

Jha, P. (2013). Disease Control Priorities in Developing Countries, 3rd Edition Working Paper #2. Economic Evaluation for Health,1-66. Retrieved July 3, 2018.

Johnson, J. (n.d.). Convolution Neural Networks for Visual Recognition. Retrieved July 3, 2018, from <http://cs231n.github.io/>

Kennedy, E. J., Jr. (1996). DARKFIELD MICROSCOPY FOR THE DETECTION AND IDENTIFICATION OF TREPONEMA PALLIDUM. 1-18. Retrieved July 3, 2018.

Liu, W. (2015). SSD: Single Shot MultiBox Detector. Arxiv: Computer Vision and Pattern Recognition,1-17. doi:10.1007/978-3-319-46448-0\_2

Liu, W. (n.d.). SSD:Single Shot MultiBox Detector. Retrieved July 3, 2018, from <http://web.cs.ucdavis.edu/~yjee/teaching/ecs289g-winter2018/SSD.pdf>

Lu, Z. (n.d.). Tensorflow detection model zoo. Retrieved July 3, 2018, from [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md)

Mogeni, P., & Williams, T. (2017). Detecting Malaria Hotspots: A Comparison of Rapid Diagnostic Test, Microscopy, and Polymerase Chain Reaction. The Journal of Infectious Diseases,216(9), 27th ser., 1091-1098. Retrieved July 3, 2018.

Mustamo, P. (2018). Object detection in sports: TensorFlow Object Detection API case study. University of Oulu: Faculty of Science,1-43. Retrieved July 3, 2018.

Organization, W. H. (2016). Multiplexed Point-of-Care test for acute febrile illness (mPOCT). 1-2. Retrieved July 3, 2018.

Pena, G. P., & Andrade-Filho, ;. (2009). How Does a Pathologist Make a Diagnosis? Archives of Pathology and Laboratory Medicine and Archives of Pathology,1-9. Retrieved July 3, 2018.

Regazzi Avelleira, J., & Bottino, G. (2006). Syphilis: Diagnosis, treatment and control. Continuing Medical Education,111-122. Retrieved July 3, 2018.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 1-14. Retrieved July 3, 2018.

Xu, J. (2017). Deep Learning for Object Detection: A Comprehensive Review. Towards Data Science. Retrieved July 3, 2018, from <https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>.