

CapillaryNet: An Automated System to Quantify Skin Capillary Density and Red Blood Cell Velocity from Handheld Vital Microscopy

Maged Abdalla Helmy
magedaa@uio.no
Department of Informatics
University of Oslo
Norway

Anastasiya Dykyy
ODI Medical AS
Norway

Tuyen Trung Truong
Department of Mathematics
University of Oslo
Norway

Paulo Ferreira
Department of Informatics
University of Oslo
Norway

Eric Jul
Department of Informatics
University of Oslo
Norway

ABSTRACT

Capillaries are the smallest vessels in the body responsible for the delivery of oxygen and nutrients to the surrounding cells. Various diseases have been shown to alter the density of nutritive capillaries and the flow velocity of erythrocytes. In previous studies, capillary density and flow velocity have been assessed manually by trained specialists. Manual analysis of a standard 20-second long microvascular video takes on average 20 minutes and requires extensive training. Several studies have reported that manual analysis hinders the application of microvascular microscopy in a clinical setting. In this paper, we present a fully automated state-of-the-art system, called CapillaryNet, that can quantify skin nutritive capillary density and red blood cell velocity from handheld microscopy videos. Moreover, CapillaryNet measures several novel microvascular parameters that researchers were previously unable to quantify, i.e. capillary hematocrit and intra-capillary flow velocity heterogeneity. Our system has been used to analyze skin microcirculation videos from various patient groups (COVID-19, pancreatitis, and acute heart diseases). Our proposed system excels from existing capillary detection systems as it combines the speed of traditional computer vision algorithms and the accuracy of convolutional neural networks.

CCS CONCEPTS

• **Computer systems organization** → **Neural networks;**
Data flow architectures.

KEYWORDS

Deep Learning, Convolutional Neural Networks, Automated Image Analysis, Microcirculation Analysis

1 INTRODUCTION

Capillaries are the smallest blood vessels in the body measuring less than 20 micrometers in diameter [1]. The exchange of many nutrients with the interstitial fluid occurs here including water, oxygen and other essential nutrients necessary for the maintenance of cellular metabolism [2, 3]. A network of capillaries constitutes the microcirculation of the body [4]. Studies assessing microcirculation reveal that capillary density and flow velocity is altered in several diseases [5–17]. Furthermore, some studies claim that changes in microcirculation occur early in disease progression and thus microcirculation monitoring could be used for early detection for various clinical conditions [12, 17]. Determining the density of capillaries allows us to understand the surface area available for the exchange of nutrients [3]. One result of a study suggests that capillary density can be used as an early marker of mortality rate from cardiovascular diseases [6]. A second study shows measuring capillary density can be a key component in understanding fibrotic diseases [18]. A study on chronic kidney disease has determined that a reduction in capillary density over time can be used as early detection to prompt timely interventions [19]. Quantifying the velocity of the red blood cells within the capillaries can help assess fluid homeostasis and the transcapillary fluid flux [3]. One study suggests that alterations in the velocity of red blood cells within the capillaries influence the number of supplies being delivered to the surrounding cells and the time it has to be exchanged across the microvascular wall to the surrounding tissues [20]. Another study shows that an increase in flow velocity in cerebral capillaries can be due to a body response trying to regulate a decrease in oxygen tension in the neurons [21].

A common theme among microvascular analysis articles is the time-consuming, laborious tasks that require clinicians and/or researchers to manually (or semi-automatically) select capillaries and determine the velocity of the blood flow. Such jobs are strenuous on the eyes and are susceptible to errors and observer variations across different datasets. On average it takes a trained researcher 20 minutes to analyze a 20-second long microvascular video [22]. The long analysis time and training requirements are some of the reasons that hinder microvascular microscopy to be integrated into routine clinical practice [23]. Furthermore, manual analysis limits the number of parameters that can be analyzed within a microvascular video. Flow velocity and capillary density are assessed, while intra-capillary flow heterogeneity and Capillary hematocrit are not routinely recorded [23]. Our observations suggest that all the above-mentioned parameters are altered in specific patient groups. In a series of three meetings organized by international experts in microvascular microscopy, it was concluded that an automatic assessment of the microcirculation is required to integrate microvascular microscopy into clinical practice [23].

This article proposes a novel system that can automatically assess the microcirculation in microvascular videos as per the following requirements.

- Detect and quantify capillaries density and capillary hematocrit;
- Track the intra-capillary flow heterogeneity and direction; and
- Classify the velocity of red blood cell flow.

We aim to lift the burden of the manual job from the professionals and allow them to focus on the method and hypotheses development. The input and outputs of CapillaryNet are shown in Figure 1. We compare our proposed architecture with other state-of-the-art capillary detection methods. We also compare the output of CapillaryNet with that of a trained researcher. We show that our approach can be more accurate and efficient than the annotations of the trained researchers and other state-of-the-art capillary detection architecture. This system is used part of a larger system for microcirculation analysis by various institutions and projects around the world.

2 RELATED WORK

In this section we review the current microcirculation systems in the literature.

Dobble et. al [24] uses a frame averaging method to remove the plasma and white blood cells gaps within the capillary before using an algorithm to detect capillaries. Using frame averaging can lead to a lower overall density calculation since capillaries with a majority of gaps or not enough blood flow will be disregarded. Furthermore, Dobble et. al [24] removes

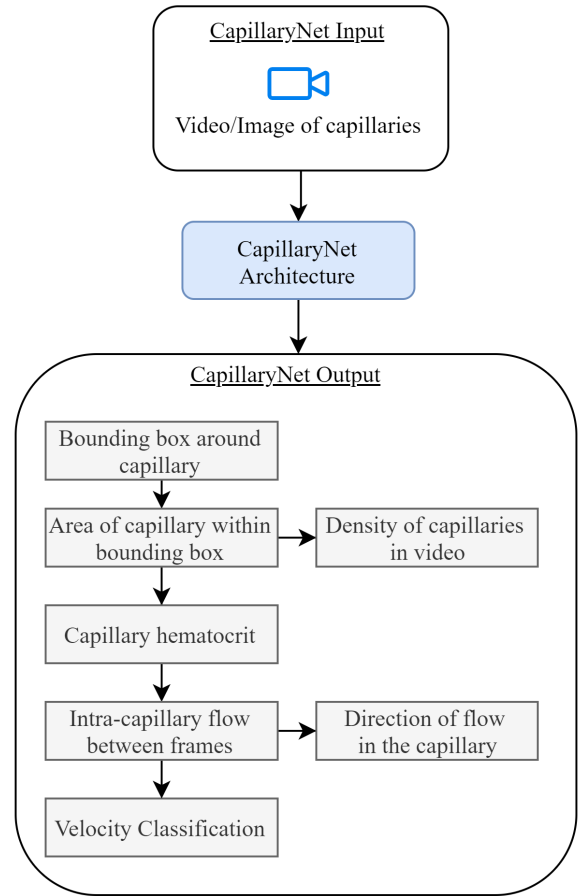


Figure 1: The input and outputs of CapillaryNet is shown above. CapillaryNet takes a video from the microscope as an input, and gives out the shown microcirculation parameters and values

capillaries that are out of focus, since they consider it to add noise to the frame averaging method. From our experiments with handheld microscopy, the nature of the rounded lens may lead to 40% out-of-focus images on both edges of the video. It is very challenging to have a fully focused video the whole time and some parts can always be out of focus. Therefore, this will further significantly reduce the capillary density values.

Hilty et. al [25] has a very similar flow as Dobble et. al [24] with minor changes. Hilty et. al [25] detects capillaries by first generating a mean image across all frames and then passing the resulting image to two pipelines, firstly classifying vessels of 20–30 μm in diameter as capillaries and secondly any vessel of up to 400 μm in diameter as venules. The capillaries are then passed to a modified curvature-based region detection algorithm [26] to an image that has been stabilized and equalized with an adaptive histogram. The result is a

vessel map that contains centerlines across structures that are between 20–30 μm wide. As stated by the authors of the curvature-based region detection algorithm [26], this type of detection is unintelligent and can lead to detecting artifacts such as hair or stains with similar sizes. Furthermore, due to the challenges in skin profile stated above, the mean of the images across the whole video is not always the best representation value since different parts of the video might have different lighting or capillaries can be out of the optimal focus. Moreover, videos that have slight motion will have to be completely disregarded since the central line is calculated across all frames instead of per frame.

Similar to Dobble et. al [24], Bezemer et. al [27] improves the method by using 2D cross-correlation to fill up the blood flow gaps caused by plasma and white blood cells. This is a better method since the number of frames to be disregarded is reduced. However, 2D cross-correlation assumes a uniform flow of blood and does not take into account the dynamic change of flow between the gaps which can inherently decrease the accuracy of prediction.

Tam et. al [28] detect capillaries through a semi-automated method which requires the user to select points on the image. The algorithm then decides if there is a capillary present. Since this method relies on the user to select the capillaries it cannot be used in a clinical environment due to the time of analysis of a microscopy video.

Geyman et. al [29] takes more of a manual approach by first using software to click away from the major blood vessels and then applying hardcoded calculations to detect the total number of capillaries based on the number of pixels in the region of interest. This is quite a manual approach and highly susceptible to observer variations across different datasets.

Demir et. al [30] uses a Contrast Limited Adaptive Histogram Equalization method (CLAHE) [31] with a median filter and an adjustable threshold to detect capillaries on the weighted mean of five consecutive frames. However, these methods need to be adjusted accordingly depending on the illumination on the video and thickness of the skin. This introduces a manual job where the user has to find the right combination of values for different videos, or the same video with different illumination.

Cheng et al. [32] applies an image enhancement step followed by the manual highlighting by the user of the capillaries. The image enhancement process darkens the capillaries and increases the brightness of the background using a best fit histogram method. Using their system, the user can further increase the contrast and smoothen the images manually to increase the differentiation of the capillaries from the background. Macros of this modifications can then be generated and applied to all future captured microscopy videos. However this macro generation assumes that the videos will be

captured with the same brightness and thickness of skin. Moreover, the image used is in the grayscale therefore if there are any artifacts it can be mistaken for capillaries.

