Master's Thesis

An SSD Multi-box Approach to Automated Malaria Diagnosis on Thick Blood Smears

BY

Natali Alfonso Burgos

in fulfillment of the requirements for the degree of

Master of Science in Artificial Intelligence

Supervisors:

Internal	Internal	External
Tom Heskes	Johan Kwisthout	Leticia Estrella
Department of Data Science	Department of Artificial Intelligence	Machine Learning Chief
Radboud University	Radboud University	Orikami

Radboud University

Department of Artificial Intelligence
Radboud University
The Netherlands

Submitted: August 3, 2018

Contents

List of Figures

List of Tables

Abstract

In 2017, the World Health Organization (WHO) reported 216 million cases of malaria worldwide (91 countries), an increase of about 5 million cases over the year 2015. An early diagnosis is the best prevention against malaria, and microscopy remains to date the gold standard for malaria diagnosis. Yet, in most affected regions, shortage in expertise and costly laboratory equipment grant limited access to an adequate diagnosis. This thesis introduces a high-performing, state-of-the-art, real-time object detector (SSD Multi-box) to the problem of malaria detection. Using deep convolutional networks, it responds to the presence of trophozoites of *P. falciparum* and white blood cells in Field-stained thick blood smears. The detection of white blood cells allows for parasitaemia estimation, a requirement to assess the severity of the infection, and in turn, the type of treatment. SSD Multi-box localizes and recognizes *P. falciparum* (recall: 87.52%; precision: 90.72%) and white blood cells (recall: 97.54%; precision: 81.27%) on a single feed-forward pass of network at a rate of 9 FPS. Our model outperforms results from similar multi-class studies, yet offering unprecedented time performance even on low-res image sets, such as the one used here. Transfer learning did not show great improvement, yet results suggest that transferring features extractors from early layers can be beneficial.

Acknowledgements

I would like to thank Tom Heskes and Leticia Estrella for their excellent supervision, patience and encouragement throughout the entire project. To Leticia Estrella, again, for guiding me through the technical intricacies of the project, and to Orikami for the opportunity to be part of this project.

I want to express special gratitude to Tjeerd Dijkstra, for his advisory role, and countless proof-readings and feedback on this manuscript. To my family, for their unconditional love and support. To my friends and classmates, that helped me keep my sanity.

Last but not least, I must express my very profound gratitude to my partner Isa, for her absolute support and continuous encouragement through the entire research process. This would not have been possible without you.

1 Introduction

1.1 Background: malaria etiology and diagnosis

Malaria is a mosquito-borne, life-threatening disease caused by parasitic microorganisms of the *plasmodium* type. These microorganisms infect female *Anopheles* mosquitos, where they grow into *sporozoites* that will infect the next host. Plasmodium species infect humans, primates, mammals, birds and reptiles. Mosquitoes act as vectors that carry them from host to host. In Africa, plasmodium parasites that infect human hosts appear in four distinct species: *falciparum*, *vivax*, *ovale* and *malariae*. *Falciparum* is the most frequent strain in the vast sub-saharan region and, unfortunately, is the most lethal: it causes almost every malarial death [WHO, 2008].

Once in the bloodstream, parasites grow and multiply in red blood cells (RBC). They first turn into ring-shaped organisms or *trophozoites*, mature into *schizonts* that break open releasing *merozoites* to later on differentiate, only some, into *gametocytes* (sexual stage). When a female *Anopheles* mosquito takes a blood meal from an infected human, *gametocytes* are collected and a new cycle of growth and multiplication starts in the mosquito.

Plasmodium parasite hosts present flu-like symptoms, typically including fever, fatigue, vomiting, and headaches. Due to the generality of these symptoms, symptom-based diagnosis is often unclear and ambiguous. Blood testing is indispensable to confirm the infection. The most widespread diagnostic techniques to date are (a) microscopic examination of blood films and (b) antigen-based rapid diagnostic tests (RDT).

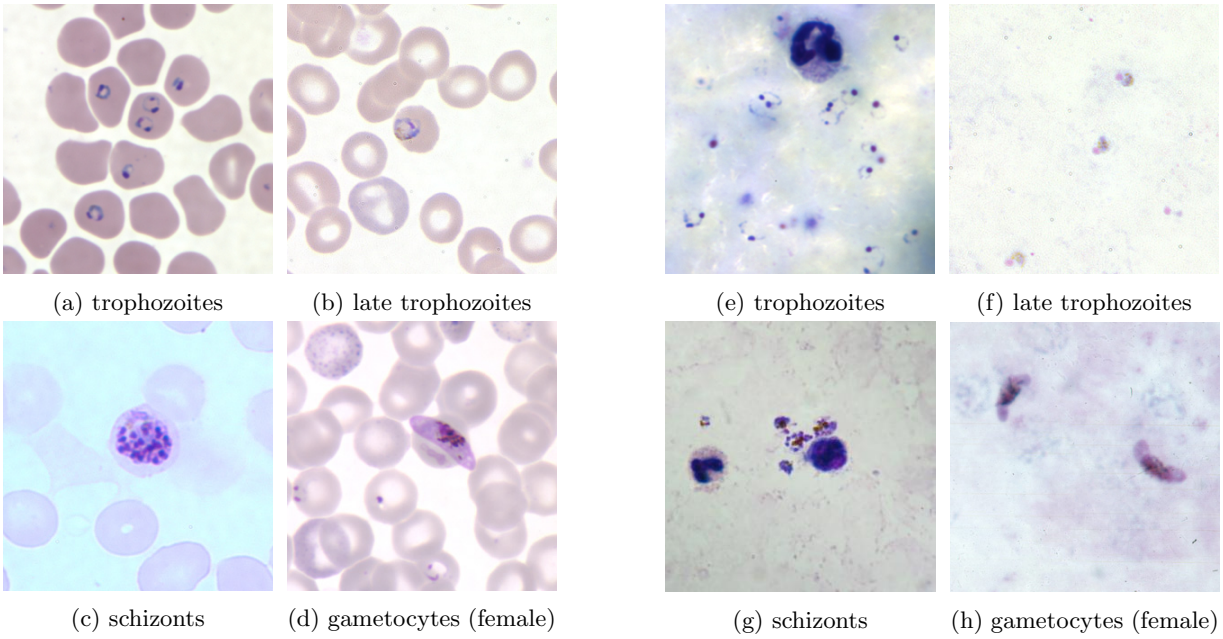
1.1.1 Microscopy: preparation of blood films

Microscopy is the "gold standard" for malaria detection [Caraballo and King, 2014]. Microscopic tests involve the collection, staining and (light) microscopy examination of a blood smear. Blood samples are typically acquired via venipuncture, that is a finger prick or an ear lobe stab. A few drops of blood (a drop of blood is estimated to be $\sim 50\mu l$) are collected on a laboratory glass slide or film[1], and prepared into either a thin or a thick smear [Makler et al., 1998].

A thin blood smear is a drop or two spread across a large area of a film such that the thickness decreases progressively toward the "feathered" edge. In this edge, blood cells are disperse, arranged side-to-side into a single layer of blood. Before staining, blood smears are air-dried. It takes approximately 10 minutes for a thin smear to dry. Immediately after, dried samples are dipped into 100% methanol. The staining exposes RBC, white blood cells (WBC), platelets and malaria parasites (contained in RBC).

Differently, a thick blood smear is a drop spread with a circle movement on a glass slide, such that the diameter of the drop widens. In thick smears there are approximately 6 to 20 stacked layers of blood, which results in a larger volume of blood to be examined per field-of-view[2]. Unlike thin smears, thick smears are not fixated with methanol. It takes approximately 30 minutes for a thick smear to dry before one can carry forward with its staining. RBC are destroyed by oxygen while air-drying and are not visible in thick smears. The aftermath, then, is a sample where WBC, platelets and malaria parasites can be spotted.

[1] glass slide and film will be used interchangeably in this text
[2] A view-field or field-of-view is the area of the sample that can be seen through the microscope.



(a) trophozoites (b) late trophozoites (e) trophozoites (f) late trophozoites

(c) schizonts (d) gametocytes (female) (g) schizonts (h) gametocytes (female)

Figure 1: **Thin and thick blood smears of P. falciparum.** Different life stages of P. falciparum can be observed both in thin and thick smears. Figures (a-d) correspond to images of thin blood smears, whereas figures (e-h) correspond to thick blood smears [Center for Disease Control and Prevention].

Field stain is a common choice for the staining of thick blood films. Though Giemsa stain is preferred in malaria diagnosis, in particular in the staining of thin smears, often times delivers unsatisfactory results due to unclean utensils or imprecise preparation of the solution. Field staining is faster and more stable. If well-stained, malaria parasites show deep red chromatin and pale blue cytoplasm. White blood cells nuclei stain purple and the background, pale blue. *Schizonts* and *gametocytes* can be recognized under Field stain too, if present. However, is frequently used for the recognition of *trophozoites* of *P. falciparum*. The staining procedure consist of dipping the blood film into two different solutions (Field A and Field B), each time for 10-30 seconds. Remaining chemicals are washed out gently under running water in between applications, and let to air dry for a few minutes. Preparation of solutions and their applications can take up to 30-45 minutes [WHO, 2010a].

The staining procedure requires cautious execution, and adequate tools and materials to assure minimum quality conditions for malaria identification. Unfortunately, these requirements are often not met in remote, low-resource areas. The lack of skilled personnel and the poorly equipped clinical facilities account for the main causes of misdiagnoses of malaria infections [Caraballo and King, 2014].

1.1.2 Microscopy: examination of blood films

The time spent on examining a blood smear has a large impact on the efficacy of microscopy. Thick smears are often used for detecting the presence of malaria parasites, since it allows for a faster diagnosis. Despite the preparation of thick smears being comparably time-consuming (about an hour), it is 10-30 times more sensitive [3] than thin smears even with very low parasitaemia[4] (detection in thick smears is lower-bounded by 15 parasites/μl). In thin

[3]Sensitivity as a measure of *discriminability*, not as a measure of detection performance. That is, how hard or easy is it to detect the target stimuli from background events. It is closely related to the concept of signal-to-background ratio.

[4]Parasitaemia is the quantitative content of parasites in the blood. It is used as an indication of the degree of an active parasitic infection

smears with a parasitaemia lower than 150 parasites/μl, parasites are rarely detected since, theoretically, it would require on average 30 times longer to find a parasite [Thellier et al., 2002]. The examination time needed on a thin smear in order to obtain a good parasitaemia estimate, in practice, becomes unfeasible.

According to WHO guidelines, parasitaemia estimates are a requisite in malaria diagnosis, even more so when the infection is caused by *P. falciparum* [Filippov and Glazunova, 1988]. Parasitaemia reports lay out the severity of the infection, and provide lead guidance on drug treatment and follow-up procedures [Tangpukdee et al., 2009].

Standard clinical protocols suggest that a minimum of 200 distinct fields-of-view of a thick smear need to be examined before a slide is regarded as malaria-negative [WHO, 2010b]. This takes approximately 5 to 10 minutes before the technician can offer a diagnosis. An alternative rule of thumb is WBC count. There are several ways of estimating parasitaemia by WBC count. Yet, the time required is roughly the same.

In any case, further examination of thin smears is often carried out. Thin smears allow for the identification of malaria species and their stages. That is, it allows to determine the severity of the infection (parallel to parasitaemia estimates), and to rule out alternative diagnosis based on the patient's symptoms and the life cycle of the parasite. Seemingly, this same task in a thick smear is greatly dependent on the microscopist' experience [Yitbarek et al., 2016]. The RBC rupture caused by oxygen during the air-drying stage, leaves a distorted appearance of parasites while other relevant visual cues vanish (e.g. Schuffner's dots) [Thellier et al., 2002; WHO, 2010a]. A WHO-accredited, *level 1* microscopist expert is able to detect and identify parasites with 90% accuracy in thick smears, even with low parasitaemia [WHO, 2016]. Non-WHO-accredited laboratory professionals, accounting for most malariologists, perform significantly worse than *level 1* experts: in a study in Ethiopia, in 2016, participants (60) score a sensitivity of 41.3 % on parasite detection, agreeing only on 51.1% of the cases [Yitbarek et al., 2016]. The shortage of expert microscopists in clinics and other sanitary institutions causes Malaria diagnosis to be, either extremely time-consuming or often misdiagnosed (time-accuracy/time-sensitivity trade-off).

The elevated human and financial costs of laboratory diagnosis led the least resourceful regions to over-diagnosis and over-treatment: a mere history of fever suffices as an indication of malaria. Over-treatment accounts for the main cause for rapid drug resistance, resulting in the long run to greater costs in the fight against malaria. In African countries it is estimated to result in losses of US$12 billion a year due to increased health-care costs, loss of work-force and due to negative effects on tourism [Amexo et al., 2004].

In sum, the collection, staining and inspection of blood films is a time-consuming procedure. Malaria outbreaks are seasonal and massive. And because of the workload that it brings along, time becomes a critical factor for malaria diagnosis in several different ways. (a) It affects the quality of the staining: if staining steps are not complied with, thick blood smears can drop by as much as 50% in quality. (b) It increases improves parasitaemia estimation, as more time is spent on microscopic examination [Dowling and Shute, 1966; Thellier et al., 2002]. (c) And it has a great impact on malaria eradication: an early detection is the best prevention against malaria.

RDTs commercially available are fast, low-cost and sometimes more accurate than microscopy at predicting the presence of malaria parasites. However, their diagnostic sensitivity and specificity varies greatly across manufacturers, does not recognize species, and are unable to estimate parasitaemia [Wilson, 2012]. Thus, all things considered, microscopy remains the customary diagnostic method.

1.2 Background: convolutional networks

Convolutional Networks (ConvNets) are feed-forward, multi-layered artificial neural networks first introduced by LeCun et al. [1998] but popularized by Krizhevsky et al. [2012] with the introduction of the AlexNet network in 2012. ConvNets are designed to perform feed-forward passes through the network more efficiently and to reduce vastly the number of parameters in the network. The idea encoded in ConvNets is inspired by neural correlates of the visual system of mammals. Each neuron in the visual cortex responds to stimuli present in a restricted region of the visual field known as *receptive fields*. Adjacent neurons exhibit overlapping receptive fields that together encompass the entire visual field (see Figure 2)

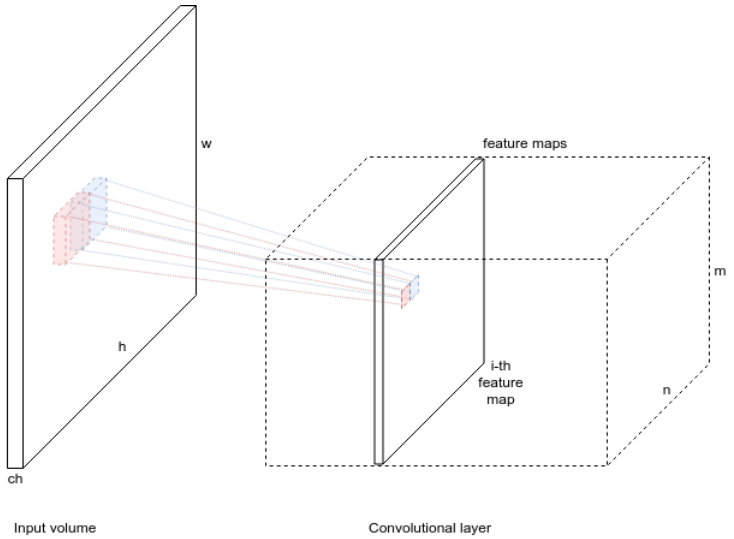


Figure 2: **Receptive fields** of two adjacent neurons in a convolutional layer (red and blue). Trainable weight matrices are shared across activation (or feature) maps extracted from the input. These maps become input volumes for the next layer.

Similarly to the visual system, convolutional layers are composed of processing units (or neurons) that connect only to a local region of the input volume. The connections are local along width and height, but full along the depth of the input volume. Neurons in a convolutional layer share weights and biases among them reducing greatly the parameters of ConvNets. Weights are learnable small 2D (or 3D) tensors refer to as *kernels*[5] that act as feature detectors of input volumes. The size of these kernels determine the size of the receptive field of a neuron.

A convolutional layer is composed of a set of kernels, where each kernel is triggered by a particular feature of the input. Kernels produce feature representations of the input and collect them into *feature maps* or *activation maps*[6]. There are as many feature maps as there are kernels in a layer. Extraction is performed by *convolving* kernels with local regions of the input volume. Given the center of an input region, a convolution operation essentially performs a sum of all local neighbors and itself, weighted by the values of the kernel (for an illustration, see Figure 3). This form of convolution preserves the spatial relationship between pixels in the image and extracts spatial invariant features, like circles or edges.

