

Article

Classification of Red Blood Cells Using Time-Distributed Convolutional Neural Networks from Simulated Videos

Samuel Molčan , Monika Smiešková , Hynek Bachratý , Katarína Bachratá  and Peter Novotný * 

Department of Software Technology, Faculty of Management Science and Informatics, University of Žilina, 010 26 Žilina, Slovakia; samuel.molcan@fri.uniza.sk (S.M.); monika.smieskova@fri.uniza.sk (M.S.); hynek.bachraty@fri.uniza.sk (H.B.); katarina.bachrata@fri.uniza.sk (K.B.)

* Correspondence: peter.novotny@fri.uniza.sk

Abstract: The elasticity of red blood cells (RBCs) plays a vital role in their efficient movement through blood vessels, facilitating the transportation of oxygen within the bloodstream. However, various diseases significantly impact RBC elasticity, making it an important parameter for diagnosing and monitoring health conditions. In this study, we propose a novel approach to determine RBC elasticity by analyzing video recordings and using a convolutional neural network (CNN) for classification. Due to the scarcity of available blood flow recordings, computer simulations based on a numerical model are employed to generate a substantial amount of training data. The simulation model incorporates the representation of RBCs as elastic objects within a fluid flow, allowing for a detailed understanding of their behavior. We compare the performance of different CNN architectures, including ResNet and EfficientNet, for video classification of RBC elasticity. Our results demonstrate the potential of using CNNs and simulation-based data for the accurate classification of RBC elasticity.

Keywords: red blood cell; video classification; computer simulation; convolutional neural network



Citation: Molčan, S.; Smiešková, M.; Bachratý, H.; Bachratá, K.; Novotný, P. Classification of Red Blood Cells Using Time-Distributed Convolutional Neural Networks from Simulated Videos. *Appl. Sci.* **2023**, *13*, 7967. <https://doi.org/10.3390/app13137967>

Academic Editor: DaeEun Kim

Received: 19 June 2023

Revised: 30 June 2023

Accepted: 4 July 2023

Published: 7 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Red blood cells (RBCs) play a crucial role in the transportation of oxygen within the blood. The health of RBCs is vital to their ability to efficiently flow through blood vessels, facilitated by their deformable membrane [1]. Elasticity is a key characteristic of healthy RBCs and is influenced by various natural factors, including the age of the cells. However, the impact of different diseases such as malaria [2], leukemia [3], diabetes [4], or sickle cell disease [5–8] significantly affects RBC elasticity.

Classifying RBCs can provide valuable insights and benefits in various scientific and medical contexts. Firstly, the accurate classification of RBCs enables the identification and characterization of different cellular types, such as normal RBCs, sickle cells, and other abnormal cell morphologies [9]. This classification can aid in the diagnosis and monitoring of various blood disorders, including sickle cell disease, thalassemia [10], and hereditary spherocytosis [11]. By distinguishing between different types of RBCs, healthcare professionals can better understand disease progression, assess the severity of conditions, and tailor treatment strategies accordingly. A more general overview of the techniques and importance of capturing cells of different elasticity can be found in [12,13].

Furthermore, RBC classification can contribute to the understanding of physiological and pathological processes within the human body. For instance, variations in RBC morphology may be indicative of certain underlying health conditions or physiological changes [14]. By analyzing and classifying RBCs, researchers can investigate the impact of factors such as nutrition, disease states, genetic variations, and environmental exposures on RBC characteristics. This knowledge can provide valuable insights into the mechanisms underlying various diseases and conditions, as well as facilitate the development of novel therapeutic approaches.

Moreover, RBC classification has implications beyond clinical and research settings. Industries involved in blood transfusion services and blood banking heavily rely on accurate identification and classification of RBCs to ensure compatibility and safety in transfusion procedures. Classifying RBCs according to specific markers or characteristics allows for precise matching between donors and recipients, minimizing the risk of adverse transfusion reactions.

This study aims to determine the elasticity of RBCs by analyzing the geometric characteristics captured in video recordings. Unfortunately, there is a scarcity of available blood flow recordings, and the limited number of videos is insufficient to train a neural network (NN) adequately, even with data augmentation techniques. One alternative approach is to employ computer simulations based on a numerical model that can be tailored to the experimental requirements, such as the desired blood flow properties and structure. Computer simulations offer a more detailed understanding of the behavior of the investigated blood flow compared to limited video recordings. Therefore, the integration of computer simulations in the study of hemodynamics and the development of new diagnostic techniques holds significant value. The simulation model used in this study is an extension of ESPResSo and incorporates the representation of blood cells as elastic objects within a fluid flow. The validity of the model is continuously assessed by comparing simulation results with laboratory experiments [15]. By utilizing simulations, a substantial amount of data for training NNs can be generated, particularly from video analyses of such experiments. The simulation model employed in our study controls the elasticity of RBCs using five parameters, which are described in detail in Section 3.

The determination of RBC elasticity through simulation experiments has been investigated in prior works [16,17]. However, this paper proposes a novel approach capable of identifying various levels of RBC elasticity associated with different diseases from video recordings. The classification of videos is accomplished using a convolutional neural network (CNN).

The simulation model of blood flow brings new possibilities for the analysis of blood flow and its properties by NN. We can observe the phenomena resulting from the current knowledge about the behavior of blood cells and their properties on videos of real blood flow experiments, recorded on cameras. However, these recordings are always only two-dimensional, and information about the third dimension can only be obtained by adding a recording of the same flow from another, independent camera, or from other shots. If we could create a sufficiently accurate simulation model, we could see how exactly we can identify cell properties from a 2D image of this simulation and, more specifically, how the information and characteristics obtained from the video may differ from the characteristics obtained from the complete model information. In this paper, we do not primarily analyze how well the simulation models the real experiment, but investigate the possibilities and success of using NN to compare the success of categorizing cells of different elasticities when we know the complete information (the exact position of the surface particles, all three dimensions) with the case when we only know the video recording; that is, the two-dimensional information. The use of CNN to recognize the elasticity of a moving blood cell is our second attempt to deal with this topic. In previous work, we used statistical methods and multidimensional data analysis [18]. We applied CNN in blood flow analysis to predict the trajectory of red blood cells in [19].

