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

Maged Abdalla Helmy\textsuperscript{a}, Tuyen Trung Truong\textsuperscript{b}, Anastasiya Dykky\textsuperscript{c}, Paulo Ferreira\textsuperscript{a}, Eric Jul\textsuperscript{a}

\textsuperscript{a}Department of Informatics, University of Oslo, Gaustadalleen 21, magedaa,paulofe,ericbj@uio.no Oslo, 0373, Norway
\textsuperscript{b}Department of Mathematics, University of Oslo, Gaustadalleen 21, tuyentt@math.uio.no Oslo, 0373, Norway
\textsuperscript{c}ODI Medical AS, Gaustadalleen 23, anastasiya.dykky@odimedical.com Oslo, 0373, Norway

Abstract

Capillaries are the smallest vessels in the body responsible for delivering oxygen and nutrients to surrounding cells. Various life-threatening diseases are known to alter the density of healthy capillaries and the flow velocity of erythrocytes within the capillaries. In previous studies, capillary density and flow velocity were manually assessed by trained specialists. However, manual analysis of a standard 20-second microvascular video requires 20 minutes on average and necessitates extensive training. Thus, manual analysis has been reported to hinder the application of microvascular microscopy in a clinical environment. To address this problem, this paper presents a fully automated state-of-the-art system to quantify skin nutritive capillary density and red blood cell velocity captured by handheld-based microscopy videos. The proposed method combines the speed of traditional computer vision algorithms with the accuracy of convolutional neural networks to enable clinical capillary analysis. The results show that the proposed system fully automates capillary detection with an accuracy exceeding that of trained analysts and measures several novel microvascular parameters that had eluded quantification thus far, namely, capillary hematocrit and intracapillary flow velocity heterogeneity. The proposed end-to-end system, named CapillaryNet, can detect capillaries at $\sim 0.9$ seconds per frame with $\sim 93\%$ accuracy. The system is currently being used as a clinical research product in a larger e-health application to analyze capillary data captured from patients suffering from COVID-19, pancreatitis, and acute heart diseases. CapillaryNet narrows the gap between the analysis of microcirculation images in a clinical environment and state-of-the-art systems.

Keywords:
Microcirculation Analysis, Early Detection of Diseases, Automated Image Analysis, Convolutional Neural Networks

2000 MSC: 68T35, 92C50, 68N01

1. Introduction

Capillaries are the smallest blood vessels in the body, measuring less than 20 micrometers in diameter \cite{1}. In capillaries, several nutrients, including water, oxygen, and other nutrients essential to the maintenance of cellular metabolism, are exchanged with the interstitial fluid \cite{2,3}. A network of capillaries constitutes the microcirculation architecture of the body \cite{4}. Studies on the assessment of microcirculation have revealed that various diseases alter the capillary density and flow velocity in patients \cite{5,17}. Furthermore, in some studies, microcirculation has been claimed to alter during an early stage of disease progression; thus, microcirculation monitoring may be used for early detection of various clinical conditions \cite{12,17}. The density of capillaries is an indicator of the surface area available for the exchange of nutrients \cite{3}. The capillary density can be used as an early metric for the mortality rate associated with cardiovascular diseases \cite{5}. Measuring the capillary density is a crucial component of understanding fibrotic diseases \cite{13}. In a study on chronic kidney diseases, a reduction in the capillary density over time was determined to be useful as an early indicator to prompt timely interventions \cite{19}. Moreover, measurement of the velocity of red blood cells within capillaries can help assess fluid homeostasis and the transcapillary fluid flux \cite{3}. Alterations in the velocity of red blood cells within capillaries influence the volume of nutrients delivered to surrounding cells and the duration of exchange through microvascular walls to surrounding tissues \cite{20}. The flow velocity in cerebral capillaries
may be increased owing to a body’s response to regulate a decrease in oxygen tension in neurons [21].

A common theme in microvascular analysis studies is the time-consuming and laborious nature of the tasks, requiring clinicians to select capillaries and manually determine blood flow velocity. Such tasks are ophthalmologically strenuous and susceptible to errors and observer variations across different datasets. On average, a trained researcher requires 20 minutes to analyse a 20-s-long microvascular video [22]. Time-consuming analyses and considerable training requirements hinder the integration of microvascular microscopy into routine clinical practice [23]. Moreover, although the flow velocity and capillary density are assessed regularly, intra-capillary flow heterogeneity and capillary hematocrit are not routinely assessed owing to the difficulty in their manual analysis. Thus, the aforementioned drawbacks also limit the number of parameters that can be analysed from microvascular videos [23].

The aim of this study is to relieve trained researchers of the burden of manual analysis, thereby enabling the trained researchers to focus on the development of methods and hypotheses. The requirements of the system were determined based on two sources—a set of interviews conducted by the authors with medical doctors working at ODI Medical AS (MedTech company responsible for the e-health industrial application), with several years’ experience in microcirculation analysis, and the literature provided in three meetings organised by international experts in microvascular microscopy. This series of meetings concluded that an automatic assessment of microcirculation is essential to integrate microvascular microscopy into clinical practice [23].

Thus, this paper presents a novel system for automatically assessing microcirculation by using microvascular videos. The primary requirements for a production-grade system are as follows:

1. reduction in the time required for microcirculation analysis from 20 minutes (manual analysis) to couple of seconds;
2. an accuracy of 93% exceeding that of a trained researcher, which is 84%;
3. detection and quantification of capillary density;
4. detection and quantification of capillary hematocrit;
5. tracking intra-capillary flow heterogeneity and direction of flow within capillaries;
6. classification of the velocity of red blood cell flow within the capillaries;
7. low power consumption, so that it can be used in battery-powered devices in hospitals (e.g., it should run on a CPU, rather than only on power-intensive GPUs).

Relevant methods previously reported in the literature were either slow (requiring several minutes to analyse microcirculation videos) or exhibited poor accuracy that was unsuitable for clinical application. The proposed method is unique because it combines traditional methods with deep neural networks (DNNs) to obtain a relatively fast system with high accuracy. The proposed method has already been applied and evaluated in various clinical studies (e.g., on COVID-19, pancreatitis, and acute heart diseases) [24].

The remainder of the paper is structured as follows. Section 2 presents the previous systems developed to analyse microcirculation images. Section 3 introduces the method of the proposed system, CapillaryNet. Section 4 describes the system implementation and the data acquisition and annotation methods. Section 5 provides the results obtained by using the proposed system and a discussion on the findings. Section 6 concludes the paper.

2. Related Work

Dobble et al. [25] 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 capillaries with most gaps or not enough blood flow will be disregarded. Furthermore, Dobble et al. [25] removes capillaries that are out of focus which were considered 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. [26] has a very similar flow as Dobbleet al. [25] with minor changes. Hilty et al. [26] 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 400µm in diameter as venules. The capillaries are then passed to a modified curvature-based region detection algorithm [27] to an image that has been stabilized and equalized with an adaptive histogram. The result is a vessel map containing centerlines across structures between 20–30µm wide. As stated by the authors of the
curve-based region detection algorithm [27], this type of detection is unintelligent and can lead to detecting artifacts such as hair or stains of similar sizes. Furthermore, due to the challenges in the skin profile stated above, the mean of the images across the whole video is not always the best representation value because different parts might have different lighting or capillaries that can be out of the optimal focus. Moreover, slight motion videos will have to be disregarded entirely because the central line is calculated across all frames instead of per frame.

Similar to Dobble et al. [25], Bezemer et al. [28] improves the method by using 2D cross-correlation to fill up the blood flow gaps caused by plasma and white blood cells. Using cross-correlation is a better method because the number of frames to be disregarded is reduced. However, 2D cross-correlation assumes a uniform blood flow and does not consider the dynamic change of flow between the gaps that can inherently decrease the accuracy of prediction.

Tam et al. [29] 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. Because 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.

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. These methods need to be adjusted accordingly depending on the illumination on the video and thickness of the skin. Such adjustments introduce a manual job where the user has to find the right combination of values for different videos or videos 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 smooth the images manually to increase the differentiation of the capillaries from the background. These 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 skin thickness. Moreover, the image used is in 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 it was assumed it has the highest contrast between the capillaries and the background. The Top-Hat transform method was used 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 with the highest contrast.

Pretasić et al. [34] used a customized neural network to segment the foveal microvasculature. Their neural network consists of 3 Convolutional Neural Networks (CNN) blocks coupled with max-pooling and a dropout layer followed by two 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 make it unsuitable for clinical use users.

