

Article

Convolutional Neural Networks to Assess Steno-Occlusive Disease Using Cerebrovascular Reactivity

Yashesh Dasari ¹, James Duffin ^{2,3}, Ece Su Sayin ^{2,3}, Harrison T. Levine ^{2,3}, Julien Poubanc ⁴, Andrea E. Para ⁴, David J. Mikulis ^{4,5}, Joseph A. Fisher ^{2,3,5}, Olivia Sobczyk ^{3,4} and Mir Behrad Khamesee ^{1,*}

¹ Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; ydasari@uwaterloo.ca

² Department of Physiology, University of Toronto, Toronto, ON M5S 1A8, Canada

³ Department of Anesthesia and Pain Management, University Health Network, Toronto, ON M5G 2C4, Canada

⁴ Joint Department of Medical Imaging and the Functional Neuroimaging Laboratory, University Health Network, Toronto, ON M5G 2C4, Canada

⁵ Institute of Medical Sciences, University of Toronto, Toronto, ON M5S 1A8, Canada

* Correspondence: khamesee@uwaterloo.ca

Abstract: Cerebrovascular Reactivity (CVR) is a provocative test used with Blood oxygenation level-dependent (BOLD) Magnetic Resonance Imaging (MRI) studies, where a vasoactive stimulus is applied and the corresponding changes in the cerebral blood flow (CBF) are measured. The most common clinical application is the assessment of cerebral perfusion insufficiency in patients with steno-occlusive disease (SOD). Globally, millions of people suffer from cerebrovascular diseases, and SOD is the most common cause of ischemic stroke. Therefore, CVR analyses can play a vital role in early diagnosis and guiding clinical treatment. This study develops a convolutional neural network (CNN)-based clinical decision support system to facilitate the screening of SOD patients by discriminating between healthy and unhealthy CVR maps. The networks were trained on a confidential CVR dataset with two classes: 68 healthy control subjects, and 163 SOD patients. This original dataset was distributed in a ratio of 80%-10%-10% for training, validation, and testing, respectively, and image augmentations were applied to the training and validation sets. Additionally, some popular pre-trained networks were imported and customized for the objective classification task to conduct transfer learning experiments. Results indicate that a customized CNN with a double-stacked convolution layer architecture produces the best results, consistent with expert clinical readings.

Keywords: blood oxygenation level-dependent magnetic resonance imaging (BOLD-MRI); cerebrovascular reactivity (CVR); convolutional neural networks (CNNs); deep learning; medical image analysis; steno-occlusive disease (SOD)



Citation: Dasari, Y.; Duffin, J.; Sayin, E.S.; Levine, H.T.; Poubanc, J.; Para, A.E.; Mikulis, D.J.; Fisher, J.A.; Sobczyk, O.; Khamesee, M.B. Convolutional Neural Networks to Assess Steno-Occlusive Disease Using Cerebrovascular Reactivity. *Healthcare* **2023**, *11*, 2231. <https://doi.org/10.3390/healthcare11162231>

Academic Editor: Joaquim Carreras

Received: 23 June 2023

Revised: 31 July 2023

Accepted: 5 August 2023

Published: 8 August 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

Cerebrovascular diseases include conditions and disorders that affect the blood vessels and cerebral blood flow (CBF) [1]. These diseases affect a significant population worldwide, with stroke being the leading cause of long-term disability and the second-leading cause of death. Globally, over 12.2 million new stroke cases are reported annually, with 7.6 million of these categorized as ischemic stroke. Additionally, the same statistical report predicts that 1 in 4 people over the age of 25 will have a stroke in their lifetime [2]. Steno-occlusive disease (SOD) involves regional arterial occlusion (blockage) or stenosis (narrowing) in the brain and is the most common cause of stroke, with arterial blockage accounting for about 85% of stroke cases [3,4]. Research also suggests that SOD patients are at a high risk of developing recurrent ischemic stroke [5].

Blood Oxygenation Level Dependent (BOLD) imaging maps the differences in CBF when a vasodilatory stimulus is administered and assumes that metabolic activity in the brain is stable during data acquisition [6]. Cerebrovascular Reactivity (CVR) is a provocative test that measures the ability of smooth muscle in the arterial walls to adjust the diameter of the vessels, independent of the effects of vessel diameter adjustments influenced by blood pressure changes or changes in neural and glial activity [7]. This test is analogous to a cardiac stress test where the patient exercises and the change in coronary blood flow are observed.

CVR closely relates to the brain vasculature's health and can highlight cerebrovascular diseases such as Alzheimer's disease (AD), SOD, stroke, and traumatic brain injury [8]. In patients with SOD, the administration of an external stimulus leads to CBF redistribution and can result in regional steal physiology [9]. Therefore, analyzing information about the severity and spatial location of abnormal CVR at the tissue level can play a vital role in the early diagnosis and management of SOD.

Currently, CVR maps are analyzed by a team of experts who visually examine CVR distribution across brain regions and evaluate the brain vasculature conditions. Motivated by the extensive prevalence of cerebrovascular diseases and the well-established success of deep learning in medical imaging, a convolutional neural network (CNN)-based clinical decision support system to facilitate CVR assessment would be welcomed. The automatic feature extraction adds several benefits to the CVR analysis workflow. The proposed model accounts for the skull-based susceptibility artifact, which are otherwise manually removed. It is also agnostic to the orientation of the maps in the mosaic stack since the training set has samples that were flipped horizontally.

