Automatic Detection of Rouleau Formations Using Image Processing and Deep Learning

Josefin Johansson & Oscar König

Master's thesis
2021:E39

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

Abstract

Analysis of peripheral blood smears is commonly used to aid physicians in diagnosing diseases, checking for infections and evaluating the function of organs. One object of interest in such analysis is the presence of red blood cell aggregates called *rouleaux*. CellaVision's product currently automates the traditionally resource-intensive process of blood smear analysis, which leads to improved patient value. As of now, it lacks the function of detecting rouleaux formations.

The aim of this thesis project is to investigate the addition of this feature to CellaVision's product. For this aim, we have implemented and compared two object detection architectures, employing image processing and machine learning methods. One of the architectures, called *Single Shot MultiBox Detector*, has previously been used by CellaVision for similar tasks. The other was tailor-made for the task of detecting rouleaux, based on the method of region proposals employed in various other object detection architectures. We have also collected and annotated the data necessary to train and evaluate models using these architectures. Said models have been tested and evaluated on expert-annotated data for different sizes of rouleau formations.

We found that the implemented models are capable of detecting rouleaux in blood smear images collected by CellaVision's systems. However, results are not robust enough for use in production, and the definitive performance of the models requires further investigation by experts. Future work is needed to ensure accurate model evaluation and sufficient performance for use in CellaVision's product.

Acknowledgements

We would like to thank CellaVision for giving us the opportunity to do this project. Even though our time spent at the office in Lund was limited and most of the work was done remotely, we felt very welcomed and included in the team. A special thanks to Kent Stråhlén and Ida Wagnström, our supervisors at the company, for their support throughout the project. We also wish to thank Annelie McCorkindale for sharing her knowledge about rouleaux and answering all our questions, not to mention for providing us with annotated data to test our models on. It has been invaluable for our work. Lastly, we would like to thank our supervisor at the Department of Mathematics at LTH, Niels Christian Overgaard, for his guidance.

Contents

1 Introduction **6**
- 1.1 Motivation . . . 6
- 1.2 Aim . . . 7

2 Theory **7**
- 2.1 Rouleaux . . . 7
  - 2.1.1 Blood Smear Analysis . . . 7
  - 2.1.2 Morphological Characteristics of Rouleaux . . . 7
- 2.2 CellaVision's Systems for Collecting Data . . . 8
- 2.3 Artificial Neural Networks . . . 9
  - 2.3.1 Convolutional Neural Networks . . . 10
  - 2.3.2 Activation Functions . . . 10
  - 2.3.3 Max Pooling . . . 11
  - 2.3.4 Batch Normalisation . . . 11
  - 2.3.5 Early Stopping . . . 11
- 2.4 Object Detection . . . 11
  - 2.4.1 Intersection Over Union . . . 12
  - 2.4.2 Non-Maximum Suppression . . . 12
  - 2.4.3 Evaluation Metrics . . . 12

3 Method **13**
- 3.1 Data Collection . . . 13
  - 3.1.1 Splitting the Test Dataset . . . 14
  - 3.1.2 Challenges . . . 14
  - 3.1.3 Annotation . . . 15
- 3.2 Region Proposal Model . . . 15
  - 3.2.1 Overview . . . 15
  - 3.2.2 Model Stages . . . 16
  - 3.2.3 Full Model Pipeline . . . 22
  - 3.2.4 Configurations . . . 23
  - 3.2.5 Annotation Tool . . . 23
- 3.3 SSD: Single Shot MultiBox Detector . . . 25
  - 3.3.1 Architecture . . . 25
  - 3.3.2 Default Boxes and Matching Strategy . . . 26
  - 3.3.3 Loss Function . . . 27
  - 3.3.4 Prediction Processing . . . 27
  - 3.3.5 Our Model . . . 28
  - 3.3.6 Configurations . . . 30
- 3.4 Evaluation . . . 31
  - 3.4.1 Ground Truth Matching . . . 31

4 Results **32**

5 Discussion **33**

5.1 Evaluation Metrics and Aim 33
5.1.1 Precision and Recall In Relation To the Task 33
5.1.2 Evaluation Challenges 34
5.1.3 Evaluating Results 36
5.2 Training Data 37
5.3 RP Architecture 37
5.4 SSD 38
5.5 Conclusions 38
5.6 Future Work 39

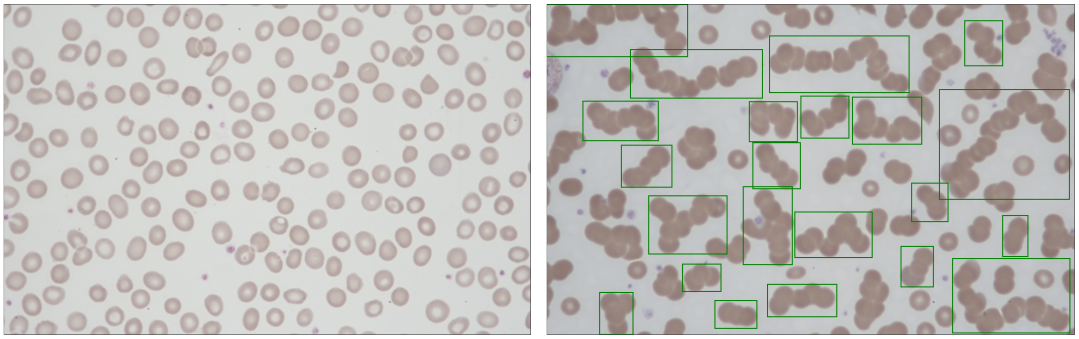
1 Introduction

1.1 Motivation

Blood tests are commonly used to help physicians diagnose diseases as well as check for infections and the function of organs. One type of blood test is called a *peripheral blood smear*, and involves analysing a thin layer of blood under a microscope. Peripheral blood smears can be used to look for abnormalities in the three main types of blood cells, namely, red blood cells (RBCs), white blood cells (WBCs) and platelets (PLTs) [1]. Traditionally, hematology labs use manual microscopy to analyse a smear, which is both time-consuming and laborious. CellaVision's technology automates this process, thus increasing the efficiency while also producing more consistent results, which leads to improved patient value [2].

Normally, RBCs in blood smears are detached from each other, as shown in Figure 1a. Nevertheless, they can form aggregates of different shapes and sizes. One such formation is the so-called rouleau (plural: rouleaux), seen in Figure 1b, which will be the focus of this project. Rouleaux are long cylindrical objects that resemble stacks of coins. The presence of rouleaux in the blood is of interest since there is a connection between high levels of rouleaux and several diseases, such as some types of cancer and diabetes, as well as many infections [3].

Currently, CellaVision's systems have the ability to analyse detached RBCs and classify them with regards to for example colour and deformations, but they are not capable of detecting rouleaux.



(a) Normal (b) Rouleaux

Figure 1: (a) shows RBCs from a healthy patient, where RBCs are detached. (b) shows a blood smear with the presence of rouleaux, where rouleaux are marked with green bounding boxes.

1.2 Aim

The aim of this thesis project is to create models capable of detecting rouleaux in blood smear images taken by CellaVision systems, so that a user may be alerted of their presence for further investigation by a hematologist. To accomplish this, we use machine learning and image processing methods to localise and classify rouleaux aggregates in individual images. Such a model can be fed a sample of cell images, wherein the count of individual rouleau detections can be used to signal their presence. Our aim is to create models that enable this.

To this aim, we have implemented a custom made object detection model architecture for the task, and compared it to an object detection model architecture previously used by CellaVision for other object detection tasks. By doing this, we investigate what types of architectures yield good performance, and whether a custom made architecture is warranted.

2 Theory

2.1 Rouleaux

To detect rouleaux, we must first understand the properties of the blood smears wherein they are found, as well as the characteristics of rouleaux themselves.

2.1.1 Blood Smear Analysis

In hematology, blood smear analysis constitutes an important part. The object of this analysis is a blood smear slide, here referred to as *slide*. A slide is a piece of glass with a drop of blood smeared on it in a thin layer, illustrated in Figure 2. Staining is applied to the blood smear to make the blood cells more visible for observation through a microscope. The slide consists of three areas: The *feathered edge*, the *monolayer* and the *body* [4], shown examples of in Figure 3. RBCs in the feathered edge, corresponding to the lightest colored area of the slide in Figure 2, are completely flattened while the body, corresponding to the darker area, is thick with many overlapping cells. The monolayer is located in between the feathered edge and the body and is characterised by mainly separated but closely packed RBCs. Although all areas can be relevant to examine, the presence of rouleaux is only of clinical interest in the monolayer, see Section 2.1.2.

2.1.2 Morphological Characteristics of Rouleaux

The RBCs in a rouleau are adhered face to face, likely due to the proteins fibrinogen and globulins [3]. Only two RBCs are required to adhere together to

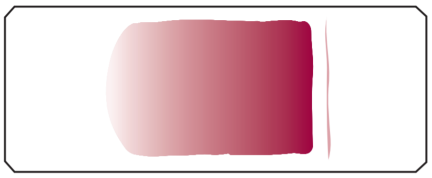
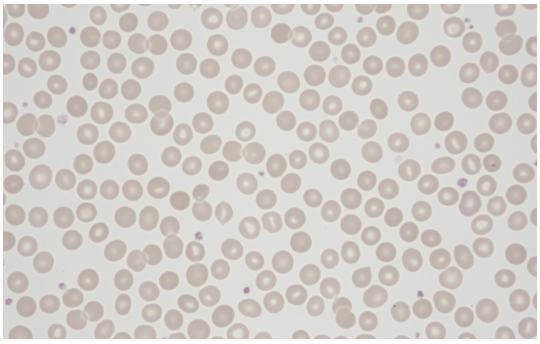
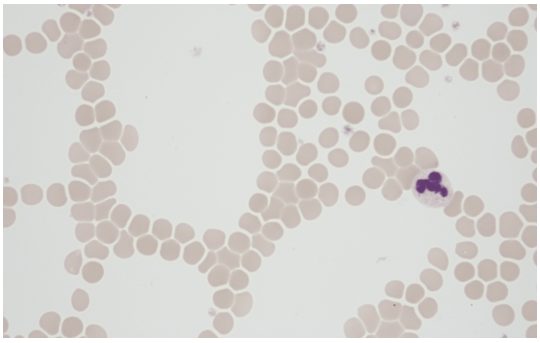


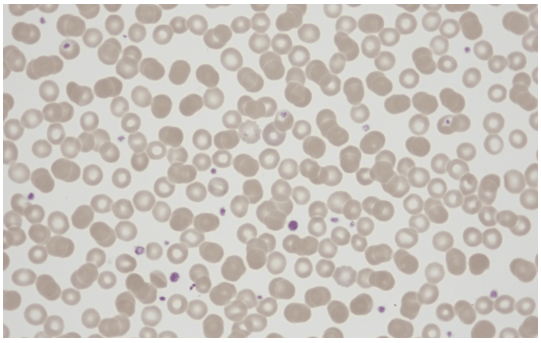
Figure 2: Illustration of a slide. Image from [5].