Nivedha et al. [35] used the green channel of the image and used a non-linear Support Vector Machine [36] 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 media and concluded that the Gaussian filter is the most suitable for their data. Furthermore, the authors compared different segmentation methods, including OTSU, K Means, watershed, and concluded that the OTSU method was the most suitable for their data. The segmented images were then passed to an SVM, and the authors achieved an accuracy of 83.3%.

Geyman et al. [37] 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. Such manual clicks are highly susceptible to observer variations across different datasets.

Java 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. Such images have to be scaled up, making this method inefficient and using more resources than needed. The authors reported 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. The sheer number of parameters can make this slow and inefficient use within a clinical setting. The training and test time was not reported in the paper.

Hilty et al. [26] uses a similar method to Cheng et al., where the authors first apply an image enhancement process followed by the highlighting of the vessels of interest to quantify the capillary. The authors use a combination of Contrast-limited adaptive histogram and a combination of the first and second-derivative Gaussian kernel convolutions [26] to quantify capillaries. This method is susceptible to detecting artifacts as capillaries in the images will not differentiate between capillaries and other objects in the image. This paper did not report the accuracy, training, and test time.

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, the authors used a spatiotemporal diagram analysis for the flow velocity calculation. This method requires white blood cells or plasma gaps to detect the velocity accurately. Therefore, capillaries that lacked such characteristics had to be disregarded, reducing the overall efficiency of velocity classification. Furthermore, the paper’s authors stated that the spatiotemporal method could 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 have to be resized to 256x256, and an accuracy of 64% was reported. This accuracy is too low for it to be used in a clinical setting.

Dai et al. [46] used a custom neural network similar to Prentašić et al. for segmentation. However, Dai et al. used five CNN blocks instead of three. Gamma correction and contrast limited adaptive histogram were used for equalization in the image enhancement step. The authors reported an accuracy of 60.94%, which is too low to be used in a clinical setting.

Both Sherry G. Clendenon et al. [47], and Sang-Ho Park et al. [48] developed methods that can segment and quantify the microvascular structure using deep learning but had to use to inject the subject with ICG injection to enhance the capillaries so it can be detected easier with the algorithm. The time for analysis was in minutes for both methods and involved an invasive procedure, meaning chemicals were added to the subject before taking a reading. Anthony D Holley et al. [49] used AVA (Version 4), which is based on Hilty et al. [26, 50] method described earlier.

In the above, the more accurate methods require semi-automatic analysis while the more automatic methods are less accurate, making such methods unsuitable for a clinical setting. In contrast, CapillaryNet is fully automatic and able to detect capillaries in ~0.9s with 93% accuracy making it suitable to be used in a clinical setting.

3. Methods

The system architecture of CapillaryNet is presented in this section. CapillaryNet comprises two primary functional stages. During the first stage, which is described in the first subsection, capillaries are detected, the area within the bounding box is calculated, and capillary hematocrit and capillary density are determined. During the second stage, which is described in the second subsection, the intra-capillary flow between frames is calculated, the direction of flow within the capillary is plotted, and the velocity within that capillary is calculated. The input and output of these two stages are shown in Figure 1.
3.1. First Stage - Capillary Detection, Area Quantification, and Density Calculation

The first stage consists of three phases—generation of the regions of interest (RoIs), classification of the RoIs using CNN, and generation of masks on the classified RoIs. RoIs are defined as bounding boxes in specific areas of the image, suggesting the presence of a capillary. RoIs are detected via two independent pipelines: the hue, saturation, value (HSV) pipeline and the Structural Similarity Index (SSIM) pipeline [51]. Then, different enhancements are applied to the RoIs and these RoIs are transmitted to two different CNNs for capillary detection. Finally, the areas of RoIs containing capillaries are quantified during the mask generation phase and the capillary density of the image is calculated.

3.1.1. HSV Pipeline of Phase 1: Generation of RoIs

The first pipeline begins by applying the following three steps to smoothen the image. In the first step, the image is convolved using a 5 × 5 normalised box filter with a stride of one. The central pixel within that window is changed to the average value of the group within the window (see equation 1). The image is then blurred using a 2D 9 × 9 Gaussian blur window (see equation 2). Gaussian filters are used because such filters do not overshoot the pixel value to a step function input while minimising the increase and decrease in pixel values. Thus, in this case, sharp edges in the image are smoothened, while avoiding excessive blurring or the loss of edge information. The image is then alpha-blended with the original input image using a Gaussian weight of -0.5 (see equation 3). Alpha blending overlays a transparent version of the smoothened image on the original image as the background. It is an arithmetic operation that minimises noise while retaining the details of the image. The transparency mask between these images is known as an alpha mask and a value of $\alpha = 1.5$ is taken in this case. The aforementioned filters were selected owing to their relatively cheap computational requirements compared to a hybridised kernel [52]. Thus, using a simple Gaussian filter, the proposed method achieved the accuracy required for industrial and clinical application, while also satisfying duration and hardware requirements. In comparison, hybridised kernels require greater computational capacity than that available in hospital environments.

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$ \hspace{1cm} (1)

where:

- $K$ = image smoothing factor

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

where:

- $x$ = distance from the origin in the horizontal axis
- $y$ = distance from the origin in the vertical axis
- $\sigma$ = standard deviation of the Gaussian distribution

$$\hat{F} = (1 - \alpha)F_0(x) + \alpha F_1(x),$$ \hspace{1cm} (3)

where:

- $F_0$ = original image
- $F_1$ = smoothing image

After the aforementioned steps are completed, the colour of the image is enhanced by a factor of three, and the contrast is enhanced by a factor of 2.5 [53]. The image is then converted to the HSV colour model, whose value represents the brightness of the colour on the spectrum from black to the average saturation value. The hue channel represents the colours, e.g., red or yellow. The saturation channel represents the degree of colour on the spectrum from grey to the pure colour of the image. In an image, the brightness, $V$, is calculated using Equation 4 [54]:

$$V = \max(R, G, B)$$ \hspace{1cm} (4)

The intermediate value, $C$, is then calculated by using Equation 5 [54]:

$$C = V - \min(R, G, B)$$ \hspace{1cm} (5)

$C$ is used to derive the hue value and is calculated using Equation 6 [54]:

$$H = 60^\circ \times \begin{cases} \text{undefined} & \text{if } C = 0 \\ \frac{G - B}{L} \pmod{6} & \text{if } V = R \\ \frac{B - R + 2}{L} \pmod{6} & \text{if } V = G \\ \frac{R - G + 4}{L} \pmod{6} & \text{if } V = B \end{cases}$$ \hspace{1cm} (6)

Pixel saturation is calculated using Equation 7 [54]:

$$S = V$$ \hspace{1cm} (7)
The HSV colour model is used in this paper instead of the red, green, blue (RGB) model because of its greater robustness against changes in illumination and shadows. While the RGB model subtracts the value at each point to obtain the colour, the HSV model combines the values of hue, value, and saturation to obtain the colour value. This fundamental difference in the determination of the colour of a pixel enables more accurate detection of capillary segments. The differences between HSV and traditional RGB colour spaces are depicted in Figure 2.

![Figure 2: Traditional RGB colour space is depicted on the left, whereas the HSV colour space is shown on the right. In RGB, the colour is determined by the resultant colour at a point within the space while in HSV the values of saturation, value and hue are combined to determine the colour. Images by Michael Horvath, available under Creative Commons Attribution-Share Alike 3.0 Unported license](image_url)

After several experiments, it was observed that the HSV colour values of capillaries were between (155, 60, 0) and (180, 255, 255). However, these values are modifiable in the code of the proposed methodology and can be modified by the user. The bitwise AND operator is used to calculate the conjunction of pixels between the specified HSV range and the image. This operation only considers standard pixels between the HSV range and the image. Subsequently, the remaining pixels are removed. The borders between the removed pixel and the HSV range are detected using a combination of the OTSU threshold and the contour approximation method.

### 3.1.2. SSIM Pipeline of Phase 1: Generation of RoIs

The second pipeline for RoI detection utilises the SSIM method, which extracts the background and subtracts it from the original image. Three sets of information are extracted from the image—luminance, contrast, and structure. The SSIM method is depicted in Figure 3.

Luminance is measured by averaging the pixel in a given window. An $11 \times 11$ window is used in this case which is denoted by $N$ in Equation 8.

$$ \mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i $$  

Contrast is measured by calculating the standard deviation of the pixel values (see Equation 9).