To optimize the assessment of the effects of SOD on cerebral blood flow regulation in individual patients, a method enabling precisely reproducible carbon dioxide (CO₂) stimuli was applied using prospective end-tidal targeting of CO₂ during high temporal resolution whole brain BOLD-MRI. The CVR maps thus generated were used for training and testing the CNNs. The proposed model can be used by participating institutions as a research tool for classifying CVR maps generated using the clinical workflow used in this work and for advancing the application of deep learning and CNNs in CVR research.

1.1. CVR: Clinical Workflow

Clinicians have administered different types of stimuli for CVR studies, including, transient reduction in mean arterial blood pressure, chemical injections, and changes in arterial partial pressure of CO₂ [10]. The CVR maps used in this work were obtained by administering a non-invasive stimulus, involving manipulation of the end-tidal partial pressures of CO₂ and O₂ gas concentrations in spontaneous breath studies (prospective end-tidal targeting) as the vasoactive stimulus and mapping the corresponding changes [11]. The gas concentration manipulation was accomplished using a computer-controlled gas blender called the RespirAct RA-MR [12], and a step-and-ramp stimulus protocol was administered [13].

CVR is measured as the ratio of the change in BOLD signal(s) (ΔBOLD) with respect to the change in end-tidal partial pressure of CO₂ (ΔP_{ETCO_2}) [14], as shown in the equation below:

$$\text{CVR} = \frac{\Delta\text{BOLD}}{\Delta P_{\text{ETCO}_2}}. \quad (1)$$

The voxel-by-voxel CVR values calculated across the brain regions are co-registered on the corresponding anatomical scans. For the CVR maps used in this work, an increase in CVR is indicated by positive values and is colored in shades of yellow, orange, and red. Similarly, reduced CVR is indicated by negative values and is colored in shades of blue. Figure 1 shows two sample CVR maps obtained from a healthy control subject and a patient with SOD, highlighting the normal and abnormal CVR distribution using the specified color scale.

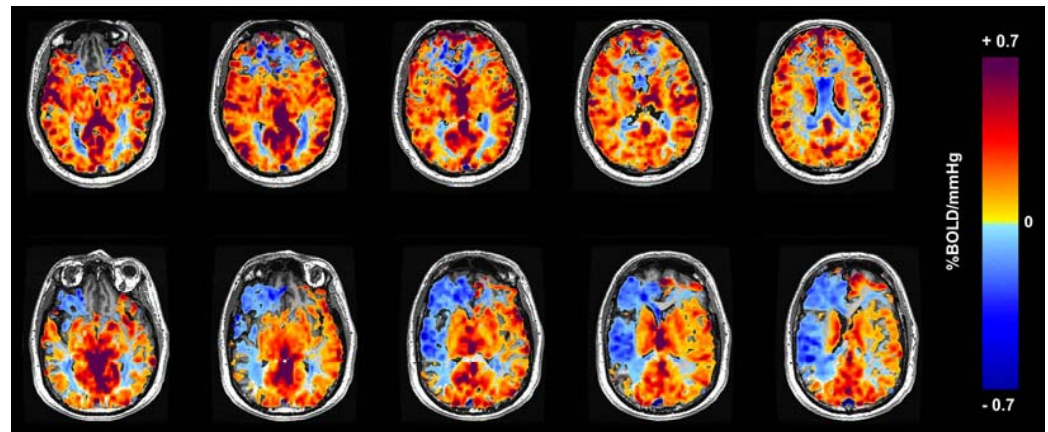


Figure 1. Colored CVR maps overlaid on the corresponding anatomic T1 weighted images (note: MRI images are displayed from the perspective of the feet looking up, so the right side of the brain is on the left side of the image). Positive CVR is yellow–red and negative CVR (steal physiology) is blue. Top row: normal healthy control. There is a susceptibility artifact from the skull base, which causes artifactually decreased CVR in the anterior inferior frontal lobes. Bottom row: patient with right–sided steno–occlusive disease and extensive steal physiology in the right middle cerebral artery (MCA) territory (blue which is on the left side of the images).

1.2. Convolutional Neural Networks

Deep Learning, a subset of Artificial Intelligence (AI), is a type of representation learning that automatically discovers important representations (called features) needed to make decisions from raw input data as a part of its learning algorithm, eliminating the need to manually pre-process the raw data [15]. At its core, deep learning models are made up of artificial neural networks that use a network of functions to establish interpretations from features and map them to a specific output [16].

Convolutional Neural Networks (CNNs) are specialized deep learning models that use a linear mathematical operator called convolution, in their convolution layers [17]. CNNs are the most popular deep learning algorithm used in visual learning tasks because it significantly reduces the number of training parameters (dimensionality reduction) while preserving local image relations. The convolution layers extract meaningful representations from the raw input data, which are then fed into the fully connected layers for the objective task. Therefore, a CNN can be described as a combination of feature extractor-classifier models.

