Technical University
of Denmark

Action Classification on the Deformation of Erythrocytes for Fetal-Maternal Haemorrhage Diagnostics

Peter Kampen, s183993

Joachim Secher, s183973

Nicklas Bruun-Andersen, s183979

Gustav Als, s184400

Special Course

Supervisor: Anders Nymark Christensen

January 12, 2022

Contents

Abstract **1**

Introduction **2**

1 Fetal Maternal Hemorrhage **3**
1.1 Introduction to fetal maternal hemorrhage . . . 3
1.2 Current methods of diagnosis . . . 3

2 Dataset **4**

3 Models and Theory **6**
3.1 Related Work . . . 6
3.2 ResNet . . . 6
3.3 Mask R-CNN . . . 7
3.4 SlowFast . . . 8
3.5 Saliency Heatmaps - Grad-CAM . . . 9
3.6 Preprocessing . . . 10
3.6.1 Detection of RBCs . . . 10
3.6.2 Annotation of RBCs . . . 11
3.6.3 Creation of videos . . . 11
3.7 Training setup . . . 12

4 Results and Experiments **13**
4.1 Results . . . 13
4.1.1 Initial model . . . 13
4.1.2 Intermediary Experiments . . . 14
4.1.3 Final results . . . 15
4.1.4 Mask R-CNN results . . . 17

5 Discussion **19**
5.1 Initial/Final results . . . 19
5.1.1 Mask R-CNN . . . 19
5.2 Difficulties with the domain . . . 20
5.3 Future work . . . 20

Conclusion **21**

Bibliography **22**

Abstract

Fetal-Maternal Hemorrhage is a common and potentially lethal condition. Diagnosis of it remains elusive from the fact that symptoms are rare and subtle, while current tests are labor intensive and cumbersome. Previous work has shown that fetal and maternal red blood cells differ in size and mechanical properties. We introduce a deep learning based method of classifying individual blood cells as they pass through a thin channel. This is done by capturing a video of each blood cell such that only that cell is present in the video. We then utilise the SlowFast [FFMH19] neural network architecture for action recognition to classify each cell. This allows us to incorporate temporal features of the cells deformation. This method yielded an accuracy of 87.02% on our test set while yielding a slightly lower accuracy on the validation set. This performance is a major improvement over those achieved in previous work [Thy20] [EF20], where the temporal features were not accounted for. Two different networks were trained with different relative weights on the spatial component. The results indicated that a significant amount of the discriminative features lies along a temporal axis. Finally we propose a full deep learning based pipeline for blood cell classification, that incorporates a Mask R-CNN architecture with the purpose of capturing the videos. Given constraints on time this was not made fully functional in this study. But previous work indicate that this could bear fruitful results.

Introduction

This project contributes to further exploring whether Deep Learning methods can be utilized in Fetal Maternal Hemorrhage diagnostics. Namely by performing action classification on the deformation of red blood cells (erythrocytes) and classifying whether they originate from the maternal or fetal blood circulation. Fetal Maternal Hemorrhage, which is simply put leakage of fetal blood into the maternal circulation, is quite troublesome to diagnose and the current methods are not ideal. An introduction to Fetal Maternal Hemorrhage and an small overview overview of the current methods is given in chapter 1. The dataset provided by the *Department of Health Technology* at DTU, was collected and processed in [Hoe18]. It consists of images captured as red blood cells moves through a narrow channel. The narrow width of the channel leads to the cells being deformed as they pass through. The dataset and how it was collected is explained in greater detail in chapter 2.

Given the video format of our data we wish to utilize the temporal aspect of our data as previous work indicate that classification using only spatial information does not achieve the desired accuracy [Thy20]. This notion of wanting to exploit the video based nature of the data mandates a neural network architecture, which can extract spatio-temporal features. Different architecture and networks were considered but ultimately the *SlowFast* architecture was chosen. The architecture chosen, *SlowFast* by Facebook AI Research (*FAIR*) is a state-of-the-art 3D CNN architecture achieving some of the best performance on well known action classification test set. *SlowFast* consists of two pathways, a slow pathway to capture the spatial structure and a fast pathway to the temporal structure. This nature seemed fitting to our data as we have a static channel and a moving blood cell. This architecture is explained in more detail in chapter 3, where we also present some related models and some of the associated theory. We will also justify and elaborate on the different approaches taken as we tuned our approach.

This leads to us presenting and discussing our experiments and their results. We will in chapter 4 first present the results of our initial model, leading into the different intermediary experiments and their results. Leading up to the main and final results found in subsection 4.1.3. The results is then discussed and commented in chapter 5 in which we also suggests ideas for further exploration.

Part 1

Fetal Maternal Hemorrhage

1.1 Introduction to fetal maternal hemorrhage

Fetal Maternal Hemorrhage (FMH) is the occurrence of a disruption to the maternal-fetal barrier that causes the fetal blood to enter the maternal circulation [JM17]. In majority of known FMH cases the cause is still largely unaccounted for. Physical trauma to the abdomen of the mother is prevalent in approximately 15% of the cases. However, in around 82% of the cases, no direct link to relevant incidences can be found and the FMH is said to arise spontaneously. [NS11]
The frequency of FMH is quite high with leakages of less than 0.05mL occurring in approximately 74% of pregnancies [ESS90]. Although the incident rate is high, most occurrences are benign and cause no distress to either mother or child. However if the leakage is large enough it can compromise the fetus which could result in stillbirth, fetal demise or giving birth to a severely anemic infant. Unfortunately the symptoms of significant FMH is often subtle and nonspecific making it difficult to identify. The severity of the condition is both linked to the volume of blood entering the maternal circulation and the speed of the hemorrhage. Most quantifications of the severity of the condition however refers to the volume. A leakage of more than 30mL appears in approx 0.33% of pregnancies with a $^{1}/_{1000}$ probability of perinatal mortality. [ESS90]

1.2 Current methods of diagnosis

Diagnosis of FMH remains difficult due to symptoms being subtle and non specific. Different method of testing for FMH exits however most being cumbersome, imprecise or labor intense. The most common test for determining the amount of fetal blood in the maternal circulation is the KB test. The KB test is labor intense as it involves processing a blood sample through different treatments to wash out the maternal cells. Then manually count the amount of present fetal cells using this to extrapolate. This method of testing is however subject to many factors which could lead to inaccurate interpretations of the result. The blood may be affected by temperature, pH and time since the sample was drawn. The fetal may not be stained as well as expected and conditions like maternal hemoglobinopathies also make the interpretation difficult. Another way to test for FMH is using Flow Cytometry. It is less labor intensive and utilize size difference between fetal and maternal RBCs however despite being less labor intensive it require specific equipment which might not be widely available. Being able to use the dynamics and size of the RBCs for classification on widely available equipment is therefore an interesting area of research. [NS11]

Part 2

Dataset

The dataset collected for this study consists of videos of RBCs being funneled through a narrow channel. The motivation for this data setup is grounded in the discrepancies between deformation properties for fetal and maternal RBCs. The dataset was collected and processed by Freja Høier as described in "Deformation of blood cells in the PolyNano Demonstrator Chip" [Hoe18].