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

Structural comparison is performed by dividing the image in terms of its standard deviation along both input signals—first along the X-axis using the following equation:

$$ \frac{(x - \mu_x)}{\sigma_x} $$

and then along the Y-axis using the following equation:

$$ \frac{(y - \mu_y)}{\sigma_y} $$

Each aforementioned step is completed by comparing the background image with the original image. The luminance comparison function shows the difference in brightness between two images (see Equation 12).

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

The $C_1$ factor in the luminance comparison equation is calculated by multiplying the pixel value by a constant (see equation 13). The value of the constant $K_1$ is empirically determined based on the dataset.

$$ C_1 = (K_1 L)^2 $$

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

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

The $C_2$ factor in the contrast comparison equation is calculated by multiplying the pixel value by a constant (see Equation 15). The value of the constant $K_2$ is empirically determined based on the dataset.

$$ C_2 = (K_2 L)^2 $$

The structure comparison function is also calculated using the standard deviations corresponding to the images (see Equation 16).
where:
\[
\sigma_{xy} = \text{see equation (17)}
\]
\[
C_3 = C_2/2
\]

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

The SSIM is then deduced by calculating the difference in luminance, contrast, and structural difference (see equation (18)). As deduced by the author of the SSIM paper [51], assuming \(\alpha = \beta = \gamma = 1\) and \(C_3 = C_2/2\), the equation can be simplified as shown in equation (19).

\[
\text{SSIM}(x, y) = \left( \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \left( \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right)
\] (19)

Following the aforementioned steps, the mean structural similarity index between the resulting image and the original image is calculated, and all differences between these images are assigned bounding boxes and considered to be RoIs. Each RoI is enhanced using pixel cumulative histogram equalization. Combination of the two aforementioned pipelines increases the probability of capillary detection, thereby improving the overall accuracy of the method. Moreover, the utilization of both pipelines is relatively inexpensive compared to purely CNN-based detectors.

The overall flowchart for phase 1 (HSV and SSIM), as described in this subsection and the previous ones, is presented in Figure 4.

In Figure 5, the RoIs detected by using HSV are highlighted in green, while those detected using SSIM are highlighted in black. Figure 5a shows a case where a greater number of RoIs is captured using the HSV pipeline than SSIM, Figure 5b depicts a case in which a greater number of RoIs is detected using SSIM compared to HSV, and Figure 5c reveals that SSIM and HSV pipelines capture RoIs in different parts of the image.

Every bounding box in the image is considered to be a RoI. Each RoI is transmitted to the CNN (in phase 2) for classification — this process is described in the next section.

3.1.3. Phase 2: CNN

The second stage begins with the transmission of the RoIs from the SSIM and HSV pipelines to the CNN of CapillaryNet for capillary detection. Each pipeline is assigned its own CNN model.

The model corresponding to the HSV pipeline comprises 7,418 parameters. The minimum size of input image to ensure high detection accuracy for this CNN was experimentally deduced to be \(15 \times 15\). The HSV CNN architecture includes two CNNs with 16 and 32 filters, respectively. This is followed by a dropout layer.
of 0.2 and 2 dense layers with 8 and 2 neurons, respectively. Both the convolutional layers and the first dense layer contain a ReLU activation function \[58\] while the last layer contains a Softmax activation function (see Equation \[20\]). The Softmax function scales the sum of the values to one, enabling the interpretation of the outcomes as probabilities when the value exceeds 0.5, the corresponding RoI is predicted to contain a capillary, while a value less than 0.5 indicates a high probability that the corresponding RoI does not contain a capillary. The CNN architecture corresponding to the HSV pipeline is displayed in Figure 6.

$$\sigma \left( x_j \right) = \frac{e^{x_j}}{\sum_{i=1}^{2} e^{x_i}}$$ \hspace{1cm} (20)

The CNN is optimised by using sparse categorical cross-entropy and trained for 20 epochs. Cross-entropy represents the loss between a label and its corresponding prediction. Sparse categorical cross-entropy is a function in Tensorflow and refers to the labelling of the input dataset. It is the same as cross-entropy loss, but takes only labels which are integers and not 1-hot vectors. It is calculated on every iteration to minimise the loss, thereby increasing the overall accuracy of the neural network, and it is expressed by Equation \[21\].

$$H(p, q) = - \sum_{x \in \text{classes}} p(x) \log q(x)$$ \hspace{1cm} (21)

where:

- \(p(x)\) is the true probability
- \(q(x)\) is the model predicted probability

The model architecture corresponding to the second SSIM pipeline includes one CNN with a ReLU activation function with eight filters \[58\], followed by one dense layer containing two neurons with a softmax activation function, as illustrated in Figure 7. The input image is taken to be of size 50 x 50, and 9,442 parameters are considered in aggregation. The CNN is optimised using Adam \[59\] and trained for 30 epochs. The loss of the model is computed in terms of cross-entropy loss between the labels and respective predictions. Table \[1\] summarises the CNN parameters for both of these models.

The outputs of the two CNNs are then combined by integrating the coordinates of the capillaries into a single list. This list contains the coordinates of all RoIs that are predicted to contain a capillary and is applied to the input image.
Table 1: Parameters for the HSV CNN model and the SSIM CNN Model

| Parameter Name      | HSV CNN Model                  | SSIM CNN Model                  |
|---------------------|--------------------------------|---------------------------------|
| Pad                 | half padded                    | not padded                      |
| Image Dimension     | 15 by 15                       | 50 by 50                        |
| Layers              | 3                              | 3                               |
| CNN Filters         | 16 and 32                      | 32 and 64                       |
| Dense Layers        | 8 with no dropout              | 64 with dropout 0.5             |
| Activation Class    | Relu and Softmax               | Relu and Software               |
| Optimizer           | Adam                           | Adam                            |
| Learning Rate       | 0.001 to 1e-10                 | 0.001 to 1e-12                 |
| Beta 1 Value        | 0.9                            | 0.9                             |
| Beta 2 Value        | 0.999                          | 0.999                           |
| epsilon value       | 1e-07                          | 1e-07                           |
| Loss Type           | Sparse Categorical             | Sparse Categorical              |
| Training Loss first Epoch | 0.3415                      | 1.1332                          |
| Training Loss Last Epoch     | 0.0783                      | 0.1605                          |
| Validation Loss first Epoch  | 0.2170                      | 0.6783                          |
| Validation Loss Last Epoch    | 0.0781                       | 0.1729                          |

3.1.4. Phase 3: Mask Generation

Detected capillaries are processed using a Gaussian blur function, which reduces the pixelation of the image. Gaussian blur is performed using a window of size $31 \times 31$. Then, the red pixels are extracted from that image by applying an adaptive Gaussian threshold function that converts all red pixels to white ones and all other pixels to black ones. The total amount of white pixels and the output corresponding to RoI including a single capillary are depicted in Figure 8a and Figure 8b, respectively. The mask is then re-calculated for every consecutive frame of the video to determine capillary density at all time instances and the capillary hematocrit. The re-calculation of the mask over the entire duration of the video enables heterogeneity of capillary perfusion, which allows the monitoring of red blood cell flows within capillaries. Via the aforementioned method, CapillaryNet detects a capillary, the area it occupies, and the blood flow through it over time. These pieces of information help clinicians assess the average capillary hematocrit and its variation over time.

3.1.5. Final Remarks on the First Stage

The detection of capillaries in microcirculation images using CapillaryNet, the calculation of the areas within the bounding boxes, and the calculation of the capillary hematocrit and capillary density as output of the first stage have been described in previous subsections. The CapillaryNet architecture achieves these goals by uniquely combining traditional and deep learning models and tailoring the parameters for capillary detection.

In previous studies, as discussed in Section 2, either traditional methods or deep neural networks had been utilised. Although traditional methods are faster than deep neural networks, traditional methods exhibit poorer accuracy as deep neural networks can learn from trained data—purely traditional methods output all non-background regions (e.g., in our case, the skin) as capillaries. On the other hand, deep neural networks are resource-intensive and time-consuming. Therefore by assigning a significant portion of the task to traditional computer vision techniques and using deep neural networks solely for detection, CapillaryNet outperforms the methods discussed in Section 2.
In the next subsection, the calculation of intra-capillary flow between frames using CapillaryNet, plotting the flow direction within the capillary, and the classification of the velocity within the capillary are discussed.

3.2. Second Stage - Velocity Classification and Determination of Flow Direction

The capillaries detected by the CNN are transmitted to the second stage of CapillaryNet. During this stage, the architecture determines the velocity and flow direction of blood within the capillaries. First, certain enhancements are applied to the RoI region from the original frame. The velocity detection algorithm requires a different set of enhancements compared to the capillary detection algorithm. Then, the image is denoised and the RoI is transmitted to the velocity detection algorithm. These steps are discussed in further detail in the following subsections.