Tama et al. [33] uses binarization followed by skeleton extraction and segmentation to quantify the capillaries. The binarization is applied to the green channel since they assume it has the highest contrast between the capillaries and the background. They use the Top-Hat transform method to reduce uneven illumination followed by Wiener filtering to remove noisy pixels and then the gaussian smoothing method to smoothen the image. The OTSU thresholding method is then applied to segment the capillaries from the background. This method relies on the user finding a reference frame from the video which has the highest contrast.

Prentašić et al. [34] used a custom neural network to segment the foveal microvasculature. Their neural network consists of 3 CNN blocks coupled with max pooling along with a dropout layer followed by 2 dense layers. Their neural network was trained in 30 hours and the segmentation took approximately 2 minutes per single image with an accuracy of 83%. The time taken and high-end hardware used to analyze a single image makes it unsuitable for clinical use since the users would like the results instantly.

Dai et al. [35] used a custom neural network similar to Prentašić et al. for segmentation. However, Dai et al. used 5 CNN blocks instead of 3. They used gamma correction and contrast limited adaptive histogram equalization for image enhancement. They reported an accuracy of 60.94%, which is too low to be useful.

Nivedha et al. [36] used the green channel of the image and used a non-linear Support Vector Machine [37] to classify the capillaries. This method involved a manual step where the user had to crop the region of interest to improve the histogram equalization. Nivedha et al performed different experiments comparing different denoising filters such as Gaussian, Wiener, Median and adaptive median and concluded that the Gaussian filter is the most suitable for their data. Furthermore, they compared different segmentation method including OTSU, K Means, watershed and concluded that OTSU method was the most suitable for their data. The segmented images were then passed to an SVM and they achieved an accuracy of 83.3%.

Javia et al. [38] modifies the ResNet18 [39] to quantify capillaries and uses the first 10-layers of the architecture. The main limitation of the ResNet architecture is that images have to be resized to 224x224 however most capillary images are less than 100x100. This means images have to be scaled up which makes this method inefficient and uses more resources than needed. They reported an accuracy of 89.45% on their data however ResNet 18 [39] has 11 million trainable parameters and with such scaling up training time can be up several hours and prediction time can be up to

several minutes. This can make this slow and inefficient to be used within a clinical setting. The training and test time was not reported in this paper.

Hilty et al. [25] uses a similar method to Cheng et al where they first apply an image enhancement process followed by the highlighting of the vessels of interest to quantify the capillary. They use a combination of Contrast-limited adaptive histogram and combination of the first and second-derivative Gaussian kernel convolutions [25] to quantify capillaries. This method is susceptible to detecting artifacts as capillaries in the images since the kernel convolutions will not be able to difference between capillaries and other objects in the image. The accuracy, training and test time was not reported in this paper.

Ye et al. [40] utilized the concept of transfer learning and used the inception Single Shot Detector V2 (ssd-inception v2) [41] to build their neural network. The ssd-inception v2 has high accuracy with reduced computational complexity making it suitable for capillary detection [42]. On the other hand, they used a spatiotemporal diagram analysis for the flow velocity calculation. This method requires white blood cells or plasma gaps in order to detect the velocity accurately. Therefore, capillaries that lacked such characteristics had to be disregarded reducing the overall efficiency of velocity classification. Furthermore, as stated by the authors of the paper the spatiotemporal method can be cumbersome and time consuming. The accuracy was not reported in this paper.

Hariyani et al. [43] used a U-net architecture combined with a dual attention module [44, 45]. The images has to be resized to 256x256 and accuracy of 64% was reported. This accuracy is low for it to be used in a clinical setting.

In the above, the more accurate methods requires semi-automatic analysis while the more automatic methods are less accurate making it unsuitable for a clinical setting. In contrast, CapillaryNet is fully automatic and able to classify microcirculation videos in $\sim 0.9s$ with 93% accuracy making it suitable to be used in a clinical setting. With a dataset of $\sim 11,000$ capillaries, the training time of capillary net is under 10 minutes.¹

3 METHOD

In this section, we explain how CapillaryNet method was developed.

3.1 Data Acquisition for CapillaryNet development

Videos were acquired on human subjects with a hand-held digital microscope (Digital Capillaroscope, Inspectis, Sweden) in the form of videos with a resolution of 1920x1080 at 30 frames per second for 20 seconds. The videos visualized

¹Trained on a Nvidia GeForce RTX 3090

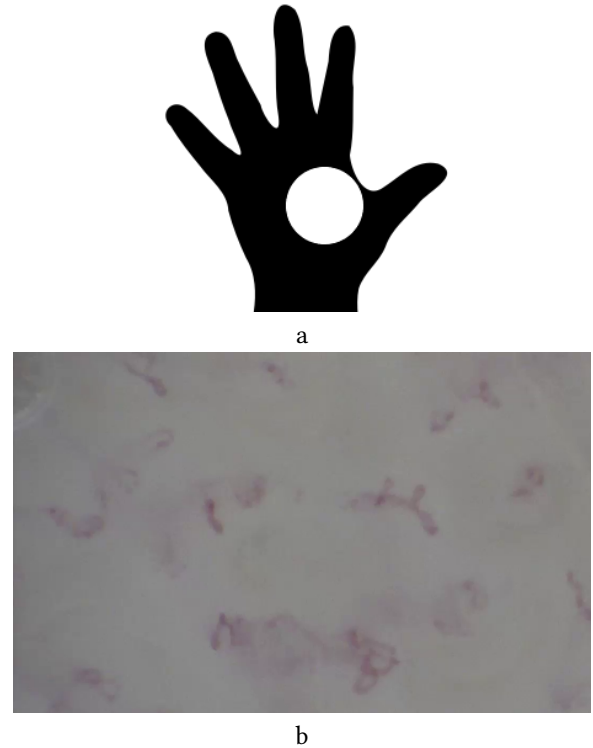


Figure 2: a) The skin region from where the microvascular videos were recorded is marked in white. b) An image from the video captured in that skin region

the nutritive capillaries in the skin papillae in the dorsum region of the hand (Figure 2a shows the region of interest and 2b the capillaries displayed by the microscope in this region). The region was coated with a layer of transparent oil before applying the microscope to the skin. 365 videos displaying the capillaries from 50 subjects were analyzed. All subjects gave their informed consent to participate in the study (IRB protocol and approval number: 2020P001987). The average age of the subjects was 57 years with a standard deviation of 17 years.

3.2 Manual Data Analysis

A trained researcher analyzed the obtained microvascular videos using in-house software for manual marking of capillaries. Capillaries visible in different frames of each video were marked. To calculate the accuracy of the algorithm the labeled bounding boxes and velocity classification were then compared with the algorithm output by the trained researcher. The algorithm was trained by extracting the capillaries within the bounding boxes which were labeled by the researcher. Regions with capillaries were compared to regions with no capillaries and regions with other skin features

(i.e. hair, skin folds and other artifacts). The algorithm was trained on ~70% of the labeled data and then validated on ~30% of the labeled data. CapillaryNet was then tested with 35 new videos acquired independently with a total of ~1000 capillaries. This test dataset was independently analyzed by two researchers and was used to calculate the reproducibility of capillary detection between two researchers.

3.3 CapillaryNet Architecture - Capillary Detection

The first part of our architecture detects the number of capillaries and calculates the density within a given area. This part consists of 2 stages, the region of interest (RoI) generation, and the CNN part. In our case, RoIs are bounding boxes in specific areas in the image suggesting the presence of a capillary. The regions of interest is detected by 2 independent pipelines, by analyzing the HSV color space of the image and by using Structural Similarity Index (SSIM) [46]. Different levels of enhancements are applied to the RoIs and passed to 2 CNNs for detecting if there is a capillary in the RoI. This way our architecture uniquely combines both traditional and Deep Learning models, and tailors it for capillary detection.

3.3.1 Stage 1: Generating RoIs. The first stage generates the RoIs. The literature (see related work section) either used traditional methods solely or deep neural networks. Although traditional methods are relatively faster than deep neural networks, they lack the accuracy of the deep neural networks since deep neural networks can learn from trained data. The RoI stage is critical since it suggests to the algorithm which parts of the image it should analyze. Using purely traditional methods will make the prediction unintelligent as anything that is not background (in our case the skin) will be suggested as a capillary. While with deep neural networks, it will have to analyze the whole image to find if capillaries exist which is resource intensive and time-consuming. For our RoIs stage we use a combination of 2 computer techniques divided into 2 independent pipelines to suggest the RoIs. We use these method instead of deep neural networks since they are relatively less computationally expensive.

The first pipeline generate RoIs by smoothing the image using 3 steps. First by convolving an image with a 5 by 5 normalized box filter and changes the central value with the average value (see equation 1). We then blur the image using a Gaussian blur window in 2D with 9 by 9 (see equation 2). We then alpha blend this image with the original input image using a Gaussian weight of -0.5 (see equation 3). Alpha blending overlays the original image with the smoothing image with a transparent look while it has a original image as the background. With this arithmetic operation we reduce the noise as much as possible while maintaining the details of the image. The transparency mask between both of these

images is known as the alpha mask and in this case we use a value of $\alpha = 1.5$.

$$K = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (1)$$

where:

K = image smoothing factor

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where:

x = distance from the origin in the horizontal axis
 y = distance from the origin in the vertical axis
 σ = standard deviation of the Gaussian distribution

$$G(x) = (1 - \alpha)F_0(x) + \alpha F_1(x), \quad (3)$$

where:

F_0 = original image
 F_1 = smoothing image

The image color is then enhanced by a factor of 3 and the contrast by factor of 2.5 [47]. We then convert the image to the hue, saturation, value (HSV) color model. The value presents the brightness of the color from black to the average saturation value. The hue channel presents the color i.e., red, yellow. The saturation presents the amount of color from gray to the pure form of the color.