CNNs, inspired by biological systems, are widely employed in deep learning for image and video classification tasks. The primary advantage of CNNs lies in their ability to analyze the semantic content of images while explicitly leveraging the spatial structure through restricted connectivity between layers (local filters), parameter sharing (convolutions), and specialized neurons for local invariance (max pooling). CNN architectures have demonstrated their ability to learn interpretable image features [20]. Consequently, these architectures effectively shift the focus from manual feature design to the design of network connectivity structures and appropriate hyperparameter choices.

2. Materials and Methods

Video classification involves the assignment of one or more labels to a video based on its content. The utilization of NNs for video classification has gained popularity due to their ability to learn intricate patterns within the data. CNNs are particularly suitable for video classification tasks as they can effectively capture spatial and temporal features. CNNs consist of multiple layers of neurons that are trained to extract relevant features from images or video frames.

Typically, NNs used for video classification are trained on a large dataset of labeled videos. These videos are partitioned into training and validation sets to assess the performance of the model. To increase the diversity of the training data, various data augmentation techniques, such as flipping, rotating, and scaling, can be employed. Additionally, transfer learning can expedite the training process by utilizing pre-trained models that were previously trained on different datasets.

However, video classification poses challenges due to variations in lighting conditions, camera angles, and object appearances. Domain adaptation techniques can address these challenges by transferring knowledge from a source domain to a target domain.

The application of video classification using NNs spans various fields, including surveillance, entertainment, and healthcare. In healthcare, video classification can aid in the analysis of medical videos such as endoscopic videos, facilitating the detection of abnormalities and assisting in diagnosis. The evaluation of video classification performance using NNs can be measured using metrics such as accuracy, precision, and recall. The choice of evaluation metric depends on the specific application and the relative importance of different types of errors. Overall, video classification using NNs is an actively evolving field with numerous promising applications and associated challenges.

2.1. ResNet

Residual Network' (ResNet) [21] is a popular type of CNN architecture that was originally designed for image classification; see Figure 1. However, ResNet can also be used for video classification by adapting it to process multiple frames of a video sequence.

In a traditional CNN, each layer processes the output of the previous layer to extract increasingly complex features from the input image. However, as the network becomes deeper, it can become harder to train and can suffer from the vanishing gradient problem. This problem occurs when the gradients used to update the weights of the network become very small, making it difficult to learn from the data.

ResNet solves this problem by introducing residual connections between layers. These connections allow the network to "skip" over certain layers, allowing for information to pass through unchanged. This helps to prevent the gradients from becoming too small, allowing the network to learn more effectively.

To use ResNet for video classification, we can apply it to each frame of the video sequence and then combine the outputs from each frame to obtain a final classification. This can be achieved by either averaging the outputs or using an attention mechanism to focus on the most relevant frames.

One popular implementation of ResNet for video classification is the Two-Stream ResNet, which consists of two ResNet networks: one for processing spatial information (the appearance of the objects in the video) and one for processing temporal information (the motion of the objects in the video). The spatial network processes each frame of the video independently, while the temporal network processes pairs of frames to capture motion information. The outputs from both networks are then combined to obtain a final classification.

Overall, using ResNet for video classification can be an effective approach due to its ability to learn complex features and its use of residual connections to prevent the vanishing gradient problem.

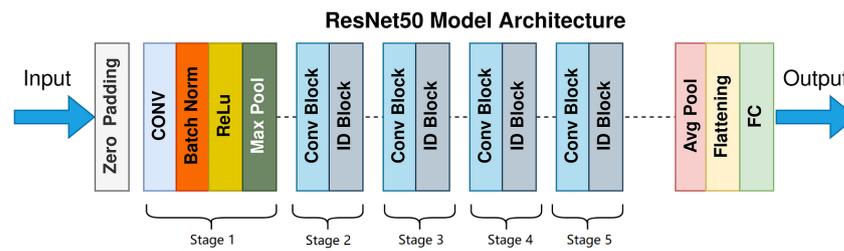


Figure 1. ResNet50 architecture [22].

2.2. EfficientNet

EfficientNet [23,24] is a family of CNNs that were specifically designed to be more efficient in terms of computation and parameter usage than existing CNNs. EfficientNet achieves this by using a novel scaling method that optimizes the network architecture (Table 1) based on the available computational resources.

EfficientNet can also be used for video classification by adapting it to process multiple frames of a video sequence. One way to do this is to use a 3D CNN architecture, which can capture both spatial and temporal information from the video frames.

To use EfficientNet for video classification, we can apply the 3D CNN to each frame of the video sequence and then combine the outputs from each frame to obtain a final classification. This can be achieved by either averaging the outputs or using an attention mechanism to focus on the most relevant frames.

One advantage of using EfficientNet for video classification is that it is highly efficient in terms of computation and parameter usage. This can be important for applications where resources are limited, such as on mobile devices or in real-time video analysis. EfficientNet can also achieve high accuracy on a variety of image classification tasks, which suggests that it may also be well-suited for video classification.

However, there are some challenges associated with using EfficientNet for video classification. One challenge is that video classification typically requires processing a large number of frames, which can be computationally intensive. Another challenge is that EfficientNet may not be as effective at capturing temporal information as other CNN architectures that were specifically designed for video classification, such as the Two-Stream CNN or the 3D ResNet.

Overall, using EfficientNet for video classification can be an effective approach due to its efficiency and high accuracy in image classification tasks. However, it may require additional optimization and tuning to achieve an optimal performance on video classification tasks.

Table 1. EfficientNet v2 B0 architecture [23].

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	224×224	32	1
2	MBCConv1, k3x3	112×112	16	1
3	MBCConv6, k3x3	112×112	24	2
4	MBCConv6, k5x5	56×56	40	2
5	MBCConv6, k3x3	28×28	80	3
6	MBCConv6, k5x5	28×28	112	3
7	MBCConv6, k5x5	14×14	192	4
8	MBCConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

3. Data Preparation

The source data used to classify the health of RBCs were obtained from multiple simulation experiments. These experiments were conducted using the open-source software ESPResSo [25], incorporating its lattice-Boltzmann and object-in-fluid modules [26].