3.2.1. Phase 1: Image Enhancement using Pixel Cumulative Histogram Equalization

First, a median blur with a kernel size of 5 × 5 is independently applied to each channel. In each channel, the central pixel value within the kernel window is re-computed based on the median value of the pixels within the kernel window. The pixels in the image range between 0 and 255 and a histogram is obtained based on the modified RoIs, which illustrates the intensity distribution of the pixels. The cumulative distribution of the pixels is then calculated based on the histogram. Finally, a cut-off threshold is applied, and all pixels with values within 10% of the cumulative increased distribution are removed. The threshold adjustment increases the overall brightness and contrast of the capillaries within the RoI, making the RoIs clearer and sharper. Adoption of this threshold is, therefore, critical for tracking the movement of pixels across different frames.

3.2.2. Phase 2: Image Denoising using Non-local Means Denoising Algorithm

The increase in the contrast and overall sharpness of each RoI is accompanied by an increase in the noise of each pixel. In this step, the noise corresponding to each pixel is reduced while retaining all information in the pixel. First, a median blur is applied to denoise each pixel locally, thereby increasing the overall smoothness of the image. As the noise value varies across pixels, some areas of an image may require more extensive denoising than others. After the computation of the noise variable as a non-local mean, the Gaussian distribution is applied and local noise is minimised. This computation is performed by converting the image to the LAB colour space - where 'L' is the lightness, 'A' (green/magenta) and 'B' (blue/yellow) are the chromatic axes. The pixels are then then separately denoised on the L and AB components, thereby ensuring that each group of pixels in the image is treated in accordance with their local noise values rather than using the global noise value for the entire image. This completes the enhancement and smoothening of RoIs. In the next step, each RoI is processed using the velocity detection algorithm of CapillaryNet, and the velocity, infra-flow direction, and flow direction within the capillaries are classified.

3.2.3. Phase 3: Velocity Detection

The Gunnar Farneback [61] algorithm is a two-frame motion estimation algorithm, which performs approximation by calculating the quadratic polynomials and polynomial expansion transforms between each pair of successive frames. The polynomial expansion coefficient is used to derive the displacement fields of each pixel, assuming pixel intensities to be constant between successive frames. The velocity vectors are obtained based on the differences in the locations of the pixels between successive images. A velocity vector represents the movement of an object from (x,y) in the direction, (dx,dy), in time, dt. This movement is illustrated in Figure. The velocity vector is used to calculate the flow direction and the intra-capillary flow between frames, and perform velocity classification. The algorithm assumes that the movement of an object can be described using a quadratic function I(x,y,t) (given by Equation 22). The quadratic function is then expanded using a hierarchical scheme of separable convolutions utilizing Taylor series approximations of the right-hand side [62], thereby yielding a system of linear equations in variables dx, dy, and dt.

The algorithm scales the images by a factor of 0.9 using a window of size, 10 × 10 with pyramid level 2. This calculation is performed over ten iterations at each pyramid level to increase the accuracy of the calculation. A polynomial size of 5 and a polynomial sigma of 1.2 are considered. The Cartesian coordinates are then converted to polar coordinates, which produces the magnitude and the angle of each 2D vector—here, the magnitude represents the absolute value of the velocity vector and the angle represents the direction of the flow. The velocity vector represents the intra-capillary heterogeneity of the flow velocity value and the angle in
the flow direction. The averaged intra-capillary heterogeneity of the flow velocity vector yields the velocity classification of the RoI. The velocity detection method is illustrated in Figure 10.

\[ I(x, y, t) = I(x + dx, y + dy, t + dt) \]  

Figure 9: Calculation of pixel displacement. A pixel at position \((x,y)\), at time \(t\), is displaced. The difference between the new and old positions, over time duration \(dt\), is given by \((dx, dy)\).

4. Implementation

This section introduces the dataset used for data training and evaluation of the CapillaryNet architecture. Moreover, the annotation of the data and the implementation of the system are also described.

4.1. Dataset

The videos are captured on human subjects using a handheld digital microscope (Digital Capillaroscope, Inspectis, Sweden) at a resolution of \(1920 \times 1080\) at 30 frames per second for 20 seconds. The nutritive capillaries in the skin papillae in the dorsum region of the hand are visualised in the videos, as depicted in Figure 11. A sample of a video frame is shown in Figure 12. The region is coated with a layer of transparent oil before applying the microscope to the skin. The total dataset consists of 365 videos captured from 50 subjects. Each video in the dataset depicts at least one capillary, with some depicting up to 70 capillaries. All subjects provided informed consent to participate in the study (IRB protocol and approval number: 2020P001987). The average age of the subjects is 57 years with a standard deviation of 17 years.

4.2. Data Annotation

A trained researcher annotated the obtained microvascular videos using in-house software. Capillaries visible in different frames of each video are marked with bounding boxes and assigned velocity labels whenever applicable. For capillary detection training, a single frame is extracted from the bounding boxes, and for velocity detection, multiple frames are extracted from within the bounding boxes. The two stated steps are performed to obtain the training, validation, and test datasets. A second trained researcher labelled the dataset, and the reproducibility of capillary detection between two researchers is calculated to be \(\sim 84\%\). Thus, the accuracy of the trained researcher is assumed to be \(\sim 84\%\). Our industrial stakeholder, ODI Medical AS (a MedTech company specializing in microcirculation analysis over the past decade), confirmed this value.

The capillary detection algorithm is trained on \(\sim 70\%\) of the dataset, validated on \(\sim 20\%\), and tested on \(\sim 10\%\).
The total dataset consists of 365 videos with ∼12,000 capillary images and ∼2,000 non-capillary images, with a ratio of 6:1. This class imbalance is considered during the design of the deep neural network, and the prediction accuracies are reflected in the F1 scores obtained. The algorithm’s accuracy on capillary detection and velocity classification is calculated by comparing the labels of the bounding boxes assigned by the trained researcher with those obtained using the proposed system.

4.3. System Implementation

The following packages were used to implement the system: imageio [v2.13.4], imutils [v0.5.4], keras [v2.7.0], Keras-Preprocessing [v1.1.2], matplotlib [v3.5.1], numpy [v1.21.5], opencv-python [v4.5.4.60] [63], Pillow [v8.4.0], scikit-image[v0.19.1], scikit-learn [v1.0.1] [64], scipy [v1.7.3] and tensorflow [2.7.0] [65]. Python 3.9 [66].

Coding and evaluation were performed in Pycharm Professional 2021.1 on a computer with Windows 10 operating system. A demo of the system can be found here [http://www.analysecapillary.space/]. The system can be cloned, modified, and used by following the instructions provided on the website. The user can then use the Python environment to use the system under the stated license.

5. Results and Discussion

In this section, the results obtained by using CapillaryNet are presented and discussed. CapillaryNet is compared with other object detection algorithms, as described in Section 5.1. CapillaryNet is compared with other microcirculation analysis methods in the literature, as detailed in Section 5.2. Sections 5.3 and 5.4 provide a discussion on the first stage results (capillary hematocrit and capillary density) and present the accuracy of the proposed method. Sections 5.5, 5.6 and 5.7 present the second stage (flow velocity of blood within capillaries and their direction) results. Sections 5.8, 5.9, 5.10 and 5.11 present an ablation study, generalisation of the proposed framework, the limitations of this paper, and future research directions.

1CapillaryNet: An Automated System to Quantify Skin Capillary Density and Red Blood Cell Velocity from Handheld Vital Microscopy © 2020 by Maged Abdalla Helmy Mohamed Abdou is licensed under Attribution-NonCommercial 4.0 International. To view a copy of this license, visit [http://creativecommons.org/licenses/by-nc/4.0/].

5.1. Capillary Detection: Comparison of CapillaryNet with other Object Detection Algorithms

Object detection is one of the most challenging problems in computer vision. The aim of this task is to detect the area occupied by an object in an image [67]. Deep learning is a powerful tool, which significantly increased the mean average precision in various object detection competitions (i.e., VOC 2007-2012 and ILSVRC 2013-2017) [68][69]. Before deep learning, salient detection methods were state-of-the-art in object detection [70][72].