In an image, the brightness V , can be calculated by using equation 4 [48]

$$V = \max(R, G, B) \quad (4)$$

The intermediate value C is then calculated by using equation 5 [48]

$$C = V - \min(R, G, B) \quad (5)$$

The value C is used to derive the hue value and is calculate by using the equation 6 [48]

$$H = 60^\circ \times \begin{cases} \text{undefined} & \text{if } C = 0 \\ \left(\frac{G-B}{C}\right) \pmod{6} & \text{if } V = R \\ \left(\frac{B-R}{C} + 2\right) \pmod{6} & \text{if } V = G \\ \left(\frac{R-G}{C} + 4\right) \pmod{6} & \text{if } V = B \end{cases} \quad (6)$$

The saturation of the pixel is calculated by using the equation 7 [48]

$$S = \begin{cases} 0 & \text{if } V = 0 \\ \frac{C}{V}, & \text{otherwise} \end{cases} \quad (7)$$

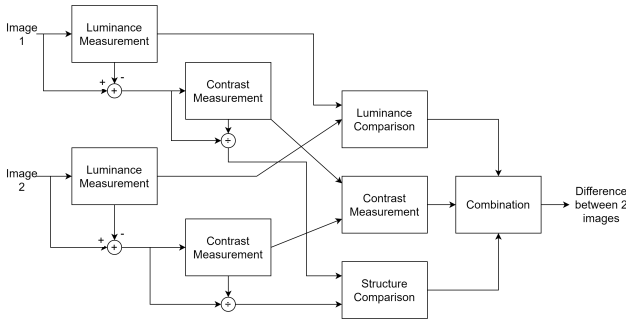


Figure 3: The Structural Similarity Index System

We use HSV color model instead of RGB because the HSV space provides robustness against light changes and shadows [48]. After several experiments, we conclude that the capillaries lay between HSV (155,60,0) and (180, 255, 255). However these values are modifiable in our code and can be changed by the user. We then use the bitwise and operator to calculate the conjunction of pixels between the specified HSV range and the image. This operation only considers pixels that are common between the HSV range and the image. The remaining pixels are removed. We then detect the borders between the removed pixel and the HSV range. The borders are detected using the OTSU threshold combined with contour approximation method [49, 50]. The input is shown in figure 4a, after the processing with the HSV, the image is shown in figure 4b and the output is shown in figure 4c. Every red box is then send to the CNN (in Stage 2) to be classified. The flow of this method is shown in figure 5.

The second pipeline utilizes the SSIM method [46]. The SSIM pipeline extracts the background and subtracts it from the original image. The SSIM method extracts three information from the image which is luminance, contrast and structure. The SSIM method is highlighted in Figure 3[46].

The luminance is measured by averaging the the pixel in a given window (equation 8). In our case, we use a 11 by 11 window.

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (8)$$

The contrast is measured by taking the standard deviation of the pixel values (see equation 9).

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (9)$$

The structural comparison is done by dividing the image with its standard deviation on both input signals, first on the x using

$$(\mathbf{x} - \mu_x) / \sigma_x \quad (10)$$

then on the signal y using

$$(\mathbf{y} - \mu_y) / \sigma_y \quad (11)$$

Each of the above is done by comparing the background image with the original image. The luminance comparison function shows the difference of brightness between 2 images (see equation 12).

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (12)$$

The C1 factor in the luminance comparison equation is calculated by multiplying a constant value with the pixel value (see equation 13). The constant value is determined by experimenting on the dataset and finding a suitable value.

$$C_1 = (K_1L)^2 \quad (13)$$

The contrast comparison between the 2 images is calculated using the standard deviation (see equation 14).

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (14)$$

The C2 factor in the contrast comparison equation is calculated by multiplying a constant value with the pixel value (see equation 15). The constant value is determined by experimenting on the dataset and finding a suitable value.

$$C_2 = (K_2L)^2 \quad (15)$$

The structure comparison function is calculated also using the standard deviation of the images (see equation 16).

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (16)$$

where:

$$\sigma_{xy} = \text{see equation 17}$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (17)$$

The SSIM is then deduced by calculating the difference in luminance, contrast and structural difference (see equation 18) however as deduced by the author of the SSIM paper[46], assuming $\alpha = \beta = \gamma$ are equal to 1 and C3 is half of C2, the equation can be simplified as shown in equation 19.

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma \quad (18)$$

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (19)$$

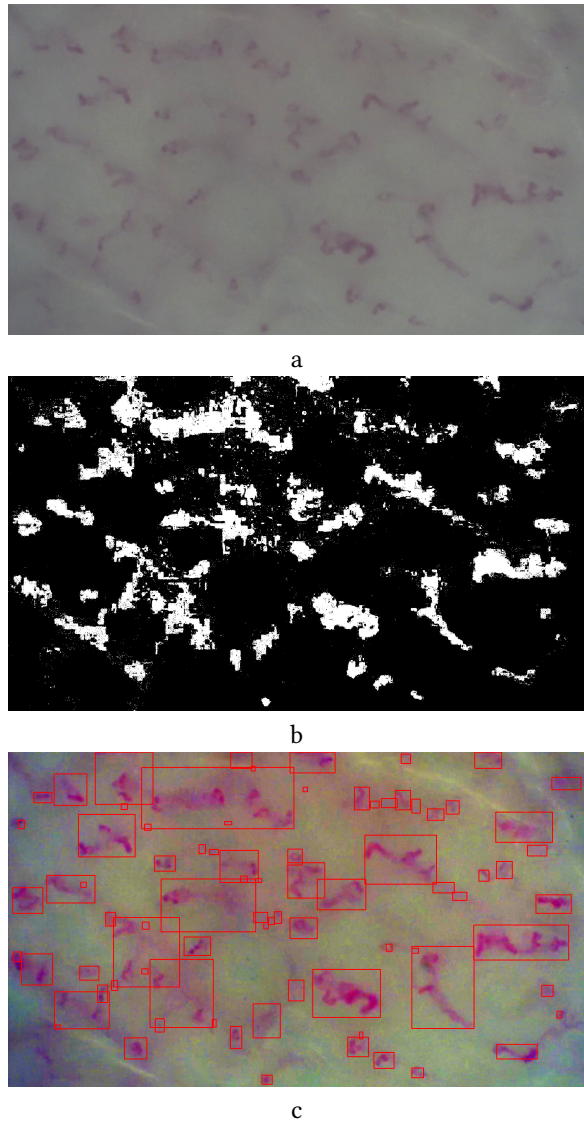


Figure 4: a) The original image input b) The image after calculating the region of interest c) the suggested RoIs on the image

To our knowledge SSIM method was used for image quality check while we are using it for the first time to detect RoIs in a capillary image.

The mean structural similarity index between the resulting image and the original image is calculated and the difference between both of these images gets a bounding box and becomes a RoI. The RoI is enhanced using pixel cumulative histogram equalization. With the combination of these 2 pipelines we are able to detect the potential capillaries in a relatively inexpensive ways in comparison to using a purely CNN based detector.

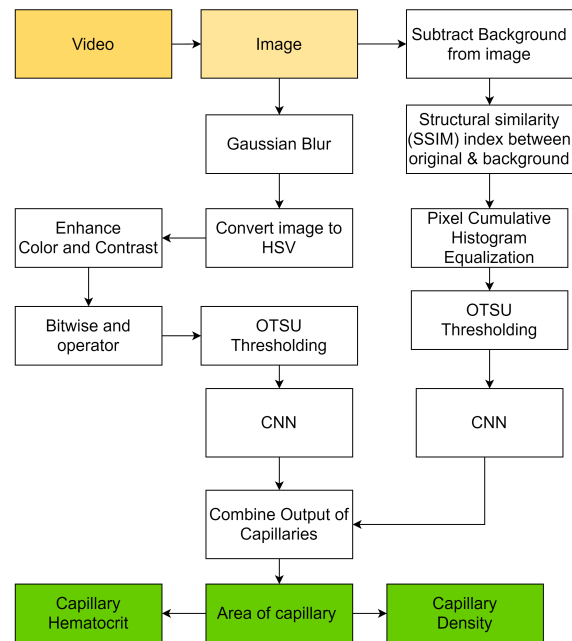


Figure 5: CapillaryNet Architecture - Capillary Detection

3.3.2 Stage 2: CNN. The second stage is passing these RoIs to a convolutional neural network part of the CapillaryNet to detect which of the RoIs are capillaries. We experimented with several architectures and our goal was to minimize the number of parameters to keep the prediction time as low as possible. Each pipeline has a CNN model. For the HSV color space pipeline, the model has a total of 6,746 parameters. The image input is 15 by 15. Our architecture consists of 1 convolutional neural networks with 16 filters followed by max pooling and a dropout layer of 0.2. The convolutional layers contain an activation function of relu [51] followed by 1 dense layer with 8 neurons and also uses an activation function relu. The architecture is shown in Figure 6. The CNN was optimized using sparse categorical cross-entropy and was trained for 20 epochs. The loss of the model was computed using cross-entropy loss between the labels and predictions.

For the second pipeline with SSIM, the model architecture consists of 2 convolutional neural networks with an activation function of relu and 32 and 64 filters respectively [51] followed by 2 dense layers with 64 and 2 neurons respectively. This is shown in Figure 7. The total number of parameters is 512,202. The CNN was optimized using Adam [52] and was trained for 30 epochs. The loss of the model was computed using cross-entropy loss between the labels and predictions.