Numerical simulations consisted of a model of an elastic object representing an RBC embedded in a flowing fluid and interactions between individual objects (cell–cell and cell–wall/obstacle). The model utilized the lattice–Boltzmann method to represent the fluid, a spring network model to simulate the cell membrane, and a dissipative version of the Immersed Boundary Method (IBM) to connect them. In addition to the fluid force, elastic forces are exerted on the cell mesh points that are evaluated from the deformation of the cell. The resultant force \vec{F}_{tot} is the driving force according to which the mesh points are propagated in space following Newton’s equation

$$m \frac{\partial^2 \vec{x}(t)}{\partial t^2} = \vec{F}_{tot},$$

where m is the mass of the mesh points. The sources of are the elasto-mechanical properties of the cell membrane, the fluid–cell interaction or possibly other external stimuli.

All simulations were performed under consistent channel and fluid flow parameters. The channel had a cuboid shape with dimensions of $104 \times 60 \times 40 \mu\text{m}$, and the fluid was discretized into a three-dimensional grid with a spatial step of $1 \mu\text{m}$. The elasticity of RBCs becomes most apparent when they come into contact with other objects. Therefore, we designed a simulated channel topology where RBCs flow through a space with obstacles. The simulated channel featured five cylinders, acting as obstacles that restricted the area of blood flow and induced the manifestation of RBC elasticity (see Figure 2). This channel design aimed to replicate a realistic laboratory environment. The kinematic viscosity of the fluid was $1.3 \times 10^{-6} \text{ m}^2/\text{s}$, and the density was $1.025 \times 10^3 \text{ kg}/\text{m}^3$. To initiate fluid flow, external forces were applied, with values chosen to achieve a maximum velocity of approximately $0.03 \text{ m}/\text{s}$.

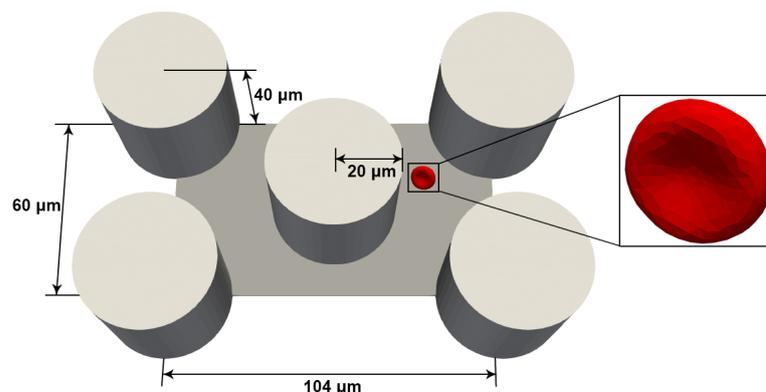


Figure 2. Scheme of the simulation microfluidic channel with cylindrical obstacles.

Cell–cell interactions were simulated using the `membrane_collision` potential, while interactions between cells and the channel walls were modeled using the `soft_sphere` potential. RBCs were represented by a surface network consisting of 374 nodes. The elastic properties of the cells were simulated using five types of elastic forces (shown in Table 2, each corresponding to a different elastic modulus.

Table 2. Overview of the used simulation parameters.

Coefficient	Value
stretching coefficient (k_s)	$5 \times 10^{-6} \text{ N}/\text{m}$
bending coefficient (k_b)	$3 \times 10^{-19} \text{ Nm}$
coefficient of local area conservation (k_{al})	$2 \times 10^{-5} \text{ N}/\text{m}$
coefficient of global area conservation (k_{ag})	$7 \times 10^{-4} \text{ N}/\text{m}$
coefficient of volume conservation (k_v)	$900 \text{ N}/\text{m}^2$

In this study, we focused on four levels of RBC elasticity. Two levels represented healthy RBCs, with the most elastic RBCs having a stiffness coefficient (k_s) of 0.005, and the least elastic RBCs representing malaria-infected cells at stage 3 of the disease with a k_s value of 0.03 (The value of $k_s = 0.03$ was chosen based on the reduced elasticity observed in malaria-infected cells at stage 3, as determined by an optical tweezers stretching experiment [27]). The remaining two levels of RBC elasticity were evenly distributed between the healthy and malaria-infected RBCs, with k_s values of 0.0133 and 0.0216, respectively.

The size of the training dataset plays a critical role in the training of machine learning (ML) models. When the dataset is small, the model may struggle to capture the intricate patterns and nuances present in the data, resulting in a poor generalization performance on unseen data. Conversely, a larger dataset provides the model with more examples to learn from, leading to better generalization and reduced overfitting where the model becomes too specialized to the training data. Therefore, it is crucial to have a sufficiently large training dataset to develop accurate and reliable ML models.

However, it is important to consider the computational resources required to train models on large datasets. Larger datasets demand more computational power and longer training times. Thus, striking a balance between dataset size and available computational resources is necessary for successful model training.

Each simulation in our study involved 36 RBCs, with nine cells representing each level of elasticity. The simulated channel has a periodic structure, meaning that when an RBC exits the channel, it reappears at the beginning. We remove the initial and final passes of RBCs due to their incompleteness. As a result, each RBC completes approximately 20–21 passes through the simulated channel (Figure 3). Consequently, each part of the train/validation datasets contains approximately

$$4 \text{ types of RBC} \times 9 \text{ from each type} \times 21 \text{ passes} = 756 \text{ samples.}$$

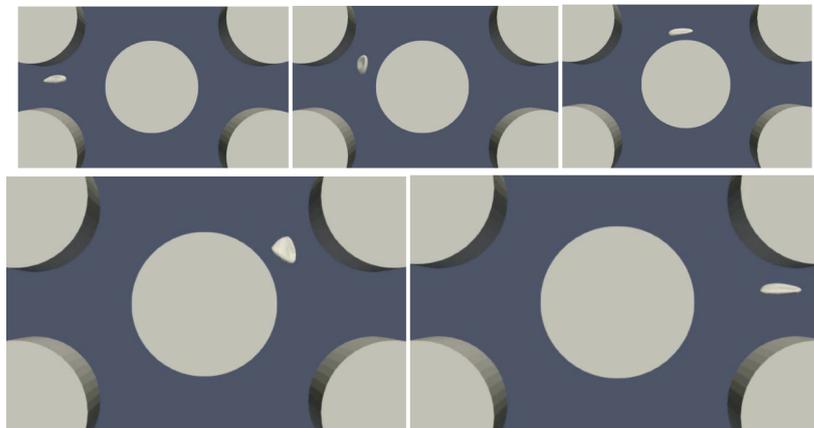


Figure 3. One pass of a red blood cell (RBC) through the simulated channel.