A convolution layer performs a convolution operation on the input image using a pre-defined filter. Other layers in a CNN can be a pooling layer, a dropout layer, a normalization layer, a loss layer, and so on [18]. A convolution operation is generally denoted with an asterisk. For input, $I(t)$ with an independent variable t , a convolution operation with a filter $K(a)$, to obtain a feature map $s(t)$ is shown in the equation below [15]:

$$s(t) = (I * K)(t). \quad (2)$$

An activation function (or transfer function) is used to adjust or map the generated output from a layer on a new scale [19]. Two types of non-linear activation functions are used in this work, the Sigmoid and the Rectified Linear Unit (ReLU) activation functions [20]. A pooling layer is typically used after the convolution layers to down-sample the feature maps, reducing their complexity. In image-related tasks, it reduces the width and height of the feature maps. Once the feature maps have extracted important representations from the input, the fully connected layer(s) compute the probability score for the objective classification [21].

Transfer learning is a widely used concept in deep learning where parametric-level knowledge is transferred between networks. The objective of such experiments is to use a

pre-trained network, originally trained on partially related or unrelated datasets, and use these weights to accomplish the target classification task [22].

1.3. Article Structure

The article is organized as follows. Section 2 discusses the recent work and advancements related to the application of deep learning in medical imaging and CVR studies, focusing on CNNs. Section 3 explains data preparation and pre-processing, network design methodology, and the deep learning concepts implemented. Section 4 presents the quantitative results obtained from the experiments and compares the networks' performance. Section 5 discusses the key findings of this research, the proposed network, and outlines the future scope. Finally, Section 6 draws conclusions from this work.

2. Relevant Work

Deep Learning has been successfully implemented in wide-ranging medical imaging tasks, for example, image segmentation and classification, and computer-aided disease diagnosis [23,24]. In radiology, the spatial structures of the organ are pivotal in classifying healthy versus unhealthy cases. Therefore, CNNs are particularly effective in this domain because they preserve local spatial relationships when filtering input images [25].

Farooq et al. developed a deep CNN-based pipeline for a four-way classification of brain MRI scans. The experiments were conducted using the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset and obtained state-of-the-art results to achieve a prediction accuracy of 98.8%. The dataset was categorized into a healthy control group, Alzheimer's Disease (AD), Mild Cognitive Impairment (MCI), and Late Mild Cognitive Impairment (LMCI) [26].

Chen et al. implemented CNNs to predict the cerebrovascular reserve in Moyamoya disease in a simultaneous [15O]-water positron emission tomography (PET)/MRI before acetazolamide administration on a dataset consisting of 24 patients and 12 healthy control subjects. Two models were used, one with both PET and MRI and another with only MRI. Both models successfully identified the regions of impaired cerebrovascular reserve, both achieving an area of 0.95 under the receiver operating characteristic curve [27].

Hussein et al. used datasets from cerebral blood flow (CBF)-based studies and proposed a multi-task learning workflow consisting of MRI-to-PET translation, followed by disease diagnosis. The work reported an average classification accuracy of 96.38% for three disorders including Moyamoya disease, SOD, and stroke [28].

Hashemzahi et al. proposed a hybrid model using a neural autoregressive distribution estimation (NADE) and a CNN for brain tumor detection. The model was tested with 3064 T1-weighted contrast-enhanced images and achieved a 95% accuracy in classifying three types of tumors [29].

Mejis et al. presented a CNN model to detect image-level intracranial anterior circulation artery occlusions using 4D-CTA imaging. The model was trained using 214 samples and obtained an accuracy of 92% on the test set of 279 samples [30].

Hou et al. used deep learning for resting-state vascular imaging to detect vascular abnormalities using cerebrovascular reactivity (CVR) and bolus arrival time (BAT) maps of the brain that were obtained using resting-state fMRI studies [31].

In addition to the detection, classification, and prediction of the cerebrovascular diseases discussed above, Zhu et al. provide a comprehensive review of deep learning-based generation and enhancement of stroke imaging, using Computed Tomography (CT) and MRI [32].

Transfer learning has also proven very effective with medical datasets. Talo et al. successfully fine-tuned ResNet34 for the binary classification of 613 MR images, using techniques such as image augmentation and optimal learning rate finder, and achieved a five-fold classification accuracy of 100% [33]. Maqsood et al. used AlexNet for the classification of 3D MRI datasets, including both segmented and unsegmented images, and achieved an accuracy of 92.85% for the multi-class classification of unsegmented

images [34]. Yuan et al. proposed a novel transfer learning-based multiparametric MR model for prostate cancer classification, achieving an accuracy of 86.92% [35].

Mohsen et al. applied transfer learning for domain adaptation for MRI scanner agnostic studies. This research investigated white matter hyperintensity segmentation using CNNs and concluded that transferring pre-trained network weights of one set of MRI images to a different MRI dataset outperformed a new network trained from scratch [36].

Many conventional machine learning models have also been successful in CVR-related learning tasks. Spencer et al. used Support Vector Machines (SVMs) to identify cerebrovascular impairment using CVR-weighted hypercapnic BOLD and other MRI techniques, achieving high performance (specificity = 0.67; sensitivity = 0.75) [37]. Kloppel et al. also used SVMs to detect AD using structural MRI and reported an accuracy of 96% [38]. Evangelia et al. investigated Linear Discriminant Analysis (LDA) with Fisher's discriminant rule, k-Nearest Neighbor (k-NN), and nonlinear SVMs to classify brain tumors using MRI datasets. The binary SVM classification for discrimination of metastases from gliomas achieved accuracy, sensitivity, and specificity of 85%, 87%, and 79%, respectively [39]. Bahadure et al. used SVMs for the binary classification of normal versus abnormal tissues using Berkeley wavelet transformation (BWT) based brain tumor segmentation and achieved 96.51% accuracy, 94.2% specificity, and 97.72% sensitivity [40].