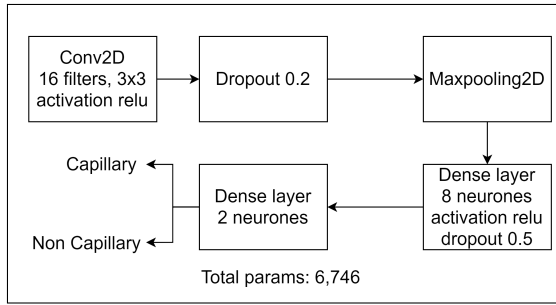


Figure 6: Convolutional Neural Network Architecture of the HSV color space pipeline- Convolutional Neural Network part of CapillaryNet

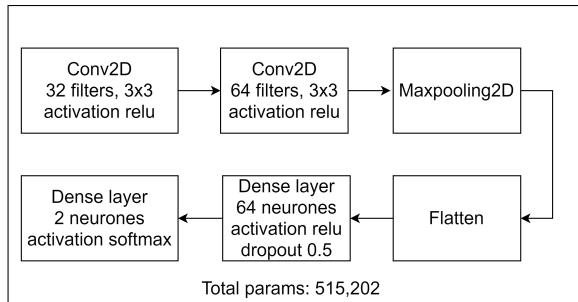
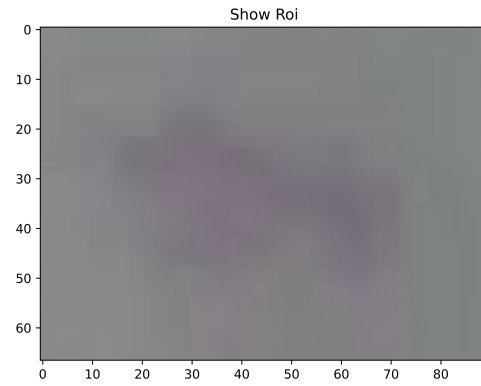


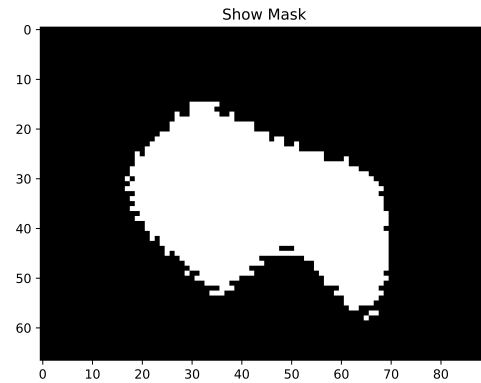
Figure 7: Convolutional Neural Network Architecture of the SSIM Pipeline- Convolutional Neural Network part of CapillaryNet

The overall accuracy of the models was 93% with 97% to the class with capillaries and 88% to the non-capillaries class. By combining the output of both of these CNNs we are able to detect more capillaries in the image.

3.3.3 Stage 3: Mask Generation. To auto-generate the mask on the capillary detected, we pass the image to a gaussian blur function which reduces the pixelation of the image. The gaussian blur has a window size of 31. The red pixels are then extracted within that image by applying an adaptive gaussian threshold function that converts all red pixels to white and the other pixels to black. The total amount of white pixels and the output of the RoI on a single capillary is shown in Figure 8a and Figure 8b respectively. The mask is then re-calculated on every consecutive frame of the video to determine the capillary density across time and the capillary hematocrit. This helps us calculate the heterogeneity of the perfusion of the capillary to monitor how much red blood cell flows through the capillary. In this way, CapillaryNet is not only able to detect the capillary and its area, but also proceed to detect how much blood flows through it across



a



b

Figure 8: a) Original image of RoI detected by CapillaryNet Architecture - deep learning part b) Area occupied by the capillary

time. This can provide the clinician with information on the average capillary hematocrit and its variation over time.

3.4 CapillaryNet Architecture - Velocity Detection, Intra-capillary Flow and Flow Direction

The capillary detected is passed to the velocity detection stage of CapillaryNet. The velocity detection method is illustrated in Figure 10. The intra-capillary heterogeneity of the flow velocity vector and direction of flow is detected using the GF algorithm [53]. The average of the intra-capillary heterogeneity of the flow velocity vector is used to deduce the velocity classification. GF is a two-frame motion estimation algorithm where the approximation is done by calculating the quadratic polynomials and polynomial expansion transform between each frame. This polynomial expansion

coefficient is used to derive the displacement fields of the pixel assuming the pixel intensities are constant between the two frames. We first apply gaussian denoising to reduce the noise algorithm to the frames; then, automatically enhance the brightness and contrast using the image histogram. We then pass these sequences of images to the GFA and obtain the difference in the location of the pixels which becomes the velocity vector. The velocity vector value determines the velocity classification. The GF method then shows the direction of flow and the intra-capillary flow between frames.

3.4.1 Image Enhancement using Pixel Cumulative Histogram Equalization. The first step is to apply a median blur with a kernel size of 5 to each channel independently. In our case, all 3 channels red, green and blue had their central pixels within the kernel window recomputed with the median value of the pixels within the kernel window. The histogram was then obtained from the modified image. The pixels in the image ranged from 0 to 255 and the histogram obtained represents the intensity distribution of the pixels. We then calculated the cumulative distribution of pixels from the histogram. Thereafter, we applied a cutoff threshold that removed the pixel values in the 10% of the cumulative distribution increasing the overall brightness and contrast.

3.4.2 Image Denoising using Non-local Means Denoising Algorithm. The median blur uses a kernel window to smoothen the image which applies local denoising to the pixels. However, if we consider noise as a variable on the pixels, this variable is not necessarily distributed equally across all the pixels. Thus, some places in the image might need more denoising in relation to other pixels in the same image. With the non-local mean, we take the noise variable as a gaussian distribution and remove it. This is achieved by converting the image to a CIELAB colorspace and then separately denoising the L and AB components [54].

3.4.3 GF Algorithm. This algorithm is based on 2 assumptions. The first assumption is that in 2 consecutive frames, the objects have the same visual values, only that one object may move in a certain direction. The second assumption is that the visual values locally around a pixel can be described by a quadratic function $I(x, y, t)$. It is mentioned in [53] that this quadratic function can be efficiently implemented in practice using a hierarchical scheme of separable convolutions technique.

Now assume that the object moves from the pixel (x,y) in a direction (dx,dy) in a small time dt . Hence, by the first assumption, we get an equality as shown in equation (20). Applying taylor series approximation of right-hand side [55] we get a system of linear equations in variables dx, dy, dt (we can choose $dt=1$), which can then be solved. An example image is shown in Figure 9.

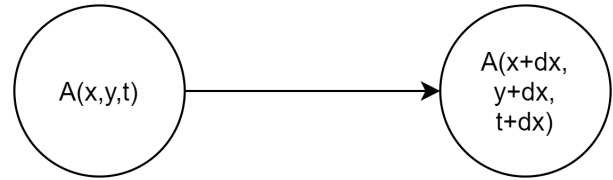


Figure 9: example of how pixel displacement is calculated. For a pixel at position x,y at point t . When displaced, the new position difference will be at dx, dy at point dt

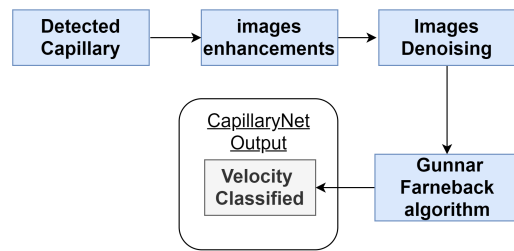


Figure 10: CapillaryNet Architecture - Velocity Detection Flow

Table 1: Benchmark of CapillaryNet capillary detection against manual analysis performed by a trained researcher on a Microcirculation video.

Name	Detection Time	Accuracy
Researcher	~120s	~84%
CapillaryNet	~0.9s	~93%

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{20}$$

For CapillaryNet we use a 2 level pyramid with a scale of 0.9. We use a window size of 10 and apply 10 iterations at each pyramid level. We use a polynomial size of 5 and a polynomial sigma of 1.2. The Cartesian coordinates are then converted to polar coordinates, essentially calculating the magnitude and angle of the 2D vector produced by the GF algorithm. The magnitude represents the velocity vector while the angle represents the direction of the flow.

4 RESULTS AND DISCUSSION

The total amount of capillaries labeled was ~11,000. To our knowledge, this is the largest dataset used for training an algorithm on skin capillaries. This data was divided into a ~70% training set and ~30% validation set for the bounding boxes. Moreover, 12 new microcirculation videos with ~1000 capillaries were acquired as

Table 2: An overview of different object detection algorithms with their size and number of parameters to tune. We also show the training time on the capillary dataset given the same dataset for capillary detection

Model Name	Size (MB)	Param (Mil)	training time
Mask RCNN	~256	~6.3	~days
DenseNet201	~7947	~18.28	~weeks
ResNet152	~9097	~58.34	~weeks
CapillaryNet	~10	~0.6	~minutes

an independent test set. The independent dataset of 12 videos was labeled by a trained researcher A. The algorithm and the trained researcher B scored ~93% and ~84% respectively. Furthermore, CapillaryNet is able to analyze 30 frames in ~4 seconds², which is the equivalent of ~0.9s per frame, while an analyst spends on average 2 minutes on the same frame to highlight the capillaries and calculate the capillary density in the image. Table 1 summarizes the main differences in capillary detection between a trained researcher and CapillaryNet.

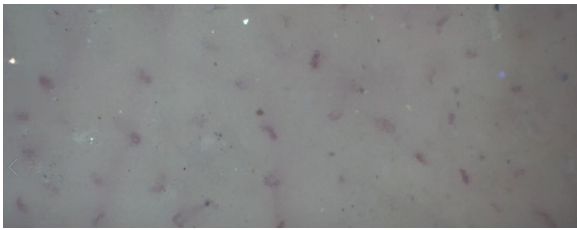


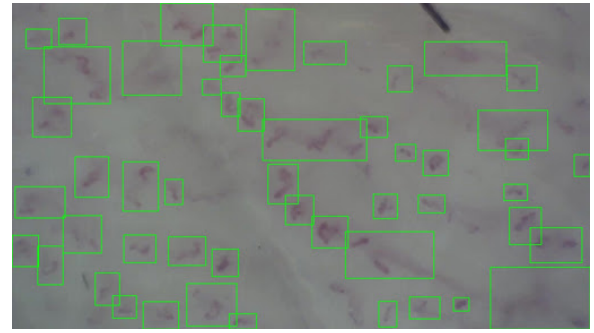
Figure 11: An image of the hand dorsum capillaries showing dirt and microscope artifacts. We expect that using purely rule-based algorithms to detect capillaries will lead to the detection of the artifacts due to similarities in sizes. On the other hand, by combining salient detection methods with convolutional neural networks we can distinguish between capillaries and artifacts.