Given the small number of training examples, we have two options to address this limitation. First, we can utilize data augmentation techniques to augment our training set and increase its size. Data augmentation involves applying transformations to the existing examples to create new variations. In our case, we can perform vertical flips (to preserve the direction of blood flow) and rotations (while considering the preservation of blood flow direction). By applying these augmentations, we can generate additional training examples and enhance the diversity of the dataset.

The second option is to leverage pretrained models, which do not require training from scratch and hence need fewer training examples. Pretrained models are pre-trained on large-scale datasets and have learned general features. We chose two pretrained models, EfficientNet v2 B0 and ResNet50, which have demonstrated successful results in various image and video tasks [28–31]. These models only require a sufficient number of training

examples to fine-tune the last few layers of the NN, making them well-suited to our small training dataset.

To prepare the dataset, we generated video recordings of individual RBCs from the simulation data. These videos serve the purpose of training and validation. Since the simulations performed in ESPResSo provide three-dimensional information about the flow of simulated RBCs, we projected these data onto a two-dimensional plane. We achieved this by generating a 2D video that captures the width and length of the channel while disregarding differences in depth. This transformation enabled us to effectively analyze and classify the RBC behavior in the videos using image and video classification techniques.

4. Results and Discussion

Our network used video samples in the shape of $N \times T \times H \times W \times C$, where N represents the batch size, T is the number of frames in the video, H is the height, W is the width, and C is the number of channels, which is 3 (red, green, blue). Then, we rescaled the video into black and white format, which reduced the number of channels from 3 to 1.

For the base of our models, we used EfficientNet v2 B0 and ResNet50, which are pretrained models. Since the two pretrained models we used are intended for image classification, we used a time-distributed layer, which enables the model to classify video recordings. The pretrained model is not trainable. This was followed by 3D Average Pooling. We finished our network with a sequence of Flatten, Dropout, and Dense layers (Figure 4), which led to an output with the number of neurons equal to the number of classes; in our case, four neurons. We used Dropout and regularization for the last dense layer to avoid overfitting. The network's optimizer is either Adam or SGD.

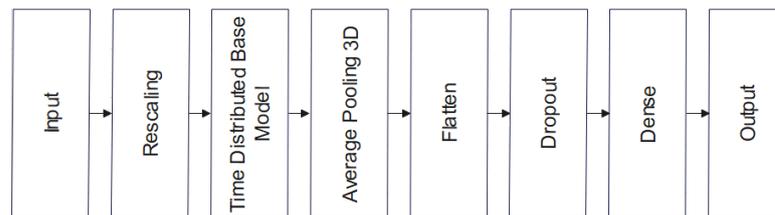


Figure 4. Diagram of the network architecture. Base model can be one of EfficientNet v2 B0 or ResNet50.

We optimized the hyperparameters of our network using the *Hyperband* class from the *keras_tuner* module [32]. The optimized hyperparameters differ slightly based on the network's optimizer, and the differences can be seen in Table 3.

The result of the hyperparameter optimization for each type of network and each type of optimizer is shown in Table 4. The best accuracy is achieved with EfficientNet v2 B0 model using Adam optimizer for the hyperparameters shown in Table 5. The validation results for each type of model are poor while the training accuracies are high, which indicates two things. First, the networks are overfitting for each architecture and optimizer, despite the regularization method used to avoid this effect. Second, the presence of overfitting suggest there is some information in the data that can be learnt.

Table 3. Hyperparameters optimized for Adam and SGD.

Hyperparameter	Adam	SGD	Options
Dropout	yes	yes	0.05, 0.1, 0.2, 0.3
Regularizer	yes	yes	L1, L2, L1 + L2
Initial learning rate	yes	yes	1×10^{-3} , 1×10^{-4} , 1×10^{-5}
Final learning rate	yes	no	1×10^{-5} , 1×10^{-6} , 1×10^{-7}
Decay steps	yes	no	1×10^4 , 1×10^5 , 1×10^6
Momentum	no	yes	0.6, 0.8, 1.0

Table 4. Validation accuracies for the models with 4 classes using optimized hyperparameters.

Neural Network Model	Optimizer	Validation Accuracy	
		4 Classes	2 Classes
EfficientNet v2 B0	Adam	55.48%	61.72%
	SGD	46.86%	56.44%
ResNet50	Adam	54.26%	55.12%
	SGD	51.45%	58.25%

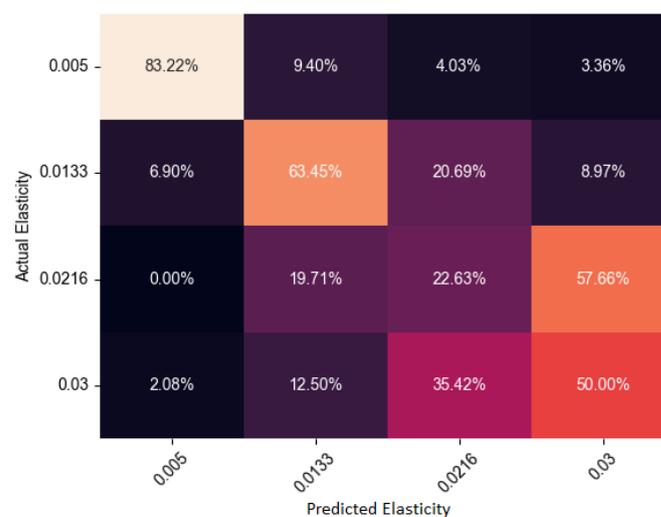
Table 5. The best set of hyperparameters from Adam optimizer.