Object detection algorithms involve millions of parameters to optimise with respect to specific datasets [67]. For example, the widely used Mask R-CNN algorithm achieved state-of-the-art object segmentation results on the COCO dataset (comprising 80 object categories with 1.5 million object instances) [67][73]. Mask R-CNN is not suitable for use in tightly constrained environments for several reasons. First, Mask R-CNN involves ∼6.3 million optimisation parameters, and our experiments show that training it on the same dataset and hardware as CapillaryNet requires approximately five days. In contrast, CapillaryNet exhibits a training time of ∼20 seconds per epoch, as its bounding box detection algorithm only requires ∼16,860 parameters. Moreover, the deployment of large models requires relatively high-end computers that utilise GPUs that are difficult to find in medical devices. Thus, CapillaryNet was designed to operate in medical environments with limited computing capabilities while simultaneously exceeding the accuracy of a trained researcher. This is enabled by the fact that CapillaryNet utilises a much shallower convolutional neural network (CNN) compared to object detection algorithms such as Mask R-CNN. Table 2 lists some commonly used models and the corresponding numbers of optimisation parameters [74]—DenseNet201 requires 18.28 million parameters, whereas ResNet152 requires 58.34 million parameters. Thus, a shallower CNN makes the proposed model more versatile by enabling faster re-training when new data is incorporated. With a training speed that is faster by a factor of 1000, a detection time that is better by a factor of 100, and a fraction of the size of the current state-of-the-art object detection algorithms, CapillaryNet is a production-ready capillary detection model that is optimised to perform microcirculation analysis in clinical environments.

Furthermore, the experiments conducted in this paper indicate that CNN-based RoI detectors, e.g., that of Mask R-CNN, do not generalise well with new capillary data. This can be attributed to the variation in illumination over the image, the blurring induced by the
size of a capillary relative to that of the image, or occlusion of the skin by hair, stains, and other artifacts, as indicated in Figure 13. The CapillaryNet architecture tackles these challenges posed by the skin profile by using the HSV colour space to detect RoIs instead of pure CNNs. Moreover, the RoI detection methods utilised by CapillaryNet are computationally less expensive than that of Mask R-CNN.

The bounding boxes obtained by using CapillaryNet are illustrated in Figure 14. Figure 14 highlights the area occupied by a capillary. Similarly, Figures 14: and depict another image obtained by using a handheld microscope and the area occupied by capillaries, respectively. As the lines are tightly wrapped around the capillaries within the bounding box, the area occupied can be calculated with high accuracy, and the capillary density is derived from that value.

Figure 13: Image of the capillaries on the dorsum part of the hand with dirt and microscopic artifacts. Using purely rule-based algorithms to detect capillaries in this case will lead to the detection of the artifacts due to similarities in size. On the other hand, by combining salient detection methods with convolutional neural networks, the algorithms can distinguish between capillaries and artifacts.

5.2. Capillary Detection: Comparison of CapillaryNet with Other Capillary Detection Methods Reported in the Literature

CapillaryNet compensates for unequal illumination using two approaches. Firstly, the application of the HSV colour space enhances the robustness of the architecture against changes in illumination. Secondly, the inclusion of out-of-focus capillaries in training, validation, and test datasets, followed by training of the CNN on the dataset labelled by a trained researcher, further reduces observer variation. Moreover, several methods in the literature highlighted the need for averaging the pixel values of several frames. In contrast, the utiliza-

| Model Name | Size (MB) | Param (Mil) | training time |
|------------|-----------|-------------|--------------|
| CapillaryNet | ~10       | ~0.6        | ~minutes     |
| Mask RCNN  | ~256      | ~6.3        | ~days        |
| DenseNet201 | ~7947     | ~18.28      | ~weeks       |
| ResNet152  | ~9097     | ~58.34      | ~weeks       |
tion of the HSV colour space combined with a CNN network enables CapillaryNet to detect capillaries from a single frame. Thus, averaging pixel values over the entire video and treating each frame uniquely become redundant. It also enables better real-time detection.

5.3. Capillary Hematocrit

Capillary hematocrit is indicative of the potential of each capillary to deliver oxygen to its surrounding tissues. When capillaries have a regular flow and are normally distributed but exhibit a low concentration of red blood cells (i.e., low hematocrit), the efficiency of oxygen delivery via microcirculation may be compromised. Additional clinical studies are required to elucidate the potential of using capillary hematocrit as an indicator for abnormal microcirculation. The proposed method calculates capillary hematocrit.

Figure 15 presents a capillary filled with red blood cells. Figures 15b and 15c display the same capillary after a few seconds with white blood cells and plasma gaps. Figure 16 shows the red blood cell distribution of this capillary derived using CapillaryNet.

![Figure 15](image1.png)

Figure 15: a) presents a capillary filled with red blood cells. (b) and (c) display the same capillary after a few seconds with white blood cells and plasma gaps. Figure 16 shows the red blood cell distribution of this capillary derived using CapillaryNet.

5.4. CapillaryNet Speed and Accuracy

CapillaryNet can analyse 30 frames in \(\sim 4\) s, which is the equivalent of a speed of \(\sim 0.9\) s per frame, whereas a trained analyst spends 2 min on average on the same frame to highlight the capillaries and calculate capillary density. As mentioned in the Implementation section, the accuracy of a trained researcher is \(\sim 84\%\), whereas that of CapillaryNet is \(\sim 93\%\). Table 3 summarises the main differences between a trained researcher and CapillaryNet in terms of capillary detection. Furthermore, it shows the F1 scores of other state of art models. The macroscopic average accuracy on the test dataset of the HSV CNN model is \(\sim 96\%\), and that on the SSIM CNN model is \(\sim 86\%\). Table 4 summarises the F1 scores of HSV Model and Table 5 summarises the F1 scores of SSIM Model. The outputs obtained from both CNNs are combined to maximise capillary detection in a microcirculation image.

![Figure 16](image2.png)

Figure 16: Graph representing the proportion of a capillary filled with red blood cells (capillary hematocrit) with respect to time

Table 3: Benchmark of CapillaryNet capillary detection against manual analysis performed by a trained researcher on a microcirculation video and other state of the art methods

| Name               | Training Time | Detection Time | F1 Score |
|--------------------|---------------|----------------|----------|
| DenseNet121 - Backbone | \(\sim 5\) minutes | \(\sim 3\) seconds | \(\sim 70\%\) |
| ResNet152 - Backbone  | \(\sim 7\) minutes | \(\sim 5\) seconds | \(\sim 73\%\) |
| Trained Researcher  | \(\sim 80\) hours | \(\sim 120\) seconds | \(\sim 84\%\) |
| Mask R-CNN          | \(\sim 30\) hours | \(\sim 100\) seconds | \(\sim 86\%\) |
| CapillaryNet        | \(\sim 2\) minutes | \(\sim 0.9\) seconds | \(\sim 93\%\) |

5.5. Velocity Detection

The quantification of red blood cell flow is a more challenging task than capillary detection. Some papers have described red blood cell flow based on manual quantification of red blood cells of different scales [75-78], which is subject to intra-individual variation. In manual quantification, each individual vessel is assigned a score representing the average flow velocity estimated by a researcher. Flow velocity scales vary between publications—some researchers have classified...
flow on a scale of 0–2 (absent, intermittent, or continuous flow) [79], others on a scale of 0–3 (absent, intermittent, sluggish, or normal flow) [79], while others on a scale of 0–5 (no-flow, sluggish, continuous very low, continuous low, continuous high, or brisk flow) [80].

More recent papers have used space-time diagrams (STDs) [26, 28, 81] to quantify red blood cell flow, which exhibit a fundamental improvement over manual analysis as STDs are independent of intra-individual variation [26, 27]. However, STDs are susceptible to slight movement due to their reliance on the position of the central line along the centre of the capillary in all frames. Thus, any movement necessitates additional manual recalibration, increasing the risk of errors and user bias. Moreover, constructing an accurate central line depends on the accurate identification of the exact width and length of the capillaries at an earlier stage. Therefore, out-of-focus capillaries need to be disregarded while fitting the central line. Moreover, STDs can only be constructed based on the whole video.

CapillaryNet provides an alternative to STDs and manual analysis, which is considered to be the gold standard for red blood cell (RBC) velocity classification [26]. It does not require a central line to deduce the velocity; which reduces the number of steps required. Further, the proposed system can be used to deduce intra-capillary flow heterogeneity. This enhances the accuracy of tracking RBCs of the proposed method.

CapillaryNet 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 [26]. The central line is not needed to deduce the velocity; therefore, the number of steps are reduced by 1 in comparison to the gold standard of detecting velocity. Furthermore, the proposed system can be used to deduce the intra-capillary flow heterogeneity. This makes the proposed method more accurate in tracking red blood cells the reliance on a central line is accurately placed between the frames.