The RBC videos were acquired using a PolyNano Demonstrator chip, a microfluidic pump combined with a high speed camera, which captures images through a microscope.

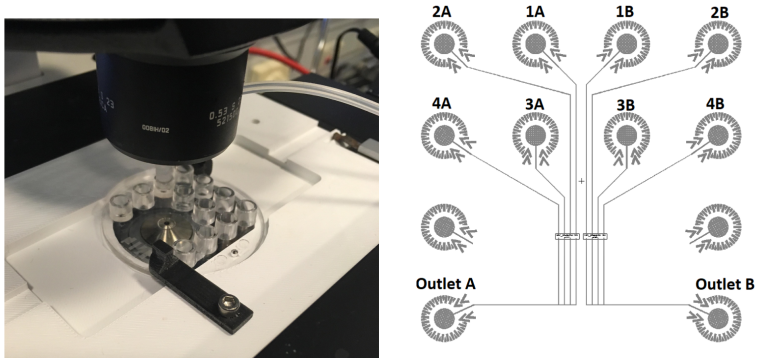

Figure 2.1 – **Left**: The PolyNano capturing setup is shown including: the chip (center), the camera (top) and the pump attached to inlet 4A. **Right**: A schematic of the PolyNano chip, where A and B are mirror image of each other.

In fig. 2.1 both the physical setup and a schematic modelling of the PolyNano chip is shown. The chip is symmetric over the vertical axis leading to an A and B part which are not discriminated over in this study. The numbering corresponds to different channel topologies which are visualized in fig. 2.2.

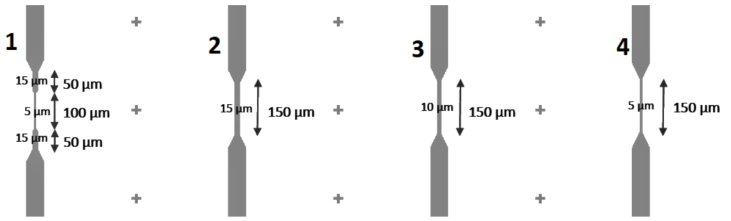

Figure 2.2 – Topology of the channels used in the PolyNano chip.

A blood sample is passed though one of the four inlets shown in fig. 2.1 then reaching the corresponding channel in fig. 2.2. A microfluidic pump supplies continuous pressure to establish a desirable flow though the channel. Finally the sample exits through the outlet. The high speed camera captures images of the center of the channels shown in fig. 2.2. The chip is transparent to allow for image capturing. The videos are captured through a microscope to allow for sufficient magnification.

The equipment used for this specific dataset is described below:

- **Pump**: Fluigent MFCS-EZ microfluidic pump
- **Microscope**: Leica DMI 3000 B microscope with 100x oil magnification
- **Camera**: AOS S-motion high-speed camera

Further for the dataset included in this study a selection of parameters were held constant for all images collected, shown in the table below:

Parameter	*Value*	*Unit*
Passage width	5	μm
Shutter Time	25	μs
Pump Pressure	6	$mbar$
Frame Rate	500	fps
Image Resolution	800×150	-
Magnification	$\times 100$	-

Table 2.1 – Setup parameters that are held constant throughout the dataset analyzed in this study.

The parameters specified in table 2.1 are chosen based on tests done by [Hoe18].
Given an average RBC width of $6 - 8\mu m$, channels with a width above this threshold will not cause sufficient RBC deformation for the purpose of this study. Channel 1 and 4 showed the most promising deformation, where channel 4 is superior due to a longer narrow channel, allowing for longer passages of detectable deformation. The frame rate in table 2.1 was chosen to achieve an average sequence of around 50-100 frames per cell traveling through the channel.

In total 43300 frames are collected for each video sequence. This sequence length is determined by an upper bound memory capacity for the AOS camera which is 5GB. Each video sequence is linked to a blood sample from a single donor: maternal or fetal, where the setup is rinsed between samples. Hence the ground truth class labels are readily determined from the label of the blood sample passed through the setup. In total the data set used in this study contains image sequences from 57 maternal donors and 35 fetal donors. An example of a maternal blood cell passing through the channel is shown in fig. 2.3:

Figure 2.3 – A maternal RBC passing through channel 4, $5\mu m$.

Part 3

Models and Theory

3.1 Related Work

Describing differences in cell deformation given a certain feature set is of great interest, as it can be used to discriminate between certain cell types e.g. fetal and maternal blood cells. In *Deformation of blood cells in the PolyNano Demonstrator Chip* [Hoe18] Hoier collected a large standardized dataset of RBCs aiming at gaining a comprehensive view of their deformation mechanics, induced by hydrodynamic shear stress. Through use of image analysis it was investigated if a descriptive parameter could probably separate single instances of a given RBC class. However no such parameter was found. It was hence of interest to investigate the inclusion of a large parameter space when classifying RBCs.
In later work by Thybo [Thy20] the classification capabilities of a ResNet based CNN, namely the Mask R-CNN [Gir17], were tested for the dataset collected by Hoier. The network based its predictions on single frame instances of either fetal or maternal RBCs. Experiments showed accuracies ranging from 54.9% up to a best case 67.13%. While this indicates a presence of discriminatory features, the precision is not adequate for diagnostic purposes. Other work that applies the Mask R-CNN architecture have managed to attain a classification accuracy of 72% [EF20].

In recent years progress has been made in the action classification space of deep learning for visual computing. Earlier implementations lacked "end-to-end" functionality and required feature engineering often specific to a certain dataset, however current state-of-the-art networks are trying to surpass this issue. Implementations tackling benchmark datasets such as Kinetics and AVA [GSR+18][KCS+17] have reached high levels of precision. Further many architectures have been made generic to an extend where they can be applied for a variety of datasets. Following these developments this study includes the SlowFast action recognition network applied to the RBC dataset. This is of interest as the mentioned earlier implementations of classification models all base their predictions on single frames. The inherent dynamic change in RBC deformation is hence not included, this however can be captured using an action recognition architecture.