Hyperparameter	Value
Dropout	0.1
Regularizer	L1 + L2
Initial learning rate	1×10^{-5}
Final learning rate	1×10^{-5}
Decay steps	1×10^4

We observe from the confusion matrix for the validation set (Figure 5) that the main problem of classification is in distinguishing between RBCs with the reduced elasticity, while its ability to distinguish between the healthy and sick RBCs is more precise. Therefore, we decided to train a binary classification NN, where the first class contains the healthy RBCs ($k_s = 0.005$) and the second class consists of all the cells with reduced elasticity. We ran the hyperparameter optimization for 2 types of NNs with 2 different optimizer options.

From the confusion matrix shown in Figure 6 we discovered that our hypothesis is not true. After changing the classification problem from 4 classes to 2 classes, the final accuracies increased, but not as significantly as the confusion matrix from Figure 5 suggests. We also tried a weighted classification for a combination of EfficientNet v2 B0 and the Adam optimizer where the final accuracy insignificantly increased to 61.74%, showing only a 0.02% increase.

There is an option to separate healthy and sick RBCs after they are classified into 4 classes, where class 0 is healthy and classes 1, 2 and 3 are sick. In this manner, we achieved a classification accuracy of 93.54%. We elaborate on this in the conclusion.

**Figure 5.** The confusion matrix represents the classification results of the validation set, which consists of samples belonging to 4 classes. The classification was performed using the EfficientNet v2 B0 model, which was optimized with the Adam optimization algorithm.

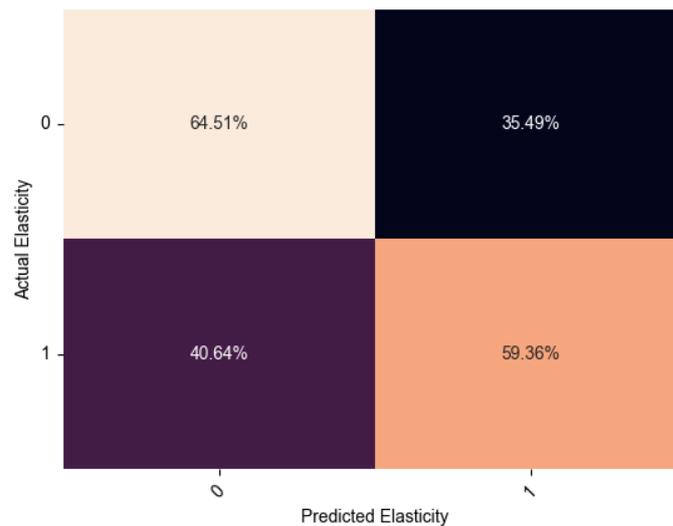


Figure 6. The confusion matrix represents the classification results of the validation set, which consists of samples belonging to 2 classes. The classification was performed using the EfficientNet v2 B0 model, which was optimized with the Adam optimization algorithm. The classes of healthy and sick are represented by the labels 0 and 1, respectively.

4.1. Adding Physical Information

In order to enhance the performance of the network, we added information about the underlying physics—specifically, about the velocity of the fluid flowing in the channel. Our hypothesis was that by adding the physical information, the NN would be able to learn to classify the RBCs better than without it.

First, using ESPReSso, we calculated the velocities of each point of the fluid (which is represented as a mesh; more information is available in [33]) in the empty channel—the channel with no RBCs in it. We obtained a 3D data of velocities, since the channel is a 3D object. Then, we took a layer $H \times W \times \lfloor D/2 \rfloor$ that corresponds to the velocities in the middle of the channel along the depth axis. This information can be used in two ways: we can either use it as it is, meaning 3D data with dimensions $W \times H \times 3$, where 3 represents the x , y , and z components of a velocity vector, or we can create a heatmap where the colors represent the velocity of the flow.

In both options, we added the physical information by creating a new branch of our NN that has the physical information about the flow as input. This information is passed through a Rescale layer, three 2D convolution layers with a number of filters 32, 16, 8, followed by a max pooling layer, a flattening layer, and a dense layer with 1024 hidden units. Then, it is concatenated with the penultimate output of the main branch of our NN, which is passed to the last dense layer.

Generally, it is not effective to add the same information to each training example for a NN. The reason for this is that NNs learn patterns and relationships in the data through the variations and differences between the examples. When all examples contain the same information, the network is unable to distinguish between them and may not learn the relevant patterns that are necessary for accurate predictions. However, there may be some cases where adding the same information to each training example can be helpful. For example, if the added information provides some contextual or background information that is relevant to all examples, it may help the network learn more effectively. In general, it is important to carefully consider the information that is added to each training example and how it may affect the network's ability to learn and generalize. We trained both versions of PINNs using the optimized hyperparameters described in Table 5, with the best performing model being EfficientNet v2 B0 with the Adam optimizer. The final results are presented in Table 6.

Table 6. The validation accuracies were evaluated for six distinct classes of models using optimized hyperparameters. The names of these neural network (NN) models consist of two parts. The first part represents the number of classes used during training, while the second part indicates the type of physical information employed for the physics-informed neural network (PINN). The abbreviations “4c”, “2c”, and “2cw” denote four classes, two classes, and two weighted classes, respectively. Additionally, “Heatmap” and “Values” signify the utilization of heatmap information and velocity vectors from the middle flow layer, respectively.

NN Model	Accuracy
4c_heatmap	87.83%
4c_values	91.48%
2c_heatmap	57.04%
2c_values	67.33%
2cw_heatmap	59.13%
2cw_values	69.78%

The obtained results highlight several important observations regarding the performance of different classification approaches. Firstly, when comparing the two-class classification (whether weighted or unweighted) to the four-class classification converted to two-class classification, it is evident that the latter achieves significantly better results. This suggests that the additional information present in the four-class classification helps to improve the overall accuracy of the model.

Furthermore, the inclusion of physics-related information in the form of a heatmap appears to be less effective than the use of velocity vectors as input. This implies that the velocity vectors carry more meaningful and discriminative information for accurate classification. The use of velocity vectors as input likely allows for the model to capture dynamic patterns and better understand the motion characteristics of the analyzed data.