To evaluate the proposed method, a trained researcher classified the velocities of about 2250 capillaries based on a sequence of 240 frames (8 seconds) captured in high resolution (1920 × 1080) at 30 frames per second (fps) into the following categories: no-flow, slow flow, normal flow, and fast flow. For each capillary classified by the analyst, CapillaryNet yields a velocity vector. This value represents the rate of change in capillary position, which is equated with speed. Figure 17 depicts a comparison of the analyst’s classification (depicted on the X-axis) and the velocity vectors obtained using the proposed algorithm (Y-axis). The velocity vectors are observed to be significantly different for each velocity class (the p-values between the pairs of classes are below 0.00001). The magnitude of the average velocity vector for no-flow is ∼0.3 mm/s, that for slow flow is ∼0.63 mm/s, that for normal flow is ∼0.88 mm/s, and that for fast flow is ∼1.66 mm/s, as recorded in Figure 17 and Table 6.

The overlap in velocity vector values between the velocity classes, no-flow and slow flow, and the classes, normal flow and fast flow, indicates the lack of precise definition of the boundaries between velocity categories in manual analysis. For this reason, analysts may classify a capillary with borderline fast-normal flow into the normal flow category or the fast flow category. Hubble et al. [82] demonstrated that inter-analyst variability can extend to up to 26% in the assessment of capillary flow in healthy controls. The accuracy is also dependent on the experience of the analyst [83]. This lack of accuracy in borderline cases creates inconsistent training data for the algorithm and results in overlapping velocity vector values between categories.

The capillaries classified by the analysts under fast
Table 6: Velocity classification and the velocity vector of the dataset

| Velocity Classification | Velocity Vector (mm/s) | Accuracy |
|-------------------------|------------------------|----------|
| No Flow                 | ~0.30                  | ~72%     |
| Slow Flow               | ~0.63                  | ~80%     |
| Normal Flow             | ~0.88                  | ~85%     |
| Fast Flow               | ~1.66                  | ~90%     |

Flow exhibited a broader range of velocity vectors compared to the other velocity classes, as shown in Figure 17. This indicates that the algorithm successfully splits the fast flow category into two categories (e.g., fast and swift flow). This novel classification is expected to provide insights into the variation of capillary flow velocity during the course of various diseases.

The algorithm is trained on the classification of a single analyst and the velocity vector boundaries between categories are refined to create a standardised method for enhancing classification consistency for capillary velocities. This reduces the intra-analyst variation in velocity classification and is not influenced by parameters such as experience of concentration of the analyst. Furthermore, the GF algorithm can detect flow in near real-time at almost 15 fps (~0.07 s per bounding box), whereas an analyst spends 1 min on average to label the speed of a single capillary and 20 min on average to annotate a single video.

5.6. Deep Neural Networks for Velocity Classification

Classification of frames at a resolution of 1920 × 1080 at 30 fps requires a deep learning model whose application in a clinical environment is not feasible. Therefore, experiments are performed at a lower resolution (1280 × 720) and lower fps (5 fps) for velocity classification. This section presents four different deep neural architectures designed to train velocity classification based on videos. This architecture is trained and developed to evaluate the velocity classification performance of CapillaryNet. The presented architectures are based on state-of-the-art papers [53–87]. In the first experimental architecture, InceptionResNetV2 [84] is used in combination with transfer learning. The top nine layers are fine-tuned on the four classes (no-flow, slow flow, normal velocity flow, and fast flow), whereas the remainder of the weights are adopted based on the ImageNet dataset [85]. The output is then transmitted to a convolutional 2D network. The second architecture is based on a time-distributed CNN, and the features of the CNN are transmitted to a recurrent neural network (RNN) [86]. For the RNN network, experiments with both long short-term memory (LSTM) and gated recurrent unit (GRU) are conducted. The architecture for a time-distributed CNN equipped with an RNN for velocity classification is depicted in Figure 18. The third architecture uses a set of 3D convolutional networks [87]. The fourth architecture is similar to the first architecture. It also utilises transfer learning of InceptionResNetV2, but unlike the first case, its output is transmitted to a convolutional 3D network. Thus, the fourth architecture is a combination of the first and third architectures. All four aforementioned architectures are observed to perform poorly, with accuracies less than 50% corresponding to each class. Moreover, these algorithms involve several million parameters to be trained, drastically increasing their energy consumption and training durations compared to CapillaryNet. Thus, classification of capillary velocity using DNNs is complex and energy-consuming, and it remains a challenging task.

5.7. Intra-capillary Flow Velocity Heterogeneity and flow direction

Flow velocity within a capillary has been calculated in previous studies only as an average value. Our empirical observations indicate that healthy subjects exhibit relatively homogeneous flow velocity, while in specific patient groups, the flow velocity varies significantly over the 20-second durations of the respective videos. Thus, a single average value correspond-
Figure 18: A deep neural architecture built to evaluate velocity classification

Table 7: Estimation of applicable parameters suggested for microcirculation analysis by Hilty et al [26] and Ince et al [23]. Two new parameters can be uniquely identified and calculated using CapillaryNet—Intracapillary 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 the capillaries is derived based on 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 RBCs is derived based on the total number of pixels occupied by the detected capillary divided by the dimension of the image |
| **per vessel** |  |
| Flow Velocity | The velocity of red blood cells—flow classified as no flow, slow flow, normal velocity flow and fast flow |
| Intra-capillary heterogeneity of flow velocity | Derived from the velocity part of CapillaryNet where the the variation in pixel movement is plotted across all frames |
| Capillary Hematocrit | Derived using a rule-based algorithm where the number of red blood cells across the frames are plotted |

Figure 19: (a) CapillaryNet applied to a capillary video publicly available to assess capillary flow direction and heterogeneity. Three lines were plotted in the video—the black line is approximately aligned with the centre of the capillary, the green line is placed around the capillary, and 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 (average velocity vector value is approximately 3), the green line (average velocity vector value is 0.2), and the blue line (average velocity vector value is 0.1).

Locality heterogeneity can be used as an additional indicator of compromised microcirculation. CapillaryNet is capable of tracking moving pixels that represent RBCs—corresponding to bounding boxes surrounding capillaries, the pixel displacement of RBCs can be calculated. Thus the proposed method presents an alternative to STDs [26] and manual analysis.

Moreover, the Intra-capillary Flow Velocity prediction accuracy of CapillaryNet is evaluated on a publicly available capillary video [88], as depicted in Figure 19b. The measurements from three different regions of the video show that CapillaryNet can detect the movement of RBCs within capillaries. The results are recorded using a graph in Figure 19b. The highest velocity vector values are observed at the capillary centre, highlighted using the black line. The lowest velocity is observed along the blue line, where there is no flow at all. Figure
Figure 20: Flow direction in a capillary automatically derived using CapillaryNet

20 highlights the flow of the RBCs within a capillary.

5.8. CapillaryNet Ablation Study

Each part of the two pipelines, namely, the HSV pipeline and the SSIM pipeline, contributes to the improvement of the overall accuracy of CapillaryNet. In this section, the effectiveness of each component in RoI generation is detailed.

In the HSV pipeline, the first step involves the application of Gaussian blur. This is critical as it reduces the pixelation in the image and smoothens it, which reduces the number of RoIs from several thousands to several hundreds. Next, HSV conversion is applied. This is beneficial, as capillary detection information is spread across the entire spectra of all the RGB channels while, in the HSV colour space, most of the information is stored in saturation values lying between 60 and 255. Further, unlike RGB, HSV separates the luma, yielding more accurate colour information corresponding to the image. This is essential for the application of histogram equalization on a colour image, which enables the preferential increase of the intensity of capillaries compared to that of background pixels irrespective of variation in illumination or the presence of undesirable artifacts such as shadows or oil bubbles on the skin. Enhancement of the colour and contrast further assists the HSV colour space to differentiate between capillaries and non-capillaries. The Bitwise operator “AND” and OTSU thresholding are mathematical operations that extract the RoI coordinates and transmit the RoIs to the CNN. Then, the CNN filters the RoIs and outputs only those RoIs that contain a capillary. The first step of the SSIM Pipeline subtracts the background image (skin with an average pixel value of 127) from the current frame. This generates as many as 200 RoIs. As in the previous step, the OTSU threshold is applied to extracts the RoI coordinates and transmit the RoIs to the CNN. Finally, the CNN filters the RoIs and outputs only those RoIs that are predicted to contain capillaries in accordance with the accuracy and parameters discussed in Section 5.4.