3. Materials and Methods

3.1. Data Analysis and Pre-Processing

The input dataset used for training the networks contained 2-dimensional RGB images of the CVR maps in axial view, stacked in a 5×5 mosaic format. Each input image displayed CVR maps across the brain with a 5-slice spacing, plotted in the standard coordinate space, and had a resolution of $1280 \times 1280 \times 3$ pixels [41]. Using the CVR maps in a mosaic format allowed the networks to be trained with comprehensive and sufficient details to make judgments on the overall brain vasculature's health. Training the networks using individual slices of CVR maps may lead to the network learning localized attributes, native to individual slices. The input dataset was obtained from a diverse patient cohort from studies conducted at our sponsoring institute and was labeled by a team of experts. Figure 2 shows four sample input images, two belonging to healthy control subjects (a, b), and two to SOD patients (c, d).

The CVR dataset was obtained by administering a step-and-ramp CO₂ stimulus protocol using prospective end-tidal targeting [13]. The original dataset contains 68 healthy control subjects and 163 patients, which was divided into three sub-categories to train, validate, and test the networks, split in a ratio of 80%-10%-10%, respectively, as shown in Figure 3.

Deep learning models heavily rely on the quantity of data available for training the networks [42]. The limited training samples available for this study can be considered a small dataset. Therefore, to address the data scarcity, artificial features were added to the training dataset using data augmentation techniques. Additionally, to accommodate some of the real-world differences in CVR maps that arise due to experimental settings, small augmentations were applied to the validation set as well. However, to maintain a standardized CVR assessment protocol, the augmentations in both sets were small-scaled and did not affect the original attributes of the mosaic images. The augmentation techniques used included rotation (0–20%), shearing (0–20%), brightness modifications (30–100%), and horizontal flips. The test set was unaltered. The final number of samples in each dataset after data augmentation is shown in Table 1.

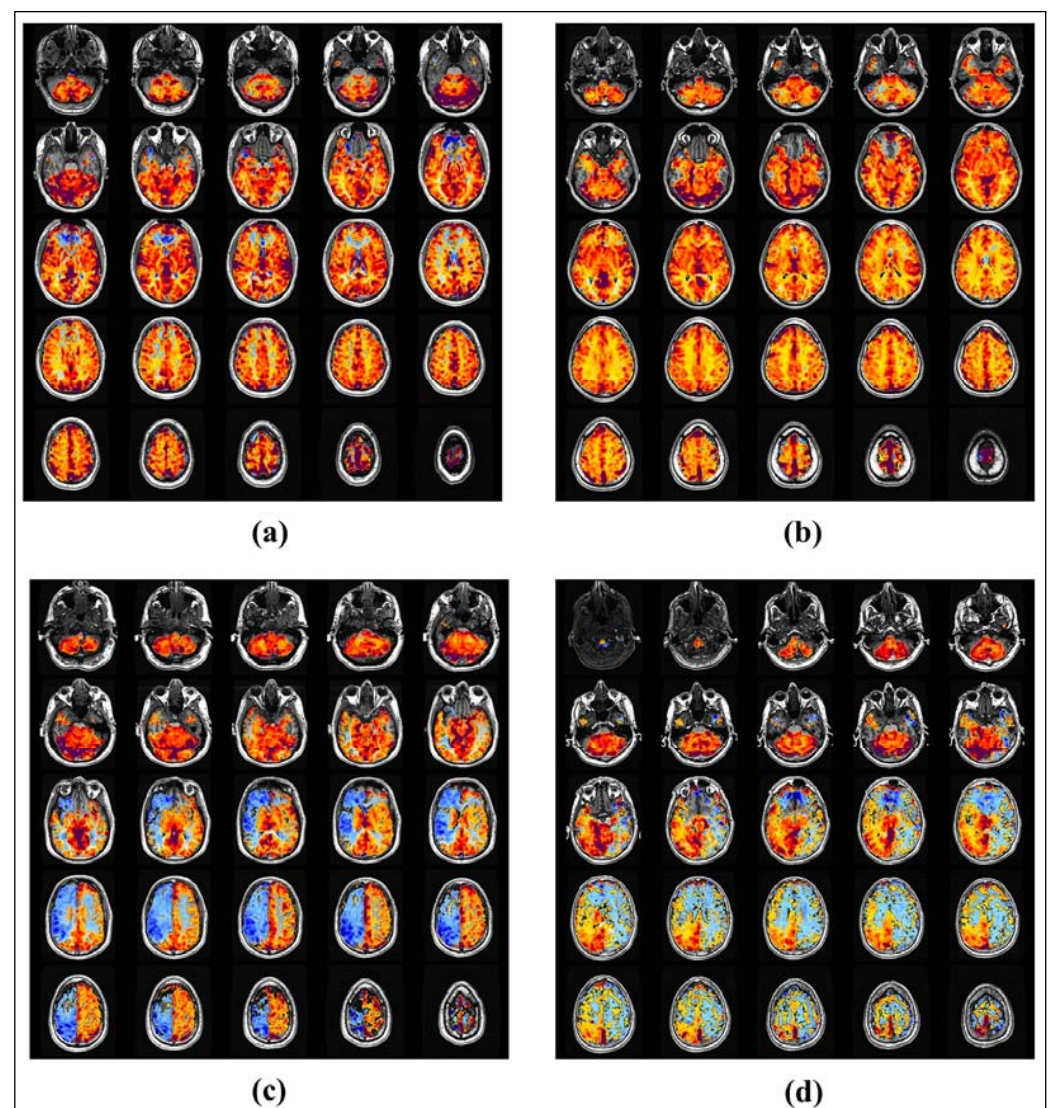


Figure 2. 5×5 mosaic images of the CVR maps from the input dataset. CVR maps in samples (a), and (b) were measured in healthy subjects and the maps in samples (c), and (d) were measured in SOD patients. As shown, there is a reduced CVR response in the right and left hemispheres of the brain in samples (c), and (d), respectively.