The retrieval of feature maps is often followed by an element-wise, non-linear activation function, such as ReLu [Livni et al., 2014]. An additional popular choice in ConvNets is to apply a *spatial pooling* layer after a convolutional layer. The purpose is to reduce the dimensionality of feature maps (down-sample) and alleviate the computational

[5]Sometimes refer to as *filters* in the literature
[6]The terms *feature map* and *activation map* will be used interchangeably throughout this thesis.

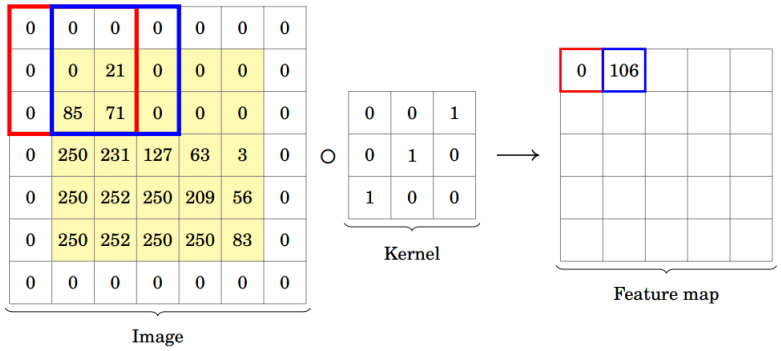


Figure 3: **Convolution operation.** Illustration of the convolution of input regions and a kernel from a convolutional step in the network as described in the text. The aftermath is a spatial invariant feature map.

load. Pooling layers preserve important spatial information of feature maps and help the convolutional layers to remain invariant to translation and rotation. Therefore, state-of-the-art ConvNets alternate convolutional and pooling layers to obtain a network that is invariant to spatial aspects of the input.

ConvNets have proven very effective in the analysis of images and are widely used in computer vision applications [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014]. ConvNets progressively compute more powerful invariants (e.g., scale invariance) as they get deeper. Generally, the more convolutional steps the network has, the more complex high-level features it will be able to recognize. Layers closer to the input learn more elementary features, e.g. edges, since less information falls under the receptive field of their neurons. Deeper layers combine many of these raw features from feature maps into more complex ones, e.g. face contours, obtaining ever more high-dimensional abstract features. Here, we'll make use of the recognition power of deep convolutional networks for the detection of malaria parasites.

1.3 Background: automated malaria detection

In the last two decades, there have been several efforts to automate malaria diagnosis. With the recent rise of intelligent systems in medical imaging [Müller et al., 2004], the application of a diverse set of computer vision and machine learning techniques to the detection, classification and parasitaemia estimation of malaria was imminent. To date, there are several image-based systems developed that derive from the analysis of digitalized malaria-infected blood films [Tek et al., 2009]. These systems need to deal with differences in position, shape and color of parasites while reducing the number of false positives caused by artifacts.

From a computer vision standpoint, this problem has been approached by different researchers following a common pipeline: segmentation of the image to be analyzed, followed by a feature extraction step (extraction of color, shape, size, texture, etc.) and classification into malaria-positive or -negative based on the extracted features.

Even though thin blood smears are discouraged for malaria diagnosis given their low sensitivity in comparison to thick smears, most attempts to automate diagnosis are based on thin smears [Linder et al., 2014; Tek et al., 2010, 2006; Das et al., 2013; Ross et al., 2006; Tek et al., 2009]. It must be due to the fact that parasites on thin smears are disperse, side-by-side in uncomplicated backgrounds. The segmentation step for thin-smear-based detection systems consists of detecting infected RBC, extracting their morphological features, and classifying them. Thin smear-based systems generally achieve high performance scores in a single digitalized field-of-view.

Examples of such systems are presented by Linder et al. [2014], who classified plasmodium falciparum with a Support Vector Machine (SVM) achieving a 95 % sensitivity [7] and 100 % specificity. Or by Das et al. [2013], who employed SVMs and Naive Bayes classifiers on both plasmodium falciparum and plasmodium vivax images detecting with up to 84% accuracy. Tek et al. [2006] and Tek et al. [2010] used the K-Nearest Neighbor algorithm (KNN) on extracted features of plasmodium falciparum images as well. Tek et al. [2006] reported 74% sensitivity, 98% specificity, 88% positive prediction and 95% negative prediction values. Interestingly, Tek et al. [2010] opt for multi-class classification including all stained structures (WBC, platelets, artifacts) and all different parasite stages as different classes. They reported no false detections even at 0.1% parasitaemia levels. Liang et al. [2016] adopted a ConvNet-based approach to single-cell classification of RBC into malaria-infected or malaria-free cells, achieving a sensitivity of 96.99 % and specificity of 97.75%. By visualizing activations maps, Sivaramakrishnan et al. [2017] claimed to outperform (98.61 % accuracy) the latter while reducing its complexity and computational time. Lastly, one of the most complete studies on thin smears was carried out by Ross et al. [2006]. An ensemble of neural network classifiers was applied in the classification of all fours strains of plasmodium. Neither inference times, nor parasitaemia estimations were reported on any of these studies.

Several authors have worked on malaria detection on thick smears, for instance Kaewkamnerd et al. [2012], Quinn et al. [2014], Quinn et al. [2016] and Rosado et al. [2016b]. Quinn et al. [2014] employed decision tree algorithms, such as Random Forest (RF), and reported ROC AUC of 0.97. Kaewkamnerd et al. [2012] model extracted candidates according to the image tonal histogram. Later on, these candidates were classified using template-matching: they were regarded as positives if their size matched the chromatin of the parasite of reference. Yet, only in a single study do they have claimed to have experimented with deep learning techniques, in particular convolutional networks, on the detection of malaria reporting good performance: 99% sensitivity and 100% specificity [Quinn et al., 2016]. Note, that the segmentation problem on thick smears was resolved either by a sliding-window sampler [Quinn et al., 2014, 2016] or by feature-engineering of pixels or group of pixels [Kaewkamnerd et al., 2012; Rosado et al., 2016b]. In every case, only detection of plasmodium falciparum was pursued except in a study conducted by Rosado et al. [2016b], where WBC objects were also annotated for detection. They used an SVM algorithm on 314 image-extracted features from each candidate in a total of 194 images captured with a mobile phone camera. The model achieved 80.5% sensitivity and 93.8% specificity in the detection of parasites, whereas in the detection of WBC it achieved 98.2% sensitivity and 72.1% specificity.

Automated malaria diagnosis on thick smears, albeit promising, shows some practical limitations. Studies discussed above were conducted in small-sized, single-source datasets of images. Image aspects vary greatly across microscope type, and across quality of the blood sample preparation and staining. Thus, it is unknown if such systems will maintain their the predictive power under different circumstances in an end product application. Also, to our knowledge, none of these studies have been explicit about detection times, nor have they offered an estimation of parasitaemia, which according to WHO is a compulsory step in malaria diagnosis and treatment.

A full review of automated malaria detection in microscopic images up and until the year 2016 can be found in Rosado et al. [2016a].

[7] Here, and for the rest of the section, sensitivity is understood as a measure of performance. More precisely, as the rate of true positives events.

1.4 Aim of this thesis

From a computer vision standpoint, malaria detection is posed as an object localization and recognition problem. Namely, an image is understood as a scene where objects of interest or events are contained and where everything else present in the image is background. Hence, object localization is the task of searching for the presence and location of events in a image. While object recognition is concerned with identifying these events.

Here, we investigate whether the detection of trophozoites of plasmodium falciparum (Pf) and white blood cells (WBC) on malaria-infected images from thick smears can be approached by a class of deep learning algorithms designed to solve location-variant object detections in a single shot. Our intention is to explore a faster method for malaria diagnosis, where detection time is a pressing factor. For that, we make use of a public image set built from digitalized malaria-infected thick smears made available by Quinn et al. [2014]. Images in the set contain Pf trophozoites and WBC objects, where only Pf objects are annotated.

State-of-the-art Single-Shot Multi-Box Detector (SSD) is chosen for the task at hand [Liu et al., 2015]. SSD is a ConvNet-based algorithm that proposes a set of boxes bounding objects of interest and identifies them in a single shot. More notoriously, it discretizes the input space efficiently in a way that allows for the detection of objects at different scales. As a result, real-time detection is achieved (59 FPS with mAP 74.3% on VOC2007 dataset[8]) by exploiting the highly parallel and distributed computation, and the recognition power of convolutional networks. What makes SSD Multi-box a well-suited approach for this problem is two-fold:

(i) object inference is real-time. The most accurate method to date is a ConvNet-based sliding-window detector. These type of detectors, albeit powerful, show very large inference times.

(ii) it allows for the detection of multi-class objects of various sizes. The detection of WBC is necessary for parasitaemia estimation. We extend the set of annotations to WBC objects and aim to detect them as well.

SSD found applications in different detection problems: vehicles[Kim et al., 2016], aerial vehicles [Xia and Li, 2018], vehicles and pedestrians [Qiong and LIAO, 2017], ships [Nie et al., 2017], to name a few; yet the detection of small objects still remains a challenge. The detection of small-scale objects was reported as SDD main performance flaw. Nonetheless, there's been several attempts to improve SSD performance in general [Jeong et al., 2017; Xie et al., 2017a,b; Zheng et al., 2018], and in the detection of small objects in particular [Fu et al., 2017; Liu et al., 2015; Cao et al., 2018]. Even further, SSD has been applied successfully to datasets of medical nature, though these applications are not abundant (breast tumor detection [Cao et al., 2017], cell detection [Yi et al., 2017]).

At sight of this empirical evidence, the SSD approach is conjecture to detect and classify Pf and WBC accurately, while scoring real-time results. Ultimately, the goal is to optimize for mobile devices and thus, offer a mobile solution to Malaria diagnosis.

[8]FPS is short for *frames per second* and mAP is the *mean average precision*, both popular unit measures for object detection performance

2 Methods and Materials

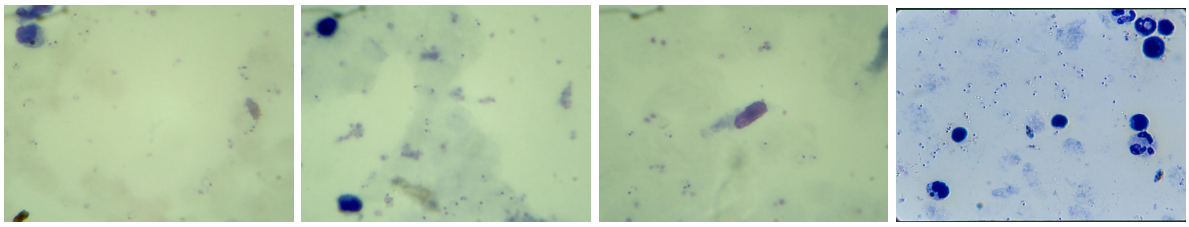
2.1 Dataset

The raw dataset consist of images of 768×1024 pixels from lossy-compressed (JPEG) digitalized, malaria-positive thick smears made available by Quinn et al. [2014]. It contains 2703 images taken from 133 Fields-stained thick blood smears. Images were captured using a Motic MC1000 camera mounted on a Brunel SP150 microscope with an oil immersion objective lens at $1000\times$ magnification. Annotations were created by a team of four experienced laboratory technicians and consisted on 50p-sided square boxes bounding 50,255 Pf parasites. Boxes bounding proximal parasites overlap. White blood cells and other artifacts can be identified in the images, but were not annotated.

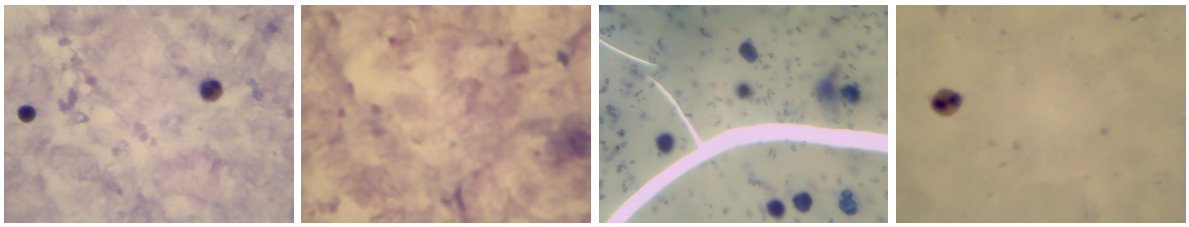
With the intent of making the data consistent and valuable for training, a few preprocessing steps were applied to the set:

2.1.1 Data cleaning

Images were hand-picked based on the discriminability of the objects of interest in them (staining quality, blurry images, etc.). Exactly 768 images were discarded. Examples of "good" and "bad" images are collected in Figure 5.



(a) Good quality images.



(b) Bad quality images.

Figure 5: **Base image set samples.** A sample of selected (a) and discarded (b) images from Quinn et al. [2014] dataset.

2.1.2 Data integration

An additional 16 images of lossy-compressed (JPEG) digitized, Field-stained, malaria-positive thick smears with a high count of Pf trophozoites were added to the set (Figure 6). Pictures were captured using a HP ScanJet 8200 camera mounted on a microscope with an oil immersion objective lens at $1000\times$ magnification, which resulted in image files of $\sim 2300 \times 1900$ pixels. These images were facilitated by AMREF [9] and no annotations were provided. We created 567 new annotations (bounding boxes) with the assistance of an experienced biologist. Boxes bounding

[9] https://www.amref.nl/

of proximal parasites overlap.

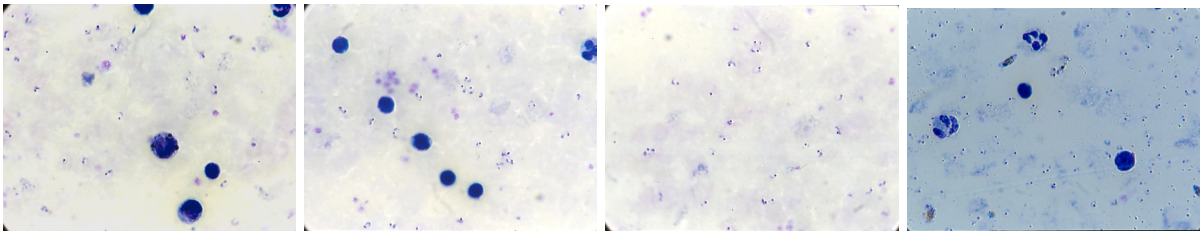


Figure 6: A sample from the AMREF image set.

2.1.3 WBC annotations

In addition, annotations bounding white blood cells (WBC) were created for both sets, again with the assistance of an experienced biologist. Class frequencies and bounding box sizes for each dataset are gathered in Table 1.

(a) Class frequency.

Class	AMREF	Quinn
Pf	524	43,222
WBC	66	3,631

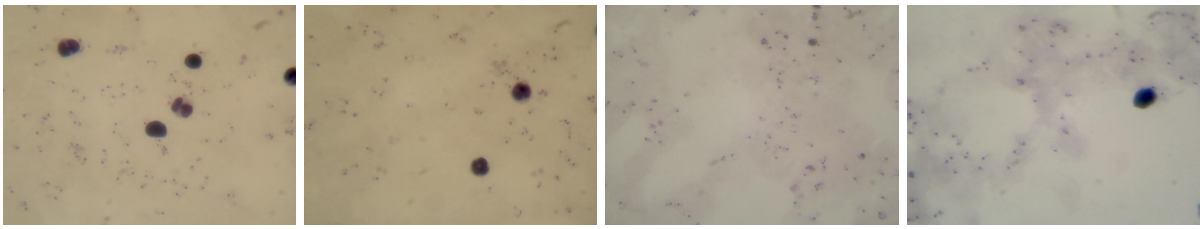
(b) Bounding box size (in pixels).

Class	AMREF	Quinn
Pf	130.00 ± 37.09	50.00 ± 0.00
WBC	280.00 ± 22.28	100.00 ± 0.00

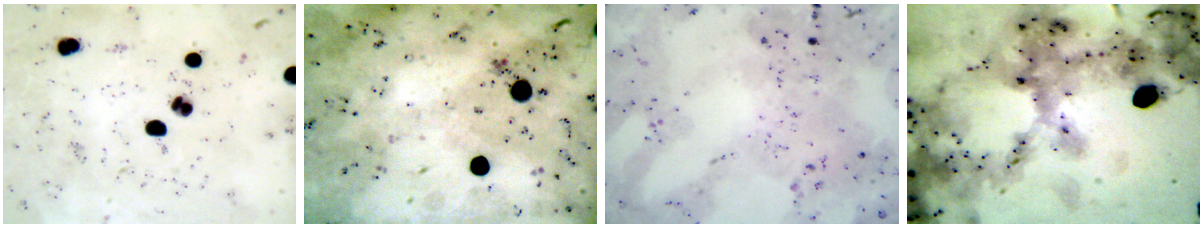
Table 1: Frequency and size of class object annotations. Note: values in (b) are given as median $\pm$ std. dev.