5.9. Additional Benefit: More Parameters with CapillaryNet

Table 5.6 lists the estimated values of the microvascular parameters suggested by Hilty [26] and Ince [23] obtained using CapillaryNet. In addition, two new parameters that have not been previously monitored in microvascular videos—capillary hematocrit and intracapillary flow velocity heterogeneity—are also monitored. Thus, the architecture of CapillaryNet is a unique combination of DNNs with salient object detection algorithms and two-frame motion estimation techniques. It is expected to motivate the development of a unified automated method for near real-time bedside analysis of microcirculation.

5.10. System Generalization

The system functions and classes of CapillaryNet are designed following the design patterns of microservices. This design philosophy allows a user of the system to replace any pipeline, phase, or stage from the proposed system with their desired methodology. For example, the deep learning module in CapillaryNet can be replaced simply by pointing the function in the CapillaryNet code to another algorithm. Further instructions are available in the README file in the GitHub repository. In other cases, CapillaryNet is expected to exhibit similar accuracy, irrespective of the recording device as long as the magnification and image quality is retained.

5.11. Limitation and Future Work

The velocity detection performance of the proposed system has certain limitations. When the flow of RBCs is not perpendicular to the recording direction of the camera, the displacement of pixels across successive frames cannot be accurately measured. Moreover, the presence of white blood cells or plasma gaps within the capillaries is essential to the accurate tracking of the flow. The authors intend to resolve this issue in future work. Another direction is to explore extraction of information using Temporal Convolutional Networks (TCNs) [89]. TCNs operate using low-level computing.
features by encoding spatio-temporal information and transmitting these features to an RNN, which deduces the corresponding high-level temporal information. Because of the distinct difference in temporal information between capillaries and the background, this method can reduce the number of RoIs generated, thereby accelerating the proposed system and increasing its overall accuracy.

6. Conclusion

This paper presents a state-of-the-art automated method for microcirculation analysis. The proposed method enables the monitoring of microvascular parameters that cannot be monitored using existing systems. These novel parameters can be indicative of the capability of each capillary to deliver oxygen to its surrounding tissues. Moreover, the proposed method standardises microcirculation analysis by eliminating intra-analyst variability and reducing the required duration. In this study, CapillaryNet is used to assess skin microcirculation in the dorsum area. CapillaryNet can automatically quantify the area occupied by a capillary, calculate the capillary density, determine the average flow velocity and the intra-capillary flow velocity heterogeneity, and quantify the capillary hematocrit. The capillary detection speed is observed to be \( \sim 0.9 \) seconds per frame, and its accuracy is \( \sim 93\% \). Thus, CapillaryNet reduces the required duration of microcirculation analysis from 20 min to a few seconds by combining DNNs with traditional computer vision techniques. Because of the design, the proposed system is suitable for low-power computers in clinical environments. This is because the neural networks can be run on a CPU and the number of neural networks is minimised, while ensuring satisfactory accuracy. The proposed system brings the state-of-the-art much closer to real-time microcirculation analysis in a clinical environment. This claim is further reinforced by the use of the proposed system in a product in an industrial environment, where it is used to calculate and track capillary changes in patients with pancreatitis, COVID-19, and acute heart diseases.

7. Disclosure Concerning Competing Interests

The authors declare the following financial interests/personal interests that may be considered as potential competing interests: The Research Council of Norway provided 50% of the funds for Maged Helmy’s Industrial Ph.D. (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.

8. 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.

9. Code

A demo of the system can be found here [http://www.analysecapillary.space] All information related to the code link will be available on the website.

References

[1] D. De Backer, S. Hollenberg, C. Boerma, P. Goedhart, G. Büchele, G. Ospina-Tascon, I. Dobbe, C. Ince, How to evaluate the microcirculation: report of a round table conference, Crit. Care 11 (5) (2007) R101.
[2] A. Cassoobhoy, What is the definition of capillaries? (2020). URL [https://www.webmd.com/heart-disease/heart-failure/guidelines/what-is-the-definition-of-capillaries]
[3] A. C. Shore, Capillaroscopy and the measurement of capillary pressure, Br. J. Clin. Pharmacol. 50 (6) (2000) 501–513.
[4] G. Guven, M. P. Hilty, C. Ince, Microcirculation: Physiology, pathophysiology, and clinical application, Blood Purif. 49 (1-2) (2020) 143–150.
[5] M. J.S. Parker, N. 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, pp. 1–13.
[6] V. Nama, J. Onwude, I. T. Manyonda, T. F. Antonios, Is capillary rarefaction an independent risk marker for cardiovascular disease in south asians?, J. Hum. Hypertens. 25 (7) (2011) 465–466.
[7] A. J. H. M. Houben, R. J. H. Martens, C. D. A. Stehouwer, Assessing microvascular function in humans from a chronic disease perspective, J. Am. Soc. Nephrol. 28 (12) (2017) 3461–3472.
[8] J. C. de Graaff, D. T. Ubbink, J. A. van der Spruit, S. M. Lagarde, M. J. H. M. Jacobs, Influence of peripheral arterial disease on capillary pressure in the foot, J. Vasc. Surg. 38 (5) (2003) 1067–1074.
[9] B. Fagrell, M. Intaglietta, Microcirculation: its significance in clinical and molecular medicine, J. Intern. Med. 241 (5) (1997) 349–362.
[10] P. M. Houtman, C. G. Kallenberg, A. A. Wouda, T. H. The, Decreased naifold capillary density in raynaud’s phenomenon: a reflection of immunologically mediated local and systemic vascular disease?, Ann. Rheum. Dis. 44 (9) (1985) 603–609.
[11] H. Schmeling, S. Stephens, C. Goia, C. Manlihot, R. Schneider, S. Luthra, E. Stringer, B. M. Feldman, Nailed capillary density is importantly associated over time with muscle and skin disease activity in juvenile dermatomyositis, Rheumatology 50 (5) (2011) 885–893.
[12] B. D. Dušcha, W. E. Kraus, S. J. Keteyian, M. J. Sullivan, H. J. Green, F. H. Schachat, A. M. Pippen, C. A. Brawner, J. M. Blank, 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) (1999) 1956–1963.
[13] J. L. Robbins, W. S. Jones, B. D. Dušcha, J. D. Allen, W. E. Kraus, J. G. Regensteiner, W. R. Hiatt, B. 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) (2011) 81–86.
[14] M. Moeini, X. Lu, P. K. Avti, R. Damseh, S. Belanger, F. Palacios, D. Boas, A. Kakkar, F. Lesage, Compromised microvascular oxygen delivery increases brain tissue vulnerability with age, Sci. Rep. 8 (1) (2018) 8219.
[15] A. López, J. C. Grignola, M. Angulo, I. Alvez, N. Nin, G. Lacuesta, M. Baz, P. Cardinal, I. Prestes, J. P. Bouchacourt, J. Riva, C. Ince, F. J. Hurtado, Effects of early hemodynamic resuscitation on left ventricular performance and microcirculatory function during endotoxic shock, Intensive Care Med Exp 3 (1) (2015) 49.
[16] D. De Backer, J. Creteur, J.-C. Preiser, J.-M. Dubois, J.-L. Vinet, Microvascular blood flow is altered in patients with sepsis, Am. J. Respir. Crit. Care Med. 166 (1) (2002) 98–104.
[17] T. Wester, Z. A. Awan, T. S. Kvernebo, G. Salerud, K. Kvernebo, Skin microvascular morphology and hemodynamics during treatment with veno-arterial extra-corporal membrane oxygenation, Clin. Hemorheol. Microcirc. 56 (2) (2014) 119–131.
[18] M. S. Goligorsky, Microvascular rarefaction: the decline and fall of blood vessels, Organogenesis 6 (1) (2010) 1–10.
[19] A. Edwards-Richards, M. DeFreitas, C. P. Katsoufis, W. Seeherunvong, N. Sasaki, M. Freundlich, G. Zilleruelo, C. L. Abitbol, Capillary rarefaction: an early marker of microvascular disease in young hemodialysis patients, Clin. Kidney J. 7 (6) (2014) 569–574.
[20] C. C. Michel, Starling: the formulation of his hypothesis of microvascular fluid exchange and its significance after 100 years, Exp. Physiol. 82 (1) (1997) 1–30.
[21] H. S. Wei, H. Kang, J.-Y. D. Rasheed, S. Zhou, N. Lou, A. Gershtein, E. D. McConnell, Y. Wang, K. E. Richardson, A. F. Palmer, C. Xu, J. Wan, M. Nedergaard, Erythrocytes are Oxygen-Sensing regulators of the cerebral microcirculation, Neuron 91 (4) (2016) 851–862.
[22] M. P. Hilty, P. Guerci, Y. Ince, F. Toraman, C. Ince, MicroTools enables automated quantification of capillary density and red blood cell velocity in handheld vital microscopy, Commun Biol 2 (2019) 217.
[23] C. Ince, E. C. Boerma, M. Cecconi, D. De Backer, N. I. Shapiro, J. Duranteau, M. R. Pinsky, A. Artigas, J.-L. Teboul, I. K. 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) (2018) 281–299.
[24] K. Kvernebo, [Old medical clinical applications](2022).
[25] J. G. Dobbe, G. J. Streekstra, B. Atasever, R. Van Zijderveld, C. Ince, Measurement of functional microcirculatory geometry and velocity distributions using automated image analysis, Medical & biological engineering & computing 46 (7) (2008) 659–670.
[26] M. P. Hilty, P. Guerci, Y. Ince, F. Toraman, C. Ince, Microtools enables automated quantification of capillary density and red blood cell velocity in handheld vital microscopy, Communications biology 2 (1) (2019) 1–15.
[27] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, L. Shapiro, Principal curve-based region detector for object recognition, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.
[28] R. Bezemer, J. G. Dobbe, S. A. Bartels, E. C. Boerma, P. W. Elbers, M. Heger, C. Ince, Rapid automatic assessment of microvascular density in sidestream dark field images, Medical & biological engineering & computing 49 (11) (2011) 1269–1278.
[29] J. Tam, J. A. Martin, A. Roorda, Noninvasive visualization and analysis of parafocal capillaries in humans, Investigative ophthalmology & visual science 51 (3) (2010) 1691–1698.
[30] S. U. Demir, R. Hakimzadeh, R. H. Hargraves, K. R. Ward, E. V. Myer, K. Najarian, An automated method for analysis of microcirculation videos for accurate assessment of tissue perfusion, BMC medical imaging 12 (1) (2012) 1–13.
[31] A. 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) (2004) 35–44.
[32] C. Cheng, C. W. Lee, C. Daskalakis, A reproducible computerized method for quantitation of capillary density using nailfold capillaroscopy, Journal of visualized experiments: JoVE 10379153088 (104) (2015).
[33] A. Tama, T. R. Mengko, H. Zakaria, Nailfold capillaroscopy image processing for morphological parameters measurement, in: 2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), IEEE, 2015, pp. 175–179.
[34] P. Prenta ş, V. M. Heisler, Z. Mammo, S. Lee, A. Merkur, E. Navajas, M. F. Beg, M. Sarunic, S. Loncaric, Segmentation of the foveal microvasculature using deep learning networks, Journal of biomedical optics 21 (7) (2016) 075008.
[35] R. Nivedha, M. Brinda, K. Suma, B. Rao, Classification of nailfold capillary images in patients with hypertension using non-linear svm, in: 2016 International Conference on Circuits, Controls, Communications and Computing (14C), IEEE, 2016, pp. 1–5.
[36] W. S. Noble, What is a support vector machine?, Nature biotechnology 24 (12) (2006) 1565–1567.
[37] L. S. Geyman, R. A. Garg, Y. Suwan, V. Trivedi, B. D. Krawitz, S. Mo, A. Pinhas, A. Tantwaworasin, T. Y. Chui, R. 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) (2017) 1261–1268.
[38] P. Javia, A. Rana, N. Shapiro, P. 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), IEEE, 2018, pp. 607–611.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[40] F. Ye, S. Yin, M. Li, Y. Li, J. Zhong, In-vivo full-field measurement of microcirculatory blood flow velocity based on intelligent object identification, Journal of biomedical optics 25 (1) (2020) 016003.
[41] C. Ning, H. Zhou, Y. Song, J. Tang, Inception single shot multibox detector for object detection, in: 2017 IEEE International Conference on Multimedia Expo Workshops (ICEMEW), 2017, pp. 549–554. doi:10.1109/ICEMEW.2017.8026312.
[42] L. Barba-Guaman, J. Eugenio Naranjo, A. Ortiz, Deep learning framework for vehicle and pedestrian detection in rural roads on an embedded arm, Electronics 9 (4) (2020) 589.
[43] Y. S. Hariyani, H. Eom, C. Park, Da-capnet: Dual attention deep learning based on u-net for nailfold capillary segmentation,
G. Corliss, Y. Chang, Solving ordinary differential equations using taylor series, ACM Transactions on Mathematical Software (TOMS) 8 (2) (1982) 114–144.