However, it is interesting to note that even though velocity vector information is incorporated, it does not lead to a substantial improvement in the final accuracy of the network. Specifically, the NN that solely relies on the provided input without any additional physical information achieves an accuracy of 93.56%. In contrast, the network utilizing velocity vectors achieves a slightly lower accuracy of 91.48%. This suggests that while the inclusion of velocity vectors may provide some useful information, it does not necessarily translate into a significant boost in classification performance.

4.2. Up-Scaling of the Healthy Examples

Our simulations include four types of RBCs, of which three types have reduced elasticity and are considered sick. Although the ratio of healthy to sick RBCs is 1:3, this does not correspond to the observed ratio in reality [34]. Therefore, we used data augmentation to upscale the minority class and balance the class sizes. As described in Section 3, we used horizontal flip and rotation to add two new training examples for each original example. This augmentation, together with the original healthy examples, makes the healthy class three times larger.

We train the best performing model from our previous experiments, a four-class classification NN with no additional information, which is then reduced to a two-class classification. We augmented the training dataset and the validation dataset to maintain consistency between the class distribution of the datasets. After training and validating the model on the dataset with the same distribution, we cross-validated the model on the dataset with a different class ratio.

Table 7 provides a comparison of three models: the original model (O_4x1) and the best-performing model from previous experiments, along with two models trained on datasets with equal ratios of healthy and unhealthy RBCs. The first model uses unmodified class weights (A_3_1_1_1), resulting in equal weights for each class. The second model (AW_3_1_1_1), on the other hand, uses class weights proportional to the sizes of the

classes. We observed that the second model outperforms the first model in terms of classification accuracy.

The results indicate that the original dataset achieved the highest accuracy on the original dataset for all three models. The O_4x1 model achieved an accuracy of 93.54%, while the A_3_1_1_1 and AW_3_1_1_1 models achieved lower accuracies of 88.88% and 81.55%, respectively.

When comparing the accuracies on the augmented dataset, it can be observed that the A_3_1_1_1 model performed slightly better, with an accuracy of 93.91%, compared to the 91.01% accuracy of the O_4x1 model. However, the AW_3_1_1_1 model achieved an accuracy of 92.21%, which was slightly lower than both the original dataset and the A_3_1_1_1 model.

Based on these findings, it can be concluded that augmenting the dataset through upsampling had a mixed impact on the model performance. While the A_3_1_1_1 model showed a slight improvement, the addition of class weights in the AW_3_1_1_1 model did not yield significant improvements and even resulted in a slightly decreased accuracy compared to the original dataset.

Table 7. The validation accuracies of three models with identical architectures were assessed. The first model, referred to as O_4x1, was trained on the original dataset, which consisted of a roughly equal number of examples per class. The second model, denoted as A_3_1_1_1, was trained on an upsampled dataset, while the third model, labeled as AW_3_1_1_1, utilized the upsampled dataset with additional class weights proportional to their respective proportions.

Validation	O_4x1	A_3_1_1_1	AW_3_1_1_1
original dataset	93.56%	88.88%	81.55%
augmented dataset	91.01%	93.91%	92.21%

4.3. Four-Class to Two-Class Classification

The obtained results reveal important insights into the performance of various classification approaches. One notable observation is the comparison between the two-class classification and the four-class classification converted to two-class classification. The two-class classification refers to the classification task where the model distinguishes between two specific classes (e.g., healthy and sick), while the four-class classification converted to two-class classification involves combining multiple classes into two broader categories.

The results indicate that the four-class classification converted to two-class classification yields a significantly better performance than two-class classification. This suggests that the inclusion of additional classes in the training process provides valuable information that enhances the overall accuracy of the model. By training the model on a dataset consisting of multiple classes, it can capture a wider range of patterns, variances, and characteristics present in the data. This broader understanding and increased complexity of the model contribute to an improved classification accuracy when distinguishing between the two broader categories in the four-class classification converted to two-class classification scenario.

The enhanced performance achieved with the four-class classification converted to two-class classification demonstrates the importance of considering a more comprehensive representation of the data during model training. By incorporating additional classes, the model can learn more nuanced and discriminative features, leading to better differentiation between the target categories. This finding highlights the value of leveraging the full range of available classes and their associated information when constructing classification models.

The obtained results highlight the importance of carefully designing the classification task and selecting the appropriate representation of classes to achieve optimal performance. By leveraging the additional information provided by the four-class classification, the model gains a deeper understanding of the underlying data, resulting in improved accuracy when

distinguishing between broader categories. This underscores the significance of considering the relevance and inclusion of additional information in classification tasks.

Furthermore, the findings also underscore the importance of selecting appropriate input features in classification models. In this case, the inclusion of velocity vectors as input features was found to be more effective compared to the use of a heatmap representation. This suggests that the choice of input features plays a critical role in capturing relevant patterns and characteristics for accurate classification.

Moreover, the study highlights that the network's architecture and optimization techniques are crucial factors influencing the final performance. It is important to consider the interplay between the model's architecture, training algorithms, and the specific classification task at hand. Fine-tuning these components and exploring alternative approaches may further improve the accuracy and performance of the classification model.

Overall, the results emphasize the need for the careful consideration of various factors, including the classification task design, representation of classes, selection of input features, network architecture, and optimization techniques, to achieve optimal classification accuracy. Further analysis, experimentation, and refinement of methods are warranted to advance our understanding and improve the accuracy of classification tasks.

4.4. Discussion

Accurate physical and numerical measurements of RBC deformation rate are very complicated and difficult to implement when testing a blood sample that is not technologically and financially demanding. Therefore, in medical experiments, there is a prevailing tendency to replace the measurement of elastic properties with the observation of other, more easily detectable and correlated properties of RBC behavior in the flow of a blood sample in a suitable microfluidic device [15,35].

The use of CNN and ML in the field of research and simulation of RBC properties and behavior is not the ultimate method. Our intention is to gradually verify the potential of this approach, which we addressed in [17,19]. Similarly, in this article, our goal was to find out whether NNs can compensate the loss of accuracy and dimension of the image data when processing video recordings with computational complexity. The issue of using NNs in biology and medicine is the lack of training and verification data, and the need for manual labeling. In our case, this difficulty is circumvented by the use of a verified simulation model that produces a large amount of data with known parameters of the used RBCs. After verifying the correctness of the predictions of the CNN trained in this way, the next step will be to determine whether the data obtained exclusively from image recordings of real experiments are equally sufficient.