2.1.4 Data transformation

In an attempt to increase signal-to-background ratio, we experimented with image enhancing techniques. **Contrast stretching** yielded the best results. It "stretches" the range of pixel intensity values of an image by linearly mapping the minimum pixel value to 0, and the maximum to 255. For images with a wider range of intensities, we cut off the top and bottom 2% of pixel intensity values so the stretching effect is more noticeable. Examples of preprocessed pictures are collect in Figure 8.



(a) Before contrast stretching.



(b) After contrast stretching.

Figure 8: **Contrast stretching.** Comparison of a sample of images from the integrated set before (a) and after (b) contrast stretching.

2.1.5 Data partition

The new image set was partitioned, at random, into training and testing sets. A customary 10% (196) of a total of 1951 images were designated to the test set. Details are collected in Table 2.

Class	Training	Testing
Pf	39,571	4,175
WBC	3,286	411

Table 2: Number of class labels in training and testing sets.

Training and test sets are fixed in all experiments. Results were reported on the test set.

2.2 Approach: Single-Shot multi-box Detector (SSD)

SSD is an object detection algorithm successfully applied to the classification of objects in natural images [Kim et al., 2016; Nie et al., 2017; Xia and Li, 2018; Qiong and LIAO, 2017; Yi et al., 2017; Cao et al., 2017]. As defined by Liu et al. [2015], "*[...] is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes [...]*". Prior to SSD, two-stage training approaches such as region proposal methods [Gu et al., 2009; Uijlings et al., 2013] followed by a separate per-region classification stage [Girshick et al., 2014; Girshick, 2015; Erhan et al., 2014] have prevailed in most recent, state-of-the-art object detectors. They are not only overly-complex and hard to optimize (multi-step model training), but too slow for real-time applications.

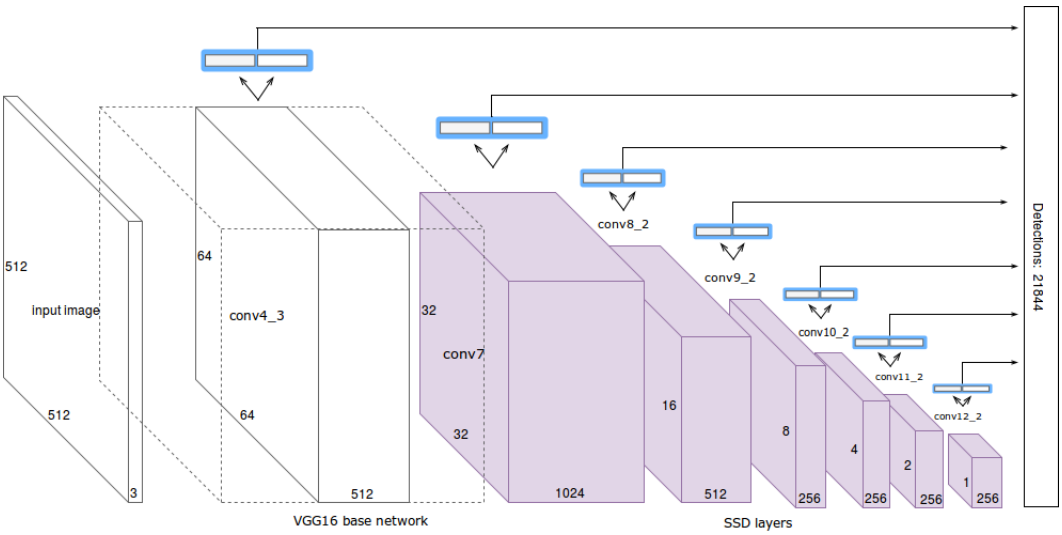


Figure 9: **SSD Multi-box.** The SSD approach consist of appending multi-scale feature maps (purple), or **SSD layers**, to the end point of a base network (for instance, VGG-16), and applying **predictor modules** (blue icon) to detection layers in order to detect and classify objects at different scales.

SSD provides an alternative "unified" framework for fast object detection. Contributions by Liu et al. [2015] to modern object detectors are two-fold: (1) the effective use of low- and high-end feature abstractions of the input in order to detect objects at different scales, and (2) the multi-class classification and bounding box proposal of objects in a single feed-forward pass of the network (hence the name, "single-shot"). Albeit heavily inspired by previous work on unified methods [Erhan et al., 2014; Redmon et al., 2015; Szegedy et al., 2014; Sermanet et al., 2013], SDD achieves real-time performance with significantly better classification accuracy.

In a nutshell, the SSD approach consist of a *meta-network* of feature maps of decreasing resolution appended to the end point of a base network. Well-known ConvNet architectures, such as VGG-16 or ResNet, truncated before the dense classification layers, are used as base networks. Feature maps in the meta-network that share the same resolution are referred to as *SSD layers*, and are used to detect objects at various scales and classify them into object categories (Figure 9). Feature maps from VGG-16 are also used for detection.

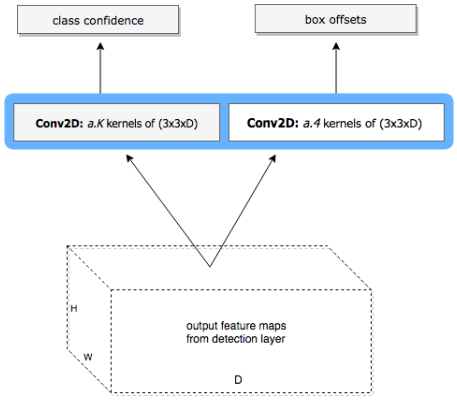


Figure 10: **Predictor module.** Set of two convolutional layers that learn class confidence scores (on the left) and offsets in default boxes (on the right). Given a anchor boxes and K object class categories, there are $a \cdot K$ class predictors (kernels) and $4 \cdot a$ box offset predictors (cf. Appendix A).

Predictor modules perform the detection of objects. Modules are composed of neurons in convolutional layers

that, in essence, evaluate activation maps into a fixed set of pre-defined bounding boxes known as *anchor boxes*.[10] They do so by defining a anchor box types of different sizes and aspect ratios centered on the receptive field of each predictive neuron, and using feature abstraction from designated activation maps to detect objects in the input image. Note that center locations of anchor boxes create a grid of cells, where cells adjacent to any given point in the grid span the receptive field of a predictive neuron. As we go deeper into the network, the grid has less crossing points and cells broaden (and less anchor boxes are defined onto them). Stacked grid of cells drawn from different activation maps effectively cover the input space and improve the search for objects of interest. Designated activation maps are referred to as *detection layers* of the network.

More precisely, predictor modules consist of two separate convolutional layers: *box predictors* that regress box coordinate offsets [11] to target boxes, and *class predictors* that compute class confidence scores for each box. Thus, there are $a \cdot 4$ kernels in a box predictor module, one for every anchor box and for every box coordinate (x, y, *width*, *height*). Likewise, there are $a \cdot K$ kernels in a class predictor module, one for every anchor box and for every object class category (Figure 10). The result is a collection of superposed anchor boxes on the input image at various locations, scales and aspect ratios, i.e. the anchor boxes of the network.

Predictor modules extend previous work in Szegedy et al. [2014] and Erhan et al. [2014] known as the **Multi-box** approach, where object detection is formulated as a regression problem to the coordinates of ground truth bounding boxes in a class-agnostic manner. For a detailed explanation on how module predictors learn object class probabilities (classification) and offsets (regression) in default box coordinates refer to Appendix A.

In sum, predictor modules are applied to detection maps at different depths of the network, which allows the SSD to detect at multiple scales of the input. An efficient discretization of the space for region detection is obtained as a result (a complete comparison of input discretization in object detection can be found in Cai et al. [2016]). Better classification performances were reported too, since objects are classified based on a large range of feature abstractions [Liu et al., 2015]. As to date, SSD Multi-box exhibits one of the best speed-accuracy trade-offs among modern object detection systems [Huang et al., 2016].

2.3 Training & Network

Training of an SSD Multi-box network is non-trivial. Fine-tunning is complicated and tedious in highly complex models with many degrees of freedom (number of hyper-parameters) as is the case with SSD. In addition, SSD training is multi-objective (*c.f.* Appendix B). The network is hard-wired to share parameters between task objectives that are usually in conflict, and thus a trade-off term needs to be fine-tuned. Because of time and material limitations for a complete experimentation program, we use default hyper-parameter values reported to offer good performance in Liu et al. [2015]:

(a) **Ground truth matching index:** `0.5`. The supervised nature of the problem demands target samples to be learned from. The ground truth assignment strategy followed to build target samples during training as detailed in Appendix B.

(b) **Hard-negative mining:** `1:2`. Ratio of positive-to-negative class box proposals that contribute to the loss . See Equation 5 in Appendix B.

[10] Also know as *default boxes* in the literature.
[11] as in "translation" or "shift"

(c) **Trade-off term (λ):** `1.0`. Controls in which proportion each objective contributes to the loss. See Appendix B for more details.

(d) L_2 **normalization scale:** `20`. Normalization layer applied to detection layer. Use to scale feature maps of different scales and norm. The scale is learned during back propagation. Only the first layer is normalized.

(e) **Scaling factors:** `(0.1, 0.1, 0.2, 0.2)`. Scaling factors of $(x, y, width, height)$ coordinates. See Algorithm 2 in Appendix B for more details.

(f) **Center-box offset:** `0.5`. Relative offset of center-box grids. The offset is computed w.r.t the input image.

Nonetheless, training an SSD on a new dataset requires that, at least, task-dependent hyper-parameters are tunned:

(a) **Input size:** We use crops of training images collected in a sliding-window fashion. Image dimensions in our dataset are variable and too large to be used as input (large computational overhead). Note that we do not down-sample, but instead use full-resolution images. Huang et al. [2016] discovered that down-sampling images by a factor of two consistently lowers accuracy by 15.88% on average. Full-resolution images allow for small objects, like malaria parasites, to be resolved.
The reference size of the detector window is set to `(512, 512, 3)`, as suggested in the original paper. We investigate performance with window sizes `(256, 256, 3)` and `(768, 768, 3)`, that is $\times 0.5$ and $\times 1.5$ the reference size, respectively. Intuitively, a (not too) small crop size has a low background-to-signal ratio and can aid the network discern objects of interest better. We investigate the effect of the window size on the SSD detection performance, and its impact in the total inference time.

(b) **Detection layers:** Refers to the selection of feature maps of the network use for localization and classification. Our task here is to detect small objects from real-world image data. This is a challenging task in the field of object detection. In fact, Liu et al. [2015] reported the detection of small objects as the principal short-coming of the SSD Multi-box. We know that ConvNets learn hierarchical levels of abstraction of the input data, ranging from local low-level representations in early layers to high-level, semantic descriptors in intermediate and late layers. And is semantic information that helps ConvNets detect objects and tell them apart. Small objects are low-res and lack sufficient pixel information for ConvNets to extract semantic descriptors from it. Then, it follows that the detection of objects at small scales rely, for the most part, on early layers of the network, and that discriminating them into classes will be an arduous task for the network. To verify the extend of this, we experiment with different choices of detection layers than that propose in Liu et al. [2015]. We guide the choice of detection layers by visualizing activation maps of the network. Quiver package, an interactive ConvNet visualization tool for Keras, is used for this purpose [Bian, 2016].

(c) **Size and aspect ratios of anchor boxes:** Guided by ground truth aspect ratios (*c.f.* Table 1), anchor boxes are strictly squares $(1:1)$ and their dimensions range, approximately, from the size of the smallest (50p-sided boxes) to the median size (230p-sided boxes) of the largest object of interest. Anchor sizes are chosen in congruence with their theoretical receptive field size. The intuition behind it is that predictive neurons in predictor modules of the network would not be able to detect any objects larger than or not contained in their receptive field. It would simply not have the necessary information to guide neither the localization of that object (shift anchor boxes towards it) nor discern which class instance does it pertain to. Thus, keeping this in mind, we set box sizes for each detection layer manually.

Data augmentation is used to prevent over-fitting. Images are rotated 90°, 180° and 270° at random, and are flipped horizontally and vertically also at random. That is, these transformations can occur simultaneously for a single image in order to increase variability. Note that class imbalance is addressed during loss computation, governed by the ratio of positive-to-negative class box proposals (hard-negative mining). "Hard to learn" negative class box proposals are preferred over "easy" cases as a bootstrapping strategy. Details on how the network learns can be found in Appendix B.

Last but not least, VGG-16 serves as the base network or feature extractor, and SSD layers in the meta-network are defined as detailed in Liu et al. [2015]. Whether layers in the base network or in the meta-network are used in the final model will depend on the choice of detection layers. Model specifications are gathered in Tables 3a and 3b.

Layer	N. Filters	Kernel Size	Stride
Input	-	-	-
Conv2D	64	(3, 3)	(1, 1)
BatchNorm	-	-	-
Conv2D	64	(3, 3)	(1, 1)
BatchNorm	-	-	-
MaxPool2D	-	(2, 2)	(2, 2)
Conv2D	128	(3, 3)	(1, 1)
BatchNorm	-	-	-
Conv2D	128	(3, 3)	(1, 1)
BatchNorm	-	-	-
MaxPool2D	-	(2, 2)	(2, 2)
Conv2D	256	(3, 3)	(1, 1)
BatchNorm	-	-	-
Conv2D	256	(3, 3)	(1, 1)
BatchNorm	-	-	-
MaxPool2D	-	(2, 2)	(2, 2)
Conv2D	512	(3, 3)	(1, 1)
BatchNorm	-	-	-
Conv2D	512	(3, 3)	(1, 1)
BatchNorm	-	-	-
MaxPool2D	-	(2, 2)	(2, 2)
Conv2D	512	(3, 3)	(1, 1)
BatchNorm	-	-	-
Conv2D	512	(3, 3)	(1, 1)
BatchNorm	-	-	-
MaxPool2D	-	(3, 3)	(1, 1)
Flatten	-	-	-
Dense	4096	-	-
Dense	1024	-	-
Dense	3	-	-

(a) **VGG-16 network**. Consist of five convolutional blocks of decreasing resolution (blocks are color-coded). We truncate before the classification layers (light gray) and use it as a base network for the SSD Multi-box.

Layer	N. Filters	Kernel Size	Stride
Conv2D	512	(3, 3)	(1, 1)
Conv2D	1024	(1, 1)	(1, 1)
Conv2D	1024	(1, 1)	(1, 1)
Conv2D	256	(1, 1)	(2, 2)
BatchNorm	-	-	-
Conv2D	512	(3, 3)	(2, 2)
BatchNorm	-	-	-
Conv2D	128	(1, 1)	(2, 2)
BatchNorm	-	-	-
Conv2D	256	(3, 3)	(2, 2)
BatchNorm	-	-	-
Conv2D	128	(1, 1)	(2, 2)
BatchNorm	-	-	-
Conv2D	256	(3, 3)	(2, 2)
BatchNorm	-	-	-
Conv2D	128	(1, 1)	(2, 2)
BatchNorm	-	-	-
Conv2D	256	(3, 3)	(2, 2)
BatchNorm	-	-	-
Conv2D	128	(1, 1)	(2, 2)
BatchNorm	-	-	-
Conv2D	256	(4, 4)	(2, 2)

(b) **Meta-network.** SSD Multi-box adds 7 extra convolutional blocks to the base network (in the table, separated with thick horizontal lines). The first block (purple) corresponds to the 2^{nd} convolutional layer in the 4^{th} block of the VGG-16.Yellow-colored layers are used as detection layers for module predictors in Liu et al. [2015]

2.4 Inference

The network is designed, partly, to deliver box coordinate offsets for anchor boxes of the network. The number of anchor boxes is often in the order of thousands per input image. Given the large size of the output, an additional post-processing stage is necessary. Post-processing consist of decoding boxes into pixel coordinates, and removing the ones with low predictability.

First, a decoding function, $\varphi^{-1}(.)$, takes in box coordinate offsets to propose bounding boxes relative to the input image, such that $\varphi^{-1}(\text{anchor box}, \text{offsets}) = \text{box proposal}$. Note that anchor boxes and box proposals are expressed in pixels. Naturally, $\varphi^{-1}(.)$ reverses the encoding function, $\varphi\ (.)$, use to build target samples during training. For a definition of $\varphi\ (.)$ refer to Algorithm 2 in Appendix B.