G. Bradski, A. Kaehler, OpenCV, Dr. Dobbs’s journal of software tools 3 (2000).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine Learning research 12 (2011) 2825–2830.

M. Abadi, Tensorflow; learning functions at scale, in: Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016, pp. 1–1.

G. Van Rossum, et al., Python programming language., in: USENIX annual technical conference, Vol. 41, 2007, p. 36.

Z.-Q. Zhao, P. Zheng, S.-s. Xu, X. Wu, Object detection with deep learning: A review, IEEE transactions on neural networks and learning systems 30 (11) (2019) 3212–3232.

A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.

W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

J. Borji, M.-M. Cheng, H. Jiang, J. Li, Salient object detection: A benchmark, IEEE transactions on image processing 24 (12) (2015) 5706–5722.

J. Borji, M.-M. Cheng, Q. Hou, H. Jiang, J. Li, Salient object detection: A survey, Computational visual media 5 (2) (2019) 117–150.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

Dnn model optimization series part i: What’s the drill? (2021).

E. C. Boerma, K. R. Mathura, P. H. van der Voort, P. E. Spronk, C. Ince, Quantifying bedside-derived imaging of microcirculatory abnormalities in septic patients: a prospective validation study, Critical care 9 (6) (2005) 1–8.

R. C. Arnold, J. E. Parrillo, R. P. Dellinger, M. E. Chansky, N. I. Shapiro, D. J. Lundy, S. Trzeciak, S. M. Hollenberg, Point-of-care assessment of microvascular blood flow in critically ill patients, Intensive care medicine 35 (10) (2009) 1761–1766.

V. S. K. Edul, C. Enrico, B. Laviolle, A. R. Vazquez, C. Ince, A. Dubin, Quantitative assessment of the microcirculation in healthy volunteers and in patients with septic shock, Critical care medicine 40 (5) (2012) 1443–1448.

A. Dubin, M. O. Pozo, C. A. Casabella, F. Pálizas, G. Murias, M. C. Mosenico, V. S. K. Edul, E. Estenssoro, C. Ince, Increasing arterial blood pressure with norepinephrine does not improve microcirculatory blood flow: a prospective study, Critical Care 13 (3) (2009) 1–8.

D. De Backer, S. Hollenberg, C. Boerma, P. Goedhart, G. Büchele, G. Osipina-Tascon, I. Dobbe, C. Ince, How to evaluate the microcirculation: report of a round table conference, Critical care 11 (5) (2007) 1–9.
newborn infants, Pediatric research 79 (6) (2016) 902–906.

[81] M. P. Hilty, J. Pichler, B. Ergin, U. Hefti, T. M. Merz, C. Ince, M. 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) (2017) 1–13.

[82] S. M. Hubble, H. L. Kyte, K. Gooding, A. C. Shore, Variability in sublingual microvessel density and flow measurements in healthy volunteers, Microcirculation 16 (2) (2009) 183–191.

[83] S. Fredly, D. Fugelseth, T. Wester, E. Haggblad, K. Kvernebo, Skin microcirculation in healthy term newborn infants—assessment of morphology, perfusion and oxygenation, Clinical hemorheology and microcirculation 59 (4) (2015) 309–322.

[84] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017, pp. 1–12.

[85] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[86] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.

[87] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 922–928.

[88] M. V. Volkov, N. B. Margaryants, A. V. Potemkin, M. A. Volynsky, I. P. Gurov, O. V. Mamontov, A. A. Kamshilin, Video capillaroscopy clarifies mechanism of the photoplethysmographic waveform appearance, Scientific reports 7 (1) (2017) 1–8.

[89] Y. Xu, S. Shan, Z. Qiu, Z. Jia, Z. Shen, Y. Wang, M. Shi, I. Eric, C. Chang, End-to-end subtitle detection and recognition for videos in east asian languages via cnn ensemble, Signal Processing: Image Communication 60 (2018) 131–143.