5. Conclusions

In this study, we investigated the classification of RBC elasticity using video recordings and CNNs. We addressed the scarcity of available blood flow recordings by employing computer simulations based on a numerical model. Our simulation model successfully captured the behavior of RBCs as elastic objects within a fluid flow, generating a large amount of training data for CNN-based classification. By analyzing the geometric characteristics captured in video recordings, we classified RBC elasticity using different CNN architectures, including ResNet and EfficientNet.

Our results demonstrate that CNNs can effectively classify RBC elasticity, providing a potential diagnostic tool for various diseases impacting RBC health. The integration of computer simulations and CNN-based video analysis offers a valuable approach to understanding the behavior of RBCs and developing new diagnostic techniques. The use of simulation-generated data provides a rich source of training examples, enabling the training of CNNs even with limited video recordings.

Furthermore, our study compared the performance of different CNN architectures, and the results showed that the EfficientNet model achieved the best accuracy for RBC elasticity classification. However, we observed that overfitting was a challenge for both ResNet

and EfficientNet architectures, indicating the need for further regularization techniques to improve the model's generalization performance.

We have observed that training the network in a higher target space (four classes) and subsequently mapping the results to a lower target space (two classes) yields the best classification performance for our problem. Another important finding is that cross-validation, where a model trained on a dataset with a different ratio of class examples is validated on another dataset with a different ratio of class examples, results in a significantly lower performance compared to validation on the same type of dataset. This observation highlights the importance of training the model on a dataset with a class ratio that matches the ratio observed in real-world scenarios. In our experiments, the RBC health is determined only by elasticity, but in reality, other RBC properties can also change, such as volume, the shape of the blood cell, etc. For this reason, further research is necessary.

In conclusion, our study highlights the potential of combining computer simulations, video analysis, and CNNs for the classification of RBC elasticity. Further research and optimization of CNN architectures and regularization techniques are warranted to enhance the accuracy and robustness of the classification model. The development of accurate and reliable diagnostic tools for assessing RBC health can have significant implications for disease diagnosis and monitoring, ultimately contributing to improved healthcare outcomes.

Author Contributions: Conceptualization, K.B., H.B. and M.S.; methodology, S.M. and M.S.; software, S.M. and M.S.; validation, S.M. and M.S.; formal analysis, S.M.; investigation, S.M., M.S. and P.N.; resources, M.S. and H.B.; data curation, M.S.; writing—original draft preparation, S.M., H.B. and M.S.; writing—review and editing, H.B., K.B., S.M., M.S. and P.N.; visualization, S.M. and M.S.; supervision, K.B.; project administration, K.B.; funding acquisition, K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the contract No. VEGA 1/0369/22.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are freely available in the GIT repository: https://github.com/molcan23/RBC_NN (accessed on 13 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ESPreSo	Extensible Simulation Package for Research on Soft Matter
CIF	Cell in Fluid, Biomedical Modeling & Computation Group
CNN	Convolutional Neural Network
ML	Machine Learning
NN	Neural Network
RBC	Red Blood Cell
SGD	Stochastic Gradient Descent

References

1. Mills, J.P.; Qie, L.; Dao, M.; Lim, C.T.; Suresh, S. Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers. *Mol. Cell. Biomech.* **2004**, *1*, 169–180. [[CrossRef](#)]
2. Rizzuto, V.; Mencattini, A.; Álvarez-González, B.; Di Giuseppe, D. Combining microfluidics with machine learning algorithms for RBC classification in rare hereditary hemolytic anemia. *Sci. Rep.* **2021**, *11*, 13553. [[CrossRef](#)]
3. Diehl, L.F.; Ketchum, L.H. Autoimmune disease and chronic lymphocytic leukemia: Autoimmune hemolytic anemia, pure red cell aplasia, and autoimmune thrombocytopenia. *Semin. Oncol.* **1998**, *25*, 80–97. [[PubMed](#)]
4. Agrawal, R.; Smart, T.; Nobre-Cardoso, J.; Richards, C. Assessment of red blood cell deformability in type 2 diabetes mellitus and diabetic retinopathy by dual optical tweezers stretching technique. *Sci. Rep.* **2016**, *6*, 15873. [[CrossRef](#)] [[PubMed](#)]