3.2 ResNet

The ResNet architecture was introduced by Zhang et al. [HZRS15] in 2016 as an approach to avoid the phenomenon of vanishing or exploding gradients which is a problem that occurs when training deeper neural networks. This results in the accuracy being saturated, and the performance on the test data may decrease, but not as a result of overfitting.

The problem is addressed by introducing a so called residual block, which implements an identity mapping, also called *skip connections*. An example of the structure of a residual block can be seen in fig. 3.1 If we let $\frac{\partial \mathcal{L}}{\partial \mathcal{X}_\ell}$ denote the gradient in a particular layer ℓ, and

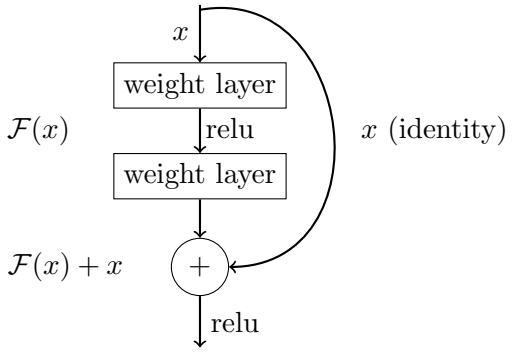

Figure 3.1 – An example of a residual block.

let then L and l denote two layers in the network, where L is a deeper layer than l. Then it can be show that using a residual block structure the gradient in an given layer can be decomposed to

$$\frac{\partial \mathcal{L}}{\partial \mathcal{X}_l} = \frac{\partial \mathcal{L}}{\partial \mathcal{X}_L}\frac{\partial \mathcal{X}_L}{\partial \mathcal{X}_l} = \frac{\partial \mathcal{L}}{\partial \mathcal{X}_L}\left(1 + \frac{\partial}{\partial \mathcal{X}_l}\sum_{i=l}^{L-1}\mathcal{F}\left(\mathcal{X}_i, \mathcal{W}_i\right)\right) \tag{3.1}$$

as show in [AK21]. This shows that the gradient in layer L can be propagated to layer l without any perturbations, and thus mitigating the problem of vanishing and exploding gradients.

The backbone of SlowFast consist of a ResNet-50 network. Each of the two pathways contains four residual blocks, and both pathways implements a modified 3D ResNet [FPW16], which is a variation of ResNet aimed at action classification.

3.3 Mask R-CNN

The Mask R-CNN structure was introduced by He et al. [Gir17] in 2018, as an improvement to the Faster R-CNN structure. It must be noted that it is only of interest in the scope of this analysis to extract the outputs that the Mask R-CNN share with the Faster R-CNN. However, we chose the Mask R-CNN implementation for the eventuality that we would require pixel wise classification. The necessity of this feature has not presented itself, hence we shall in this section primarily focus on the bounding-box part of the network outputs.

The architecture is structured into 4 parts, which function somewhat concurrently, but optimized to share a lot of the parameters. The initial step, which is shared. Is a fully convolutional network[1] with the purpose of extracting feature maps from the given image. Prior to this a collection of partially overlapping anchors or bounding boxes are proposed. These, along with the extracted feature maps from the CNN, are fed into a Region Proposal Network (RPN). The RPN is still a fully convolutional layer. It utilises the original feature

[1]Fully convolutional means that the head of the network is also a convolutional layer.

maps to yield a measure of the probability that an anchor is foreground or background. This measure comes from Intersection over Union (IoU), where the training bounding boxes are used as foreground, while everything else is background. Let δ denote the measure function of the area of a region in the image.

$$IoU = \frac{\delta(A \cap B)}{\delta(A \cup B)} \tag{3.2}$$

All anchors with an IoU above 0.7 are then passed further into the network. From there the network splits into a bounding box regressor and classifier. These work with a combined loss. Let $\tilde{p}_i$ denote the ground truth class, and $\tilde{t}_i$ the ground truth position of a bounding box. The loss is then given by

$$L(\{p_i\}, \{t_i\}) = c_1 \sum_{i \in \Omega} L_{cls}(p_i, \tilde{p}_i) + c_2 \sum_{i \in \Omega} L_{reg}(t_i, \tilde{t}_i) \tag{3.3}$$

Where $c_1 \approx c_2$ are predetermined normalizing parameters. t_i denotes a parameterization of the coordinates and L_{cls}, L_{reg} denote the classification and regression loss respectively. L_{cls} is simply the binary cross entropy, while

$$L_{reg}(t, \tilde{t}) = R(t - \tilde{t}) = \begin{cases} \frac{1}{2}(t - \tilde{t})^2 & \text{If } |t - \tilde{t}| < 1 \\ |t - \tilde{t}| - 0.5 & \text{Otherwise.} \end{cases} \tag{3.4}$$

The anchors are then considered as regions of interest (RoI), and subsequently split into k rectangles of the same size. Using non-max suppression the network cut down on the number of these rectangles. Thus we obtain the needed rough segmentation. The actual network then finalizes with a fast R-CNN to perform classification on each of the regions. However, since we only have one class we shall not dwell on it any further.

3.4 SlowFast

In 2019 Feichtenhofer et. al. introduced a two stream end-to-end network architecture for video recognition [FFMH19]. The main idea of the paper is a structure with two pathways: a "Slow" pathway that focuses on extracting spatial features from the input, and a "Fast" pathway which takes a high temporal resolution input stream and aims at capturing motion related context. The authors were inspired by the findings on the mechanisms of the ganglion cells in the human eye. Here two cell types were found to supplement each other, namely P-cells and M-cells. P-cells focus on spatial detail and color, where the M-cells have higher temporal frequency. In fig. 3.2 the SlowFast network architecture is visualized:

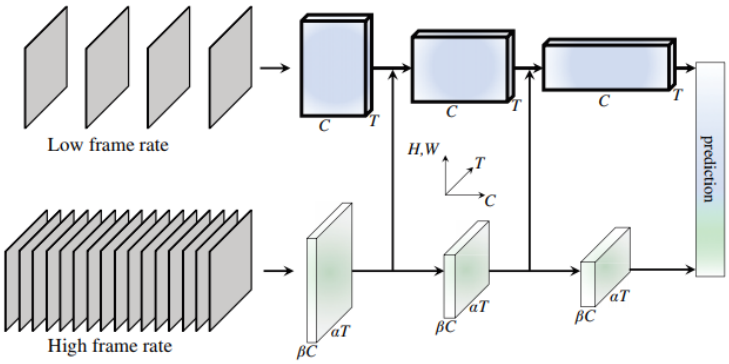

Figure 3.2 – The two stream SlowFast network architecture developed for video recognition [FFMH19].