Table 1. The total number of samples in the train, validation, and test sets after data augmentation was applied to the original input data.

Dataset	Healthy Samples	Unhealthy Samples
Train	594	650
Validation	168	186
Test	6	16

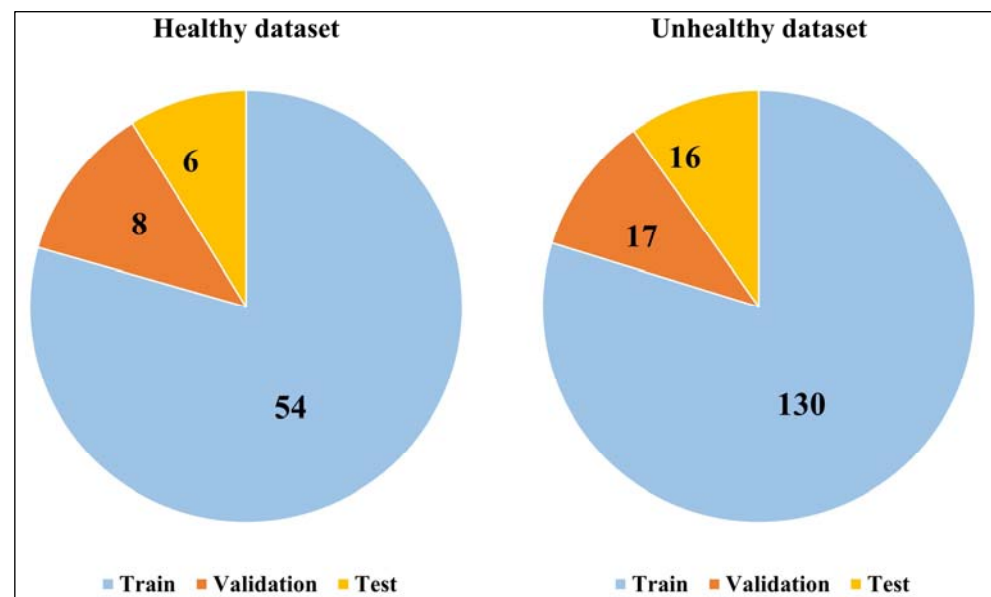


Figure 3. Distribution of the original dataset (68 healthy, and 163 unhealthy samples) into the train, validation, and test subsets. The splitting ratio within each category (healthy and unhealthy) is roughly 80%-10%-10%, as shown.

3.2. CNN Design

The optimization of CNNs mainly involves fine-tuning the network's hyperparameters (configurable values set before the training begins) during successive trials of training. The network automatically updates its parameters (weights and biases) during the learning process [43].

The CNN design and optimization were conducted using Python (v. 3.10.4), TensorFlow (v. 2.9.1) [44], and Keras (v. 2.9.0) [45] libraries. The computational setup consisted of an Intel X2 Gold 5218 dual-core 2.3 GHz processor, with 256 GB RAM, and a 64-bit Windows 10 operating system, and the experiments were run as CPU processes. The design strategy involved training a small, shallow CNN, and using its performance as a benchmark to improve the network configurations and settings. The network hyperparameters were further fine-tuned to develop more optimized networks. The trained model details such as the network architecture, weights, and state of the trained models were saved for testing and future use.

All the customized networks investigated were trained using the Adam optimization algorithm [46]. ReLU activation function [47] was used with the convolution layers, and the max pooling operation [48] was used after the convolution layer(s), with a pool size of 2×2 . After the feature extraction process, a flatten layer was used to flatten the multi-dimensional input tensor into a single dimension that could be fed into a fully connected layer. The hidden fully connected layers used the ReLU activation function, and the output fully connected layer used a sigmoid activation function. The binary cross-entropy loss function was used, and the accuracy was monitored after each epoch. The networks were trained for up to 200 epochs, with the early stopping callback. For early stopping, the validation loss was monitored with a patience of 10 epochs [49].

With limited training samples, the relationships that a neural network learns can be the result of sampling noise, which does not exist in real test data. This leads to overfitting [50]. To address this, dropout layers were added to the network architecture. Dropout [51] is a widely used regularization technique where units along with their connections are dropped from the network. This happens only during training. Another effective normalization technique that reparametrizes deep networks called batch normalization (also called batch norm) was investigated. It applies transformations to normalize the inputs fed to the subsequent layer [52].

The different hyperparameters experimented with and fine-tuned during the network optimization are listed in Table 2.

Table 2. The different hyperparameter settings that were experimented with as a part of the network architecture optimization process. The best values for each hyperparameter for the dataset considered are shown in bold font, wherever possible. Note: The hyperparameter combinations used were based on the results from sensitivity studies and therefore not all combinations are discussed.