Lastly, box proposals are discarded based on their class confidence scores. Box proposals scoring equal to or greater than a certain detection threshold are kept. However, further action is necessary to reduce the number of redundant detections. The network is trained to propose multiple bounding boxes per ground truth box, thus making *non-maxima suppression* (NMS) indispensable during inference. In the context of object detection, NMS is a technique use to single-out detections triggered by the same target object [Rothe et al., 2014]. The most common approach is a greedy, local maxima strategy. That is, over a set of adjacent detections, the one with the highest confidence score is preferred. Detections are considered adjacent to each other based on their Jaccard index (defined in section 2.5).

Needless to say, negative-class box predictions are discarded from the start. This measure reduces the number of boxes to be processed to approximately 2% of the total amount.

2.5 Performance measures

Given the locations of known targets, a set of target estimations and a proximity criterion, we can determine which fraction of the targets were detected and adequately localized (true positives), which were not (false negatives) and which estimations do not meet the proximity criterion of any target (false positives). In the context of object detection, the **Jaccard index** (or **Intersection Over the Union (IoU)**) is a popular coefficient to measure proximity and is defined as the intersection over the union of two bounding boxes. In this study setting, we make use of the Jaccard index to determine valid detections.

Malaria detection is a multi-target diagnostic task, where the maximum number of non-targets is virtually unbounded (large number of negative-class boxes). **Precision** and **recall** are useful measures of object detection success for tasks with pronounced class imbalance. In a multi-class case, precision and recall are computed per class. Precision is defined as the fraction of valid detections, whereas recall is the fraction of detected targets. Namely, a high precision score is associated with low false positive rate and a high recall score is associated with a low false negative rate (misses). We use the precision-recall curve to evaluate the trade-off between precision and recall at increasing confidence thresholds (or decision boundaries).

mean Average Precision (mAP) is used as measure of global performance. It is defined as the grand mean of the average precision (AP) score per class. To compute AP, we average across precision scores at different thresholds and then compute the mean across classes to obtain mAP. Note that AP can be understood as the Area Under the Curve (AUC) of the precision-recall curve. Therefore, mAP is a more unbiased measure of the object

detector's performance.

A suitable approach to expose the relation between detections and the number of errors made is the **free-response receiver operating characteristic (FROC)** curve. The FROC curve is a monotonically-increasing curve that, at any given point, tells what the average number of false negatives per sample is for a recall score. We use the cumulative sum over recall scores to build the curve. Note that the frequency (of false positives) is preferred over fraction due to the virtually unbounded number of false positives per sample, strongly dependent of the discretization of the input space [Zou et al., 2011].

Last but not least, we compute the **normalized confusion matrix** to visually aid classification assessment.

2.6 Hardware

A high-performance computer cluster running RHEL 7 (OS) with dedicated graphic processing unit (GPU) nodes was at our disposal. Each node offers 2×Intel Xeon E5-2630v4 Broadwell (2.4GHZ, 28 cores) processors, 128GB RAM, and 2×Nvidia Tesla K80 GPUs. The parallel processing power of the GPUs speeds up the training process by several orders of magnitude.

2.7 Study Design

As pointed out in Section 2.3, we fine-tune and modify the original SSD network to better fit our case. The series of experiments are clustered in two consecutive blocks:

(i) **Fine-tunning and modification of the network** The set of experiments performed here are, for the most part, concerned with variations in the choice of detection layers and the size of anchor boxes. As mentioned earlier, we guide the choice of detection layers by visualizing activation maps of the network. Best performing models are reported and compared to each other.

(ii) **Transfer learning of VGG-16 feature extractors** Transfer learning is applied in a second series of experiments. Transfer learning refers to the action of re-purposing, or *transferring*, weights (kernels) from a base network trained on a base dataset to a target network in order to improve generalization and/or reduce training time on a target dataset [Yosinski et al., 2014]. Pre-trained weights of a VGG-16 network trained on benchmark dataset ImageNet (2012)[12] are transferred to the base network of the best-performing model from the first series of experiments. During training, we load and "freeze" some pre-trained weights and train the rest from scratch. We experiment with freezing up to the 1st, the 2nd, the 3rd, the 4th, and the 5th VGG-16 blocks, and report their performance.

Lastly, we analyze their time performance, in particular in their trade-off with detection and classification performances. Training experiments last around 2.5 days each, and given the extensive hyper-parameter search space, we could not afford validation techniques such are cross-validation.

[12]Ready-to-use, pre-trained weights are made available in Keras library.

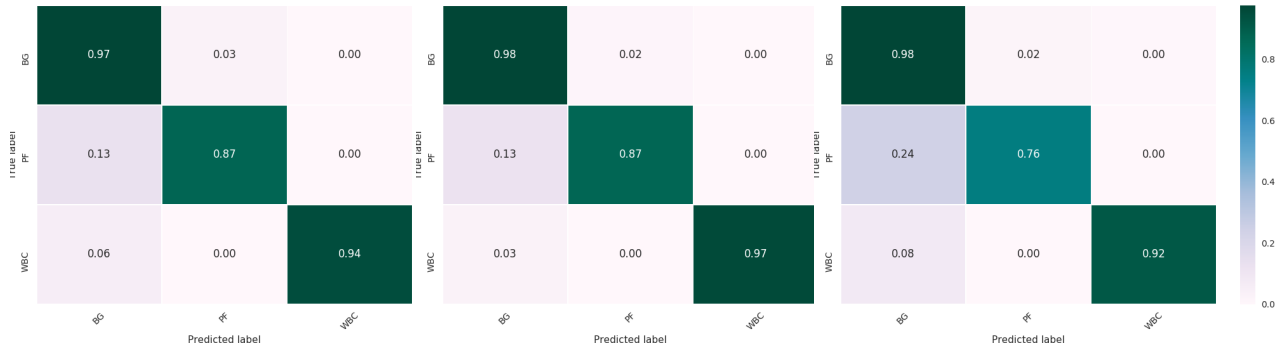
3 Results

In this section, results of best-preforming models are reported and analyzed. Every model is trained for 120 full-passes of the data in mini-batches of 8 input images. Batch normalization is used to accelerate convergence. During inference, NMS is applied to adjacent detections with $IoU(\text{box}_1, \text{box}_2) \geq 0.4$ to reduce the number of double detections. A detection is considered valid if it exhibits $IoU(\text{true}, \text{pred}) \geq 0.3$ w.r.t the closest ground truth annotation. Every model is evaluated against the test set described in Table 2, and reported in terms of precision, recall and false positive count.

3.1 SSD fine-tunning: window size, detection layers and anchor box sizes.

After much experimentation, results of best-performing models are collected in Table 4. We report one model per window size variant: SSD-256, SSD-512 and SSD-768. Model specifications, such as choice of detection layers and size of anchor boxes are contained in Table 7 in the Appendix. Note that the original implementation uses principally layers in the meta-network (*c.f.* Table 3b) as detection layers, and anchor boxes increase evenly in size along the network ranging from the smallest to the largest object in the dataset. Surprisingly, this only seemed to offer good results with reference input size ($512 \times 512 \times 3$). Detection layer choices for ($256 \times 256 \times 3$) and ($768 \times 768 \times 3$) input sizes used almost exclusively layers from the base network for detection.

3.1.1 Detection performance



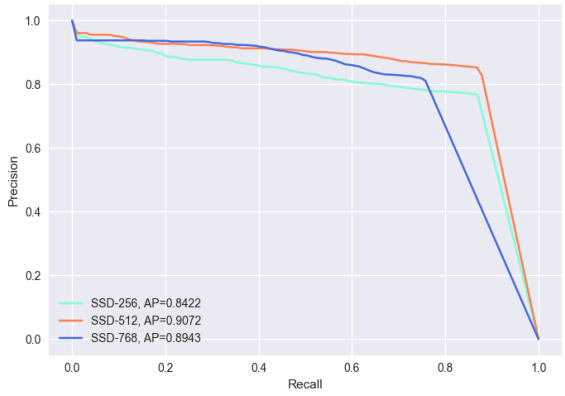
(a) SSD-256 (b) SSD-512 (c) SSD-768

Figure 11: **Confusion matrices of SSD-256, SSD-512 and SSD-768** Class confidence threshold is set to 0.5. Class frequencies are normalized over true labels.

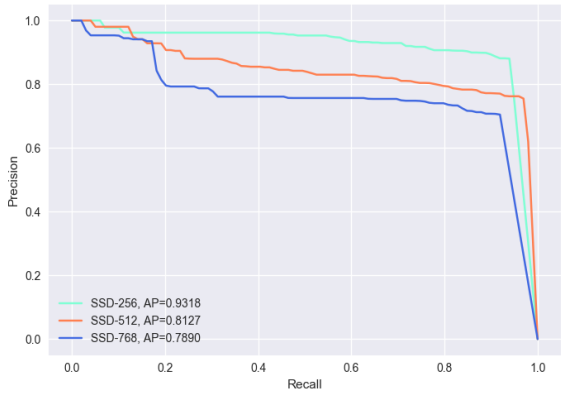
	Average Precision			**Recall@AP**		
	Weighted Mean	PF	WBC	Weighted Mean	PF	WBC
SSD-256	85.10 %	84.21 %	**93.17 %**	87.58 %	**86.89 %**	94.10 %
SSD-512	**89.92 %**	**90.72 %**	81.27 %	**88.46 %**	**87.52 %**	**97.54 %**
SSD-768	88.52 %	**89.42 %**	78.90 %	77.07 %	75.57 %	91.89 %

Table 4: Performance comparison of results under different sizes of the input.

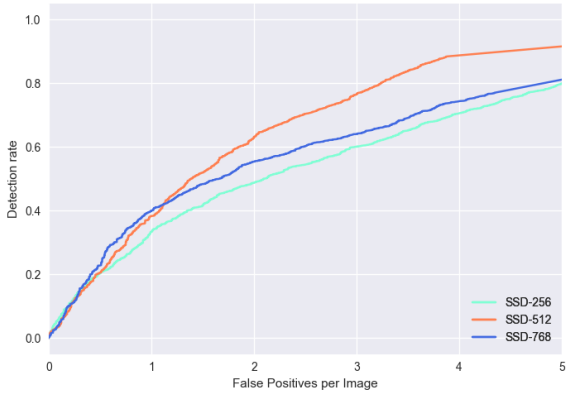
As disclosed in Table 4 and supported by Figure 11, SSD-512 showed the best classification performance (mAP: 89.92%, mean Recall@AP: 88.46%). Overall performance is measured by the weighted sum across each class, weighted by its respective class frequency. There are ×10 more Pf labels (4175) than WBC labels (411), and so correct detection of parasites are rewarded with a higher weight. Despite SSD-512 resulting in the most precise and sensitive model in the detection of objects of interest, SSD-256 appears to be +12.77% more precise when detecting WBC (see Figure 12b). Interestingly, on the same task, SSD-512 shows a higher recall (+3.53%) suggesting more WBC detection at the expense of a higher number of false positives. Figure 12d supports this finding, where the trade-off between recall and false positive frequency is depicted per class for all three models (FROC curves). Figure 12d reveals that for a low false positive frequency (between 0 and 1), SSD-256 scores higher recall levels for the WBC class (*turquoise dashed curve*). However, visually, the area under the FROC curve, an indicator of overall performance, is greater for Pf and WBC in the case of SSD-512. In the task of detecting Pf, SSD-512 and SSD-768 are the most precise models (SSD-512: 90.72%, SSD-768: 89.42%), although SSD-512 is +15.81% more sensitive to Pf. This suggests SSD-768 to have a more conservative performance, i.e. missing more parasites rather than misclassifying them (see Figure 11c).



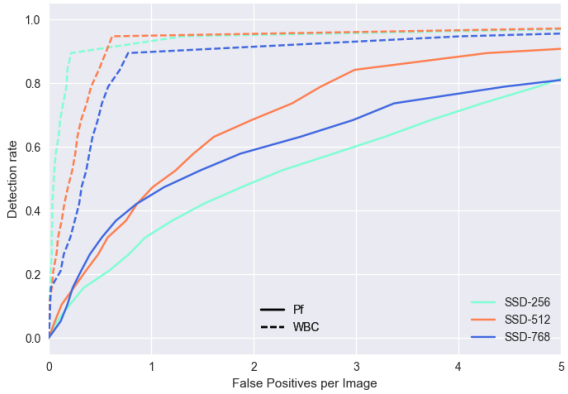
(a) Precision-Recall curves, Pf class



(b) Precision-Recall curves, WBC class



(c) Class-agnostic Recall vs. FP p/ image



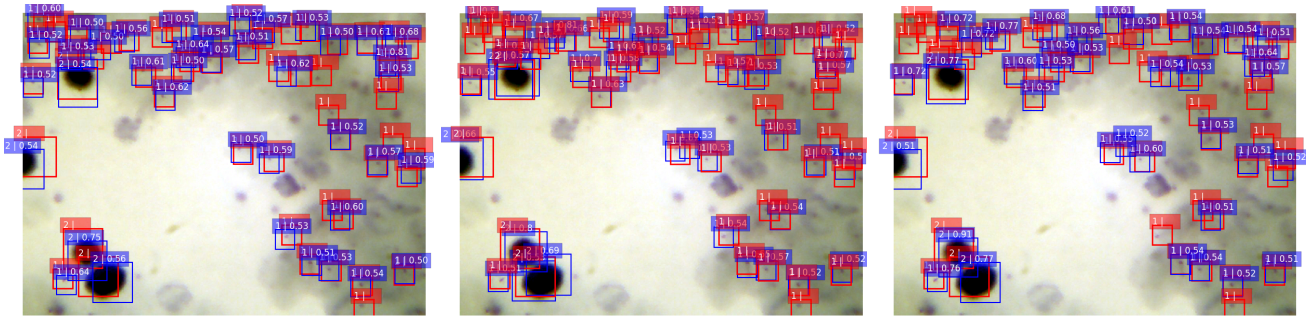
(d) Per-class Recall vs. FP p/ image

Figure 12: **Detection performance** of SSD-256, SSD-512 and SSD-768 models.

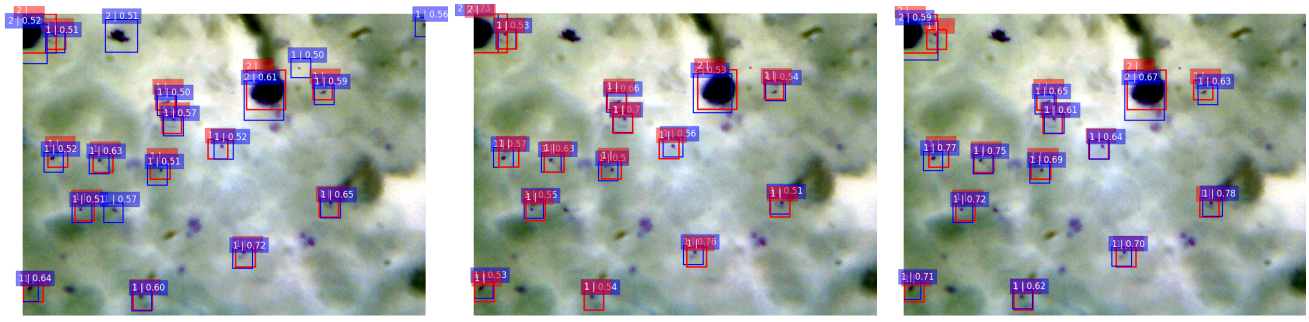
In sum, SSD-512 shows the best precision and recall overall. Figure 12c confirms these results: SSD-512 offers, by a significant margin, the best trade-off between detection of positive class objects (Pf + WBC) and false positive count p/ image. However, SSD-256 performs better at the correct detection of WBC (see WBC counts for

detections in Figure 13).