Slow pathway. This stream uses 3D ResNet blocks adapted to process spatio-temporal input data. The authors introduce the hyperparameter τ to control *temporal stride*, i.e. the pathway processes one out of τ frames. τ hence regulates the temporal frequency of the incoming frames. Feichtenhofer et. al. found a configuration of $\tau = 16$ to be optimal for large datasets like *Kinetics*. In this paper however the limited amount of frames per passing RBC imposes a restriction for high values of τ, hence in this study we mainly apply $\tau = 8$.

Fast pathway. Again a 3D ResNet block is used, now with a *temporal stride* of τ/α. α represents the frame rate ratio between the fast and slow pathway, and is set to $\alpha = 8$ in this study. To limit computational cost, the fast pathway has lower channel capacity, with a ratio of $\beta = 1/8$ channels of the slow pathway.

The information of the two pathways are fused using lateral connections. A unidirectional fusion is used where features from the fast pathway is propagated into the slow pathway. The fusions are included between different "stages" of the network, specifically for a ResNet-50, four lateral connections are present. Due to the discrepancy in temporal dimension, a transformation is done, where the authors experimented with *Time-to-channel*, *Time-strided sampling* and *Time-strided convolution*. The default transformation in their implementation is *Time-strided convolution*.
The SlowFast architecture achieved state-of-the-art in 2019 on video recognition datasets such as *Kinetics* and *AVA*[GSR+18][KCS+17], and is used in this paper as a benchmark video recognition architecture for RBC classification.

3.5 Saliency Heatmaps - Grad-CAM

It was of interest in this study to consider the spatial contributions to the networks choices of classification. To that end we consider saliency heatmaps as generated by Grad-CAM. This is a method of assigning importance to each element (pixel) in the input. While several different methods exists, both parametric and non-parametric, we employed a gradient based method. The purpose is to find the class discriminative localisation map $Ł^c_{\text{Grad-CAM}} \in \mathbb{R}^{t \times h \times w}$ with respect to the class c, where h, w denotes the height and the width of the input respectively. While t in the case of action classification, is the temporal component, as these maps will of course change over time. In order to estimate $L^c_{\text{Grad-CAM}}$ we consider the activations of the last convolutional layer $A^k_{t,i,j}$. It must be noted that the slowfast structure indeed has two *last convolutional layers*, hence in practice we consider

both $A^{\text{Slow}}_{t,i,j}$ and $A^{\text{Fast}}_{t,i,j}$. There is no theoretical difference between the two, hence we shall consider the Grad-CAM for an arbitrary network. The backpropagation algorithm yields the gradients $\frac{\partial y^c}{\partial A^k_{t,i,j}}$ over which a global average pool is performed, thus yielding

$$\alpha^c_{t,k} = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{t,i,j}}. \tag{3.5}$$

Thus $\alpha^c_{t,k}$ is a linear measure of the impact each feature map k has on the classification. Since convolutional layers maintains relative spatial importance we let

$$L^c_{t,i,j} = \text{ReLU}\left(\sum_k \alpha^c_{t,k} A^k_{t,i,j}\right) = \max\{0, \sum_k \alpha^c_{t,k} A^k_{t,i,j}\}. \tag{3.6}$$

This compresses the spatial importance over a $h \times w$ grid to a $i \times j$ grid, where $(i, j) < (h, w)$. The usage of the element wise ReLU activation entails that we only consider features with a positive impact on the classification. Other methods uses the ℓ^2 approximation of the L^2 norm over the image. However this would provide a absolute measure of which areas that have a large impact on the classification. Since the dimensionality of the output is much less than that of the input we use trilinear interpolation to visualize the actual locations (since there is a temporal axis too).

3.6 Preprocessing

3.6.1 Detection of RBCs

Methods from classical image analysis, such as BLOB detection, was used to detect the individual RBCs and generate bounding boxes for them, which allowed for all images not containing any RBCs being removed from the analysis.
To perform BLOB detection the images has to be converted to a binary images, in such a fashion that the foreground consist of the RBCs, and everything else is treated as background. To achieve this, it is used that only the RBCs change position across the all the frames in for specific donor, and every other part of the image can be assumed to be static. An easy to compute representation of the static information in the image is to compute a median image.

The procedure used to perform this analysis for a single donor (a single video sequence), was to compute an approximate median image of a random sample of 100 images for the given donor. To account for any small variations and noise that may still occur, the 100 images were filtered using a gaussian filter with a kernel size of $\sigma = 4$ beforehand.

Then each frame for the given donor is processed. First the median image is subtracted from the current image, resulting in every part of the image which are similar to the median image getting a value close to zero. Note that at this point we have a pseudo image, where values of the pixels do not have any clear interpretation, as they can be both positive and negative. A binary version of the image is then created simply using a threshold value, T, where if the absolute value of the pixel is less than T the pixel is set as background, and else it is foreground. It was found that a value of $T = 2$ gave the best results on dataset A. The process is visualized in fig. 3.3.
The individual BLOBs in the binary image are then detected using the measure package from `SciPy`, which automatically gives a lot of information about the BLOBs, including bounding boxes. The binary image is not without noise, and as such some very small

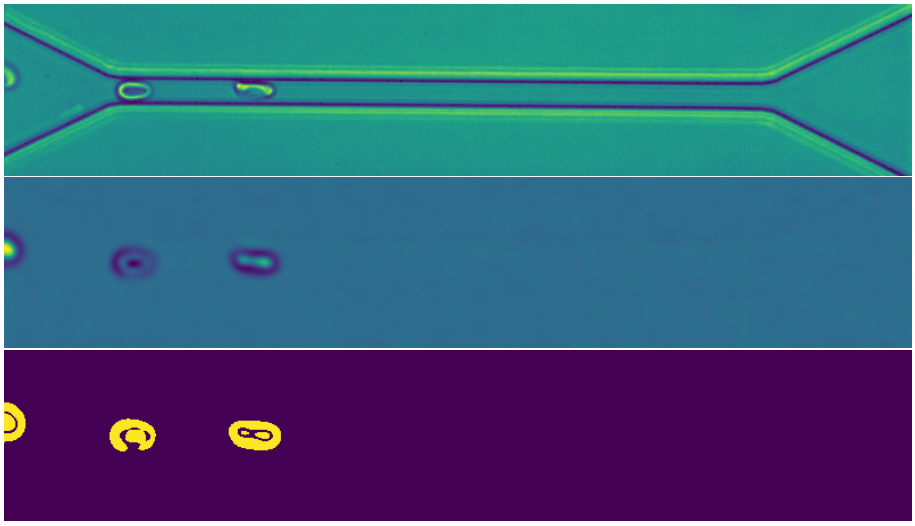

Figure 3.3 – From top to bottom - Raw image from donor D128. Image with gaussian filter and median image subtracted. Binarized image.