Hyperparameter	Values
Batch Size	8, 16, 32 , 64
Batch Normalization	Yes /No (with default settings)
Dropout Rate	0% (No dropout), 20%, 50%
Network Depth (number of layers)	2, 3 hidden layers
Network Width (filters/nodes in each layer)	8, 16, 32, 64, 128
Learning Rate	0.01, 0.001 (default) , 0.0005
Kernel Size of Convolution Layer	3 × 3 , 5 × 5, 7 × 7

3.3. Transfer Learning

Some of the popular pre-trained networks, trained on the ImageNet dataset (ImageNet Large Scale Visual Recognition Challenge; ILSVRC) [53], were used to implement transfer learning using the CVR dataset. The networks investigated were EfficientNetB0 [54], InceptionV3 [55], ResNet50 [56], and VGG [57].

These networks were imported with their pre-trained weights using Keras libraries, and then modified to suit the objective binary classification task by freezing all the layers and replacing the classifiers. This workflow allowed leveraging the features learned by the pre-trained models and making predictions on the CVR dataset [58].

The original images were downsized to 224×224 pixels, and a batch size of 32 was used while using the early stopping callback that monitored the validation loss with a patience of 10 epochs.

For InceptionV3, the base model was instantiated with the pre-trained weights and all the layers were frozen, and the ImageNet classifier was excluded. A flatten layer was added after the last fully connected layer, making the output shape (None, 51200). Following this, another fully connected layer was added with 1024 neurons and a ReLU activation function, followed by a dropout layer (dropout rate = 50%), and an output layer with a sigmoid activation function.

Similarly, for EfficientNetB0, ResNet50 and VGG16, the pre-trained networks were imported while excluding the classifier. Only one fully connected output layer was added on top of the frozen layers with a sigmoid activation function.

4. Results

As discussed in Section 3, the customized CNNs were trained using the mosaic CVR dataset for up to 200 epochs, with a learning rate of 0.001, the Adam optimizer, and an early stopping callback. While investigating the effects of batch size, it was observed that increasing the batch size improved the model performance and reduced the noise in the validation loss. Based on the results and recommendations from Bengio [59], a batch size of 32 was used.

4.1. Baseline Experiments

The baseline model consisted of an input layer, a convolution layer with 8 filters, a fully connected layer with 8 neurons, and an output layer with 1 neuron. The first series of experiments investigated the effect of input image resolution, comparing downsized resolutions of 256×256 and 512×512 pixels. The results indicated that an input resolution of 256×256 pixels produced optimal results.

However, the overall performance of this network was poor, and it achieved training and validation accuracy of 52.3% and 52.5%, respectively.

4.2. Dropout Regularization and Batch Normalization

To address the overfitting, regularization techniques were introduced in the network architecture. When dropout was proposed, it was only used with fully connected layers and not convolution layers [51]. However, more recent studies have demonstrated that this could indeed improve performance [60].

Two dropout rates were investigated with the baseline model: 0.2 (dropping 20% nodes), and 0.5. A dropout layer was also added after the convolution layer to test the contradicting rules about this practice. Experiments demonstrated that using dropout layers with a dropout rate of 0.5 after both the convolution layer and the hidden fully connected layer produced the best results.

Finally, adding batch normalization layers after the convolution layer and the hidden fully connected layer further improved the results. The configuration of the dropout layer, batch norm layer, and activation function is again a topic of debate [61]. This study presents a configuration best suited for the dataset in consideration.

The best-performing network had the following architecture: input layer, convolution layer (8 filters), batch norm layer, dropout layer (rate = 0.5), fully connected layer (8 nodes), batch norm layer, dropout layer (rate = 0.5), and output layer.

4.3. Network Width and Depth

Using the inferences from the baseline experiments, the network width and depth were investigated next. In this series of sensitivity analyses, the widths of existing convolution and hidden fully connected layers were increased from 8 to 64. For network depth experiments, additional convolution layers and fully connected layers, along with dropout and batch norm layers were added.

4.4. Advanced Hyperparameters

Some of the advanced hyperparameters were experimented with using the best-performing networks. The results indicated that the kernel size had an insignificant effect on the overall network performance. Therefore, a 3×3 kernel size is recommended for this CVR dataset.

Three learning rates were investigated: 0.001 (default), 0.0005, and the ReduceLROnPlateau [62] callback. ReduceLROnPlateau reduced the learning rate when the selected performance metric stops to improve. Based on the results, the default learning rate of 0.001 is recommended for the CVR dataset used in this work.

4.5. Network Performance Comparison

A comparison of the accuracy and loss values obtained from experimenting with the hyperparameter settings is shown in Table 3. The table describes the network architecture, the total number of parameters, the number of epochs the models trained for, and the performance metrics.

The generalization capabilities of a network can be evaluated using its performance on unseen data in the test set. In a binary classification task, this is carried out using the metrics discussed below.

Based on the prediction made by the classification model on an input, the outcomes can be classified into four groups:

1. True Positive (TP): Number of samples classified correctly as yes or success.
2. True Negative (TN): Number of samples classified correctly as no or failure.
3. False Positive (FP): Number of samples classified incorrectly as a yes or success.
4. False Negative (FN): Number of samples classified incorrectly as no or failure.