(a) Monolayer



(b) Feathered edge



(c) Body

Figure 3: Appearance of different parts of a blood smear. The monolayer (a) is located in between the feathered edge (b) and the body (c), which correspond to the lightest and darkest coloured areas of the slide in Figure 2, respectively.

be considered a rouleau but during this project we have assumed the minimum number to be three. Furthermore, large rouleaux are commonly branched. RBC aggregates often form in the body of the slide during the preparation process. However, we are only interested in naturally formed rouleaux, which are found in the monolayer, so analysis is restricted to that particular area.

2.2 CellaVision's Systems for Collecting Data

For model evaluation and training, as well as general analysis, cell images have to be collected, which is done using CellaVision systems. CellaVision provides multiple systems for automatic analysis of blood cells. These systems are equipped

with a digital camera microscope that captures digital images of cells from slides. There is also supporting image analysis and machine learning software. Data for this project was collected using one of CellaVision's DC-1 systems. When a slide is loaded into the DC-1, it automatically identifies the monolayer and finds focus. It then takes photographs of the monolayer with 100x magnification and a resolution of 1920×1200 pixels in BMP-format.

2.3 Artificial Neural Networks

State of the art image analysis uses machine learning methods to process image data, especially neural networks.

Deep learning is a machine learning technique based on artificial neural networks (ANNs). ANNs are so called because they are inspired by biological neural networks in the human brain. They consist of connected (artificial) neurons, typically organised in layers, with a weight corresponding to each connection. Figure 4 shows an example of a simple neural network. A neuron takes the weighted sum of its inputs as well as an additional bias term and creates the output by passing the weighted sum through an activation function. Mathematically, the output from a neuron, $\hat{y}_i$, with m inputs, $\{x_1, \ldots, x_m\}$, can be formulated as

$$\hat{y}_i = \varphi \left(\sum_{j=1}^{m} w_{ij} x_j + b_i \right),$$

where w_{ij} is the weight for input x_j, b_i is the bias and φ is the activation function. When training an ANN, the goal is to update the weights such that the loss between the output of the model and the ground truth $\boldsymbol{y}$ is minimised according to some loss function $\mathcal{L}$:

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{y}, \varphi(\boldsymbol{x}, \boldsymbol{w})).$$

This is done using backpropagation and an optimisation algorithm, for example stochastic gradient descent or Adam.

The most basic type of ANNs are fully connected feed-forward networks. In feed-forward networks, the information flows from the input layer, through the hidden layers to the output layer without any outputs being fed back into the network. As in the case of deep neural networks, there are often multiple hidden layers in between the input and output layer but it can be sufficient with only one layer or even none, depending on the task. A network that is fully connected has all neurons in two adjacent layers connected to each other.

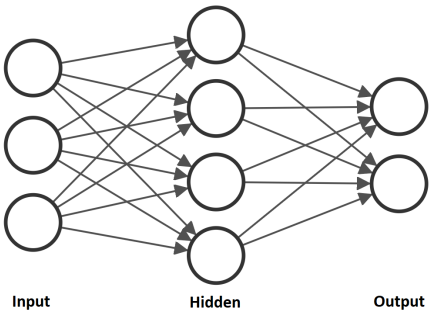


Figure 4: An example of a simple fully connected feed-forward neural network.

2.3.1 Convolutional Neural Networks

Fully connected networks can lead to a large amount of parameters to learn for data such as images. To reduce the number of parameters, it can be effective to let some connections share weights. A practical way of implementing weight sharing for images is by using the convolution operation. Convolutional neural networks (CNNs), a type of feed-forward network, employs this operation in at least one of the layers. Convolution is performed by letting a kernel of a predetermined size containing learnable parameters traverse the image. At each location, the kernel is multiplied elementwise with the pixel values of the image with the same dimensions as the kernel. The products are then summed up, finally resulting in a feature map.

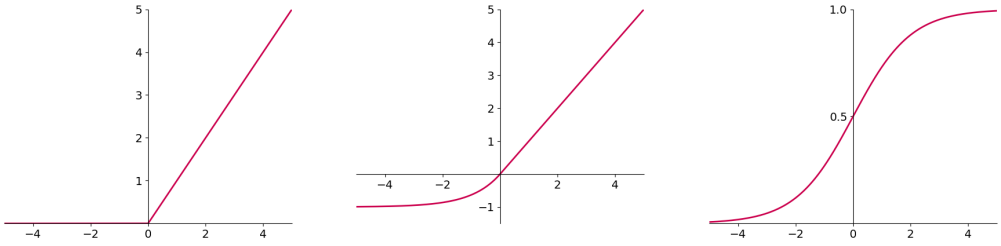
2.3.2 Activation Functions

Activation functions introduce non-linearity into neural networks by transforming the node inputs, thus making the network capable of performing more complex tasks. A common activation function is the ReLU (Rectified Linear Unit), defined by (1). ELU (Exponential Linear Unit), defined by (2), is an activation function similar to ReLU but can produce negative outputs. It also includes a hyper-parameter $a \geq 0$. The sigmoid function, defined by (3), is another popular activation function. Plots of the functions are shown in Figure 5.

$$\text{ReLU}(x) = \max(0, x) \tag{1}$$

$$\text{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ a(e^x - 1) & \text{otherwise} \end{cases} \tag{2}$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{3}$$



(a) ReLU (b) ELU with $a = 1$ (c) Sigmoid

Figure 5: Three common activation functions.

2.3.3 Max Pooling

Pooling is commonly applied after a convolutional layer and activation to reduce the dimensionality but still keep the most important information. Max pooling is one type of pooling operation, which involves sliding a kernel over a feature map and selecting the maximum value from the region covered by the kernel.

2.3.4 Batch Normalisation

Batch normalisation is a method used in ANNs to make them faster and more stable [6]. This is accomplished by normalising the input to a layer to have zero mean and a variance of 1 for each batch.

2.3.5 Early Stopping

Early Stopping is a technique used when training a network to prevent overfitting. After each epoch, the model is evaluated on the validation set, and if the validation performance starts to decline, training is stopped, since this indicates overfitting. Naturally, there is variance in the validation performance, thus one might want to see the validation performance decrease for several epochs before stopping. This number of epochs is called the *patience*. Often, the validation performance after each epoch is compared to the lowest validation loss achieved so far.

2.4 Object Detection

The task of detecting rouleaux in cell images is a type of *object detection* task. Object detection combines image classification and object localisation. Image classification is the task of assigning a label to an image while object localisation involves identifying objects and drawing bounding boxes around them. Accordingly, the output of an object detection model is bounding box coordinates for the detected objects and confidence scores for each detected object that indicate

the certainty of a correct detection. Moreover, object detection models can be divided into two main classes: One-stage methods and two-stage methods. Two-stage methods first propose regions with possible objects and then classify these proposals. One-stage methods perform both steps simultaneously. Generally, two-stage methods have a higher detection accuracy than one-stage methods but the detection speed is slower [7].

2.4.1 Intersection Over Union

Intersection over Union (IoU) measures the similarity between two bounding boxes. It can be employed as a part of an object detection model and for model evaluation. IoU is defined as the area of the intersection divided by the area of the union of the two boxes, illustrated in Figure 6.

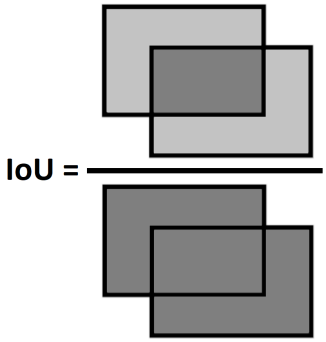


Figure 6: Intersection over Union is the area of intersection divided by the area of the union.

2.4.2 Non-Maximum Suppression

Many object detection models generate a large number of bounding boxes. Ideally, only one bounding box is predicted for each object in an image. To reduce the number of boxes and particularly to avoid multiple predictions for the same object, a technique called non-maximum suppression (NMS) is used. Starting with the box with the highest confidence score, it is kept as a final prediction and the IoU with the remaining boxes is calculated. If the IoU is above a set threshold, the boxes are regarded as too similar and the lower scoring box is removed. This is then repeated for the next highest scoring box and so on, until there are no boxes with too much overlap left.

2.4.3 Evaluation Metrics

In binary classification problems, training examples are labelled with one of two classes. One of these classes can be said to be the *positive* class, while the other is said to be *negative*. When predicting the label of an example, the prediction is either correct or incorrect, which is denoted *True* or *False*, respectively. Thus, each training example falls into one of four categories: True Positive (TP), False

Positive (FP), True Negative (TN) or False Negative (FN). These categories are illustrated in Table 1.

Table 1: Predictions fall into one of four classes depending on the predicted label (positive or negative) and the correctness of the label (true or false).

	Positive	Negative
Correct	TP	TN
Incorrect	FP	FN

These quantities can be used to define metrics that are used to evaluate the model, namely *precision* and *recall*. Precision is a measure of how accurate positive predictions are. It is defined as the ratio between the number of True Positives and the number of total positive predictions, which is equal to the sum of True Positives and False Positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

Recall is a measure of how well the model identifies positively labelled examples. It is defined as the ratio between the number of True Positives and the number of positively labelled examples, which is equal to the sum of True Positives and False Negatives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

The *F1-score* is another common measure that indicates the balance between precision and recall. It is defined as the harmonic mean of the precision and recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Perfect precision and recall results in the highest possible score of 1 while precision or recall of 0 results in the lowest score of 0.