BLOBs which are not RBCs are being detected. This is mitigated by removing all BLOBs with a size of less than 50 pixels.
The remaining BLOBs are then classified as RBCs, and their position is known from the bounding boxes of the BLOBs.

It was however observed, that in some cases a part of edge and center of the RBC was close to the same pixel intensity as the background, when the RBC was inside the canal. In the process of creating the binary image, a single RBC are divided into two separate BLOBS.
These cases are detected from their lack of circularity, and overlapping bounding boxes, as it result in long, thin shapes. These BLOBS are then grouped back together, as a single BLOB, despite the BLOBs not being connected.

3.6.2 Annotation of RBCs

Every RBC is tracked across the frames in which they are present. They are then annotated with a unique identifier, regardless of the number of RBCs in a given frame. An RBC is followed from one frame to the next by applying the heuristic that movement from frame to frame is smaller than the distance to the nearest other RBC. Hence the proximity of two BLOB centers is a good metric for tracking a RBC. Additionally it is enforced that a BLOB can only move from left to right. Given these measures the RBCs are annotated.

3.6.3 Creation of videos

Given the annotated images, the data can be processed to extract the video format used in subsection 4.1.3. The bounding boxes for each individual RBC are used to create the videos, in such a way that a square video only consisting of the area inside the bounding box. A RBC video starts when the RBC enters the inlet, and it stretches until it reaches the far right side of the channel outlet.

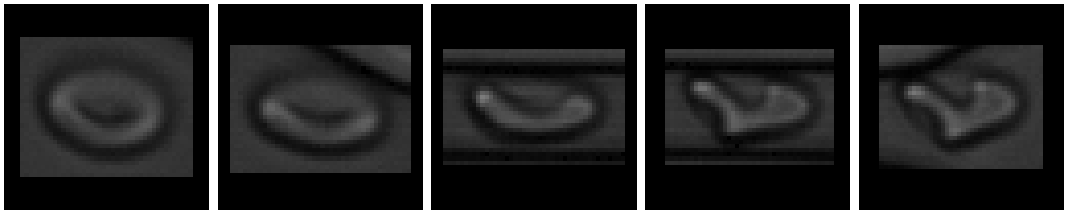

Figure 3.4 – **Frame 1**: RBC enters inlet, in the left part of the frame. **Frame 2**: RBC reaches channel. **Frame 3 & 4**: RBC is funneled through channel, showing deformation. **Frame 5**: RBC leaves channel.

3.7 Training setup

For all training performed in this report the following was used: 70% of the data used in training, 20% in validation and 10% for testing. Two distinct subsets of the original RBC dataset are used throughout this study. A smaller dataset was used for exploratory testing and initial results and in the subsequent experiments. This is denoted dataset A. For the final results a more complete dataset is used, denoted dataset B. In table 3.1 the fetal/maternal distribution of the two subsets are shown.

Table 3.1 – Subsets of the original RBC dataset, used in this study. Numbers refer to the number of donors in each dataset.

	Maternal	Fetal
Dataset A	6	6
Dataset B	57	35

To gain a meaningful accuracy metric, a uniform class distribution is maintained in the test and validation sets. As optimization algorithm, stochastic gradient descent (SGD) is used. During training we apply a learning rate of 0.1, learning momentum of 0.9 and a weight decay of 10^{-4}. Binary cross entropy is used as only two classes (fetal or maternal) exist. For the available GPU memory a batch size of 8 is the maximum possible and is hence used. For the SlowFast specific parameters the following are used, $\alpha = 8$, $\beta = 1/8$ and $\tau = 8$ (and one model with $\tau = 2$). In table 3.2 hyperparameters that change for different models, are displayed.

Table 3.2 – Different hyperparameter configurations for SlowFast models.

	Initial/Experiments	Final $\tau = 8$	Final $\tau = 2$
Epochs	196	280	340
ResNet	50	101	101

All models are trained on the DTU Compute cluster. The GPU types are NVIDIA GTX 2080 Ti, NVIDIA GTX 1080 Ti, Titan X and Titan V, with either 11 or 12 GB dedicated memory. During training the incoming frame is resized such that the shorter size is randomly sampled from $[256, 320]$, as required by the SlowFast implementation. Subsequently a 224×224 random crop is taken and passed into the network.
For inference/evaluation, 10 frames are sampled along the temporal axis for every input video. The frames are resized such that the shorter spatial axis is scaled to 256, and three 256×256 crops are taken. The softmax scores are averaged for final prediction.

Part 4

Results and Experiments

4.1 Results

We shall in this section outline some of the results achieved in the study. Particularly we emphasise the initial results and some of the issues that the majority of our energy was spent solving. The final method are further presented. The individual experiments leading to the final method are outlined in subsection 4.1.2.

4.1.1 Initial model

Initially we considered dataset A table 3.1. A ResNet-50 backbone was chosen to cut down on training time, while validating during training, every 2 epochs. The Top-1 error curves on the training and validation sets can be seen in fig. 4.1. The corresponding final Top-1 error for all three splits are tabulated in table 4.1.

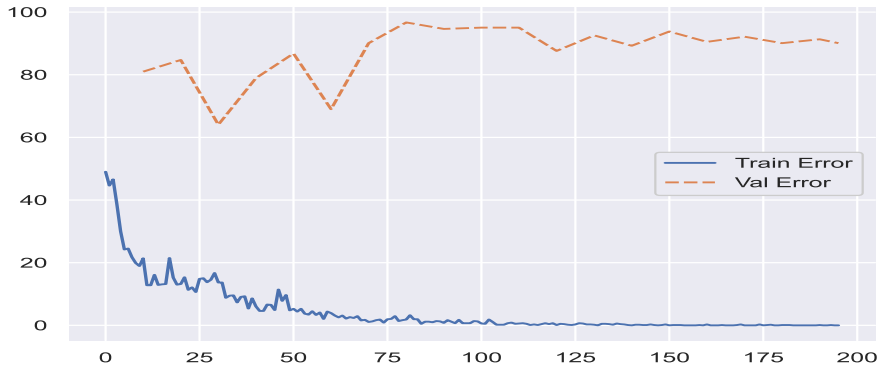

Figure 4.1 – Top-1 error on the train and validation splits during training.