5. Brandão M.M.; Fontes, A.; Barjas-Castro, M.L.; Barbosa, L.C.; Costa, F.F.; Cesar, C.L.; Saad, S.T. Optical tweezers for measuring red blood cell elasticity: Application to the study of drug response in sickle cell disease. *Eur. J. Haematol.* **2003**, *70*, 207–211. [[CrossRef](#)] [[PubMed](#)]
6. Faivre, M.; Renoux, C.; Bessaa, A.; Da Costa, L.; Joly, P.; Gauthier, A.; Connes, P. Mechanical signature of red blood cells flowing out of a microfluidic constriction is impacted by membrane elasticity, cell surface-to-volume ratio and diseases. *Front. Physiol.* **2020**, *11*, 576. [[CrossRef](#)]
7. Suresh, S. Mechanical response of human red blood cells in health and disease: Some structure-property-function relationships. *J. Mater. Res.* **2006**, *21*, 1871–1877. [[CrossRef](#)]
8. Faivre, M.; Bessaa, A.; Costa, L.D.; Joly, P.; Connes, A.G. Impact of surface-area-to-volume ratio, internal viscosity and membrane viscoelasticity on red blood cell deformability measured in isotonic condition. *Sci. Rep.* **2019**, *9*, 6771. [[CrossRef](#)]
9. Klinken, P.S. Red blood cells. *Int. J. Biochem. Cell. Biol.* **2002**, *34*, 1513–1518. [[CrossRef](#)]
10. Advani, R.; Sorenson, S.; Shinar, E.; Lande, W.; Rachmilewitz, E.; Schrier, S.L. Characterization and comparison of the red blood cell membrane damage in severe human alpha-and beta-thalassemia. *Blood* **1992**, *79*, 1058–1063. [[CrossRef](#)]
11. Hassoun, H.; Palek, J. Hereditary spherocytosis: A review of the clinical and molecular aspects of the disease. *Blood Rev.* **1996**, *10*, 129–147. [[CrossRef](#)] [[PubMed](#)]
12. Bayareh, M. Active cell capturing for organ-on-a-chip systems: A review. *Biomed. Eng.-Biomed. Tech.* **2022**, *67*, 443–459. [[CrossRef](#)] [[PubMed](#)]
13. Nilsson, J.; Evander, M.; Hammarström, B.; Laurell, T. Review of cell and particle trapping in microfluidic systems. *Anal. Chim. Acta* **2009**, *649*, 141–157. [[CrossRef](#)] [[PubMed](#)]
14. Huisjes, R.; Bogdanova, A.; van Solinge, W.W.; Schiffelers, R.M.; Kaestner, L.; van Wijk, R. Squeezing for life—Properties of red blood cell deformability. *Front. Physiol.* **2018**, *9*, 656. [[CrossRef](#)] [[PubMed](#)]
15. Kovalčíková, K.; Cimrák, I.; Bachratá, K.; Bachratý, H. Comparison of Numerical and Laboratory Experiment Examining Deformation of Red Blood Cell. In *Bioinformatics and Biomedical Engineering, Proceedings of the 7th International Work-Conference on Bioinformatics and Biomedical Engineering, Granada, Spain, 8–10 May 2019*; Springer: Cham, Switzerland, 2019; pp. 75–86. [[CrossRef](#)]
16. Bachratá, K.; Buzáková, K.; Chovanec, M.; Bachratý, H.; Smiešková, M.; Bohiniková, A. Classification of Red Blood Cell Rigidity from Sequence Data of Blood Flow Simulations Using Neural Networks. *Symmetry* **2021**, *13*, 938. [[CrossRef](#)]
17. Molčan, S.; Smiešková, M.; Bachratý, H.; Bachratá, K. Computational Study of Methods for Determining the Elasticity of Red Blood Cells Using Machine Learning. *Symmetry* **2022**, *14*, 1732. [[CrossRef](#)]
18. Bachratá, K.; Bachratý, H.; Slavík, M. Statistics for comparison of simulations and experiments of flow of blood cells. In *Proceedings of the Experimental Fluid Mechanics—EFM16, Marienbad, Czech Republic, 15–18 November 2016*. [[CrossRef](#)]
19. Chovanec, M.; Bachratý, H.; Jasenčáková, K.; Bachratá, K. Convolutional Neural Networks for Red Blood Cell Trajectory Prediction in Simulation of Blood Flow. In *Proceedings of the Bioinformatics and Biomedical Engineering—IWBBIO 2019, Granada, Spain, 8–10 May 2019*. [[CrossRef](#)]
20. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *Proceedings of the Computer Vision—ECCV 2014, European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014*. [[CrossRef](#)]
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, CA, USA, 17–30 June 2016*; pp. 770–778. [[CrossRef](#)]
22. The Annotated ResNet-50. Available online: <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758> (accessed on 16 June 2023).
23. Tan, M.; Le, Q.V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019*; pp. 6105–6114.
24. Tan, M.; Le, Q.V. Efficientnetv2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 18–24 July 2021*; pp. 10096–10106.
25. Weik, F.; Weeber, R.; Szuttor, K.; Breitsprecher, K.; de Graaf, J.; Kuron, M.; Landsgesell, J.; Menke, H.; Sean, D.; Holm, C. ESPResSo 4.0—An extensible software package for simulating soft matter systems. *Eur. Phys. J. Spec. Top.* **2019**, *227*, 1789–1816. [[CrossRef](#)]
26. Jančígová, I.; Kovalčíková, K.; Weeber, R.; Cimrák, I. PyOIF: Computational tool for modelling of multi-cell flows in complex geometries. *PLoS Comput. Biol.* **2020**, *16*, e1008249. [[CrossRef](#)]
27. Suresh, S.; Spatz, J.; Mills, J.P.; Micoulet, A.; Dao, M.; Lim, C.T.; Beil, M.; Seufferlein, T. Connections between single-cell biomechanics and human disease states: Gastrointestinal cancer and malaria. *Acta Biomater.* **2005**, *1*, 15–30. [[CrossRef](#)]
28. Marques, G.; Deevyankar, A.; de la Torre Díez, I. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Appl. Soft Comput.* **2020**, *96*, 106691. [[CrossRef](#)]
29. Nayak, D.R.; Padhy, N.; Mallick, P.K.; Zymbler, M.; Kumar, S. Brain Tumor Classification Using Dense Efficient-Net. *Axioms* **2022**, *11*, 34. [[CrossRef](#)]
30. Coccomini, D.A.; Messina, N.; Gennaro, C.; Falchi, F. Combining EfficientNet and vision transformers for video deepfake detection. In *Proceedings of the 21st International Conference on Image Analysis and Processing, Lecce, Italy, 23–27 May 2022*. [[CrossRef](#)]
31. Zhu, L.; Yang, Y. Compound memory networks for few-shot video classification. In *Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018*. [[CrossRef](#)]
32. KerasTuner. Available online: <https://github.com/keras-team/keras-tuner> (accessed on 16 June 2023).

33. Cimrak, I.; Gusenbauer, M.; Jancigova, I. An ESPResSo implementation of elastic objects immersed in a fluid. *Comput. Phys. Commun.* **2014**, *185*, 900–907. [[CrossRef](#)]
34. Wells, I.C.; Itano, H.A. Ratio of sickle-cell anemia hemoglobin to normal hemoglobin in sicklemics. *J. Biol. Chem.* **1951**, *188*, 65–74. [[CrossRef](#)] [[PubMed](#)]
35. Tsai, C.-H.D.; Tanaka, J.; Kaneko, M.; Horade, M.; Ito, H.; Taniguchi, T.; Ohtani, T.; Sakata, Y. An On-Chip RBC Deformability Checker Significantly Improves Velocity-Deformation Correlation. *Micromachines* **2016**, *7*, 176. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.