3 Method

3.1 Data Collection

Data was collected from twenty slides marked for containing rouleaux. Thirteen of these twenty slides were ultimately used. The remaining seven were discarded because of poor general image quality. The whole monolayer of these slides were

captured, resulting in around four to eight thousand images per slide. From these images, two datasets were created, *TrainVal* for training and validation, and *Test* for testing. The sets differ in size, and most importantly, in method of annotation - the first was annotated by us while the other was annotated by an expert.

For the TrainVal dataset, 250 images were randomly sampled from each slide for a total of 3250 images. Out of these, 1220 images were ultimately annotated by us using an annotation tool, with the rest being discarded. This process is detailed below in Section 3.2.5.

For the Test dataset, 30 images (not included in the *TrainVal* set) were randomly sampled from each slide for a total of 390 images. Out of these, 256 images were ultimately annotated, with the rest being discarded for poor quality, by an expert. The expert was asked to discard images unfit for analysis, and to annotate rouleaux in three different size categories corresponding to the number of RBCs in them, *three or below*, *four to six* and *seven or more*.

3.1.1 Splitting the Test Dataset

In the test data, rouleaux are annotated by size into three categories. Thus, the model performances can be evaluated on different subsets of rouleaux. The test data was split into three subsets, illustrated in Table 2. The first one, *Test 1*, contains all annotations, the second, *Test 2*, removes the smallest group (0-3) and the third, *Test 3* removes the second smallest group (4-6), leaving only the largest rouleaux.

Table 2: Table showing which size categories of rouleaux are included in which test data subsets.

	0-3	4-6	7+
Test 1	■	■	■
Test 2		■	■
Test 3			■

3.1.2 Challenges

The test data was annotated by us with guidance from a hematology expert. Analysing blood smears requires experience and there are challenges associated with identifying and annotating rouleaux even for experts.

While the theoretical definition of rouleaux is quite exact, there are no simple visual indicators which alone determine whether a group of RBCs constitute a rouleau on a case to case basis, and non-aggregated groups of RBCs often exhibit some characteristics. These characteristics also vary between slides, which is

exacerbated by the fact that the presence of rouleaux complicates the mechanics of preparing a blood smear, since the blood is not acting normally.

When identifying rouleaux, the expert puts great emphasis on the surrounding regions, since rouleaux seldom appear alone. While some cases are more characteristic than others, identifying a rouleau typically involves studying the whole image, looking for other candidates, and weighing in this information in the assessment. Especially for smaller rouleaux, these visual indicators are less clear, and our practical definitions become "fuzzier". According to an expert at CellaVision, rouleaux contain a minimum of four RBCs, but if there is an abundance of rouleaux nearby, this limit might be relaxed. The expert also takes external factors from other tests into account to determine if the patient is a "rouleaux patient".

These factors pose problems, both technical and regarding our ability to annotate accurately. The lack of simple visual indicators caused a fundamental difficulty in recognising rouleaux, resulting in faulty annotations in the training data. The context-dependency is problematic since our models use local information when learning their appearance; neither of our models are designed to account for the presence of rouleaux in remote parts of the image (or even not so remote). This means that an identical cell cluster might be labelled either positively or negatively depending on its surroundings, which can cause conflicting examples of the appearance of rouleaux used for training models.

3.1.3 Annotation

Because of greatly varying size and shape, we found training to be data intensive, and for the reasons detailed above, annotation is challenging. To facilitate the annotation of data, an annotation tool was created based using parts of the custom made architecture detailed in the next section. While the annotation tool is part of the data collection process, we choose to present it in Section 3.2, as it uses part of the model architecture discussed in this section.

3.2 Region Proposal Model

3.2.1 Overview

The custom-made architecture, henceforth *Region Proposal architecture* or *RP architecture*, is a four stage model which uses a *region proposal* approach similar to that of R-CNN [8]. Regions of Interest (ROI) are detected by a region proposal algorithm, which are then classified by a neural network. For this task, ROI are candidate regions which might contain rouleaux, and the classification means determining their presence.

Region proposals for general objects is a difficult problem because of the lack of common features for "general objects", however, the nature of our task avoids many difficulties associated with general region proposals. For one, the set of objects that might appear in our blood smear images is limited. Furthermore, in the monolayer, blood cells (if not part of aggregates such as rouleaux) are supposedly well-separated with a plain background [4], which facilitates segmentation. CellaVision's product performs RBC segmentation for unrelated goals, and thus segmentation data for collected blood smear images is already available.

This segmentation data can be used for RBC localisation, and by extension to detect overlapping RBCs, or *clusters* of RBCs. Since rouleaux consist of overlapping RBCs, these regions are candidates for containing rouleaux. Detecting these clusters and using them for rouleau candidate localisation is the basis for the RP architecture. From these ROIs, representations of the shapes of the clusters, *shape segments*, are extracted and classified by a neural network classifier. The full model architecture is illustrated in Figure 7.

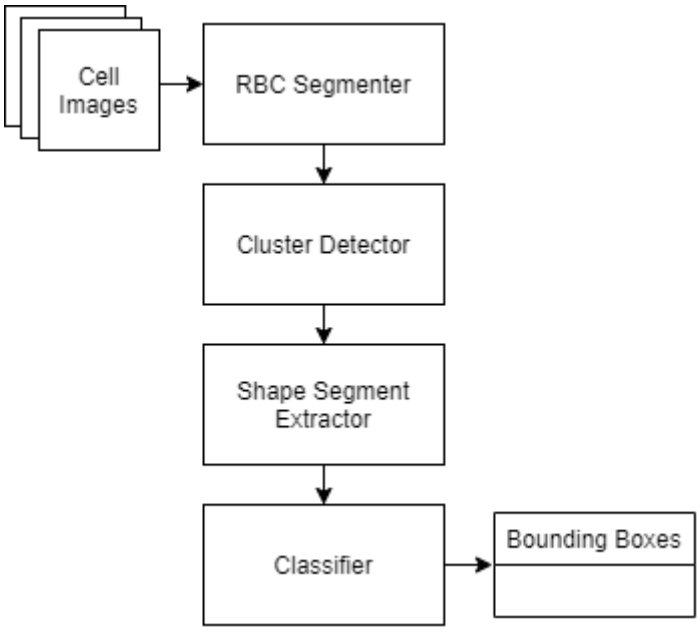


Figure 7: Model block diagram. The model takes cell images and processes them in four stages. The output is a list of bounding boxes for each image containing the position of detected rouleaux.

3.2.2 Model Stages

Segmentation Stage. In the first stage of the model, blood smear images are segmented by CellaVision's RBC Segmenter. When fed with a blood smear image, the RBC Segmenter produces a list of detected objects with localisation data (bounding boxes) and object classes. Such segmentation data is plotted in Figure 8. While the segmenter is explicitly made for RBC segmentation, it also has the ability to identify other types of objects such as WBCs. Not all

segmentation classes are of interest for cluster detection. For obvious reasons not classes corresponding to non-RBC objects, but there are also classes which denote standalone RBCs, which are by definition not part of a cluster. Thus, the segmentation data is filtered from undesired segmentation classes before it is used for cluster detection.

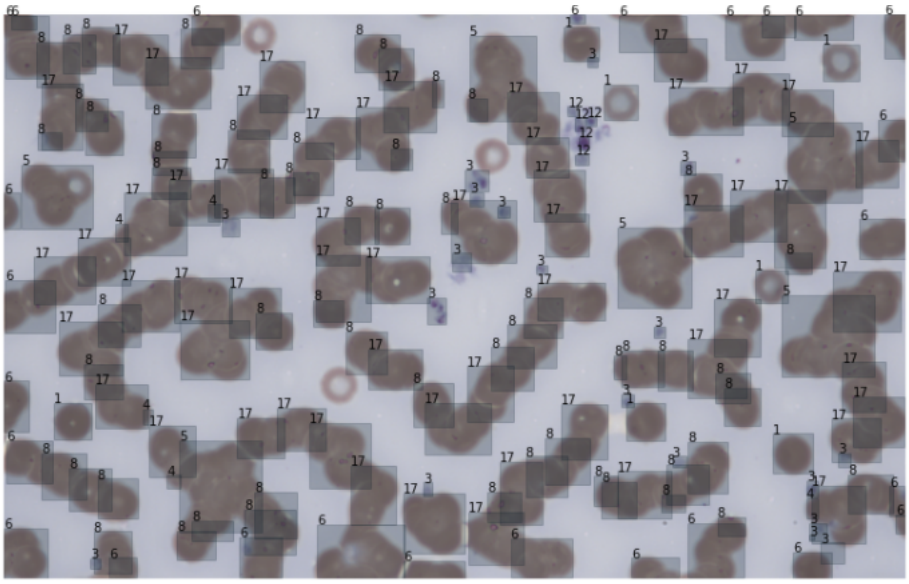


Figure 8: RBC Segmentation data visualised with all classes included. The number in each box corner indicates the class. Some RBCs have been missed in the segmentation.

Cluster Detection Stage. The second stage of the model is the region proposal stage. The candidate regions for rouleaux contain overlapping cells, such as in Figure 9, and are thus localised by finding clusters of RBCs. This stage of the model is called the *Cluster Detector*, and it outputs a list of groups of clustered Segmentation Objects for each image.

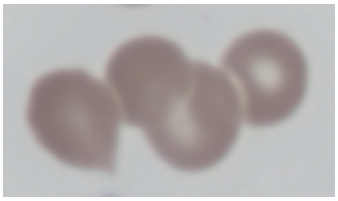


Figure 9: Four overlapping RBCs which form a cluster.

We define a cluster as the set of all by overlap connected Segmentation Objects starting from some Segmentation Object. To do this, we must define what constitutes overlap. The naive way to define overlap between two Segmentation Objects is to check for the intersection of the square bounding boxes defined by the coordinates of their localisation data. However, this approach erroneously

clusters diagonal Segmentation Objects together, as illustrated in Figure 10, since the square bounding boxes poorly match the shapes of the RBCs.