4.1 Capillary Detection: CapillaryNet vs Object Detection Algorithms

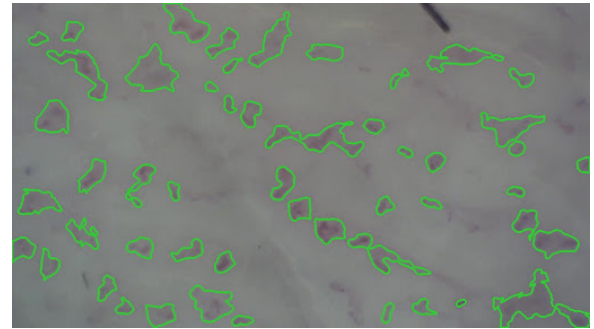
Object detection in computer vision is one of the most challenging problems where the aim is to find the area occupied by an object instance in an image [56]. Deep learning is one of the powerful tools which significantly increased the mean average precision in object detection competitions (i.e., VOC 2007-2012, ILSVRC 2013-2017) since its arrival in 2012 [57, 58]. Before deep learning, salient detection (SD) methods were the state of the art in object detection [59–61].

Object detection algorithms have millions of parameters to tune to optimize for the dataset [56]. For example, Mask R-CNN algorithm which is one of the top-ranked object detection algorithms

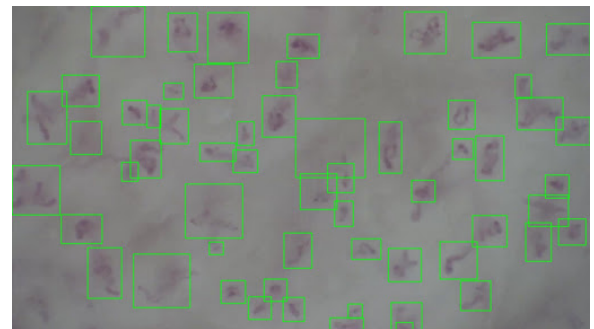
²Tested on AMD Ryzen 9 5950X with 16 processors



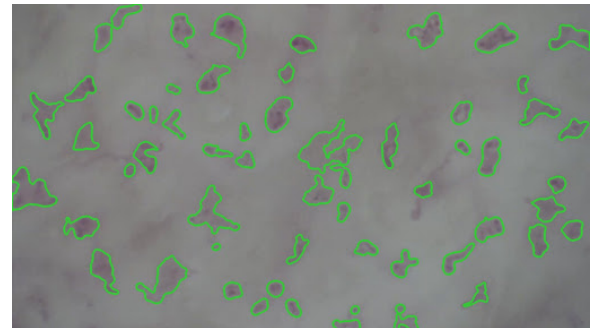
a



b



c



d

Figure 12: CapillaryNet applied to the image to detect capillaries. Areas with capillaries are marked in green. The bounding boxes on the videos are shown in 12a and 12c with green boxes highlighted around the capillaries. 12b and 12d shows the corresponding mask applied.

achieved state-of-the-art results on the COCO dataset (80 object categories with 1.5 million object instances) for object segmentation [56, 62]. Mask R-CNN architecture extends Faster R-CNN [63] by adding a branch to predict the segmentation mask of each Region of Interest (RoI) on a pixel-to-pixel basis. However, Mask RCNN has 6.3 million parameters to tune and our experiments show that it takes approximately 5 days to train given the same dataset and hardware as CapillaryNet. The power consumption to train such models is relatively high and when deployed demands disk space and ram from medical devices which can be relatively challenging.

CapillaryNet has a training time of ~ 5 seconds per epoch³ as opposed to several days which is due to the fact that the CapillaryNet bounding box detection part has ~ 0.6 million parameters which is $\sim 10\%$ of Mask R-CNN size. This makes the model feasible to run in a clinical situation without the need to upgrade to powerful or energy-hungry computers. With $\sim 10\%$ of the size of Mask R-CNN, our model can be considered more sustainable in terms of power consumption. The main reason why CapillaryNet is $\sim 10\%$ the size of Mask R-CNN and relatively much smaller than the other object detection algorithms is due to the shallower convolutional neural network (CNN) used. A shallower CNN also makes the model more versatile because we can re-train and improve the model when new data comes in with relatively fewer delays (seconds for training instead of weeks for larger CNNs). Therefore, CapillaryNet is a much smaller, faster, and more flexible production-ready capillary detection model that is optimized to address the issue for microcirculation analysis in clinical settings. Table 2 shows some commonly used models and the number of parameters to tune as compared with CapillaryNet and Mask RCNN [64].

With 1000x+ the training speed, 100x detection time and 0.12% of the size of Mask RCNN we can achieve an equivalent accuracy to that of state-of-the-art object detection.

Furthermore, from our experiments, we hypothesize that the CNN RoI-based detectors such as the one present in Mask R-CNN do not generalize well with new capillary data. This is due to varied illumination on different parts of the image, blur due to the size of a capillary relative to the image, or occlusion due to hair, stains and other artifacts on the skin as shown in Figure 11. This in turn makes it challenging for the Mask R-CNN and any other CNN RoI-based detector to generalize with accuracy equivalent to the trained researcher. The CapillaryNet architecture tackles these challenges posed by the profile of the skin by using the HSV color space to detect RoIs instead of pure CNNs. Moreover, these RoI detection methods are computationally less expensive in comparison to the Mask R-CNN RoI detection step, making CapillaryNet faster.

The output of the CapillaryNet bounding boxes is shown in Figure 12. Figures 12a shows the bounding box detected by CapillaryNet and 12b highlights the area occupied by that capillary. Similarly, 12c shows another image obtained from a handheld microscope and 12d shows the area occupied by those capillaries. With the lines finely wrapped around the capillaries within the bounding box, we can calculate with high accuracy the area occupied by the capillary and thus the capillary density.

³Tested on a Nvidia GeForce RTX 3090

4.2 Capillary Detection: CapillaryNet vs Capillary Detection Methods in the Literature

CapillaryNet compensates for the unequal illumination challenge highlighted in the related work section in 2 ways. First, we use an HSV color space that is robust to such challenges instead of the RGB color space and second our training dataset consisted of out-of-focus capillaries captured on real data to make it capable of detecting them. Furthermore, CapillaryNet reduces the observer variation factor by attaching a CNN to classify the RoIs. This CNN is trained on a validated capillary dataset labeled by trained researchers.

Moreover, several methods highlighted the need for taking the mean pixel value of several frames. In contrast, in this paper, by using HSV color space combined with a CNN network CapillaryNet is able to detect capillaries from a single frame. This way we eliminate the need to take a mean value across the whole video and treat each frame uniquely with its values. This also brings us closer to real-time detection since we do not need to wait for the whole video to be recorded to get the mean value, but rather we can start detecting capillaries from a single independent frame.

4.3 Capillary Hematocrit

Detecting capillary hematocrit can provide the clinician information on the potential of each capillary to deliver oxygen to the surrounding tissue. To our knowledge, only flow velocity and capillary density were assessed in previous studies. However, if capillaries have a normal flow and are normally distributed, but show a low concentration of red blood cells (i.e., low hematocrit) the oxygen delivery ability of the microcirculation may be compromised. Clinical studies will need to be conducted to elucidate the potential of using capillary hematocrit as a marker for abnormal microcirculation and our method provides a good tool to do so. Figure 13a shows the capillary filled with red blood cells, Figures 13b and 13c shows the same capillary a couple of seconds later with white blood cells and plasma gaps. Figure 14 shows the red blood cell distribution of this capillary derived using CapillaryNet.

4.4 Velocity Detection

The quantification of red blood cell flow is a more challenging task than capillary detection. Some papers base their red blood cell flow on manual quantification of the red blood cell with different scales [65–68] which is subject to intra-individual variation. In manual quantification, each individual vessel receives a score representing the average flow velocity estimated by a researcher. Flow velocity scales vary between publications; some researchers classifying flow on a scale from 0 to 2 (absent, intermittent or continuous flow) [69], others on a scale between 0 to 3 (absent, intermittent, sluggish or normal flow) [69], while others on a scale from 0 to 5 (no-flow, sluggish, continuous very low, continuous low, continuous high, or brisk flow) [70].

More recent papers use space-time diagrams [25, 27, 71] to quantify red blood cell flow. A space-time diagram (STD) is a fundamental improvement over the manual eye analysis since it is independent of the individual performing the analysis [24], [67]. However, it comes

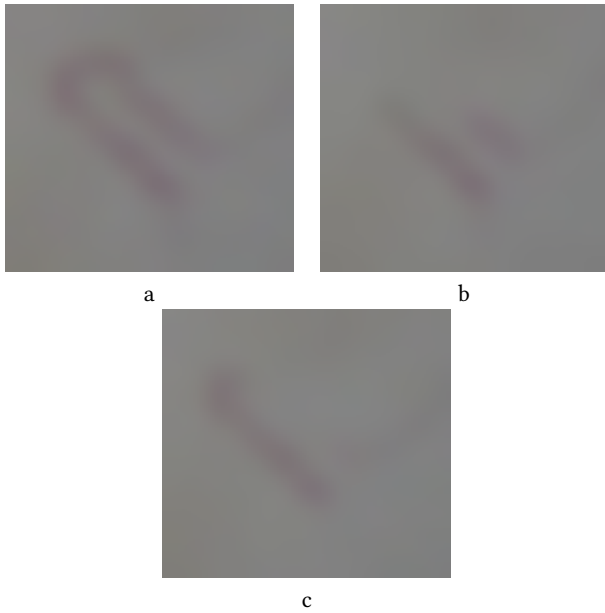


Figure 13: a) A capillary filled with red blood cells b) and c) A capillary partially filled with plasma gaps or white blood cells

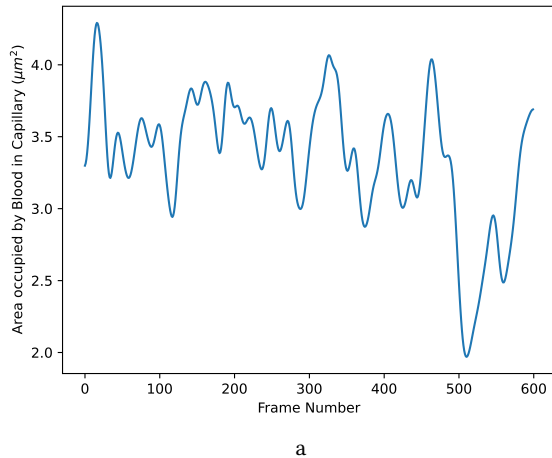


Figure 14: Graph showing how much a capillary is filled with red blood cells (capillary hematocrit) across time

with its drawback. It is very sensitive to the slightest movements since it strictly counts on the central line being at the center of the capillary between all the frames to have an accurate space-time diagram. Therefore, if the position or width of the capillary changes between the frames due to a camera shake or flow variation, the user must re-calibrate the central line to plot an accurate diagram.