Table 4.1 – Minimum and final Top-1 errors (%) for the training, validation and test split.

	Train	Validation	Test
Min. Top-1 Error	0.00	64.50	12.39
Final Top-1 Error	0.00	90.08	12.39

The results in fig. 4.1 and table 4.1 are clearly spurious. One can of course easily hypothesise overfitting, however skewed by the fact that the validation error increases almost monotonically from the beginning. One ought also to note that the initial result on the

test set was quite satisfactory. This does not stem from a well fitted model, but instead a test set skewed towards fetal RBCs, and as the classifier favors classifying this class the Top-1 error achieved is mistakenly low. The classifiers inability to explain the information contained in the domain is evident when comparing the test error to the unimpressive final validation Top-1 error of 90.08%. The blatantly obvious occurrence of overfitting observed in fig. 4.1 indicates that the discriminatory class features found by the model are in fact not correlated with the explanatory class features. It was hypothesized that the surrounding channel which the RBC passes through was included in the classification. To verify this hypothesis the Saliency Heatmap - Grad-CAM described in section 3.5 was applied to a section of a frame evaluated by the model. In fig. 4.2 the resulting heatmaps are shown.

Figure 4.2 – An arbitrarily sampled maternal cropped RBC frame with a Saliency Heatmap - Grad-CAM included. **Left**: slow pathway. **Right**: fast pathway.

In fig. 4.2 it is observed that for an arbitrarily sampled maternal RBC the slow pathway determines the lower right corner of the channel as being indicative of the class. The fast pathway includes other sections of the channel, however not detecting the RBC. This is not satisfactory for RBC classification. To combat the model extracting information from the channel a series of experiments were conducted.

4.1.2 Intermediary Experiments

To remedy the overfitting phenomena linked to the channel found in fig. 4.1 and fig. 4.2 different regularizing techniques were applied.

Disabling slow pathway: As explained in section 3.4 the slow pathway in the SlowFast architecture is included to effectively capture the spatial information contained in the frames. However since the channel is temporally invariant the need for a dedicated spatial pathway could be hypothesized redundant. Hence turning off the slow pathway in the network could theoretically lead to the model including mainly temporal features, RBC movement, and ideally avoid the channel affinity from fig. 4.2. This was implemented by passing zeros to the slow pathway, while passing the original frame to the fast pathway, thereby solely applying the fast pathway for training and evaluation. Both the Top-1 error and Saliency heatmap were observed to be similar to what was observed in fig. 4.2 and table 4.1. Thus it is clear that disabling the slow pathway does not improve the models classification capabilities, hence further experiments were conducted.

Additional data augmentations: In the space of Deep Learning one of the most reliably effective techniques for mitigating overfitting is to perform data augmentation when

training a model. This is especially true for high variance datasets with many discriminatory features. However for the RBC dataset included in this study, the structural changes in each RBC which the model should capture are quite subtle. Hence applying distorting augmentations on the blood cell was approached with care. Scale jittering and random horizontal flipping were thus included as possibly regularizing augmentation strategies. Regrettably both the Top-1 error showed little improvement, table 4.2, in comparison to the results from table 4.1.

Video Blackening: As neither of the previous attempts successfully minimized overfitting more invasive and non-online strategies were included. It can be assumed that no information required for classification is contained in the channel, hence the preprocessing pipeline was extended by setting all pixel values to 0 outside of a 100×100-pixel square with the RBC in the center. This was implemented using the bounding boxes described in subsection 3.6.1. An example frame is shown in fig. 4.3.

Figure 4.3 – Example RBC illustrating a frame where all but a 100×100-pixel radius is blackened.

Training a model on data where the preprocessing shown in fig. 4.3 had been applied proved to be unfruitful. The training error did not converge table 4.2. This stems from the model not being able to learn enough information. This follows from the fact that 91.66% of each frame now contains no information (pixel values are 0). Thus passing no information through the model.

Table 4.2 – Final top 1 errors (%) for the training, validation and test split in the three experiments.

	Train	Validation	Test
Disabling SlowFast	0.00	91.23	12.49
Data Augmentation	0.00	87.23	14.21
Video Blackening	37.14	48.92	44.58

Given the conclusions found for the exploratory experiments described above, a final strategy was composed to effectively handle channel overfitting.

4.1.3 Final results

To exclude as much information regarding the channel from the final videos as possible, each frame is now centered around the RBC, and only the pixels contained in the bounding box are passed to the model. This format can be seen in fig. 3.4. With this cropping strategy the amount of redundant channel related information contained in each frame is evidently reduced.
Applying the cropping strategy to the preprocessing pipeline initially lead to poor validation and test results on dataset A, table 3.1. Subsequently the preprocessing pipeline was applied to dataset B, table 3.1. Two models were trained on this dataset, one using $\tau = 2$

for 340 epochs and another using $\tau = 8$ for 280 epochs. The training and validation Top-1 errors are shown over every epoch for the $\tau = 8$ model in fig. 4.4.

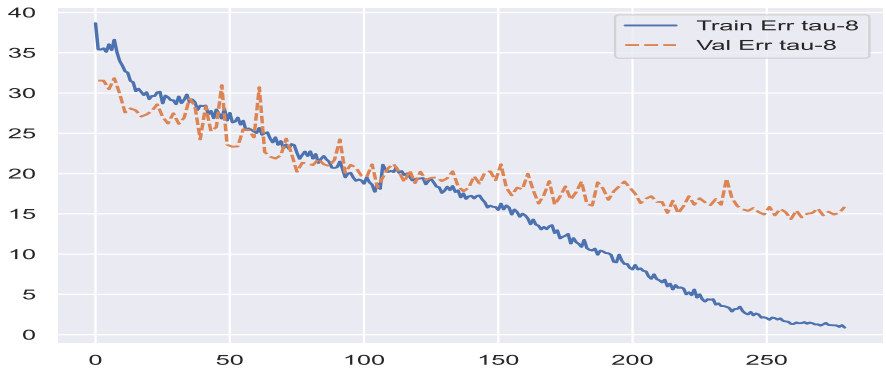

Figure 4.4 – Training and validation Top-1 error for all epochs for the $\tau = 8$ model, trained using the preprocessing strategy described in subsection 4.1.3.

The minimum Top-1 errors for the models is shown in table 4.3.