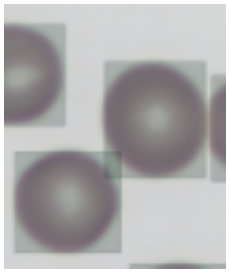


Figure 10: Non-overlapping RBCs with overlapping square bounding boxes.

To perfectly cluster objects together, we would need high resolution segmentation data (masks), however, by fitting ellipses in the bounding boxes, we get a good approximation of the Segmentation Object shapes, as seen in Figure 11. We can then define overlap by the intersection of these ellipses.

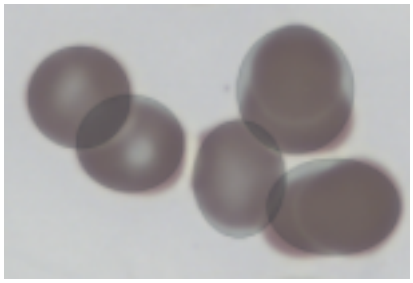


Figure 11: By fitting ellipses to square Segmentation Object bounding boxes, we get a good approximation of the Segmentation Object shapes. This allows for better clustering by checking for ellipse intersection.

We can adjust how much overlap is required for two RBCs to be clustered together by shrinking or expanding these ellipses before checking for intersection, as illustrated in Figure 12. By shrinking ellipses before checking for intersection, we require more overlap, since lightly overlapping ellipses will no longer intersect after shrinking. Conversely, expanding ellipses allows us to cluster nearby objects together which do not actually overlap. Both axes of the ellipse are expanded or contracted by the same amount of pixels, which is denoted the *clustering sensitivity*.

The problem of finding clusters can be transferred to a graph problem. Each cell image corresponds to an undirected graph, where Segmentation Objects are vertices, and adjacency is defined by overlap. Clusters then correspond to connected components, which are found by using the graph search algorithm *Depth-First Search*.

After clusters have been found, it is natural to discard some clusters based on their size, since rouleaux are by definition aggregates of many RBCs. Before

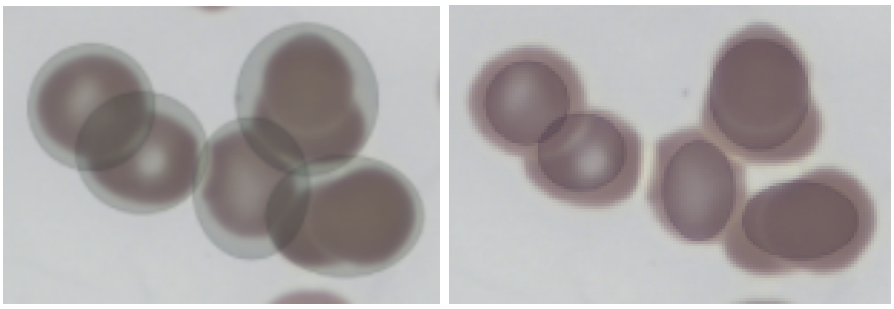


Figure 12: Ellipse fitting of some Segmentation Objects with clustering sensitivities of 10 and -10, respectively.

being sent to the third stage, the clusters are filtered by their number of cells. We define the *cell threshold* parameter as the threshold of contained cells under which clusters are discarded; when the model is used for predictions, they are marked as negatives, and they are not used for training the classification step. This is illustrated in Figure 13, where discarded clusters are plotted in red, using a cell threshold of 3.

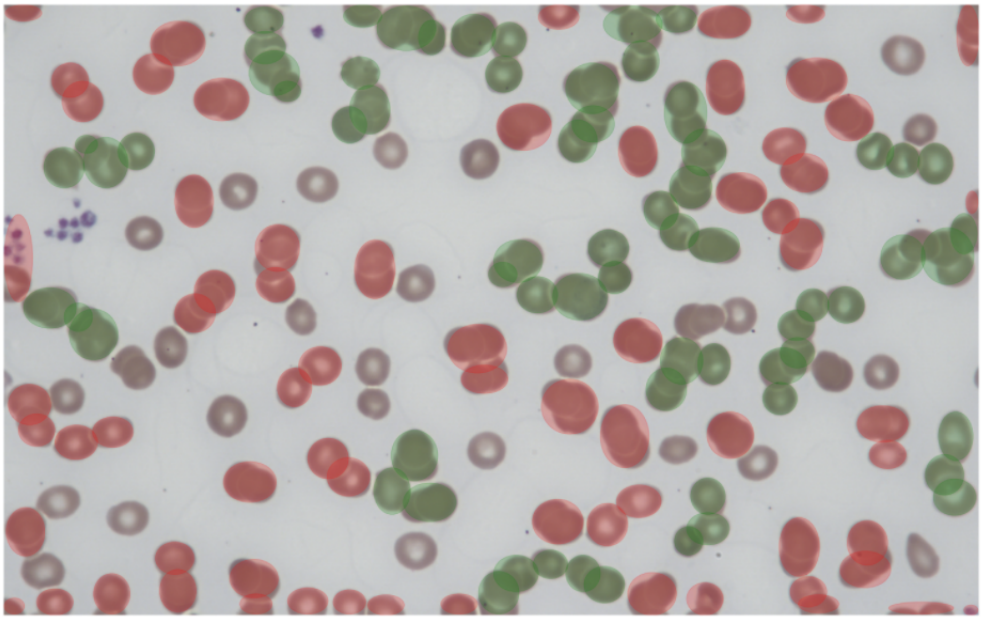


Figure 13: Segmentation Object ellipses of clusters with less than three members are marked red, and with three or more green. The unmarked RBCs have been filtered in stage one, i.e. are not Segmentation Objects.

Shape Segment Extraction Stage. For the third stage, clusters are converted to ROIs which are cropped and processed to extract the cluster *shape segments* for classification. Shape segments are masks of the cluster shape used instead of the bitmap data. A cluster can naturally be converted to a ROI

by selecting the smallest bounding box containing all its Segmentation Object bounding boxes, which is is illustrated in Figure 14.

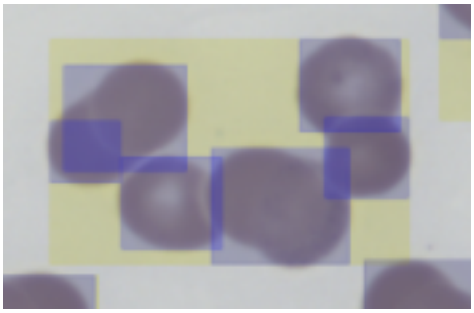


Figure 14: Cluster converted to ROI in yellow and Segmentation Object bounding boxes in blue.

The segmentation is not perfect, and sometimes cells are missed or badly bounded. To help ensure that the whole cluster is included in the shape segment, we first expand ROIs by a 30 pixels on each side, as illustrated in Figure 15. ROIs close to the image edge whose resulting bounding boxes end up outside the image borders are discarded.

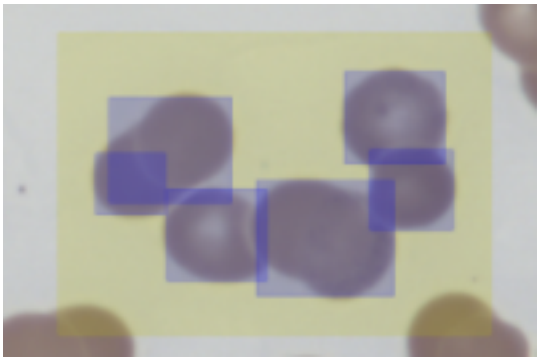


Figure 15: Cluster converted to ROI and expanded with Segmentation Object bounding boxes in blue.

Next, the image data of extracted regions are converted to grey-scale and thresholded to produce masks. The threshold is set $m + \frac{\delta}{2}$, where m is the mean bitmap pixel value and δ is the bitmap pixel value standard deviation inside the region, which was experimentally found to produce good masks. In Figure 16, we see the cluster bitmap data before and after thresholding.

Since the centre of RBCs are generally brighter, this often produces "hollow" masks, where the centres of RBCs are not included, depending on the contrast in the image. These holes are filled to produce a collection simply connected cell masks. In Figure 17, we see the same ROI as in Figure 16 after filling the holes. Any holes with an area below 1500 pixels are filled. This eliminates the influence of varying image contrasts, as well as enabling for the next step.

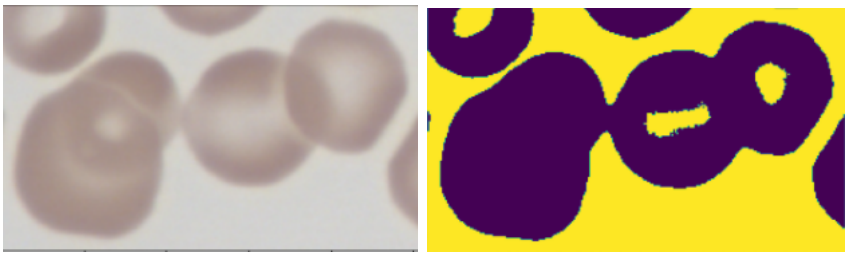


Figure 16: Extracted ROI before and after thresholding.

Figure 17: Same ROI as in Figure 16 after filling holes.

The resulting masks also contain segments of surrounding cells which are not part of the cluster, but still contained within the ROI around the edges. To filter these segments out, we mark the midpoints of Segmentation Objects contained in the cluster, and keep all segments connected to any of these points, which is illustrated in Figure 18.

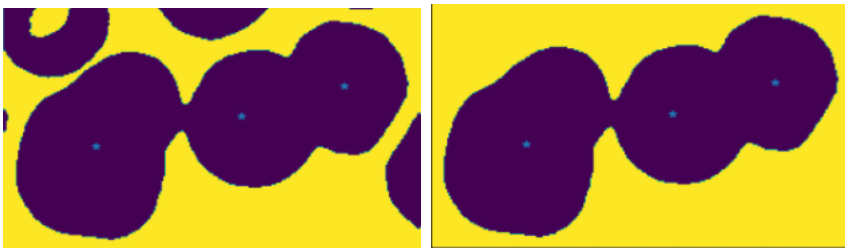


Figure 18: Segments which are connected to any of the cluster Segmentation Object midpoints are kept. Same ROI, first with marked midpoints and then with only matching segments kept.