This might add an extra task to velocity classification which can be prone to errors and user bias. Furthermore, to construct an accurate central line we are dependent on identifying the exact width and length of the capillaries in the earlier stage. Therefore, capillaries that get out of focus have to be disregarded since fitting the central line to plot the STD will not be accurate. Moreover, space-time diagrams need to go through the whole video to calculate the velocity and rely on gaps in the flow to deduce the flow.

CapillaryNet overcomes the limitation of needing the whole video by utilizing the Gunnar Farneback (GF) algorithm [53].

Thus, our velocity detection method can be considered as an alternative way to the space-time diagram and manual eye analysis which is considered as the gold standard for red blood cells (rbc) velocity classification [25]. We do not need the central line to deduce the velocity therefore we reduce the steps needed to detect velocity. Furthermore, we use the GF algorithm to deduce the intra-capillary flow heterogeneity. This makes our method more accurate in tracking red blood cells since we do not singly rely on a central line accurately placed between the frames.

For videos with high resolution (1920x1080) and 30FPS, the GF algorithm was used to classify the velocity. A sequence of 240 frames (8 seconds) was classified for its velocity by a trained researcher. The velocity of about 2250 capillaries was classified in one of the following categories: no flow, slow flow, normal flow and fast flow. For each capillary classified by an analyst, the GF algorithm provided a velocity vector value. Figure 15 displays the analyst classification (on the x axis) compared to the velocity vector value obtained by the algorithm (y axis). The velocity vectors were significantly different for each velocity class (the p-value between two classes was below 0.00001). The average velocity vector for no flow was 0.3, slow flow was 0.7, normal flow was 0.9 and fast flow was 1.5, as illustrated in Figure 15.

There was an overlap in velocity vector values between the velocity class no flow and slow flow, and between the velocity class normal flow and fast flow. This points to the lack of a precise enough definition in manual analysis of where the boundaries between velocity categories are. For instance, analysts may classify sometimes capillary with a “fast-normal” flow (borderline cases) into the normal flow category, while other times into the fast flow category. Hubble et. al have shown that inter-analyst variability can be up to 26% in assessing the flow in capillaries of healthy controls [72]. The accuracy further decreases between an experienced and a newly trained analyst [73]. This lack of accuracy for “borderline cases” creates inconsistent training data for the algorithm and results in overlapping velocity vector values between categories.

The capillaries classified by the analysts as having fast flow had a much broader spread in velocity vectors compared to the other velocity classes as shown by the fast flow on the x-axis in Figure 15. This indicates that the fast flow category could be potentially split into two categories (e.g., fast and very fast flow) by the algorithm. This new classification may bring new insights in understanding how capillary flow velocity changes in the course of various diseases.

By training the algorithm on the classification of one analyst and by refining the velocity vector boundaries between categories, we create a standardized method that has the potential to be more

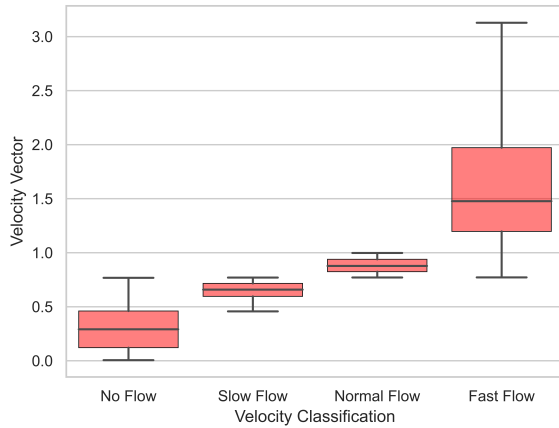


Figure 15: CapillaryNet: Velocity vector values of 2,255 capillary videos classified by GF Algorithm part in CapillaryNet

consistent in classifying capillary velocities. This reduces the bias of velocity of classification between groups of analysts and is not influenced by parameters such as analyst level of experience or how focused/distracted the analyst is on the day of analysis. Furthermore, the GF can detect the flow in near real-time with almost 15FPS (~0.07s per bounding box), while an analyst spends on average 1 minute to label the speed of one capillary, and on average 20 minutes to annotate one video.

4.5 Using Deep Neural Networks, instead of GF, for Velocity Classification: some experiments

We performed experiments to utilize a deep neural network for video classification for data with lower resolution (1280x720 instead of 1920x1080) and fps (5fps instead of 30fps). Classifying on 1920x1080 and 30FPS will create a model that is not plausible for usage in a clinical environment in terms of training and classification. We experimented with 4 different deep neural architectures to train the velocity classification model. We developed and trained 4 different deep neural architectures to test if they can challenge the GF algorithm and experimented with all sequences in increments of 10 between 10 frames to 240 frames.

The first architecture was classifying a sequence of frames. For this, we used InceptionResNetV2 [74] and the concept of transfer learning. The top 9 layers were fine-tuned on the 4 classes (no flow, slow flow, normal velocity flow, fast flow) while the rest of the weights were used from the ImageNet dataset [75].

The output was then passed to a convolutional 2D network. The second architecture was based on a time-distributed CNN and passing the features of the CNN to a recurrent neural network (RNN). For the RNN network, we experimented both with Long short-term memory (LSTM) and Gated recurrent unit (GRU). We demonstrate the architecture of this method in Figure 16. The third architecture

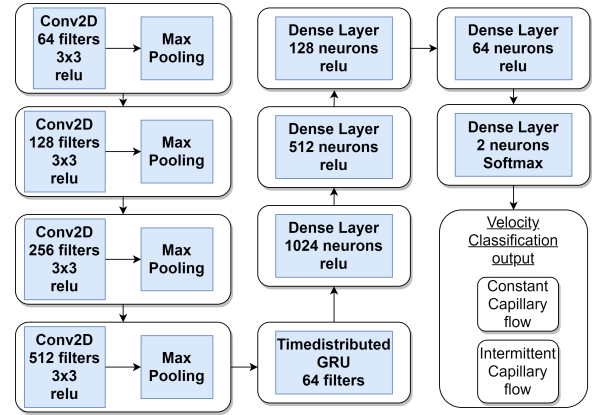


Figure 16: An example of a deep neural architecture built to test for velocity classification

was using a 3D convolutional network. The fourth architecture is similar to the first architecture where transfer learning of Inception-ResNetV2 was utilized but the output was passed to a convolutional 3D.

All the above architectures performed poorly with accuracies of less than 50% for each class. Moreover, they had several million parameters to train, consuming more energy and drastically increasing training time compared to GF (which performed it in near real-time). Using deep neural networks for velocity classification in capillaries is complex and energy-consuming. From our experiments, we deduce that using deep neural networks for video classification for such types of data remains a challenging task to be solved.

4.6 Intra-capillary Flow Velocity Heterogeneity and flow direction

Flow velocity within a capillary has been calculated in previous studies as an average value. Our unpublished empirical observations point that healthy subjects have a relatively homogeneous flow velocity, while in certain patient groups the flow velocity varies significantly throughout the 20 seconds of the video. Obtaining one average value for each capillary does not allow the clinicians to detect this difference. Therefore, measuring the intra-capillary flow velocity heterogeneity has the potential to be used as an additional marker to detect compromised microcirculation. CapillaryNet utilizes Gunnar Farneback (GF) algorithm which tracks moving pixels which in this case represents the red blood cells. Therefore, if a capillary is surrounded by the bounding box, we can calculate the pixel displacement of red blood cells. Thus, by combining deep neural networks to classify speed and GF, our velocity detection method can be considered as an alternative way to the space-time diagram and manual eye analysis which is considered as the gold standard for red blood cell velocity [25].

We show an example of the velocity vector of the GFA on a publicly available capillary video [76] in Figure 17a. We take measurements from three different regions of the video to show that the GF algorithm can detect the movement of red blood cells within

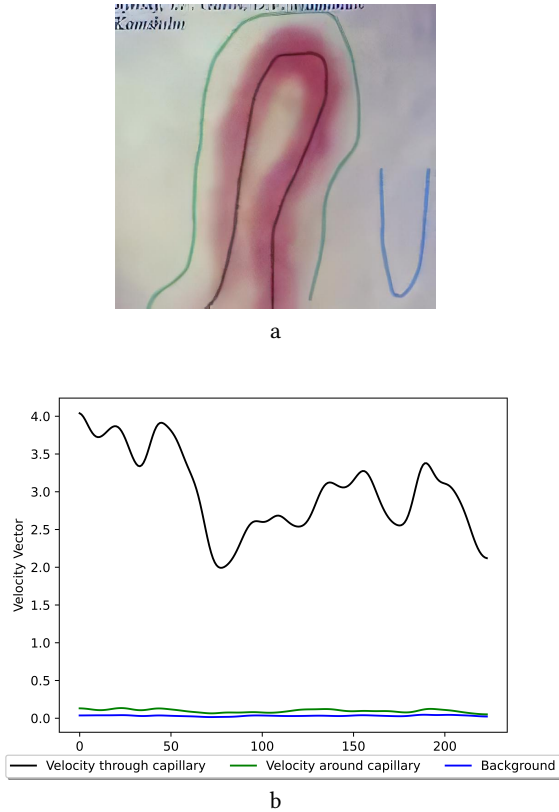


Figure 17: a) CapillaryNet applied to a capillary video publicly available to assess capillary flow direction and heterogeneity. We have plotted 3 lines in the video where we are interested to detect the velocity vectors. The black line is approximately in the center of the capillary. The green line goes around the capillary. The blue line is randomly placed next to the capillary. These lines are visible in the image. b) Results of the velocity vector across the frames of the black line which is the capillary center (average velocity vector value was 34.9), a green line which is located on the outside of the capillary (average velocity vector value was 0.1), a blue line which is randomly placed at the side of the capillary (average velocity vector value was 0.7)

the capillary. We present the results on a graph in Figure 17b. We observe that the highest velocity vector values were found at the center of the capillary which is highlighted with the black line where red blood cells are flowing through the capillary. The lowest velocity vector was found on the side of the capillary highlighted with the blue line where there is no flow. Figure 18 highlights the flow of the red blood cells within the capillary.