These values can be used to calculate the testing accuracy, sensitivity, and specificity as shown in the formulas below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5)$$

Table 3. Results obtained on the training, validation, and test datasets by the different network architectures. In the table, convX represents a convolution layer with “X” filters, fcX represents a fully connected layer with “X” nodes, drop represents a dropout layer (rate = 0.5), and BN represents a batch normalization layer. Note: Each convolution layer is followed by a 2×2 2D max pooling layer. The best performing network is highlighted in bold font.

Model Number: Architecture	No. of Parameters	Epochs	Training Accuracy & Loss	Validation Accuracy & Loss	Testing Accuracy Sensitivity & Specificity
1: conv8, fc8, drop, fc1	1,032,497	16	0.8977 & 0.21	0.8835 & 0.29	0.95 0.83 & 1
2: conv8, BN, drop, fc8, BN, drop, fc1	1,032,561	42	0.9670 & 0.06	0.97 & 0.09	0.72 1 & 0.62
3: conv16, BN, drop, fc16, BN, drop, fc1	4,129,633	14	0.9991 & 0.005	0.9204 & 0.177	0.77 0.17 & 1
4: conv16, BN, drop, fc32, BN, drop, fc1	8,258,561	53	0.8531 & 0.3465	0.9176 & 0.3149	0.27 1 & 0
5: conv16, BN, drop, fc64, BN, drop, fc1	16,516,673	16	1.00 & 0.0007	0.9631 & 0.1581	0.27 1 & 0
6: conv32, BN, drop, fc32, BN, drop, fc1	16,517,313	10	1.00 & 0.0054	0.5284 & 0.8745	0.27 1 & 0
7: conv32, BN, drop, fc64, BN, drop, fc1	33,033,601	10	1.00 & 0.0018	0.5250 & 2.3896	0.32 1 & 0.06
8: conv64, drop, fc64, drop, fc1	66,066,305	23	0.9867 & 0.0316	0.9488 & 0.2335	0.95 0.83 & 1
9: conv64, fc128, drop, fc1	132,131,585	12	1.00 & 0.0018	0.5284 & 1.1150	0.63 1 & 0.5
10: conv8, conv8, BN, drop, fc8, BN, drop, fc1	246,905	31	0.9884 & 0.0285	0.9204 & 0.2554	0.95 0.83 & 1
11: conv8, conv8, drop, fc16, drop, fc1	492,969	36	0.9975 & 0.0191	0.9318 & 0.19	0.31 1 & 0.06
12: conv16, conv16, BN, drop, fc16, BN, drop, fc1	986,993	40	1.00 & 0.0043	0.9715 & 0.0776	0.81 1 & 0.75
13: conv16, conv16, BN, drop, fc32, BN, drop, fc1	1,971,153	27	0.9983 & 0.0095	0.9176 & 0.3017	0.68 1 & 0.64
14: conv32, conv32, BN, drop, fc32, BN, drop, fc1	3,946,721	46	1.00 & 0.0006	0.9801 & 0.0671	0.95 1 & 0.9375
15: conv64, conv64, BN, drop, fc64, BN, drop, fc1	15,784,385	10	1.00 & 0.0024	0.7642 & 0.5937	0.7272 1 & 0.625

As shown in Table 3, Model 14 achieved the best performance on all three datasets, obtaining accuracy of 100% and 98% in training and validation datasets, respectively, after training for 46 epochs, and an accuracy of 95% on the test set. It stacks two convolution

layers with 32 filters in each, followed by a batch norm layer and a dropout layer with a rate of 50%, a hidden fully connected layer with 32 neurons, another series of batch norm and dropout layers, and the output layer.

4.6. Generalization Capabilities

As shown in Table 1, the test set comprised 6 healthy samples and 16 unhealthy samples. The proposed model (Model 14) labeled the healthy group samples correctly, TP = 6; however, incorrectly labeled one out of the 16 unhealthy samples, TN = 15 (out of 16). Therefore, the prediction accuracy of this model is:

$$Accuracy = \frac{6 + 15}{6 + 15 + 0 + 1} = 0.9545 \text{ (or 95.45\%)}. \quad (6)$$

The confusion matrix for the proposed network, Model 14, is shown in Figure 4.

Total samples = 22		Predicted: SOD	Predicted: Healthy
Actual:	SOD	TN = 15	FP = 1
	Healthy	FN = 0	TP = 6

Figure 4. Confusion matrix summarizing the performance of the best-performing network on the test set. The matrix shows the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the network on the test set.

4.7. Ablation Study

The key components of the proposed network are convolution layers, batch normalization layers, dropout layers, and fully connected layers. An ablation study was conducted to verify the effects of each component on the model performance metrics [63,64]. Additionally, two types of adaptive optimizers, RMSProp (Root Mean Square Propagation) and Adam (Adaptive Moment Estimation) were investigated. Table 4 shows the performance metrics obtained by the different model architectures.

Based on the results, the Adam optimizer (model A1) achieves a better performance as compared to the RMSProp optimizer (model A2). Comparing models A4–A7, adding a batch normalization layer and a dropout layer after the stacked convolution layers has a positive effect on the performance and improves overfitting. However, the overfitted models perform better on the test set. The loss curves obtained from training the networks with and without the batch normalization and dropout layers (models A1, and A7) are shown in Figure 5.

Model A4, where the batch normalization and dropout layers after the first fully connected layer are dropped, suggests that these layers have a negligible impact on the model's performance on training and test sets. However, validation results are improved by adding these regularization layers.