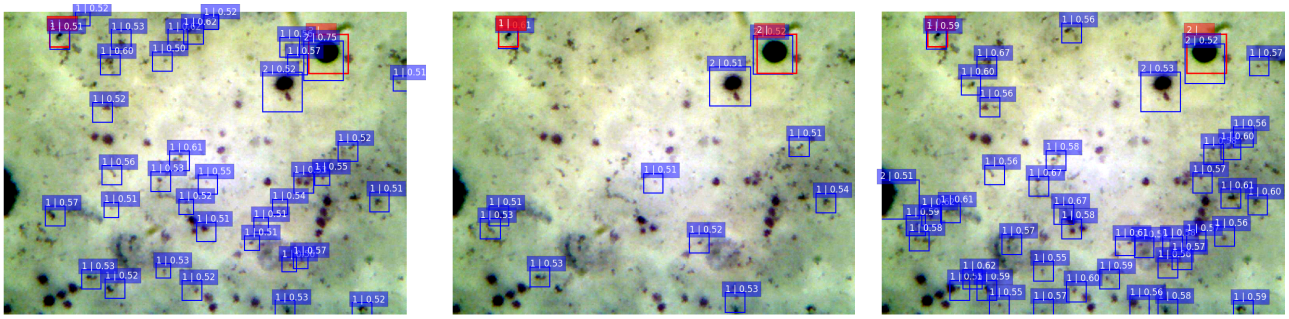
3.1.2 Predictions



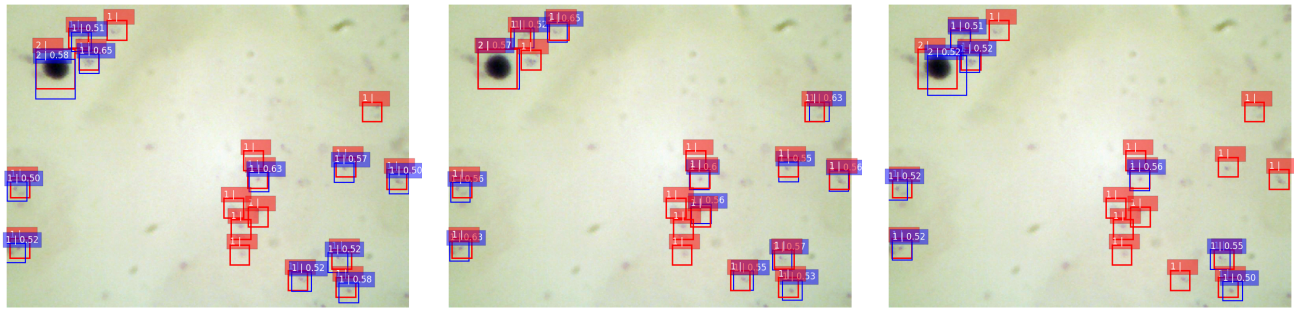
(a) GT: 44 (Pf), 4 (WBC). **Left:** TP: 38, FP: 0, FN: 10. **Middle** TP: 35, FP: 0, FN: 13. **Right:** TP: 31, FP: 0, FN: 17.



(b) GT: 13 (Pf), 2 (WBC). **Left:** TP: 15, FP: 4, FN: 0. **Middle** TP: 15, FP: 0, FN: 0. **Right:** TP: 14, FP: 0, FN: 1.



(c) GT: 1 (Pf), 1 (WBC). **Left:** TP: 2, FP: 35, FN: 0. **Middle** TP: 2, FP: 9, FN: 0. **Right:** TP: 2, FP: 41, FN: 0.



(d) GT: 17 (Pf), 1 (WBC).**Left:** TP: 12, FP: 0, FN: 6. **Middle** TP: 14, FP: 0, FN: 4. **Right:** TP: 9, FP: 0, FN: 9.

Figure 13: **Detections.** Ground truth (**red**) and predictions (**blue**) of SSD-256 (**left**), SSD-512 (**middle**) and SSD-768 (**right**) models on test images. In the sub-captions, ground truth (GT) and detection error (FP and FN) counts. Class 1: Pf. Class 2: WBC.

3.1.3 Time performance

SSD Multi-box showcases one of the best inference time performances among object detectors. We compare input size variants of the SSD mentioned above in terms of execution times. The data pipeline during inference is end-to-end, and consists of an input data processing stage (pre-processing) before a forward-pass through the network (inference), followed by a post-processing stage. The pipeline is implemented in Tensorflow (C++) using Tensorflow's Python API. Times are reported for one image input sample, feeding the network in batches of 4 image crops during the inference stage.

Figure 14 shows CPU and GPU execution times (in milliseconds) of each model. As expected, execution times increased jointly with the size of the input, and models benefited largely from GPU parallelization, in particular during the forward-pass of the input through the network (SSD-256: 46 FPS, SSD-512: 9 FPS, SSD-768: 6 FPS). ConvNets are highly-parallelizable and show huge gains in performance when executed on GPUs. Likewise, the number of detection layers have weigh in on performance as much as the size of the input have. With reference to SSD-256 inference time, SSD-512 makes use of 3 detection layers more, yet is 5× slower. SSD-768, on the other hand, uses 5 detection layers less than SSD-512, however is 0.3× times slower. The additional computational cost of detection layers is noticeable.

The pre-processing stage includes contrast stretching, cropping and scaling. Performance gains in GPU against CPU execution times are noticeable (around 40% gain) on this stage. While at first glance these operations may seem inherently serial, they benefit off efficient algorithms implemented on GPUs (c.f. `CropAndResize` and `HistogramFixedWidth` Tensorflow operations).

The post-processing stage includes combining results from input crops, sorting and ranking boxes, and suppressing redundant detections (NMS). Computations in this stage are highly loop-intensive and Tensorflow operations like NMS (`NonMaxSuppressionV2`) and top-k sorting (`TopKV2`) do not offer GPU kernels, which causes costly data transfers to and fro that slow down the pipeline. Some performance gain can be seen on GPUs as the number of crops grow, i.e. SSD-256 vs. SSD-768 comparison on CPU and GPU.

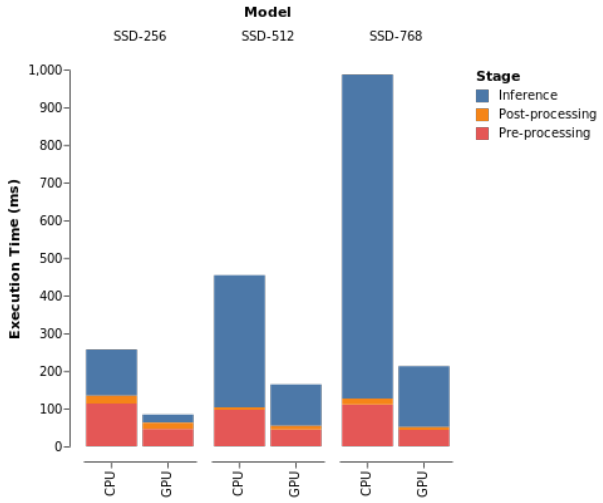
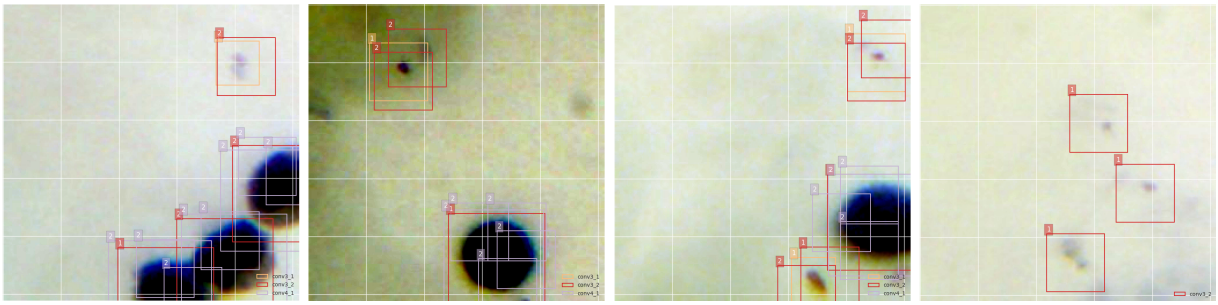


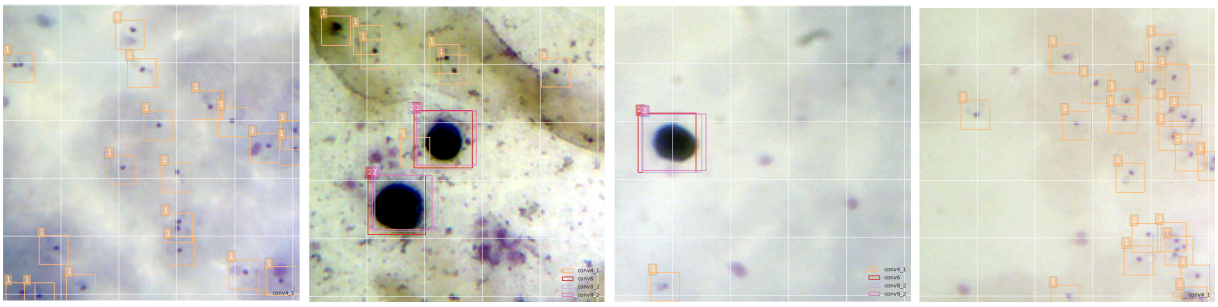
Figure 14: **Inference performance on CPU and GPU devices.** The figure shows execution times of every stage during inference for one single image sample. Time performance during the inference stage are showcased per frame (or image crop) instead of per image (as is customary for objection detection algorithms). Post-processing execution times are showcased per batch (of image crops). During the pre-processing, images are cropped into 12 (SSD-256), 4 (SSD-512) and 2 (SSD-768) sub-images and processed in batched of 4 samples in the following steps.

3.1.4 Further Analysis

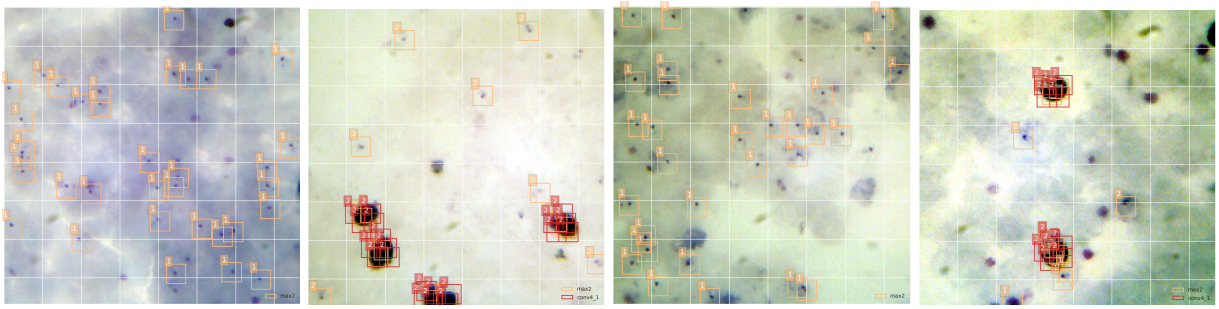
Further analysis is conducted in order to explore confusion tables, missed targets and false positives. For that purpose, we examine the source of detections for every model reported above. We do so by assembling high-scoring detections, such that ≥ 0.5, from each detection layer independently, and depicting them on the input image. Instances of this alternative depiction are shown below in Figure 15.



(a) Per-layer detections from SSD-256



(b) Per-layer detections of SSD-512



(c) Per-layer detections of SSD-768

Figure 15: **Detections clustered by source of detection layer.** High scoring detections from every detection layer are selected from SSD-256, SSD-512 and SSD-768. Detection layers are color-coded in prediction boxes. The digit on the top-left corner indicate instance class category.

Visual inspection of several image crops reveals that late layers of the SSD-512 (block10-conv2, block11-conv2, block12-conv2) do not ever contribute to detection. Detections are made on activation maps that result from block4-conv1, block6, block8-conv2 and block9-conv2. As expected, early layers are responsible for the detection of Pf objects. More specifically, on block4-conv1 (earliest) activation maps, where exclusively Pf are detected. Under the same logic, maps from block6, block8-conv2 and block9-conv2 serve to detect WBC only. Despite of counting on 3 layers for WBC detection, SSD-512 does not present the highest proportion of WBC detection overall.

Similar behavior is exhibited by SSD-256, where detection of either Pf or WBC are registered exclusively on block3-conv1 and block4-conv1, respectively. Differently, block3-conv2 contributes to the detection of both classes.

This seems to be add more to class confusion (Pf are classified as WBC, and viceversa) than, let's say, reducing miss rate scores.

In the case of SSD-768, detection layers appear to be task-specific too: activation maps from maxpool-2 respond mainly to Pf objects, whereas block4-conv1, to WBC objects. Despite being task-specific, block4-conv1 causes a few false detections by confusing WBC with PF objects. Anyhow, most false detections are still caused by Pf artifacts.

In sum, the selection of activation maps for detection seems to be critical for performance. A proper selection can help reduce the number of false positives due to class confusion. It tends to occur in early layers of the network (block3-conv2 in SSD-256 and maxpool-2 in SSD-768), where it lacks high-dimensional information to discern between classes. Still, a significantly large proportion of false positives are due to miss-classification of artifacts.

3.2 Transfer Learning: VGG-16 feature extractor

During this set of experiments, weights are transfered from a pre-trained VGG-16 to the SSD Multi-box feature extractor (also, VGG-16). Transfered weights of VGG-16 convolutional blocks are "frozen" during training in a cascade fashion; namely, for the same network, we execute as many experiments as there are VGG-16 blocks: freezing up and until blocks 1, 2, 3, 4 and 5, respectively (*c.f.* Table 3a). Trainable weights, opposed to frozen weights, are initialized using two different strategies: either, (1) using pre-trained weights, or (2) following Xavier uniform initialization strategy [Glorot and Bengio, 2010]. We run these experiments on SSD-256 and SSD-512 models, the best-performing models on the task at hand.

3.2.1 Detection performance

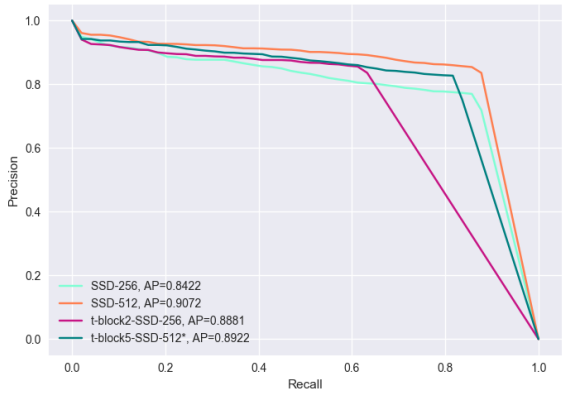
	Average Precision			**Recall@AP**		
	Weighted Mean	PF	WBC	Weighted Mean	PF	WBC
SSD-256	85.10 %	84.21 %	93.17 %	87.58 %	86.89 %	94.10 %
t-block2-SSD-256	88.31 %	**88.80 %**	82.44 %	65.13 %	62.58 %	90.66 %
SSD-512	89.92 %	90.72 %	81.27 %	88.46 %	87.52 %	97.54 %
t-block5-SSD-512	87.49 %	89.22 %	69.44 %	82.83 %	82.03 %	90.42 %

Table 5: Performance comparison of the best two performing models (SSD-512 and SSD-256) against the best performing model resulting from transfer learning.

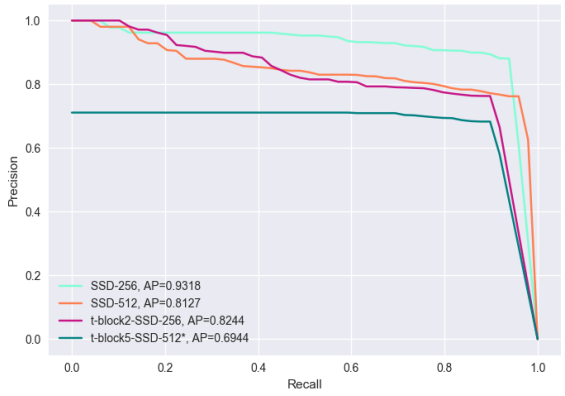
Best results for each SSD variant followed the weight initialization strategy described in (1) and are shown in Table 5. Interestingly, for SSD-256, pre-trained VGG-16 kernels transfered from convolutional blocks 1 and 2 (t-block2-SSD-256) resulted in good feature abstractions for this model. t-block2-SSD-256 shows an improvement in precision scores of Pf class objects, with an increase of +5.45% w.r.t SSD-256 performance (Figure 16a). However, in every other aspect, SSD-256 remains superior in performance (Figure 16b, 16d). Table 5 reveals t-block2-SSD-256 to perform comparably poor to every other model at detecting Pf objects.

In the case of SSD-512, it seems that transfered feature extractors did not fully capture the distinctive features of neither Pf nor WBC objects. Or alternatively, the activation maps chosen for detection did not contain the required information for convolutional predictors to discriminate class instances correctly. This is more visible for

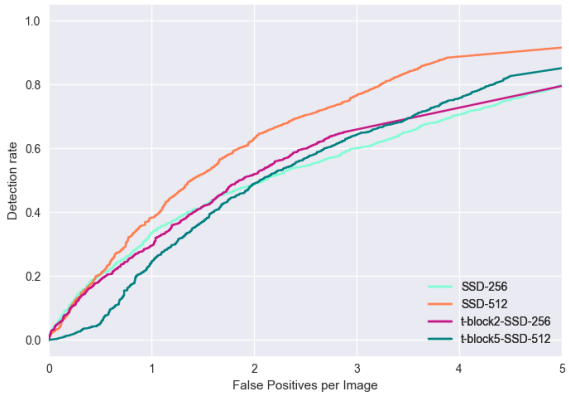
the WBC class, where precision levels drop heavily. Surprisingly, using a full pre-trained VGG-16 network (t-block5-SSD-512) shows, by a large margin, the best performance over every experimental case of the same network: using pre-trained blocks up and until 1, 2, 3 and 4 leads the network to converge to an suboptimal local minima and show random-like performance. Albeit showing the best results among every transfer learning experiment on SSD-512 model, t-block5-SSD-512 does not outperform the benchmark performance set by SSD-512.