Table 4.3 – Minimum and final Top-1 errors (%) for the training, validation and test split.

	Train	Val.	Test
Min. Top-1 Error $\tau = 8$	0.870	14.35	12.98
Min. Top-1 Error $\tau = 2$	4.93	14.35	13.40

From table 4.3 it is seen that both models have learned the training dataset, where $\tau = 8$, displays the lowest error. The validation errors are equivalent, and the test error is minutely lower for $\tau = 8$.

In fig. 4.4 a steady decrease in both training and validation error is observed over the first 120 epochs. The training error decreases further over the remaining epochs, however this is not reflected in the validation error which partially stagnates. Interestingly the validation error does not increase as would be the case for a overfitting scenario. Instead it traverses through several local minima and reaches a best epoch just after 250 epochs. The lack of overfitting could point to the fact that the feature distribution which is being learned is not completely represented in the training data. From table 4.3 it is observed that the validation set reflects the test set well for both models, as their errors are of equal magnitude. It is further evident that the information contained in the given dataset is learned fully given the near zero error displayed in table 4.3, especially true for the $\tau = 8$ model. Hence for further exploration adjusting the domain might be of interest, this will be discussed in chapter 5.
In fig. 4.5 the confusion matrices for the two models are shown for evaluation on the test set.

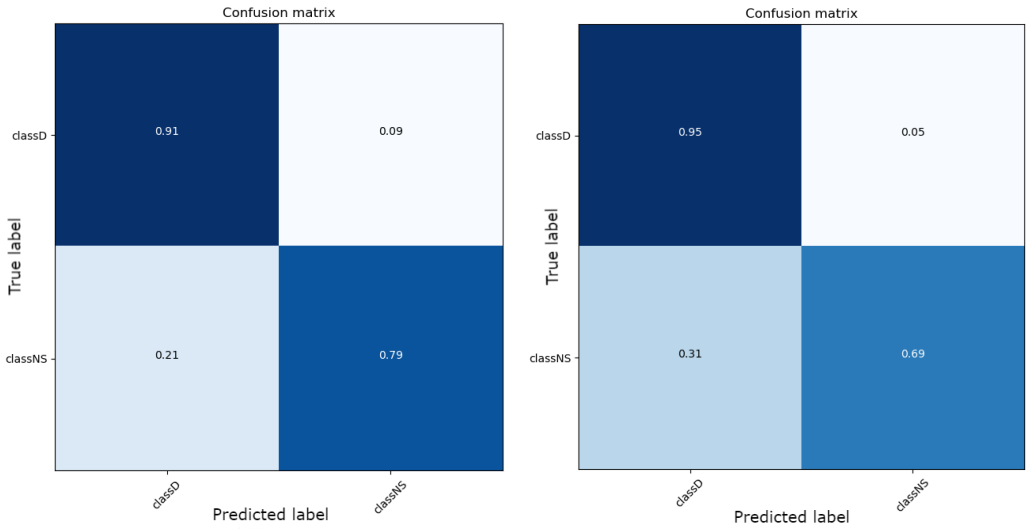

Figure 4.5 – Confusion matrices for the final models evaluated on the test set. **Left**: $\tau = 8$. **Right**: $\tau = 2$.

Both models display a partially skewed affinity in favor of maternal bloods cells, which results in 21% false negatives for fetal blood cells for $\tau = 8$ and 31% for $\tau = 2$. This is not ideal due to the nature of the classification problem, where classification of fetal cells have the highest weight. The issue might be partially attributed to a training set skewed towards maternal blood cells.

4.1.4 Mask R-CNN results

The goal of applying a R-CNN to our dataset amounted to gaining a more reliable set of bounding boxes for the RBC dataset. Further it was of interest to have a bounding box accuracy metric (IoU) such that thresholding would be possible for the proposed RoIs. Given the static nature of the RBC dataset it was hypothesized that a smaller subset of the full dataset could be used for training the bounding box regressor. Subsequently the trained R-CNN can be evaluated on the full dataset. Finally the sequential heuristics applied for annotating RBCs for a video in subsection 3.6.2 can be applied.
The R-CNN used is a COCO pre-trained network with a ResNet-101 backbone. The network was trained on dataset A, table 3.1. In table 4.4 the training hyperparameters are presented.

Table 4.4 – Hyperparameter configuration used for training the Mask R-CNN network.

Hyperparameter	Value
Epochs	10
Batch size	2
Weight decay	10^{-4}
RoI detection threshold	0.7
non-max suppression threshold	0.3
Learning momentum	0.9
Learning rate	10^{-4}

The loss function described in eq. (3.3) is applied. A small batch size is used due to GPU memory constraints, and no batch normalization is applied since performance gain

is negligible for small batch sizes. In fig. 4.6 the training and validation loss (L_{reg} and L_{cls}) is shown for each epoch.

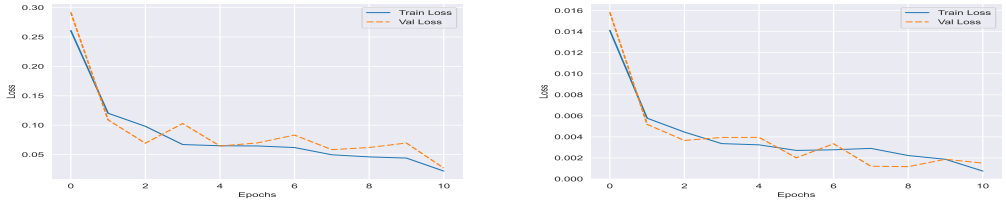

Figure 4.6 – The bounding box regression loss L_{reg} (Top) and the classification loss L_{cls} (Bottom) over the 10 epochs of training.

From fig. 4.6 it is observed that the initial loss is relatively low, likely due to a successful application of transfer learning achieved by applying a COCO pre-trained model. The final validation loss is $(1 - IoU) = 0.02701$. The trained R-CNN is evaluated on the full dataset, excluding the 12 donors used for training and validation. While fig. 4.6 indicates a well-fitted model, this evaluation provided spurious bounding boxes which proved no improvement when comparing with the BLOB based bounding boxes extracted using the method described in section 3.6. Due to this result, and as a consequence of time constraints, improvement on the R-CNN was pursued no further. For a complete RBC classification pipeline where efficient data exploitation is of interest, a functioning R-CNN appears to be an obvious component. However this is left for future work.