(If the shape segments are extracted for training, they are randomly flipped and rotated at this point.)

These masks can be used to measure the area of the corresponding cluster, measured in pixels. This is done by summing the elements of the binary mask. Each marked pixel adds one to the area. This provides another tool to filter which clusters make it through to the classification stage. Before being sent to the final stage, shape segments are thresholded by their mask area, and discarded segments are assigned to the negative class and not used for training the classification stage. We call this parameter the *area threshold*.

Clusters, and thus ROIs, differ wildly in size, so the masks of the resulting shape segments are of different dimensions. To avoid resizing and keep the original scale, the canvas of these segments are then expanded or cropped around its centre to a uniform square size of 448×448, which are ultimately classified by the NN classifier. The reason for this choice of dimension is that it is twice the width and height of the input size of the convolutional neural network used in the final stage, meaning masks can be easily downsampled for the final step.

Classification. The fourth and final stage of the model classifies the shape segments as rouleaux or not using a convolutional neural network. The network consists of a VGG16 base network pre-trained on ImageNet for feature extraction, followed by a 256 node fully connected ReLU classifier layer, and finally a single node output layer with sigmoid activation. The network architecture is illustrated in Figure 19. As a pre-processing step, shape segments are downsampled by a factor 2 to dimension 224×224.

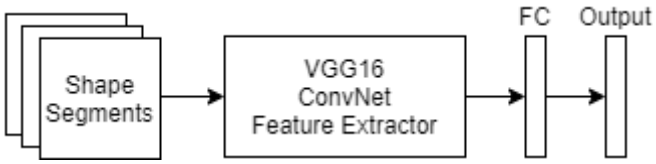


Figure 19: Network architecture. The network consists of a VGG16 convolutional base network followed by a 256 node fully connected layer, and single node output layer.

The network is trained by feeding annotated shape segments in batches of 16, which are filtered by cell and area threshold parameters of previous stages. The creation of training data is detailed next in Section 3.2.5. Only the fully connected layers are trained; the convolutional base network weights are locked. RMSProp is used for optimisation with Binary Cross Entropy loss and early stopping. The early stopping stops the training after the validation loss has not improved from the previous best for five consecutive epochs.

When making predictions, a batch of arbitrary size of shape segment masks can be fed through the network which are then scored by the classifier between 0 and 1. Shape segments with scores over 0.5 are classified as rouleaux.

3.2.3 Full Model Pipeline

When making predictions, the full model first performs the segmentation for all input images. After this, each image is fed individually through the remaining stages, outputting bounding boxes for the positively classified shape segments.

3.2.4 Configurations

Each stage is configured by parameters, meaning different configurations have been created and tested, which are detailed below in Table 3.

Table 3: Parameter values for the tested RP model configurations.

Model #	Cell threshold	Clustering sensitivity	Area threshold
1	3	0	10000
2	3	0	20000
3	3	0	40000

3.2.5 Annotation Tool

When developing the Cluster Detector stage of the RP model, we noted that it performed well on blood smear images containing rouleaux, and that, often, a clear majority of detected clusters in these were seemingly rouleaux. In these cases, the Cluster Detector output could be used as the basis for correct annotation. Based on this observation, an annotation tool was created to facilitate the annotation of data.

The annotation tool is based on the first two stages of the RP architecture, see Section 3.2.2. Segmentation data from the CellaVision RBC Segmenter is loaded into memory. The tool loads and plots corresponding images and feeds these through the Cluster Detector, presenting the user with detected clusters. These are all initially marked as positive or negative, and clusters under a user-specified size are filtered. The user has the option to discard the image for a range of reasons discussed in the next section. If the image is kept, the user navigates the annotations with the arrow keys and corrects mislabelled cases. The user also has the option to merge clusters together, should the Cluster Detector fail to gather the whole cluster. When the image is correctly annotated, the user presses *Done* to save the annotations and load the next image. The annotation tool is shown in Figure 20.

There are obvious flaws with this method of annotation, deriving from the use of the Cluster Detector discussed in Section 3.2.2. The detection is by no means perfect; the segmentation will occasionally miss RBCs, but mainly the ellipse approximation in the cluster algorithm causes the detection to join clusters which are really disconnected or separate a cluster into several parts. The clustering especially fails in very RBC dense images, where RBCs can be "squished together" in such a way that ellipses are not good approximations of their shapes. Additionally, the size filter means smaller rouleaux can be missed by the user accidentally filtering them away. Some examples where the annotation tool has failed are illustrated in Figure 21.

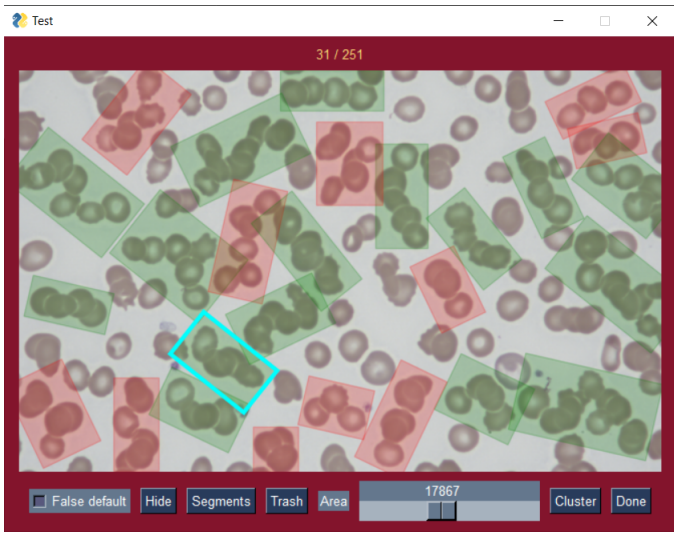


Figure 20: The annotation tool developed for annotation. Clusters are detected and filtered by area, and boxes are plotted around them. To save annotations or discard an image, *Trash* or *Done* is selected, respectively. (Note that the models do not use angled bounding boxes as shown in the image.)

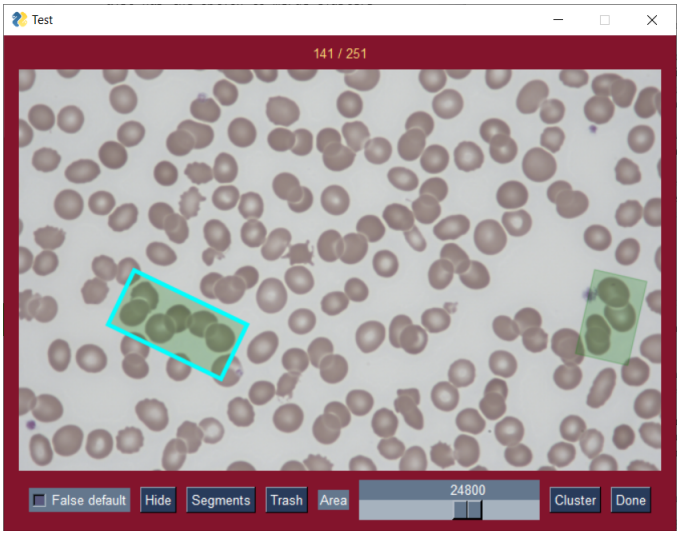


Figure 21: An image showing two cases where the clustering fails. In both cases, the a pair of clearly separated clusters have been detected as one. The clustering fails due to the inaccuracy of the ellipse approximation of segmented RBC objects.

As a consequence, our annotation is largely an approximation. We believe the quality of the data is good enough for evaluating our models, but ultimately, expert annotation could improve results.

With respect to the limitations and uncertainties discussed above, guidelines

for the discarding of images were established in order to maintain annotation quality and consistency. The first criteria for removal is bad image quality, the main reasons being

- Blurry image
- Image region not part of the monolayer
- Images with artefacts such as dirt

The second criteria for removal is bad clustering quality. Bad cluster detection causes RBCs to be missed, or extra RBCs to be included, in the annotation. However, some inaccuracies were tolerated as cluster detection was seldom completely perfect. Cases that were tolerated in small quantities were

- Small rouleau missed
- Additional RBCs in cluster bounding box
- Missed (non majority) part of rouleau
- Two (small) rouleaux in cluster bounding box

3.3 SSD: Single Shot MultiBox Detector

SSD [9] is a one-stage method for object detection. This model architecture was chosen as a contrast to the multi-stage Region Proposal architecture. Moreover, CellaVision has previously used this architecture on other tasks, such as detection of PLTs and PLT clumps.

3.3.1 Architecture

The SSD architecture consists of a base network, a standard image classification network (truncated before any classification layers), and additional convolutional feature layers on top of the base network that produce feature maps for detection predictions. The prediction feature maps are of progressively decreasing resolution to detect objects of multiple scales. Smaller objects can be detected in higher resolution feature maps and larger objects can be detected in lower resolution feature maps. Predictions are made using 3×3 convolutional filters applied to the prediction feature maps and results in confidence scores for the object categories as well as offsets relative to *default box* coordinates. An example of the SSD architecture is shown in Figure 22.