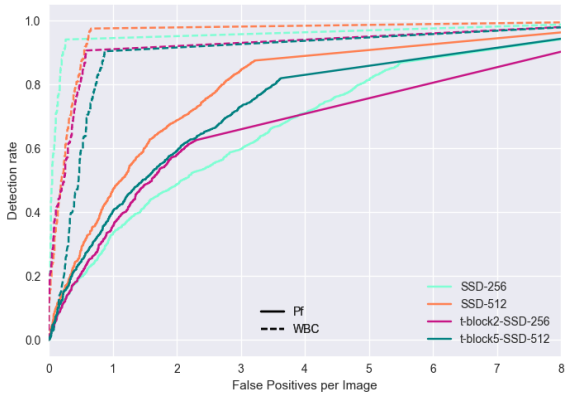
(a) Precision-Recall curves, Pf class



(b) Precision-Recall curves, WBC class



(c) Class-agnostic Recall vs. FP p/ image



(d) Per-class Recall vs. FP p/ image

Figure 16: **Detection and Classification performance** of SSD-256, SSD-512 and t-block2-SSD-256 models.

3.2.2 Further analysis

We hand-pick strongly activated feature maps of SSD-256 and t-block2-SSD-256 and depict them in Figure 17. In both cases, activation maps from blocks 3 (conv1, conv2), 4 (conv1) and 6 are used for detection. Figure 17 illustrates which elements of the input image activate neurons in these layers. We know that, for deep networks and gradient-based learning, the size of the gradient tends to vanish as it back-propagates through the network. Namely, weight updates of early layers are relatively small in comparison to late layers, and so it can take up to many training steps for early layers to converge. Here, early layers (block2-conv1, block3-conv1) from t-block2-SSD-256 benefit from efficient edge detectors in comparison to their counterparts in the SSD-256 case. Kernels in block2-conv1 and block3-conv1, trained on millions of images for hundreds of thousands training steps, distinguish the nucleus and the ring-form body of Pf objects to the detail, as well as of the WBC objects. Note that block3-conv2 shows no activation for Pf objects contained in the image sample in Figure 17. This might account for the

increase in precision of t-block2-SSD-256 for Pf class objects (due to block3-conv1), but the decrease in recall of the same class (due to block3-conv2).

Activation maps block4-conv1 and block6 (latter not shown in Figure 17), on the other hand, are used to detect WBC-sized objects. It seems that pre-trained weight initialization aids block4-conv1 to activate only regions where large objects are present, or alternatively, activate every other region except where the object is. The same seems to happen in activation maps from block3-conv2, yet much less strongly. Its counterparts in SSD-256 show a clear activation for large objects, such as WBC objects, yet with greater detail. Particularly, maps from block3-conv2 in SSD-256 get activated by internal structures from WBC objects, as opposed to only contours from maps in block4-conv1 in t-block2-SSD-256. It seems that SSD-256 have captured relevant features that describe WBC objects better that, in turn, help the network discern them from large artifacts in the pictures. This finding might be related to the drop in WBC class precision in t-block2-conv1 model.

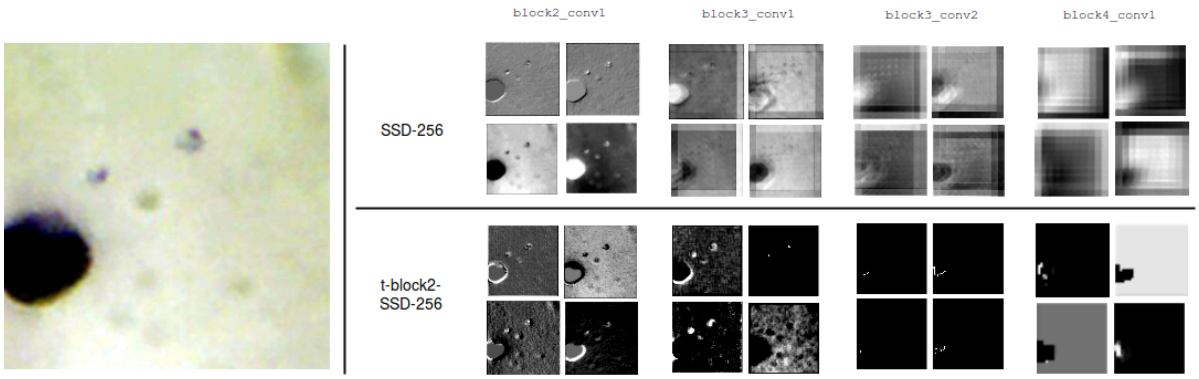


Figure 17: **Activation maps** from SSD-256 and t-block2-SSD-256.

In sum, pre-trained weights from block2 in SSD-256 have leverage the network's precision on the detection of WBC class objects. However, immediate later layers do not seem to activate to Pf objects, which might be responsible for the decrease in Pf class recall levels. Overall, SSD-256 did not benefit from transferred weights. Similarly, SSD-512 did not benefit from transfered weights either. SSD-512 still shows the best detection performance and the best trade-off between detected objects and the average FP frequency per image.

4 Discussion

4.1 General discussion

This thesis explores the application of a fast object detector (SSD Mutli-box) to the localization and classification of Pf and WBC objects in digitalized, malaria-positive images from thick blood films. Results indicate that the best performing model (SSD-512) achieves a sensitivity of 87.52% and an AP of 90.72% in the detection of Pf, whereas in the detection of WBC, it achieves 97.54% and 81.27%, respectively. This level of performance is achieved at the expense of 5 false Pf detections and 2 false WBC detections per image, in average. Results are displayed in Table 6 in comparison with results of every other publication to date on the topic.

Yet, hardly comparable in some cases due to a different metric choice or some other constraint, similar performance to that demonstrated by the SSD is published under other methods. Some studies report superior performance, at least partly, however not without important limitations. For instance, methods in Elter et al. [2011] (97%) and Ketut et al. [2013] (91%) claimed to be more sensitive to Pf than the SSD (88%) showed to be. Likewise, Quinn et al. [2016] (97%) reported higher AP scores, and Elter et al. [2011] (0.8), again, exhibited the lowest record of average FP per image, both in the task of Pf detection. In any case, the problem description is a subset of the one presented here: the binary case of classifying candidates into malaria-positive or -negative. With the exception of Rosado et al. [2016b] and Kaewkamnerd et al. [2012], automated malaria detection is defined as a binary problem. And even these exceptional studies, reduce a multi-class problem to a binary one by using a single algorithm per class category. In this thesis we propose a unified, single-shot method to solve a multi-class problem. The addition of more classes leads to a higher model complexity, that may result in poorer classification performance. In sight of this, SSD achieves promising results. Now, Rosado et al. [2016b] is comparably the most similar study to this one: they aim to detect Pf and WBC for an end mobile application. In spite of counting with higher resolution pictures, the performance achieved by SSD is superior in every aspect.

Unfortunately, some other studies in Table 6 preferred metrics, whose calculations include true negative frequencies (specificity in ROC analysis or accuracy). This supposes an obstacle for a one-to-one comparison of the results. Given the over-represented negative (background) class in our dataset, our choice of metrics did not include true negative frequencies to avoid any imbalance bias.

The size and quality of the image set are important aspects of the problem. One can say that they define the problem at hand. Malaria diagnosis is a non-trivial one (complex background, many artifacts, staining variations, smear thickness, etc), thus collecting good quality, high-res pictures for diagnosis is imperative. We made use of the image set collected and employed by Quinn et al. [2014]. Pictures in this set were captured by a Motic MC1000 camera, a popular microscope digital camera that delivers images of 1280×1024 pixels. Alternative mobile-captured image sets from studies shown in Table 6 show a higher resolution than images in our set. In spite of this disadvantage, SSD proved to be highly sensitive and precise in the recognition of Pf and WBC objects contained in the set. Quinn et al. [2014] reported, likewise, good results, yet SSD offers better precision on spotting Pf and it further extends the problem to a multi-class case to potentially deliver an estimation of parasitaemia. We believe that higher resolution pictures will have a positive impact on recall levels of the SSD, and in the decrease of FP, since most positive predictions are confused with artifacts found in the background (see Figure 11). It should be noted that Quinn et al. [2014] adopted a sliding window approach followed by a ConvNet classifier. This is awfully time-consuming, and it increases as image resolution does. SSD, in turn, is a real-time detector (discussed

Author/Year	Segmentation	Feature Extraction	Classifier	Dataset Size	Performance	FP	Time
[Kaewkamnerd et al., 2012]	Adaptive thresholding	Hue histogram, Chromatin size	Template matching	20 (F)	ACC(Pf): 90, ACC(Pv): 75	–	–
[Elter et al., 2011]	Thresholding	Stats. moments (60), texture (52), color (62)	SVM	256 (F)	SE(Pf):97	Pf: 0.8	–
[Yunda et al., 2012]	AGNES, Morphological gradient techniques	Wavelet-based features, PCA	MLP	248 (F)	SE(Pv): 77	–	–
[Ketut et al., 2013]	–	Stats. moments (4), Entropy of histograms of RGB, HSV, HIS space	Genetic programming	180 (C)	SE(Pf):91	–	–
[Quinn et al., 2014]	–	Connected components, moments	Random Forest	2703 (MF)	ROC-AUC(Pf): 0.97, AP(Pf): 69	–	–
[Rosado et al., 2016b]	Otsu and adaptive thresholding	Geometry, color, texture (314 total)	SVM	194 (MF)	SE(Pf): 80, SE(WBC): 98, SP(Pf): 94, SP(WBC): 72	Pf: 8, WBC: 1	0.21 FPS
[Quinn et al., 2016]	Sliding window sampler	–	ConvNets	1182 (MC)	ROC-AUC(Pf): 1.00, AP(Pf):97	–	–
This thesis	Aggregated mesh of default bounding boxes	–	SSD Multi-box	2073 (MC)	SE(Pf): 88, SE(WBC): 98, AP(Pf): 91, AP(WBC): 81	Pf: 5, WBC: 2	9 FPS

Table 6: **Table comparison of different approaches to malaria detection in thick blood smears including ours.** Extended Table 1 in [Rosado et al., 2016a]. Every publication dedicated to the detection of malaria and/or white blood cells on digitalized images from thick blood smears. *Glossary:* **AGNES** stands for "Absence of gradients and Nernstian equilibrium stripping"; **C** stands for cropped sub-images; **F** stands for full resolution images; **M** stands for mobile or Motic cameras; **SE, SP, ACC** stand for sensitivity, specificity and accuracy, respectively; **AUC** stands for Area Under the Curve ; **Pv** stands for Plasmodium vivax.

below). Lastly, in terms of number of samples, this study registers the largest dataset of digitalized images from thick blood smears to date. We extend Quinn et al. [2014] dataset with 16 high-res images presenting a high-level of parasitaemia. That results in an addition of 524 Pf and 66 WBC labels. Though a limited addition, is still, by a large margin and jointly with Quinn et al. [2014], the largest set collected under this study subject. The size of the dataset affects the prediction power of a model. The more samples of target objects it has seen, the better it will generalize to new, unseen ones.

Last but not least, SSD-512 infers detections on `(512, 512, 3)` image crops at 9 FPS on GPU, which accounts for one to two full images in the test set (largest image: 9 crops, smallest image: 4 crops). Pre- and post-processing

stages add up for an extra 50ms per full image. Note that inference times plunge in comparison to the ones registered for the same input size: Liu et al. [2015] reported 19 FPS in average with a total of 24,564 default boxes, while our model infers at a rate of 9 FPS with lesser default boxes (10,922). In both cases a batch size of one is used. We believe the drop in time performance is mainly due to less powerful hardware in our end. Original inference times are measured on Nvidia Titan X GPUs, faster processing units than Tesla K40 GPUs used in this thesis. Comparable time performances can be found in, for instance, Yi et al. [2017], where they use a light-weighted SSD detector implemented in Keras, with an input size of `(300, 300, 3)` on Tesla K40 GPUs detecting at a rate of 10 FPS. Rosado et al. [2016b] reported SVMs inference times, and were the only authors to do so. They claim to make predictions at a rate of 0.21 FPS (4.63 seconds), that is approx. $\times 41$ slower than SSD-512 time performance. SSD yields record times with no real competitor among detectors in Table 6.

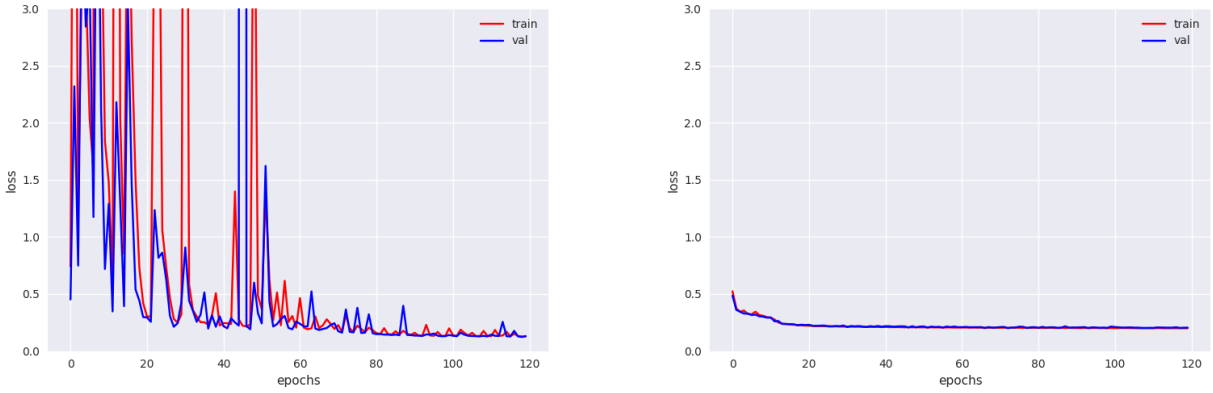
The following sub-sections are intended to discuss about performance and analysis results, limitations and other insights on the SSD Multi-box application to malaria detection.

4.2 On receptive fields and activation maps

Receptive fields refer to the region of the input space where a neuron of a convolutional layer is "paying attention to". In the SSD framework, receptive fields are the key concept to understand the inner-workings of the detection ability of the network. Under this lens, some conclusions can be drawn from empirical evidence:

(i) **Early layers work best for detection of objects at small scale.** Interestingly, the lack of semantic descriptors of feature abstractions from early layers did not seem to weigh in as much in the detection of small objects. However, we believe it had an effect on the frequency of FP. In particular, in the detection of Pf objects, where the only source of confusion (2% of all Pf detections) is in contrast with artifacts in the background. This suggests that feature abstractions in early layers suffice for localization, yet more relevant descriptors are required for better classification. We mentioned that our dataset consist of low-res pictures, and thus opting on later layers for detection will possibly not resolve this issue. This stands in congruence with results in Section 3.1.4, and other trial-runs not reported in this thesis. We believe that high-res pictures will allow for later layers of the network to extract relevant information of Pf object to better discriminate them from artifacts.

(ii) **"Objectness" learning works best if anchor boxes are contained in receptive fields.** Performance results were in agreement with our intuitions. There is virtually no limit to the size of anchor boxes of the network, yet the ones that were contained in their receptive fields delivered the best localization performance. For instance, as it the case with SSD-512 (see Table 7). It seems that having access to surrounding information of objects centered in anchor boxes has a positive impact in performance. Notwithstanding, the reverse case (anchor box larger than its receptive field) still allows the SSD to learn object localizations, yet not with some limitations. Figure 18 shows smooth-ℓ_1 localization errors of SSD-256 and SSD-768 along training epochs. Both models specify anchor boxes that are larger than their receptive fields. In the case of SSD-256, the network did not show a stable learning up to around epoch 55. We conjecture that is due to the lack of peripheral information inside anchor boxes, rather than, for instance, bad weight initialization or learning step size. In the case of SSD-768, on the other hand, localization learning seemed to monotonically-decrease without much difficulty, yet it presents the highest miss rates among detectors (Pf: 24%, WBC: 8%). In sum, specifying anchor sizes



(a) SSD-256 (b) SSD-768

Figure 18: **Localization loss** of SSD-256 and SSD-768 during training.

that are contained in their receptive fields aids the SSD to localize objects faster and effectively. Anchor boxes significantly larger (not enough information) or significantly smaller (higher noise-to-signal ratio) do not seem to work well.