Part 5

Discussion

5.1 Initial/Final results

In subsection 4.1.1 it is observed that models trained on dataset A, table 3.1, fail to learn a correct representation of the class specific features despite a seeming data abundance with 500.000 total frames. This indicates a possibly larger feature space in the RBC domain than can be explained by the data, this will be elaborated on further in section 5.2. The subset further contained misleading class specific information related to the PolyNano channel as discussed in subsection 4.1.2.

For dataset B, table 3.1, the class feature distribution explained in the training dataset is noticeably better correlated with both the validation and test sets. In fig. 4.5 it was however observed that models trained on this dataset had a clear affinity towards maternal blood cells, pushed on by a skewed dataset. For a Deep Learning classifier to be valid as a diagnostic tool used towards lethal conditions, it has to achieve low false negatives for the given illness. With this in mind the 79% true positive for fetal RBCs from fig. 4.5 is not completely satisfactory. It could thus be of interest to include a weighting scheme in which correct classification of the fetal blood cells weighted higher. This could be included as a step in the training pipeline, or a thresholding mechanism in the inference process. It is worth noting that these measures might yield better performance on fetal blood, however decreasing total accuracy.

5.1.1 Mask R-CNN

In fig. 4.6 both the training and validation errors for RBC detection are observed to drop steadily over the 10 epochs for which the model is trained, reaching a minimum of $(1 - IoU) = 0.02701$. However as explained in subsection 4.1.4 this is not reflected in stable bounding boxes when evaluating on the entire dataset. Earlier work such as [EF20] nonetheless indicates that a R-CNN can achieve high accuracy on RBC detection for similar data. Therefore for a future RBC detection and classification pipeline, inclusion of a R-CNN as the detection engine is the obvious choice. This is especially true given the high accuracy these networks achieve. This will allow for a greater exploitation of the incoming data stream, as edge cases can be included. Further stability towards detection of multiple RBCs in near proximity of on another is a requirement for a functional diagnostic engine. This is where image detection algorithms such as BLOB analysis are especially lacking, and where the Mask R-CNN instance segmentation capabilities are superior.

5.2 Difficulties with the domain

The domain yields a few challenges that might affect the classification ability of the networks. The first problem comes from the fact that we are trying to classify a inherently three dimensional motion based on a projection onto two dimensions. Thus a significant amount of potentially discriminative information might lie in this domain. This also entails that a cell might be presented in several different ways depending on its orientation in those three dimensions. The cause of the remaining error could therefore potentially be, that we simply do not have enough data to discriminate in all orientations of the blood cells.

Another potentially debilitating problem arises from the very low resolution of the image files. The average RBC is approximately a meer 30×30 pixels in the image. Prior to being passed to the network the image is resized to approximately 5 times larger, to be compatible with the slowfast implementation, where of course no new information is added. Thus if we consider an activation A^k in the first convolutional layer. Then A^k is given by a linear combination of $(5 \times 7 \times 7)$ elements in the input. However since most of these are linear combinations of much fewer elements, then the node is unable to encapsulate much of the structure. This entails that each activation in the first layer is able to encapsulate less input from the data. The initial layers would therefore probably be used to pass the information along until it has been downsized sufficiently, rather than learning meaningful things. Thus ideally we would have a resolution that yielded a RBC of roughly 224×224 pixels.

5.3 Future work

In order to achieve a better performing model, a few explorations could be done. The obvious first experiment would be to make an ensemble of networks. We do not have a large variety of different backbone within the slowfast framework available. Hence we would propose achieving the variability of each model in the ensemble by varying the sample rate in the slow pathway.

As we saw in fig. 4.5 networks with different sampling rates perform quite different, however with an overall similar accuracy. Furthermore, for each of these networks methods such as stochastic weight averaging or simply sampling versions of the networks at different stages in training could lead to a larger and/or better ensemble. Finally if other types of networks were to be employed in an ensemble or simply as stand-alone, it might improve the performance of the pipeline. In particular transformer based networks such as Microsoft's SWIN transformer might yield interesting results for this application.

It might also be conceivable that an improved version of the pipeline outlined in this paper could be extended to other domains. Further an investigation into whether this pipeline for detection could be extended to other type of cells would be of interest. Focused on movement/deformation of cells.

Conclusion

In this study we have successfully incorporated the SlowFast architecture in order to do single RBC classification. The incorporation of temporal features led to a significant performance increase over previous work that relied only on spatial features. Using the SlowFast architecture we have achieved an accuracy of 87.02%, which is notably higher than the previous best results achieved by [EF20]. A significant amount of the work in this study was in the preprocessing. The basis of this was a simple method utilising classical image analysis tools. It was shown in previous works that neural network architectures such as Mask R-CNN could yield very robust results. However time did not permit us to incorporate it fully into the pipeline. Finally we saw in table 4.3 that a lower sample rate ($\tau = 8$) in the slow pathway yielded slightly better results than for a higher sample rate ($\tau = 2$). The fact that the class accuracies, fig. 4.5, were noticeably different, points towards the possibility that ensembling could yield a performance increase. Since we did not utilise all data, future work ought to incorporate that in the training process. This along with an improved pipeline provides an opportunity for improved results. We can therefore conclude that a large amount of the discriminative features lie along a temporal axis.

Bibliography

[AK21] Gustav Ragnar Støttrup Als and Peter Johannes Tejlgaard Kampen. Skin lesion classification using deep neural networks on dermoscopic images, 2021.

[EF20] Rasmus Alkestrup Eskesen and Theis Friis. Hydrodynamic deformability-based classifcation of fetal and adult red blood cells using deep learning, 2020.

[ESS90] H. F. POLESKY E. S. SEBRINAGN. Reviews fetomaternal hemorrhage: incidence, occurrence, and clinical risk factors, time of effects. 1990.

[FFMH19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. 2019.

[FPW16] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition, 2016.

[Gir17] Kaiming He Georgia Gkioxari Piotr Dollar Ross Girshick. Mask r-cnn. 2017.

[GSR+18] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. 2018.

[Hoe18] Freja Hoeier. Deformation of blood cells in the polynano demonstrator chip, 2018.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[JM17] Shigeru Ohki Jun Miyahara, Hiroshi Sugiura. Survival of an infant with massive fetomaternal hemorrhage with a neonatal hemoglobin concentration of 1.2 g/dl without evident neurodevelopmental sequelae. 2017.

[KCS+17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. 2017.

[NS11] Eric W. Reynolds Nino Solomonia, Karen Playforth. Fetal-maternal hemorrhage: A case and literature review. 2011.

[Thy20] Jonathan Thybo. Detection of fetal-maternal haemorrhage using deep learning-based image classification of erythrocytes, 2020.