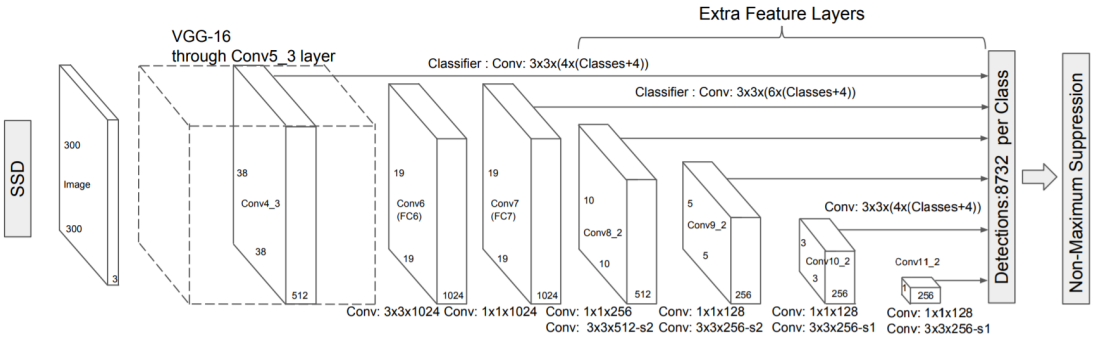
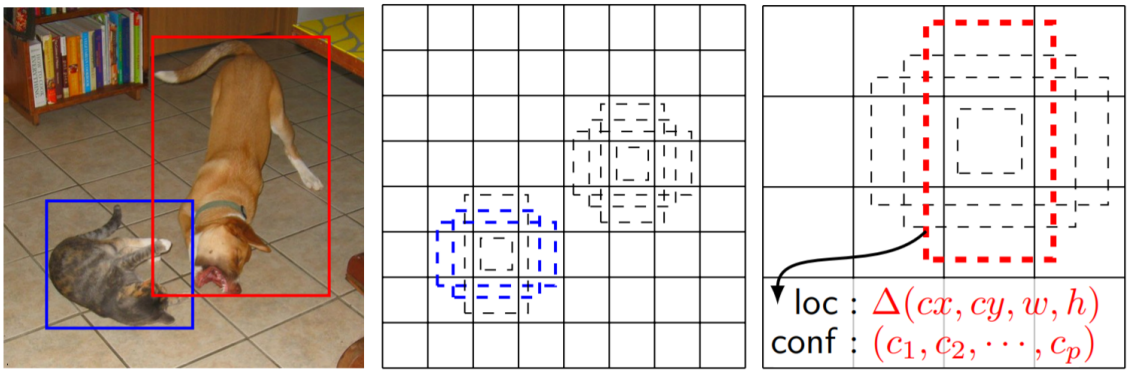


Figure 22: Example of SSD architecture. Image from [9].

3.3.2 Default Boxes and Matching Strategy

Default boxes are bounding boxes placed at every location in the input image corresponding to the locations of the prediction feature maps. They vary in scale and aspect ratio, parameters set before training, to better allow for detection of objects of different shapes and sizes. During training, first, each ground truth box is matched to the default box with the largest IoU. Then, the remaining default boxes are matched with their respective most similar ground truth box. If the IoU is above 0.5, the match is considered *positive*, otherwise it is *negative*. As an example, Figure 23 shows two feature maps of size 8×8 and 4×4, respectively. In the 8×8 feature map, two default boxes are matched with the cat and one default box is matched with the dog in the 4×4 feature map.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 23: Example showing the matching of ground truth boxes and default boxes. Two default boxes are matched with the cat, (b), and one is matched with the dog, (c). The matches are considered as positives and the rest as negatives. Image from [9].

3.3.3 Loss Function

The loss function is a combination of the confidence loss $\mathcal{L}_{\text{conf}}$ and localisation loss $\mathcal{L}_{\text{loc}}$. Let

$$x_{ij}^p = \begin{cases} 1 & \text{if IoU} > 0.5 \text{ between default box } i \text{ and ground truth box } j \text{ of class } p, \\ 0 & \text{otherwise.} \end{cases}$$

The confidence loss is defined as the *Softmax loss* over confidences c:

$$\mathcal{L}_{\text{conf}}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0), \text{ where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)},$$

where N is the number of positive matches and class 0 is the background. Since there are significantly more negatives than positives, which may have an undesirable effect on the learning of the model, the number of negatives used is limited. Choosing the negatives with the highest confidence loss, the ratio between the negatives and positives is at most 3:1. This is called *hard negative mining*.

The localisation loss is defined as a *Smooth L1 loss* between the predicted box l and the ground truth box g. With the offsets for the centre cx, cy of default box d and its width w and height h we have:

$$\mathcal{L}_{\text{loc}}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right),$$

where

$$\text{smooth}_{\text{L1}}(a) = \begin{cases} 0.5a^2 & \text{if } |a| < 1, \\ |a| - 0.5 & \text{otherwise.} \end{cases}$$

The final loss function is a weighted sum with weight factor α:

$$\mathcal{L}(x, c, l, g) = \frac{1}{N}(\mathcal{L}_{\text{conf}}(x, c) + \alpha \mathcal{L}_{\text{loc}}(x, l, g)).$$

3.3.4 Prediction Processing

During inference, the predictions are processed in two steps. First, predicted boxes below a confidence threshold are filtered out. Then, non-maximum suppression is performed. At most, the top 200 predicted boxes are kept per image.

3.3.5 Our Model

Architecture. As base network, ResNet50 v2 [10] is used. On top of the base network, there are three convolution blocks of the form Conv - BatchNorm - ELU - MaxPool and a fourth block with the same structure but without the pooling layer. As an input step, the images are resized to 512×512. The convolution layers use kernels of size 3×3 and 64, 48, 48 and 32 channels, respectively. The max pooling is done with a 2×2 kernel. Predictions are based on the feature maps generated before the pooling layer in each block. The resolution of the prediction feature maps are 16×16, 8×8, 4×4 and 2×2, respectively.

The scale values and aspect ratios chosen for the default boxes should be representative of the sizes and shapes of the objects we want to detect. Moreover, there is one scale value for each prediction layer, in this case a total of four scales, and boxes of all aspect ratios are created for each scale. Additionally, for aspect ratio 1, boxes with scales slightly larger than the scales set for each layer are created. These scale values are computed as the geometric mean of two scales corresponding to two consecutive layers. The last scale, however, is based on the scale for the final layer and a value set only for this computation. Thus, one more scale value than the number of prediction layers must be determined.

The aspect ratios were set based on observations of the data. To determine the scales, two different methods was used. One method is based on experiments. The other method involved statistical analysis of the sizes of the bounding boxes in the test data. The scales of the test data boxes were approximated by the square root of their area relative to the image size. Given the distribution of the scales, the 0.05, 0.35, 0.65 and 0.95 quantiles were chosen as the scales for the prediction layers and the fifth scale was arbitrarily set to 1.0. Because of the challenges with the annotation of the TrainVal data previously outlined, it is reasonable to believe that the bounding boxes in the test data more accurately describe the true sizes of rouleaux. Moreover, assuming that the sizes of rouleaux in the test data are representative of their sizes in general and given that they are only used as guidelines, there is little risk of overfitting the test data.

Training. For the training, the data was randomly split into 70% train and 30% validation sets. Image augmentation was applied to the training data to make the model more robust and expand the training set. The specific techniques used were horizontal and vertical flip, rotation by 90, 180 and 270 degrees and colour adjustment. Colour augmentation was used to account for differences in staining between slides. Moreover, transfer learning and fine-tuning were applied. First, weights pre-trained on the ImageNet dataset were loaded into the base network and all base layers were frozen so that only the added top layers were trained. Then, the whole network was trained with a lower learning rate. The learning rate was initiated to 0.001 and reduced if the validation loss stopped decreasing.

Early stopping with patience 10 was used to end the training and Adam was used as optimisation method. The weight factor in the loss function was set to 0.1 and a minimum number of negative boxes per batch used to calculate the loss was set to 5 for a batch size of 4.

Training Data Subsetting. Test data was split into sets of different sizes and different SSD variants were created which specialise in detection of these sizes. The only mechanism for differentiating between different rouleaux sizes for SSD is to subset the training data. This was done by thresholding using the cell counts of annotations provided by the segmentation data in the annotation tool. Compare this to the RP architecture, where candidates are discarded during testing at earlier stages independent of the training of the classification stage.[1]

Each variant uses annotations with a minimum cell count of three, four and seven, mirroring the cell counts of the test sets. Annotations with lower cell counts are discarded, i.e. marked as negative.

Data Processing. The SSD architecture used is designed for input images of size 512×512. Consequently, non-square images are "squished". If the 1920×1200 images in our data (with significantly non-square aspect ratio) were to be used directly, the characteristics of rouleaux in the resized images would differ depending on their orientation in the image. Thus, the full 1920×1200 images are converted into two square images. The bottom 240 pixels are cropped resulting in a 2:1 aspect ratio. Then, the image is split in the middle along the vertical axis, resulting in two square images. This is illustrated in Figure 24.

This results in annotated bounding boxes partially or completely outside the image bounds. Any bounding boxes whose inner edges are less than 100 pixels from the cropped edge are removed. (Kept boxes are cropped so that they are fully inside image bounds.) This processing is performed for training and evaluation.

Prediction Thresholds. In the final step of the SSD model, as described in Section 3.3.4, boxes are filtered out based on their confidence score after which NMS is applied. For this, a confidence threshold as well as an IoU threshold for the NMS must be determined. Based on the matching described in Section 3.4.1, the F1-score of the models on the validation data was computed with varying confidence thresholds and IoU thresholds to find optimal values. The IoU values tested were between 0 and 0.5 with 0.05 intervals and the confidence thresholds

[1]Note that the training data of the RP model is subset by the same criteria used during testing.

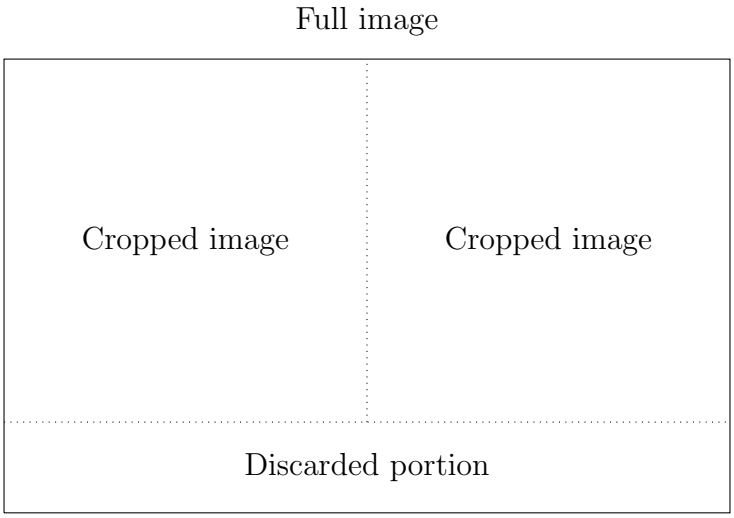


Figure 24: Illustration of the cropping and splitting used to produce square images for SSD. The bottom portion is cropped away and discarded, leaving a 2:1 aspect ratio image, which is split in two to produce square images.

were between 0 and 0.95. A heat map visualising the evaluation of Model 1, specified below in Section 3.3.6, is shown in Figure 25.