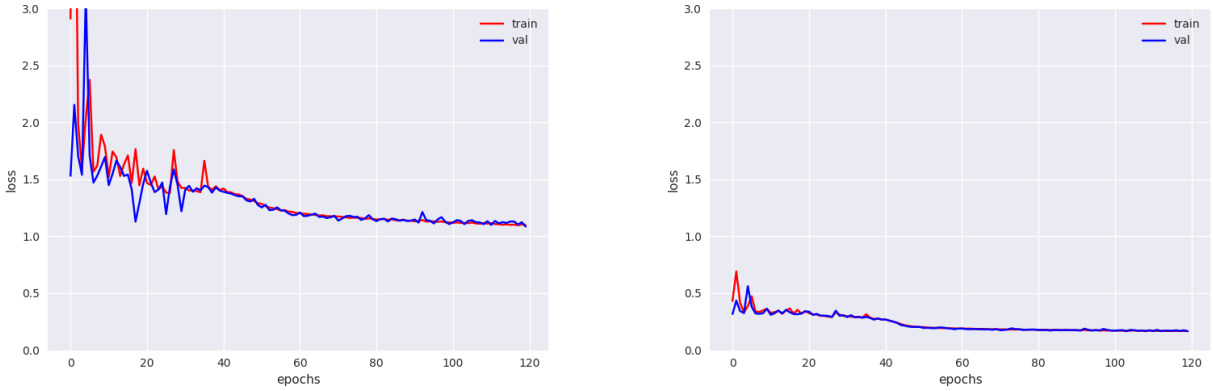
(iii) **A light-weighted version of SSD can deliver similar performance if activations maps are carefully chosen.** Results in Section 3.1.4 show the contribution of every detection layer of the network to detection. These depictions suggest that, for the problem of malaria detection, fewer layers than that of the original implementation were required for detection. Layers in block 10, 11, and 12 from SSD-512 did not recognize any objects, and can be easily be left out. SSD-256 and SSD-768 showed competitive results with 4 and 2 detection layers, respectively. Visual inspection of activation maps and detections per layer proved to be a great way of deciding on relevant detection layers, and to spare those ones that do not contribute to detection at all. Certainly, we can benefit from computational gains with a light-weighted version of SSD without loss of detection performance. Other techniques can potentially provide with great gains in performance, such as kernel prunning [Howard et al., 2017] or efficient base networks, like MobileNet [Li et al., 2016].

(iv) **Variations in crop size did not improve detection, but showed great gains in time performance.** Our intuition is that smaller, overlapping crops of input images could rise the signal-to-background ratio, and in turn, increase classification performance. However, results indicate that this might, in fact, not be the case. There is no indication that performance increases as input size decreases. Variables like anchor box sizes or choice of detection layers proved to be more determinant in SSD detection ability. In any case, results are inconclusive. Time and computational limitations to cross-validate performance under variations of the window size make it hard to rule out the possibility of an effect.

Yet, benefits were observed in time inferences. The window size determines the number of anchor boxes of the network. SSD-256 detects at a rate of 46 FPS with $18,944$ anchor boxes, $\times 5$ faster than SSD-512 with only $\times 0.57$ more anchor boxes. Pre- and post-processing stages add approx. 60ms to the total time per full image (12 image crops). This suggest that, with better fine-tunning of the network, great gains in time performance can be achieved with small(er) input sizes.

4.3 On SSD multi-objective loss

SSD Multi-box attempts to minimize two objectives: localization (regression) and identification (classification) of targets. It does so by aggregating them into a scalar loss expression, $L = L_{conf} + \lambda\ L_{loc}$, where λ is a free parameter such that $\lambda > 0$. Multi-objective learning is a non-trivial optimization problem that presents possibly many optimal solutions. Scalarizing the function does not render the problem convex (single optimal solution), but constraints the solution to a single value (the scalar value) under a particular choice of λ. A disadvantage of scalarization is that it offers little insight into the problem to be optimized, and that it depends greatly on λ. This becomes particularly relevant if the objectives are in conflict with each other, and no single solution exists that yields every objective optimal.



(a) Confidence loss

(b) Localization loss

Figure 19: **Multi-objective loss.** Localization and Confidence training loss of SSD-512. Note that they follow an similar trend throughout training epochs: after epoch 40, both objectives register a steep(er) decrease and slowly start to converge.

It is hard to tell whether objectives in our problem are in a conflictive relationship. Empirical evidence from training sessions suggest the opposite. Figure 19 depicts the training loss of the SSD-512 reported above, where every objective contributes equally to the total loss ($\lambda = 1.$). Both objectives canonically decrease in value, and start converging almost simultaneously after several training steps (epoch 40). This suggest not only the ease of the network to learn localization faster than object recognition, but it also suggests a cooperative association between the objectives. Note that, intuitively, classification performance can only thrive once the localization of targets is in place. We believe that, given this apparent cooperation among objectives, much can be gained from dynamically fine-tunning λ. For instance, lowering lambda after the localization loss remains unchanged for a certain number of training epochs [Jin and Sendhoff, 2008].

Moreover, multi-objective learning poses a regularization side-effect on the network that results highly beneficial in the case of complex models. In the SSD framework, multi-objective learning is proposed under a "hard parameter sharing" method, where hidden layers of the network are shared between objectives (Figure 20). This reduces greatly the risk of over-fitting. The model seeks for a representation that captures relevant features for every task, and in doing so reduces the risk of over-fitting on the main task.

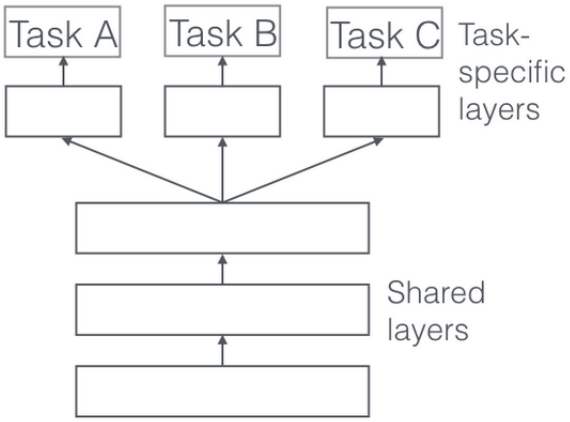


Figure 20: **Hard parameter sharing.** An illustration of the method of hard parameter sharing in multi-objective learning in deep neural networks. Source: `http://ruder.io/multi-task/`.

4.4 On transfer learning

Weights from a VGG-16 network pre-trained on ImageNet 2012 set were transfered to the base network of SSD detectors to benefit from knowledge extracted from long training sessions. Unfortunately, transfer learning did not translate to the problem of malaria detection as expected. Experiments with SSD-512 under-performed in every occasion, whereas SSD-256 improved only on Pf precision rates when weights from block 1 and 2 of convolutional layers are initialized (and not trained) with pre-trained weights. In sight of this and by visual inspection of Figure 17, one can realize the benefit in performance that pre-trained weights from early layers could imply for the SSD. Weights in early layers seem excellent edge detectors obtained after long training sessions on hundreds of thousands of image samples. As laid out sooner in this thesis, gradient-based weight updates in early layers are small, and achieving good low-level detectors can take too many training steps. It should be noted that we did not observe the benefits of transferring learning to early layers in the SSD-512 case. We believe a plausible explanation is concerned with convergence to a sub-optimal local minima due to an unfortunate case of bad weight initialization of the rest of the layers. In any case, transferring learning to late layers, either as a weight initialization method or for direct inference, led both models to converge but to detect poorly. SSD seemed to recognize some objects of interest, though missing many of them, and to confuse class identities quite regularly. These results are not reported since only the best-performing models of every series of experiments were. Nonetheless, this suggests that task-specific information of the problem at hand to be essential for correct discrimination of objects. Unfortunately, ImageNet 2012 image set contains very different objects than the ones of interest in this thesis.

5 Conclusion

This thesis introduces a high-performing, real-time malaria detector. It localizes and identifies plasmodium falciparum (recall: 87.52%; precision: 90.72%) and white blood cells (recall: 97.54%; precision: 81.27%) on a single feed-forward pass of an SSD Multi-box network, a state-of-the-art deep learning algorithm for object detection. In the multi-class case (Pf + WBC), SSD offers the best detection performance to date, even when detecting in low resolution images. However, there is a lot of room for improvement. Here, we disclosed the importance of the choice of model layers for detection, and receptive field considerations on the size of anchor boxes of the network. We believe that further experimentation under these considerations can increase detection performance. We also made remarks on the effect of the image set quality and size. We believe that higher resolution images can obtain huge gains in performance. SSD did not benefit greatly from transfer learning techniques, yet results suggest that transferring low-level features from early layers can be beneficial.

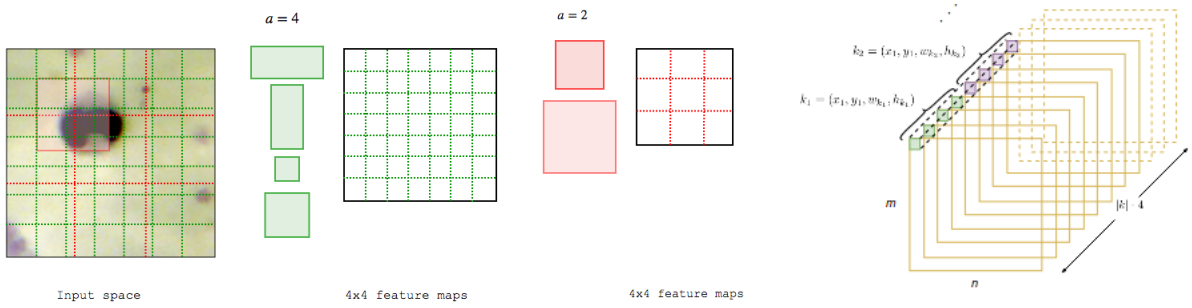
In sum, the contribution of this application is two-fold: (a) it offers an easy way to estimate parasitaemia in a blood smear by counting parasites and white blood cells per field-of-view. This feature truly allows for a real-world application to malaria diagnosis. And (b) inference is real-time. SSD delivers an unprecedented time performance. Blood smear examination is the bottleneck of malaria diagnosis. This application can accelerate diagnosis to as fast as it takes a clinician to capture pictures of a field-of-view.

6 Appendix

A Module predictors: computing class confidence scores and bounding boxes offsets.

As aforementioned, module predictors consist of two convolutional layers, whose neurons either evaluate to sigmoids to compute class confidence scores (class predictors), or evaluate to offsets in center-box coordinates, width and height of anchor boxes (box predictor). To do so, module predictors divide the spatial dimension of activation maps into $m \times n$ grids. Points in the grid are center coordinates of anchor boxes, and coincide with the center location of receptive fields of predictive neurons in modules (Figure 21a). Note that the output of predictors is of the same resolution as their input feature maps. Kernels in predictors are 3×3 in size, and perform "valid" convolutions with a stride of 1 to input regions of designated activations maps. There are $a \cdot K$ number of kernels in class predictors, and $a \cdot 4$ in box predictors. That is, a whole spatially invariant map for each box coordinate, box dimension or class confidence (Figure 21b).

Box predictors start off with a fixed set of anchor boxes, $\{D_m\}_{m=1}^{M}$, where $d_i = (x_i, y_i, w_i, h_i) \in D_m$ for M detection layers, and propose "adjustments" to them to better match target boxes. Adjusted anchor boxes are referred to as box proposals of the network. Given an input set r of aspect ratios [13], there are a anchor boxes types (w_a, h_a) on a point in the grid drawn on any m detection layer, where $a = \{1, \cdots, |r_m| + 1\}$. The definition of anchor boxes are illustrated in Algorithm 1. Note that $|\{D_m\}_{m=1}^{M}|$ is of fixed size for a particular input size and network configuration.



(a) Stacked center-box grids from (8 × 8) and (4 × 4) detection feature maps.

(b) Box offsets computations of point (1, 1) of the grid.

Figure 21: **Operation of anchor boxes of the network.** **(a)** Center-box grids project to the input space in the same way receptive fields do. In this way, an effective discretization of the input space is achieved. Aspect ratios and anchor sizes are defined per detection layer. **(b)** At every (x_n, y_m) of a detection feature map, a anchor boxes offsets are computed. Each box uses 4 maps for its calculation $(x, y, width, height)$.

Class predictors, on the other hand, compute K object class categories per anchor box. Concretely, they compute $c_i = (c_i^1, c_i^2, \cdots, c_i^K)\ \forall\ d_i$, where c_i is the softmax across K classes.

[13] Refers to the proportional relationship between the width and height of a rectangle box (W:H); e.g. in $r = \{1, 2, \frac{1}{2}, 3, \frac{1}{3}\}$, $r = 1$ defines a square (1:1), $r = 2$ defines a horizontal rectangle (2:1), and so forth.

Algorithm 1: Define anchor boxes $(x, y, width, height)$ of the network.

Function `defineAnchorBoxes`(s, r, M)**:**

- $s \leftarrow (s_{min}, s_{max})$ scale of objects to be detected across M maps
- $r \leftarrow$ anchor box aspect ratios
- $M \leftarrow$ number of detection layers
- $\{(s^m_{min}, s^m_{max})\}^M_m \leftarrow linspace(s_{min}, s_{max}, M)$
- **foreach** $m \in M$ **do**
  - `// every location along the width and height of detection volume` M
  - $y_m, x_m \leftarrow (w_m \times h_m)$
  - **foreach** $r^m_i \in r$ **do**
    - $w_1, h_1 \leftarrow \frac{\sqrt{s^m_{min} \cdot s^m_{max}}}{w_m}, \frac{\sqrt{s^m_{min} \cdot s^m_{max}}}{h_m}$
    - $w_{2..|r|+1}, h_{2..|r|+1} \leftarrow \frac{s^m_{min}}{w_m} \cdot \sqrt{r^m_{2..|r|+1}}, \frac{s^m_{min}}{h_m \cdot \sqrt{r^m_{2..|r|+1}}}$
  - **end**
  - append $\{x_i, y_i, w_i, h_i\}^{|r|+1}_i \rightarrow D_m$
- **end**
- **return** $\{D_m\}^M$

In sum, the objective of the network is to learn class categories and offsets w.r.t target boxes guided by features learned from the data. The result is a collection of box proposals with assigned class predictions for objects in them.

B Multi-task loss function & ground truth encoding

The learning objective of an SSD network is the weighted sum across multi-task objectives $\{l_m\}_{m=1}^{M}$, where l_m is the loss computed at detection layer m (see equation 1). During training, network parameters $\mathbf{W}$ are learned from a set of training samples $\{(X^i, Y^i)\}_{i=1}^{N}$, where $Y^i = (c^i, g^i)$ is the pair of target bounding box coordinates $g^i = (x^i, y^i, w^i, h^i)$ and its associated class label $c^i \in \{0, 1, 2, \cdots, K\}$, and X^i is the training image area spanned by g^i [Cai et al., 2016].

$$\mathcal{L}(\mathbf{W}) = \sum_{m=1}^{M} \sum_{i \in S_m} \alpha_m l_m(X^i, Y^i | \mathbf{W}) \tag{1}$$

The loss l_m is the the joint contribution of confidence and localization losses [Cai et al., 2016; Liu et al., 2015]. We begin with matching ground truth boxes $g \in G$ with anchor boxes $d \in D$ that exhibit a Jaccard index higher than a threshold (0.5)[14]. Matched boxes $\hat{g} \in \hat{G}$ are parameterized as originally proposed in Girshick et al. [2014]:

Algorithm 2: Build training targets by matching with ground truth boxes g with anchor boxes d of the network.

Function `encodeGroundTruth(`G, D`)`**:**

  foreach $g \in G$ **do**

    foreach $d \in D$ **do**

      if $IoU(g, d) \geq 0.5$ **then**

        $x_{\hat{g}}, y_{\hat{g}} \leftarrow \frac{x_g - x_d}{w_d \cdot x_s}, \frac{y_g - y_d}{h_d \cdot y_s}$ `// scale-invariant translations of center coordinates`

        $w_{\hat{g}}, h_{\hat{g}} \leftarrow \frac{\log\left(\frac{w_g}{w_d}\right)}{w_s}, \frac{\log\left(\frac{h_g}{h_d}\right)}{h_s}$ `// log-space translations of the width and height`

        `/* where` (x_s, y_s, w_s, h_s) `are scaling factors */`

        append $\hat{g}$ to $\hat{G}$

    end

  end

  return $\hat{G}$

This parameterization encodes pixel coordinates into translations (or "offsets") relative to their matched anchor boxes, i.e. precisely what the network is attempting to learn. It follows that, during the inference phase, box offsets $o \in O$ require a decoding step wherein "adjusted" anchor boxes, or box proposals, are mapped back into pixel space[15] (see Figure 22).