Table 3: We describe how CapillaryNet can calculate and derive the applicable parameters suggested for microcirculation analysis by Hilty et al [25] and Ince et al [23]. In addition we introduce two new parameters that can be uniquely identified and calculated by CapillaryNet: Intra-capillary heterogeneity of flow velocity and capillary hematocrit.

Parameter	CapillaryNet Detection Description
per image or video	
Total Vessel Density for Capillaries	The sum of the area occupied by capillaries is derived by the total number of pixels occupied by the detected capillary divided by the dimension of the image
Functional Capillary Density	Sum of the area occupied by capillaries that contain moving red blood cells which are derived by the total number of pixels occupied by the detected capillary divided by the dimension of the
per vessel	
Flow Velocity	The velocity of red blood cells flow classified as slow flow, normal velocity flow and fast flow
Intra-capillary heterogeneity of flow velocity	Derived from GF algorithm where we plot the variation in pixel movement across all frames
Capillary Hematocrit	Derived from a rule-based algorithm where we plot the number of red blood cells across the frames

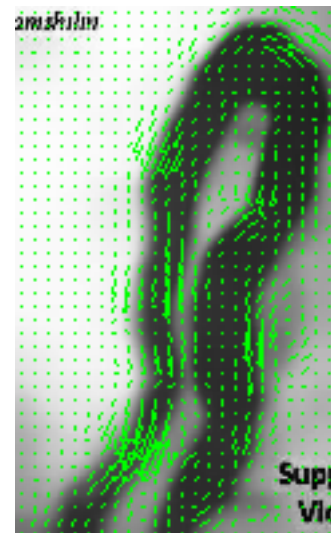


Figure 18: Flow direction of capillary automatically derived by CapillaryNet

4.7 CapillaryNet Parameters

In Table 3 we show how CapillaryNet can calculate and derive the microvascular parameters suggested by Hilty [25] and Ince [23]. In addition, we introduce 2 new parameters that have not been previously monitored in microvascular videos: capillary hematocrit and intra-capillary flow velocity heterogeneity. CapillaryNet is a unique architecture that combines deep neural networks with salient object detection and two-frame motion estimation techniques to detect capillaries and estimate the velocity of red blood cells. Our architecture paves the way to a unified automated method for near real-time bedside analysis of microcirculation.

5 CONCLUSION

In this paper, we are not only pushing forward the state of the art in capillary detection by presenting an automated architecture for microcirculation analysis but also increasing the number of microvascular parameters that can be monitored beyond what is currently possible today. This can provide novel information on the potential of each capillary to deliver oxygen to the surrounding tissue. Moreover, we present a method that can standardize velocity classification eliminating intra-variability between researchers and reducing the time taken for microcirculation analysis. In this study CapillaryNet has been used to assess skin microcirculation. CapillaryNet can automatically quantify the area occupied by a capillary, calculate the capillary density, derive the average flow velocity, derive the intra-capillary flow velocity heterogeneity, and quantify the capillary hematocrit. With a unified architecture that can detect capillaries at ~ 0.9 seconds per frame with $\sim 93\%$ accuracy, we are closer to real-time microcirculation analysis in a clinical environment than ever.

6 COMPETING INTERESTS

The authors declare the following financial interests/personal interests which may be considered as potential competing interests:

The Research Council of Norway provided 50% of the funds for Maged's Industrial PhD (First Author - Industrial Ph.D. project nr: 305716.). His other 50% is funded by ODI Medical AS.

Anastasiya Dykyy is 100% funded by ODI Medical AS under BIA project nr: 282213.

ODI Medical AS has provided the hardware, funds and expertise to collect and analyze the medical data.

7 RESEARCH FUNDING

The Research Council of Norway provided the necessary funds for this project: Industrial Ph.D. project nr: 305716 and BIA project nr: 282213.

8 CODE

The code will be available for verification and validation purposes on Github. The link will be added soon.

ACKNOWLEDGMENTS

We would like to acknowledge Stein Bleivik for his continuous support of this project.

REFERENCES

- [1] Daniel De Backer, Steven Hollenberg, Christiaan Boerma, Peter Goedhart, Gustavo Büchele, Gustavo Ospina-Tascon, Iwan Dobbe, and Can Ince. How to evaluate the microcirculation: report of a round table conference. *Crit. Care*, 11(5):R101, 2007.
- [2] Arefa Cassoobhoy. What is the definition of capillaries?, 2020.
- [3] A C Shore. Capillaroscopy and the measurement of capillary pressure. *Br. J. Clin. Pharmacol.*, 50(6):501–513, December 2000.
- [4] Goksel Guven, Matthias P Hilty, and Can Ince. Microcirculation: Physiology, pathophysiology, and clinical application. *Blood Purif.*, 49(1-2):143–150, 2020.
- [5] Matthew J.S. Parker and Neil W. McGill. The established and evolving role of nailfold capillaroscopy in Connective-Tissue disease. In *Connective Tissue Disease - Current State of the Art [Working Title]*. IntechOpen, 2019.
- [6] V Nama, J Onwude, I T Manyonda, and T F Antonios. Is capillary rarefaction an independent risk marker for cardiovascular disease in south asians? *J. Hum. Hypertens.*, 25(7):465–466, July 2011.
- [7] Alfons J H M Houben, Remy J H Martens, and Coen D A Stehouwer. Assessing microvascular function in humans from a chronic disease perspective. *J. Am. Soc. Nephrol.*, 28(12):3461–3472, December 2017.
- [8] Jurgen C de Graaff, Dirk Th Ubink, Joost A van der Spruit, Sjoerd M Lagarde, and Michael J H M Jacobs. Influence of peripheral arterial disease on capillary pressure in the foot. *J. Vasc. Surg.*, 38(5):1067–1074, November 2003.
- [9] B Fagrell and M Intaglietta. Microcirculation: its significance in clinical and molecular medicine. *J. Intern. Med.*, 241(5):349–362, May 1997.
- [10] P M Houtman, C G Kallenberg, A A Wouda, and T H The. Decreased nailfold capillary density in raynaud's phenomenon: a reflection of immunologically mediated local and systemic vascular disease? *Ann. Rheum. Dis.*, 44(9):603–609, September 1985.
- [11] Heinrike Schmeling, Samantha Stephens, Cristina Goia, Cedric Manliot, Rayfel Schneider, Sanjeev Luthra, Elizabeth Stringer, and Brian M Feldman. Nailfold capillary density is importantly associated over time with muscle and skin disease activity in juvenile dermatomyositis. *Rheumatology*, 50(5):885–893, May 2011.
- [12] B D Duscha, W E Kraus, S J Keteyian, M J Sullivan, H J Green, F H Schachat, A M Pippen, C A Brawner, J M Blank, and B H Annex. Capillary density of skeletal muscle: a contributing mechanism for exercise intolerance in class II-III chronic heart failure independent of other peripheral alterations. *J. Am. Coll. Cardiol.*, 33(7):1956–1963, June 1999.
- [13] Jennifer L Robbins, W Schuyler Jones, Brian D Duscha, Jason D Allen, William E Kraus, Judith G Regensteiner, William R Hiatt, and Brian H Annex. Relationship between leg muscle capillary density and peak hyperemic blood flow with endurance capacity in peripheral artery disease. *J. Appl. Physiol.*, 111(1):81–86, July 2011.
- [14] Mohammad Moeini, Xuecong Lu, Pramod K Avti, Rafat Damseh, Samuel Bélanger, Frédéric Picard, David Boas, Ashok Kakkar, and Frédéric Lesage. Compromised microvascular oxygen delivery increases brain tissue vulnerability with age. *Sci. Rep.*, 8(1):8219, May 2018.
- [15] Alejandra López, Juan Carlos Grignola, Martín Angulo, Ignacio Alvez, Nicolás Nin, Gonzalo Lacuesta, Manuel Baz, Pablo Cardinal, Ivana Prestes, Juan P Bouchacourt, Juan Riva, Can Ince, and Francisco Javier Hurtado. Effects of early hemodynamic resuscitation on left ventricular performance and microcirculatory function during endotoxic shock. *Intensive Care Med Exp*, 3(1):49, December 2015.
- [16] Daniel De Backer, Jacques Creteur, Jean-Charles Preiser, Marc-Jacques Dubois, and Jean-Louis Vincent. Microvascular blood flow is altered in patients with sepsis. *Am. J. Respir. Crit. Care Med.*, 166(1):98–104,