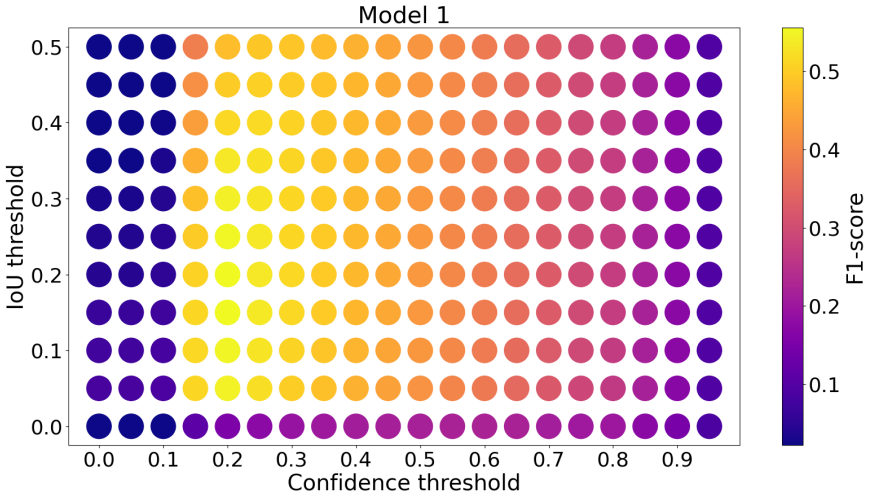


Figure 25: Heat map showing F1-scores for a grid of IoU and confidence threshold values.

3.3.6 Configurations

Tables 4 and 5 present the model configurations evaluated. The models in Table 4 use default box aspect ratios {0.5, 2/3, 1.0, 3/2, 2.0} and the scales are determined based on statistical analysis. In Table 5, the aspect ratios are {0.5, 1.0, 2.0} and the scales are experimentally determined.

Table 4: Models using aspect ratios {0.5, 2/3, 1.0, 3/2, 2.0} and statistically determined scales.

Model #	Cell count	Scales	Conf. thresh.	IoU thresh.
1	3+	{0.12, 0.17, 0.22, 0.39, 1.0}	0.20	0.15
2	4+	{0.14, 0.19, 0.24, 0.41, 1.0}	0.20	0.25
3	7+	{0.21, 0.27, 0.33, 0.51, 1.0}	0.15	0.10

Table 5: Models using aspect ratios: {0.5, 1.0, 2.0} and experimentally determined scales.

Model #	Cell count	Scales	Conf. thresh.	IoU thresh.
4	3+	{0.14, 0.22, 0.30, 0.60, 0.96}	0.20	0.20
5	4+	{0.16, 0.24, 0.32, 0.64, 0.96}	0.20	0.20
6	7+	{0.22, 0.30, 0.38, 0.76, 0.96}	0.15	0.10

3.4 Evaluation

3.4.1 Ground Truth Matching

Evaluation in object detection problems is more involved than in e.g. classification problems, where labels can be directly compared to ground truth. The prediction output format (of both models) is a list of bounding boxes for each image of positive predictions, i.e. denoting where rouleaux have been detected. (Any hypothetical bounding box not predicted is implicitly a negative prediction.) To decide which predictions qualify as correct and incorrect, i.e. true positive or false positive, these predicted bounding boxes must be *matched* to ground truth bounding boxes according to some set of rules.

Matching bounding boxes means deciding which predicted bounding boxes correspond to which ground truth bounding boxes, if any. This is similar to the default box matching in the SSD, however, the goal is different. This matching can be done in any variety of ways, so it is essential to match such that boxes are matched if and only if the model prediction corresponds to the actual annotated object (or part of it).

In our evaluation, prediction boxes are matched sequentially, and to be matched with a ground truth box, they must satisfy an IoU threshold of 0.5. Predictions are first sorted by confidence score and then in turn matched to their highest IoU ground truth box not already matched to a prediction and satisfying the IoU threshold. If no such ground truth box exists, the prediction is classified as a false positive.

After the predictions have been matched to ground truth, their precision, recall and F1-score can be calculated, which we use to evaluate the models.

4 Results

In Tables 6, 7 and 8 we present the precision, recall and F1-scores of our tested models for the three test sets. The cell with the best score for each metric is marked with green background.

Table 6: Performance of all models on Test 1. The best score for each metric is marked with green background.

Test 1				
Type	**Model #**	**Precision**	**Recall**	**F1-Score**
RP	1	0.53	0.54	0.53
	2	0.64	0.34	0.44
	3	0.69	0.12	0.29
SSD	1	0.50	0.40	0.44
	2	0.45	0.24	0.32
	3	0.37	0.048	0.085
	4	0.53	0.43	0.47
	5	0.58	0.30	0.40
	6	0.55	0.077	0.14

Table 7: Performance of all models on Test 2. The best score for each metric is marked with green background.

Test 2				
Type	**Model #**	**Precision**	**Recall**	**F1-Score**
RP	1	0.48	0.64	0.57
	2	0.64	0.45	0.53
	3	0.69	0.16	0.25
SSD	1	0.45	0.43	0.44
	2	0.45	0.29	0.35
	3	0.37	0.059	0.10
	4	0.50	0.48	0.49
	5	0.58	0.36	0.45
	6	0.55	0.094	0.16

Table 8: Performance of all models on Test 3. The best score for each metric is marked with green background.

Test 3				
Type	**Model #**	**Precision**	**Recall**	**F1-Score**
RP	1	0.27	0.76	0.40
	2	0.40	0.73	0.52
	3	0.66	0.40	0.50
SSD	1	0.19	0.30	0.23
	2	0.27	0.29	0.28
	3	0.37	0.095	0.15
	4	0.23	0.37	0.28
	5	0.34	0.35	0.34
	6	0.53	0.15	0.23

5 Discussion

5.1 Evaluation Metrics and Aim

While the balance between precision and recall varies a lot between different models for each test set, the best F1-score for each test set is in the range between 0.50 and 0.60; the best models for each set seem to perform roughly the same. The best performance according to the F1-score was found on Test 2 for RP Model 1, with an F1-score of 0.57. For a general object detection model, this is not a very robust result. You would not want your self-driving car to miss half of the stop signs, and see stop signs where there are not any just as often (which would correspond to precision and recall, and thus F1-score, equal to 0.50). However, with regards to the aim, the requirements are different, and these metrics alone do not necessarily paint the full picture. To interpret the results, we will first present a qualitative discussion on the interpretation of low precision and recall values in relation to the task.

5.1.1 Precision and Recall In Relation To the Task

Theoretically, a model with low recall could be accurately used to indicate the presence of rouleaux in a blood smear given some idealised circumstances. If the distribution of features that are used to successfully identify rouleaux do not vary (among rouleaux) between blood smears, the detections of a model with low recall will still correlate closely with the actual amount of rouleaux. To illustrate this with an example, imagine a model which only detects S-shaped rouleaux (and does so perfectly). The model will consequently miss most rouleaux (since they are not usually S-shaped), but the number of detections is proportional to

the true amount of rouleaux, meaning it can be used as an indicator of the relative concentration. However, lower recall might imply that some "types" of rouleaux may be unrecognisable to the model. The "type" of rouleaux might correlate with other factors in the blood smear. Thus, the model might be able to accurately detect the presence of rouleaux in some patients or blood smears, but completely miss them in others, potentially giving the user a false sense of security. Lower recall also means that the variance in number of detections increases, causing the amount of data needed to be analysed for accurate estimation of the relative amount of rouleaux to go up. This is especially problematic if trying to detect rouleaux in very low concentrations.

Low precision can be more detrimental than low recall, since it implies that the model has not correctly learned an accurate representation of what a rouleau is. Another problem with precision as a metric is that it is highly dependent on the concentration of rouleaux in the test set of the model. The number of true positives is proportional to the rouleaux concentration, and given that the number of false positives is independent of the concentration, precision is asymptotically inversely proportional to the concentration by definition. Depending on the natural variance of rouleau-like objects, a lower precision model could still be used to detect rouleaux in higher concentration areas. If the number of false positives per area is relatively constant between slides, the signal of the amount of true positives can still be found among the noise consisting of false positive if there are enough occurrences.

Not only is the interpretation of these metrics highly dependant on the quality of errors, but the notion of "rouleau-like" objects leads us to a second important factor for precision (and partially for recall). As previously discussed, what is classified as a rouleau is highly context-dependent, meaning that models might have learned "local representations" of what rouleaux look like well, with precision suffering from positive detections of aggregates which might be considered rouleaux in a different context. Such false positives are not independent of the rouleau concentration, quite the opposite. Detection of such objects is proportional to the presence of rouleaux, but is not reflected by precision, which is contingent on "actual rouleau classification", and not rouleau-likeness. This problem stems from the fact that neither model takes region context into account, however, it is also a question of evaluation, independent of model design. There are a slew of other factors affecting the values of evaluation metrics, which are only due to the design of the evaluation process.