Only box proposals whose matching anchor boxes are labeled as ground truth contribute to the loss. Unmatched anchor boxes are discarded. Let $A_{ij}^{k} \in \{1, 0\}$ denote the assignment of the i^{th} default box to the j^{th} ground truth

[14]The Jaccard index is the intersection over the union (IoU) of the volumes of a pair of boxes.

[15]Decoding function $\varphi^{-1}(O, D)$ in Fig. 22 reverses the transformation detailed in Algorithm 2 for center coordinates $\left(x_d \cdot x_l \cdot x_s + x_d, y_d \cdot y_l \cdot y_s + y_d\right)$ and for the width $\left(\exp(h_l \cdot h_s) \cdot h_d\right)$ and height $\left(\exp(w_l \cdot w_s) \cdot w_d\right)$

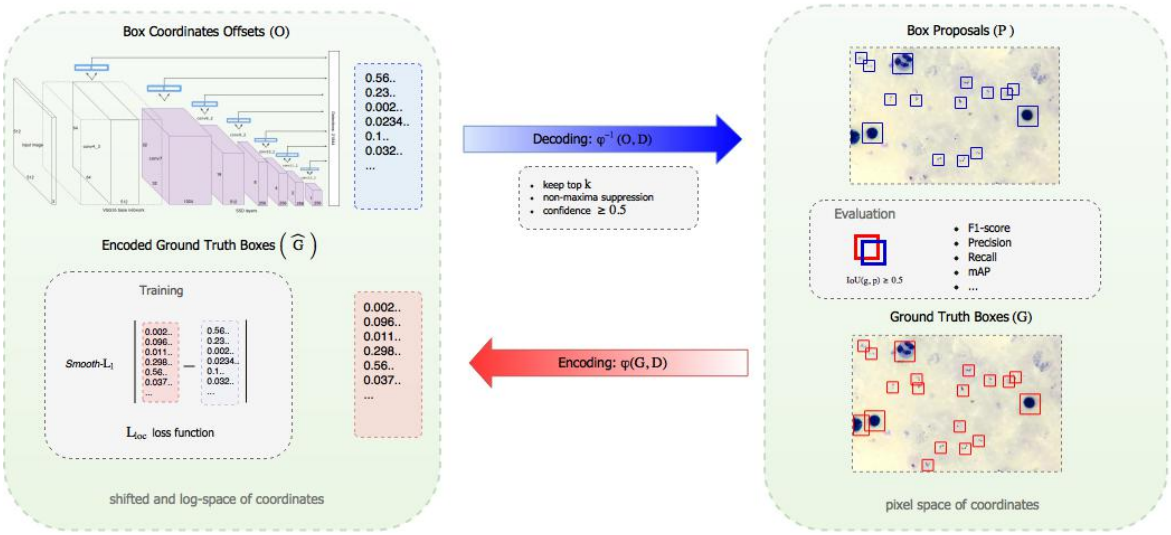


Figure 22: **Encoding-Decoding scheme.** Illustrates the encoding $\varphi(G, D)$ and decoding $\varphi(O, D)$ transformations of bounding boxes. Ground truth boxes coordinates are reduced to a translated logarithmic space of their coordinates, where it is used for training. Next, these transformations are reversed back to pixel space, where assessments on input images are in order.

box of an object class category $k \in K$. Then, the joint loss for layer m is defined as,

$$l(X, Y|\mathbf{W}) = \frac{1}{N}\big(L_{conf}(A, c) + \lambda\ L_{loc}(a, O, \hat{G})\big) \tag{2}$$

where $c = (c^0, c^1, .., c^K)$ is the class predictor output, $N = \sum_{i,j,k} x_{ij}^k$ is the number of positive samples, and λ is a trade-off coefficient between classification and localization objectives.

As first proposed by Erhan et al. [2014], we regress towards target offsets for center box coordinates (x, y) of anchor boxes, and for its width (w) and height (h). The localization loss L_{loc} is the Smooth ℓ_1 loss between anchor box offsets and target offsets. Then,

$$L_{loc}(A, O, \hat{G}) = \sum_{i,j,k} a_{ij}^k \Big(\frac{1}{4} \sum_{b \in (x,y,w,h)} \text{smooth } \ell_1(o_i^b - \hat{g}_j^b)\Big) \tag{3}$$

where, $\hat{g}_i$ is the transformed match of a ground truth annotation to an anchor box, and

$$\text{smooth } \ell_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{4}$$

Least absolute deviations (ℓ_1-norm) are chosen for its robustness in that that is less sensitive to outliers than, say, ℓ_2-norm, since residuals are not squared. In our case, regression targets are virtually unbounded, therefore can require careful fine-tunning of the training loss if trained with ℓ_2. Smooth ℓ_1 adds yet more robustness to plain ℓ_1 loss and makes the training more stable (ensures differentiability).

The classification objective L_{conf} is the categorical cross-entropy over multiple classes. Given that the negative class (or background) represents the majority class by a large amount, we select only a reduced amount of negative samples to contribute to the loss. The intention is to prevent the learning process to be biased towards the majority class. This is made effective by sorting the top-ranked negative boxes in a fixed optimal positive-to-negative ratio.

Then, for class predictor output c, Liu et al. [2015]) defined L_{conf} as follows,

$$L_{conf}(A, c) = -\sum_{i,j}^{N}\sum_{k=1}^{K} A_{ij}^{k} \log(\hat{c}_i^k) - \sum_{i \in neg} \log(\hat{c_i}^0) \tag{5}$$

where, $\hat{c}_{ij}^k$ is the soft-max $\frac{\exp(c_{ij}^k)}{\sum_k \exp(c_{ij}^k)}$, and $\hat{c}_i^0$ is the class predictor output for the negative class.

Last but not least, optimal parameters $\mathbf{W^*} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{W})$ are learned using Adam algorithm, a gradient-based optimizer.

C Model Specifications.

Model	*Detection Layers*	*Feature Map Size*	(s_{min}, s_{max})	*Anchor boxes*	*Stride*	*Theoretical RF*	*Num. of Params*	$\lvert\{D_m\}_{m=1}^{M}\rvert$
SSD-256	Block-3, Conv2D-1	(64, 64)	(35.84, 63.64)	2	4	32	19,707,524	18,944
	Block-3, Conv2D-2	(64, 64)	(61.44, 89.24)	2	4	40		
	Block-4, Conv2D-1	(32, 32)	(89.6, 117.4)	2	8	76		
	Conv2D-6	(16, 16)	(145.92, 173.72)	2	16	292		
SSD-512	Block-4, conv2D-1	(64, 64)	(40.96, 71.68)	2	8	76	23,896,738	10,922
	Conv2D-6	(32, 32)	(71.68, 102.4)	2	16	292		
	Block-8, Conv2D-2	(16, 16)	(102.4, 133.12)	2	32	356		
	Block-9, Conv2D-2	(8, 8)	(133.12, 163.84)	2	32	420		
	Block-10, Conv2D-2	(4, 4)	(163.84, 194.56)	2	64	612		
	Block-11, Conv2D-2	(2, 2)	(194.56, 225.28)	2	128	996		
	Block-12, Conv2D-2	(1, 1)	(225.28, 256.0)	2	256	2020		
SSD-768	Block-2, MaxPool	(192, 192)	(50.0, 200.0)	2	4	24	3,001,064	92,160
	Block4, Conv2D-1	(96, 96)	(130.0, 250.0)	2	8	76		

Table 7: **Model specifications of SSD-556, SSD-512 and SSD-768.** Note: (s_{min}, s_{max}) refers to the minimum and maximum size of anchor boxes, in pixels. $|\{D_m\}_{m=1}^{M}|$ refers to the total number of anchor boxes of the network. *Glossary:* **RF**, stands for receptive field.

Bibliography

Amexo, M., Tolhurst, R., Barnish, G., and Bates, I. (2004). Malaria misdiagnosis: effects on the poor and vulnerable. *The Lancet*, 364(9448):1896–1898.

Bian, J. (2016). Quiver. `https://github.com/keplr-io/quiver`.

Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. *CoRR*, abs/1607.07155.

Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., and Wu, J. (2018). Feature-fused ssd: fast detection for small objects. In *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, volume 10615, page 106151E. International Society for Optics and Photonics.

Cao, Z., Duan, L., Yang, G., Yue, T., Chen, Q., Fu, H., and Xu, Y. (2017). Breast tumor detection in ultrasound images using deep learning.

Caraballo, H. and King, K. (2014). Emergency department management of mosquito-borne illness: malaria, dengue, and west nile virus. *Emergency medicine practice*, 16(5):1.

Center for Disease Control and Prevention. Laboratory identification of parasites of public health concern. Data retrieved from `https://www.cdc.gov/dpdx/malaria/index.html` on 2017-12-06.

Das, D. K., Ghosh, M., Pal, M., Maiti, A. K., and Chakraborty, C. (2013). Machine learning approach for automated screening of malaria parasite using light microscopic images. *Micron*, 45:97–106.

Dowling, M. and Shute, G. (1966). A comparative study of thick and thin blood films in the diagnosis of scanty malaria parasitaemia. *Bulletin of the World health Organization*, 34(2):249.

Elter, M., Haßlmeyer, E., and Zerfaß, T. (2011). Detection of malaria parasites in thick blood films. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5140–5144. IEEE.

Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. (2014). Scalable object detection using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2154.

Filippov, A. and Glazunova, Z. (1988). The importance of a quantitative assessment of parasitemia in tropical malaria. *Meditsinskaia parazitologiia i parazitarnye bolezni*, (4):18–21.

Fu, C., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. (2017). DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Gu, C., Lim, J. J., Arbeláez, P., and Malik, J. (2009). Recognition using regions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1030–1037. IEEE.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2016). Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*.

Jeong, J., Park, H., and Kwak, N. (2017). Enhancement of ssd by concatenating feature maps for object detection. *arXiv preprint arXiv:1705.09587*.

Jin, Y. and Sendhoff, B. (2008). Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415.

Kaewkamnerd, S., Uthaipibull, C., Intarapanich, A., Pannarut, M., Chaotheing, S., and Tongsima, S. (2012). An automatic device for detection and classification of malaria parasite species in thick blood film. *Bmc Bioinformatics*, 13(17):S18.

Ketut, E., Zakiyyah, R. F., Herry, P. M., et al. (2013). Malaria parasite identification on thick blood film using genetic programming. In *International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*.

Kim, H., Lee, Y., Yim, B., Park, E., and Kim, H. (2016). On-road object detection using deep neural network. In *Consumer Electronics-Asia (ICCE-Asia), IEEE International Conference on*, pages 1–4. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M. A., Sameer, A., Maude, R. J., et al. (2016). Cnn-based image analysis for malaria diagnosis. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 493–496. IEEE.

Linder, N., Turkki, R., Walliander, M., Mårtensson, A., Diwan, V., Rahtu, E., Pietikäinen, M., Lundin, M., and Lundin, J. (2014). A malaria diagnostic tool based on computer vision screening and visualization of plasmodium falciparum candidate areas in digitized blood smears. *PLoS One*, 9(8):e104855.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., and Reed, S. (2015). Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*.

Livni, R., Shalev-Shwartz, S., and Shamir, O. (2014). On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863.

Makler, M. T., Palmer, C. J., and Ager, A. L. (1998). A review of practical techniques for the diagnosis of malaria. *Annals of tropical medicine and parasitology*, 92(4):419–433.

Müller, H., Michoux, N., Bandon, D., and Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 73(1):1–23.

Nie, G.-H., Zhang, P., Niu, X., Dou, Y., and Xia, F. (2017). Ship detection using transfer learned single shot multi box detector. In *ITM Web of Conferences*, volume 12, page 01006. EDP Sciences.

Qiong, W. and LIAO, S.-b. (2017). Single shot multibox detector for vehicles and pedestrians detection and classification. *DEStech Transactions on Engineering and Technology Research*, (apop).

Quinn, J. A., Andama, A., Munabi, I., and Kiwanuka, F. N. (2014). Automated blood smear analysis for mobile malaria diagnosis. *Mobile Point-of-Care Monitors and Diagnostic Device Design*, 31:115.

Quinn, J. A., Nakasi, R., Mugagga, P. K., Byanyima, P., Lubega, W., and Andama, A. (2016). Deep convolutional neural networks for microscopy-based point of care diagnostics. *arXiv preprint arXiv:1608.02989*.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

Rosado, L., Correia da Costa, J. M., Elias, D., and S Cardoso, J. (2016a). A review of automatic malaria parasites detection and segmentation in microscopic images. *Anti-Infective Agents*, 14(1):11–22.

Rosado, L., Da Costa, J. M. C., Elias, D., and Cardoso, J. S. (2016b). Automated detection of malaria parasites on thick blood smears via mobile devices. *Procedia Computer Science*, 90:138–144.

Ross, N. E., Pritchard, C. J., Rubin, D. M., and Duse, A. G. (2006). Automated image processing method for the diagnosis and classification of malaria on thin blood smears. *Medical and Biological Engineering and Computing*, 44(5):427–436.

Rothe, R., Guillaumin, M., and Van Gool, L. (2014). Non-maximum suppression for object detection by passing messages between windows. In *Asian Conference on Computer Vision*, pages 290–306. Springer.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sivaramakrishnan, R., Antani, S., and Jaeger, S. (2017). Visualizing deep learning activations for improved malaria cell classification. In *Medical informatics and healthcare*, pages 40–47.

Szegedy, C., Reed, S. E., Erhan, D., and Anguelov, D. (2014). Scalable, high-quality object detection. *CoRR*, abs/1412.1441.

Tangpukdee, N., Duangdee, C., Wilairatana, P., and Krudsood, S. (2009). Malaria diagnosis: a brief review. *The Korean journal of parasitology*, 47(2):93–102.

Tek, F. B., Dempster, A. G., and Kale, I. (2006). Malaria parasite detection in peripheral blood images. In *BMVC*, pages 347–356.

Tek, F. B., Dempster, A. G., and Kale, I. (2009). Computer vision for microscopy diagnosis of malaria. *Malaria Journal*, 8(1):153.

Tek, F. B., Dempster, A. G., and Kale, I. (2010). Parasite detection and identification for automated thin blood film malaria diagnosis. *Computer vision and image understanding*, 114(1):21–32.

Thellier, M., Datry, A., Cisse, O. A., San, C., Biligui, S., Silvie, O., and Danis, M. (2002). Diagnosis of malaria using thick bloodsmears: definition and evaluation of a faster protocol with improved readability. *Annals of Tropical Medicine & Parasitology*, 96(2):115–124.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

WHO (2008). World malaria report. page 10.

WHO (2010a). *Basic Malaria Microscopy: Learner's guide*. World Health Organization.

WHO (2010b). *Basic Malaria Microscopy: Tutor's guide*. World Health Organization and Center for Disease Control.

WHO (2016). *Malaria microscopy quality assurance manual-version 2*. World Health Organization.

Wilson, M. L. (2012). Malaria rapid diagnostic tests. *Clinical infectious diseases*, 54(11):1637–1641.

Xia, F. and Li, H. (2018). Fast detection of airports on remote sensing images with single shot multibox detector. In *Journal of Physics: Conference Series*, volume 960, page 012024. IOP Publishing.

Xie, X., Han, X., Liao, Q., and Shi, G. (2017a). Visualization and pruning of ssd with the base network vgg16. In *Proceedings of the 2017 International Conference on Deep Learning Technologies*, pages 90–94. ACM.

Xie, X., Xu, X., Ma, L., Shi, G., and Chen, P. (2017b). On the study of predictors in single shot multibox detector. In *Proceedings of the International Conference on Video and Image Processing*, pages 186–191. ACM.

Yi, J., Wu, P., Hoeppner, D. J., and Metaxas, D. (2017). Fast neural cell detection using light-weight ssd neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–112.

Yitbarek, T., Nega, D., Tasew, G., Taye, B., and Desta, K. (2016). Performance evaluation of malaria microscopists at defense health facilities in addis ababa and its surrounding areas, ethiopia. *PloS one*, 11(11):e0166170.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *CoRR*, abs/1411.1792.

Yunda, L., Ramirez, A. A., and Millán, J. (2012). Automated image analysis method for p-vivax malaria parasite detection in thick film blood images. *Sistemas & Telemática*, 10(20):9–25.

Zheng, L., Fu, C., and Zhao, Y. (2018). Extend the shallow part of single shot multibox detector via convolutional neural network. *arXiv preprint arXiv:1801.05918*.

Zou, K. H., Liu, A., Bandos, A. I., Ohno-Machado, L., and Rockette, H. E. (2011). *Statistical evaluation of diagnostic performance: topics in ROC analysis*. CRC Press.