- July 2002.
- [17] Torjus Wester, Zaheer Ahmed Awan, Tobias Skylstad Kvernebo, Göran Salerud, and Knut Kvernebo. Skin microvascular morphology and hemodynamics during treatment with veno-arterial extra-corporeal membrane oxygenation. *Clin. Hemorheol. Microcirc.*, 56(2):119–131, 2014.
- [18] Michael S Goligorsky. Microvascular rarefaction: the decline and fall of blood vessels. *Organogenesis*, 6(1):1–10, January 2010.
- [19] Alcía Edwards-Richards, Marissa DeFreitas, Chryso P Katsoufis, Wacharee Seeherunvong, Nao Sasaki, Michael Freundlich, Gaston Zilleruelo, and Carolyn L Abitbol. Capillary rarefaction: an early marker of microvascular disease in young hemodialysis patients. *Clin. Kidney J.*, 7(6):569–574, December 2014.
- [20] C C Michel. Starling: the formulation of his hypothesis of microvascular fluid exchange and its significance after 100 years. *Exp. Physiol.*, 82(1):1–30, January 1997.
- [21] Helen Shinru Wei, Hongyi Kang, Izad-Yar Daniel Rasheed, Sitong Zhou, Nanhong Lou, Anna Gershteyn, Evan Daniel McConnell, Yixuan Wang, Kristopher Emil Richardson, Andre Francis Palmer, Chris Xu, Jiandi Wan, and Maiken Nedergaard. Erythrocytes are Oxygen-Sensing regulators of the cerebral microcirculation. *Neuron*, 91(4):851–862, August 2016.
- [22] Matthias Peter Hilty, Philippe Guerci, Yasin Ince, Fevzi Toraman, and Can Ince. MicroTools enables automated quantification of capillary density and red blood cell velocity in handheld vital microscopy. *Commun Biol*, 2:217, June 2019.
- [23] Can Ince, E Christiaan Boerma, Maurizio Cecconi, Daniel De Backer, Nathan I Shapiro, Jacques Duranteau, Michael R Pinsky, Antonio Artigas, Jean-Louis Teboul, Irwin KM Reiss, et al. Second consensus on the assessment of sublingual microcirculation in critically ill patients: results from a task force of the european society of intensive care medicine. *Intensive care medicine*, 44(3):281–299, 2018.
- [24] Johannes GG Dobbe, Geert J Streekstra, Bektas Ataserver, R Van Zijderveld, and Can Ince. Measurement of functional microcirculatory geometry and velocity distributions using automated image analysis. *Medical & biological engineering & computing*, 46(7):659–670, 2008.
- [25] Matthias Peter Hilty, Philippe Guerci, Yasin Ince, Fevzi Toraman, and Can Ince. Microtools enables automated quantification of capillary density and red blood cell velocity in handheld vital microscopy. *Communications biology*, 2(1):1–15, 2019.
- [26] Hongli Deng, Wei Zhang, Eric Mortensen, Thomas Dietterich, and Linda Shapiro. Principal curvature-based region detector for object recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [27] Rick Bezemer, Johannes G Dobbe, Sebastiaan A Bartels, E Christiaan Boerma, Paul WG Elbers, Michal Heger, and Can Ince. Rapid automatic assessment of microvascular density in sidestream dark field images. *Medical & biological engineering & computing*, 49(11):1269–1278, 2011.
- [28] Johnny Tam, Joy A Martin, and Austin Roorda. Noninvasive visualization and analysis of parafoveal capillaries in humans. *Investigative ophthalmology & visual science*, 51(3):1691–1698, 2010.
- [29] Lawrence S Geyman, Reena A Garg, Yanin Suwan, Vivek Trivedi, Brian D Krawitz, Shelley Mo, Alexander Pinhas, Apichat Tantraworasin, Toco YP Chui, Robert Ritch, et al. Peripapillary perfused capillary density in primary open-angle glaucoma across disease stage: an optical coherence tomography angiography study. *British Journal of Ophthalmology*, 101(9):1261–1268, 2017.
- [30] Sumeyra U Demir, Roya Hakimzadeh, Rosalyn Hobson Hargraves, Kevin R Ward, Eric V Myer, and Kayvan Najarian. An automated method for analysis of microcirculation videos for accurate assessment of tissue perfusion. *BMC medical imaging*, 12(1):1–13, 2012.
- [31] Ali M Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38(1):35–44, 2004.
- [32] Cynthia Cheng, Chadd W Lee, and Constantine Daskalakis. A reproducible computerized method for quantitation of capillary density using nailfold capillaroscopy. *Journal of visualized experiments: JoVE*, (104), 2015.
- [33] Arya Tama, Tati R Mengko, and Hasballah Zakaria. Nailfold capillaroscopy image processing for morphological parameters measurement. In *2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, pages 175–179. IEEE, 2015.
- [34] Pavle Prentašić, Morgan Heisler, Zaid Mammo, Sieun Lee, Andrew Merkur, Eduardo Navajas, Mirza Faisal Beg, Marinko Šarunic, and Sven Lončarić. Segmentation of the foveal microvasculature using deep learning networks. *Journal of biomedical optics*, 21(7):075008, 2016.
- [35] Guangzheng Dai, Wei He, Ling Xu, Eric E Pazo, Tiezhu Lin, Shasha Liu, and Chenguang Zhang. Exploring the effect of hypertension on retinal microvasculature using deep learning on east asian population. *PLoS one*, 15(3):e0230111, 2020.
- [36] R Nivedha, M Brinda, KV Suma, and Bheemsain Rao. Classification of nailfold capillary images in patients with hypertension using non-linear svm. In *2016 International Conference on Circuits, Controls, Communications and Computing (I4C)*, pages 1–5. IEEE, 2016.
- [37] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.
- [38] Perikumar Javia, Aman Rana, Nathan Shapiro, and Pratik Shah. Machine learning algorithms for classification of microcirculation images from septic and non-septic patients. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 607–611. IEEE, 2018.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [40] Fei Ye, Songchao Yin, Meirong Li, Yujie Li, and Jingang Zhong. In-vivo full-field measurement of microcirculatory blood flow velocity based on intelligent object identification. *Journal of biomedical optics*, 25(1):016003, 2020.
- [41] Chengcheng Ning, Huajun Zhou, Yan Song, and Jinhui Tang. Inception single shot multibox detector for object detection. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 549–554, 2017.
- [42] Luis Barba-Guaman, Jose Eugenio Naranjo, and Anthony Ortiz. Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded gpu. *Electronics*, 9(4):589, 2020.
- [43] Yuli Sun Hariyani, Heesang Eom, and Cheolsoo Park. Da-capnet: Dual attention deep learning based on u-net for nailfold capillary segmentation. *IEEE Access*, 8:10543–10553, 2020.
- [44] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [45] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [47] Alex Clark. Pillow (pil fork) documentation. *Readthedocs*. <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>

- 2015.
- [48] Khamar Basha Shaik, P Ganesan, V Kalist, BS Sathish, and J Merlin Mary Jenitha. Comparative study of skin color detection and segmentation in hsv and ycbcr color space. *Procedia Computer Science*, 57:41–48, 2015.
- [49] Oleh Berezsky, Oleh Pitsun, Natalia Batryn, Kateryna Berezska, Nadiya Savka, and Taras Dolynyuk. Image segmentation metric-based adaptive method. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 554–557. IEEE, 2018.
- [50] Contour approximation method.
- [51] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.
- [52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [53] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.
- [54] Amanpreet Kaur and BV Kranthi. Comparison between ycbcr color space and cielab color space for skin color segmentation. *International Journal of Applied Information Systems*, 3(4):30–33, 2012.
- [55] George Corliss and YF Chang. Solving ordinary differential equations using taylor series. *ACM Transactions on Mathematical Software (TOMS)*, 8(2):114–144, 1982.
- [56] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.
- [57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [59] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [60] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.
- [61] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational visual media*, 5(2):117–150, 2019.
- [62] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.
- [64] Dnn model optimization series part i: What’s the drill?
- [65] E Christiaan Boerma, Keshen R Mathura, Peter HJ van der Voort, Peter E Spronk, and Can Ince. Quantifying bedside-derived imaging of microcirculatory abnormalities in septic patients: a prospective validation study. *Critical care*, 9(6):1–6, 2005.
- [66] Ryan C Arnold, Joseph E Parrillo, R Phillip Dellinger, Michael E Chan-sky, Nathan I Shapiro, David J Lundy, Stephen Trzeciak, and Steven M Hollenberg. Point-of-care assessment of microvascular blood flow in critically ill patients. *Intensive care medicine*, 35(10):1761–1766, 2009.
- [67] Vanina S Kanoore Edul, Carolina Enrico, Bruno Laviolle, Alejandro Risso Vazquez, Can Ince, and Arnaldo Dubin. Quantitative assessment of the microcirculation in healthy volunteers and in patients with septic shock. *Critical care medicine*, 40(5):1443–1448, 2012.
- [68] Arnaldo Dubin, Mario O Pozo, Christian A Casabella, Fernando Pálizas, Gastón Murias, Miriam C Moseinco, Vanina S Kanoore Edul, Elisa Estenssoro, and Can Ince. Increasing arterial blood pressure with norepinephrine does not improve microcirculatory blood flow: a prospective study. *Critical Care*, 13(3):1–8, 2009.
- [69] Daniel De Backer, Steven Hollenberg, Christiaan Boerma, Peter Goedhart, Gustavo Büchele, Gustavo Ospina-Tascon, Iwan Dobbe, and Can Ince. How to evaluate the microcirculation: report of a round table conference. *Critical care*, 11(5):1–9, 2007.
- [70] Siv Fredly, Drude Fugelseth, Cathrine S Nygaard, E Göran Salerud, Tom Stiris, and Knut Kvernebo. Noninvasive assessments of oxygen delivery from the microcirculation to skin in hypothermia-treated asphyxiated newborn infants. *Pediatric research*, 79(6):902–906, 2016.
- [71] Matthias Peter Hilty, Jacqueline Pichler, Bulent Ergin, Urs Hefti, Tobias Michael Merz, Can Ince, and Marco Maggiorini. Assessment of endothelial cell function and physiological microcirculatory reserve by video microscopy using a topical acetylcholine and nitroglycerin challenge. *Intensive care medicine experimental*, 5(1):1–13, 2017.
- [72] Sheena MA Hubble, Hayley L Kyte, Kim Gooding, and Angela C Shore. Variability in sublingual microvessel density and flow measurements in healthy volunteers. *Microcirculation*, 16(2):183–191, 2009.
- [73] Siv Fredly, Drude Fugelseth, Torjus Wester, Erik Häggblad, and Knut Kvernebo. Skin microcirculation in healthy term newborn infants—assessment of morphology, perfusion and oxygenation. *Clinical hemorheology and microcirculation*, 59(4):309–322, 2015.
- [74] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [75] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [76] Mikhail V Volkov, Nikita B Margaryants, Andrey V Potemkin, Maxim A Volynsky, Igor P Gurov, Oleg V Mamontov, and Alexei A Kamshilin. Video capillaroscopy clarifies mechanism of the photoplethysmographic waveform appearance. *Scientific reports*, 7(1):1–8, 2017.