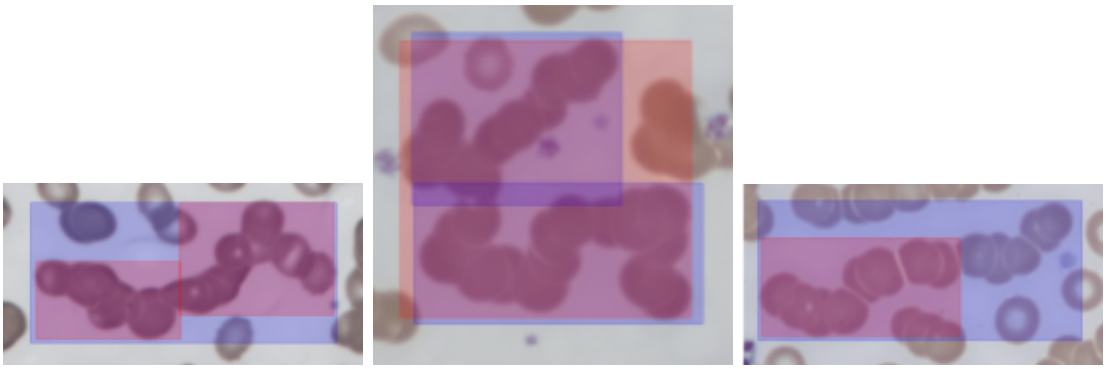
5.1.2 Evaluation Challenges

As mentioned, evaluation in object detection tasks is not as clear cut as simple classification tasks, and one has to design what constitutes a correct and incorrect prediction more closely. For simplicity, we opted to use a standard method of matching predictions to ground truth on the basis of IoU and quantifying true and false positives. However, there are some important differences to how one

might qualify what counts as a match compared to other object detection tasks. This stems from the composite quality of rouleaux; whereas a car which is split down the middle might not be considered two cars, this is at large true for rouleaux. More specifically, this results in three phenomena in prediction which cause ambiguity in evaluation:

- *Split predictions* – where a single annotated rouleau is spanned by two predictions
- *Joint predictions* – where two adjacent annotated rouleaux are spanned by a single prediction
- *Partial predictions* – where parts of a rouleau have been predicted

These are illustrated in Figure 26. While such predictions are suboptimal, one might consider them a far lesser problem than detecting a rouleau where there is not one. Furthermore, these phenomena, especially split predictions, partly stem from the ambiguity of the annotation process. As previously discussed, rouleaux are often branched and the separation of RBCs can sometimes be unclear, meaning a group of connected RBCs might be considered to be a single or multiple rouleaux depending on interpretation. These effects are especially prominent for larger rouleaux, which contain more potential for ambiguity.



(a) Split prediction (b) Joint prediction (c) Partial prediction

Figure 26: Illustration of split, joint and partial predictions.

The effect of these phenomena on precision and recall can be detrimental. A split prediction produces two false positives and one false negative (missed ground truth), and a joint prediction produces one false positive and two false negatives, meaning their effect is far greater than for example a general false positive.

The effect on evaluation metrics could be lessened by changing how ground truth matching is performed, including how false/true positives/negatives are qualified. To combat the effects of split and joint predictions, a more complex matching

strategy could be used by letting several predictions be matched to a single ground truth annotation and vice versa, allowing these predictions to be counted as true positives. However, this produces problems with accurately counting rouleaux of a given size.

To combat the effect of partial predictions, one could use a different box matching criteria, such as intersection area over prediction area. Thus, a partial prediction where the prediction box is enclosed in the ground truth box would be counted as a true positive. However, detecting only a miniscule part of a large rouleau should arguably not be counted as a "full" true positive. Alternatively, one could lower the IoU threshold for ground truth matching, but this creates the potential for incorrect predictions to be falsely counted as true positives.

Another factor that greatly impacts the precision on especially Test 3 is the detection of smaller rouleaux which are not included in the set. While detecting smaller rouleaux might not be the goal, the division of rouleaux in the sizes used is ambiguous, and the exact number of RBCs is often hard to count precisely. In the RP architecture, the main factor deciding whether they are positively classified is the shape segment area, since the cell threshold is kept low for all configurations. However, the area does not accurately correspond to the number of cells. These predictions could be counted as neither true or false positives when evaluating larger test sets, however, the evaluation would then correspond poorly to the actual behaviour of the model.

We have not performed a quantitative analysis on the effect of these factors on the results, however, we believe that these phenomena are commonly occurring, and potentially skew the results. This is especially true for the biggest class of rouleaux, meaning that the direct comparison of evaluation metrics between test sets might be misleading.

5.1.3 Evaluating Results

Taking the above discussion into account, we believe the results indicate that our models are capable of detecting rouleaux to the point where they could be used to indicate rouleaux concentrations in blood smears. However, the results are potentially not robust enough for use in production. Missed detections could lead to a false sense of security, leading users to not further investigate the presence of rouleaux. On the other hand, false positives might lead to make the feature too unreliable, ultimately leading to lack of use. This has to be further investigated before use. Doing this requires working closely with an expert, both to interpret results and to develop more sophisticated methods for evaluation.

5.2 Training Data

The architectures perform comparably for Test 1 and 2, but for Test 3, the SSD performs much worse. One reason for this is the different methods of subsetting used for training data between the two architectures. Annotations for training originate from the annotation tool, which supplies a cell count based on segmentation data from the CellaVision RBC Segmenter. However, the segmenter consistently detects groups of adhering RBCs as one object, resulting in undercounting of cell counts, especially for larger, more tightly adhering rouleaux. Adding to this, training on only large rouleaux greatly impacts the amount of positive examples in and of itself. As a consequence, the number of positive examples as well as the quality of the data suffered greatly for SSD when training on only large rouleaux. For the RP configurations, we instead opted to filter rouleaux by size (both for training and prediction) mainly by their shape segment area. The RP configurations also filter by cell count, but all models use the low value of 3.

However, we believe the RP architecture is less data intensive overall. The qualitative reason for this is that while the SSD has to "learn localisation", in the RP architecture the localisation is not learned, but instead implemented manually using task-specific information. By the same logic, we believe the SSD is more sensitive to the data overall. In the RP architecture, the learning of the object detection problem is essentially transferred to the simpler learning problem of classification, as well as enabling for more robust rotation augmentation. Ultimately, we believe the SSD has the capacity to learn representations of rouleaux without major alterations, given more training data of better quality.

5.3 RP Architecture

A fundamental problem with the RP architecture is that the CellaVision RBC Segmenter is not optimised for connected RBCs. On the contrary, one of the main objectives of the segmenter is to find detached cells for analysis. This problem is exacerbated by the fact that the clustering algorithm does not take into account the visual information indicating whether two RBCs are connected, and the ellipse approximation of cells do not perfectly match their shape. While the clustering method used provides a cheap method for ROI localisation, one could consider opting for directly using the the segment shape extractor to examine all segments. However, we believe the main motivation behind the RP architecture has merit. RBCs, as the building blocks of rouleaux, are simple to detect and recognise, and as such, they are able to provide useful information, whether for finding ROIs or in classification, compared to analysing rouleaux as a unit.

5.4 SSD

A crucial part of the SSD is the default box configurations, which is indicated by differences in performance between the models trained on the same data. All models with default box scales based on statistical analysis of the data perform worse than with experimental scales. This raises the question of the efficiency of the statistical method.

Differing centres of default boxes and ground truth boxes reduces the IoU, which emphasises the importance of the positions of the default boxes. Determining their positions as is done now, i.e. mapping the locations of the prediction feature maps back to the input image, results in cases of non-overlapping default boxes for some scales and aspect ratios. This consequently increases the risk of a ground truth box not being matched to any default box and thus the network does not learn to detect said object. This may partly be the reason the statistically chosen scales fails compared to the experimental scales. Although the boxes are equally spaced, the statistically based boxes are consistently smaller and therefore do not cover as much of the image.

To increase the frequency of the default boxes, their positions could be explicitly set. Another option is to extract one or more feature maps of higher resolution from earlier layers in the base network. Then, however, it must be considered if the higher resolution feature maps should be used in addition to the original feature maps or replace them.

5.5 Conclusions

Compared to most other cell objects of interest in blood smear analysis, rouleaux differ greatly in shape and size, which means designing object detection solutions is harder. A perhaps greater problem stems from the ambiguity in annotation and composite nature of rouleaux, making both evaluation and training machine learning networks harder. We believe both architectures form the basis for something that works, however, not even the best performing model configurations are ready for production use; results have to be evaluated together with an expert. As of now, there are factors potentially greatly impacting the performance metrics used left to be investigated.

We argue that the RP architecture is less data sensitive, and its multi-stage design makes it easier to train and tune, since different stages can be analysed separately. These are the main advantages of the RP architecture, which performs better with the set up used in the project. However, we believe that the SSD has the capacity to learn accurate representations of rouleaux given enough data. The SSD architecture is widely used and easier to implement and main-

tain, which has big practical advantages. For both architectures, optimisations can be made. However, we believe the collection of more and higher quality data is of higher priority in the current situation.

5.6 Future Work

The main area for future work is to develop methods for accurately evaluating the robustness of models in conjunction with an expert, as well as collecting more data of better annotation quality for training. Then, after models which are deemed robust for detection of individual rouleaux have been developed, a system for what concentrations and types are deemed worthy of alerting must be developed.

We also believe that investigating the ability of more context-aware predictions to mimic the classification process of experts could be beneficial. Another interesting topic for investigation is detecting what areas of blood smears to focus on for finding rouleaux.

References

[1] MedlinePlus. *Blood smear*. https://medlineplus.gov/ency/article/003665.htm. [Accessed: Feb. 15, 2021].

[2] CellaVision. *Introducing Digital Cell Morphology by CellaVision*. https://www.cellavision.com/images/pdf/CellaVision-Concept-Brochure.pdf. [Accessed: Feb. 15, 2021].

[3] R.W. Samsel and A.S. Perelson. "Kinetics of rouleau formation. I. A mass action approach with geometric features". In: *Biophysical Journal* 37.2 (1982), pp. 493–514. DOI: 10.1016/S0006-3495(82)84696-1.

[4] eClinpath. *Smear examination*. https://eclinpath.com/hematology/hemogram-basics/blood-smear-examination/. [Accessed: May. 23, 2021.

[5] Carolina Vilchez. "13 - Examination of the peripheral blood film and correlation with the complete blood count". In: *Rodak's Hematology*. Sixth Edition. St. Louis (MO): Elsevier, 2020, p. 203. ISBN: 978-0-323-53045-3. DOI: 10.1016/B978-0-323-53045-3.00022-2.

[6] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167.

[7] Xiaoyue Jiang et al. *Deep learning in object detection and recognition*. Springer, 2019. DOI: 10.1007/978-981-10-5152-4.

[8] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524.

[9] Wei Liu et al. *SSD: Single Shot MultiBox Detector*. 2015. arXiv: 1512.02325.

[10] Kaiming He et al. *Identity Mappings in Deep Residual Networks*. 2016. arXiv: 1603.05027.

Master's Theses in Mathematical Sciences 2021:E39
ISSN 1404-6342

LUTFMA-3454-2021

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/