This study validates that the originally proposed network (model A1) produces the best overall results with the given CVR dataset.

Table 4. Ablation study to verify the effect of each component of the proposed network on the overall performance. The model performance on the training, validation, and test sets are presented. In the table, convX represents a convolution layer with “X” filters, fcX represents a fully connected layer with “X” nodes, drop represents a dropout layer (rate = 0.5), and BN represents a batch normalization layer. Adam and RMSProp are the two optimizers investigated during the ablation study.

Model Name: Architecture	Epochs	Training Accuracy & Loss	Validation Accuracy & Loss	Testing Accu- racy Sensitivity & Specificity
A1: conv32, conv32, BN, drop, fc32, BN, drop, fc1; Adams	46	1.00 & 0.0006	0.9801 & 0.0671	0.95 1 & 0.9375
A2: conv32, conv32, BN, drop, fc32, BN, drop, fc1; RMSProp	19	0.9983 & 0.004	0.9090 & 0.5402	0.82 0.5 & 0.9375
A3: conv32, conv32, fc32, BN, drop, fc1	28	1.00 & 0.003	0.9232 & 0.26	0.95 1 & 0.9375
A4: conv32, conv32, BN, drop, fc32, fc1	64	1.00 & 0.075	0.9488 & 0.226	1 1 & 1
A5: conv32, conv32, BN, fc32, fc1	19	1 & 0.0004	0.9062 & 0.28	1 1 1
A6: conv32, conv32, drop, fc32, fc1	19	1 & 0.0006	0.929 & 0.15	0.95 1 0.9375
A7: conv32, conv32, fc32, fc1	16	1.00 & 0.00009	0.9233 & 0.25	1 1 & 1
A8: conv32, conv32, fc32, BN, fc1	19	1.00 & 0.0003	0.906 & 0.28	1 1 1
A9: conv32, conv32, fc32, drop, fc1	19	1.00 & 0.0006	0.93 & 0.15	0.95 1 0.9375

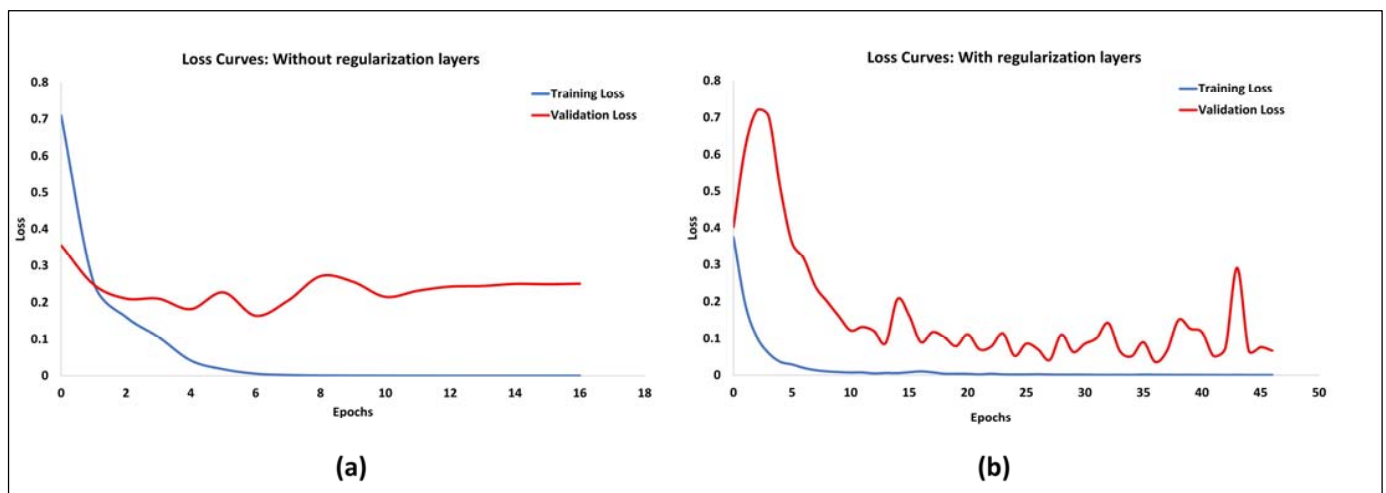


Figure 5. Comparison of the training and validation loss of the models with and without batch normalization and dropout layers. As observed in the left plot (a), the validation loss starts to increase with epochs while the training loss continues to decrease, demonstrating overfitting. This is addressed when the batch normalization and dropout layers (represented as regularization layers) are added, as demonstrated in the right plot (b).

4.8. Transfer Learning Performance

The transfer learning experiments with EfficientNetB0, InceptionV3, and ResNet50 networks achieved training and validation accuracy in the range of 50–53%. With the

early stopping criterion, these networks' performance stopped improving after 27, 14, and 85 epochs, respectively. The VGG16 model produced the best results, achieving 99.83% and 94.88% training and validation accuracy, respectively, after training for 25 epochs.

All the pre-trained networks correctly labeled the unhealthy dataset. However, they incorrectly labeled the healthy dataset, classifying them as SOD class. Therefore, the overall prediction accuracy achieved by these networks is 72.72%.

4.9. Results Analysis

A comparison of the performance of the best-performing customized CNN trained from scratch and the transfer learning experiments is presented in Figure 6.

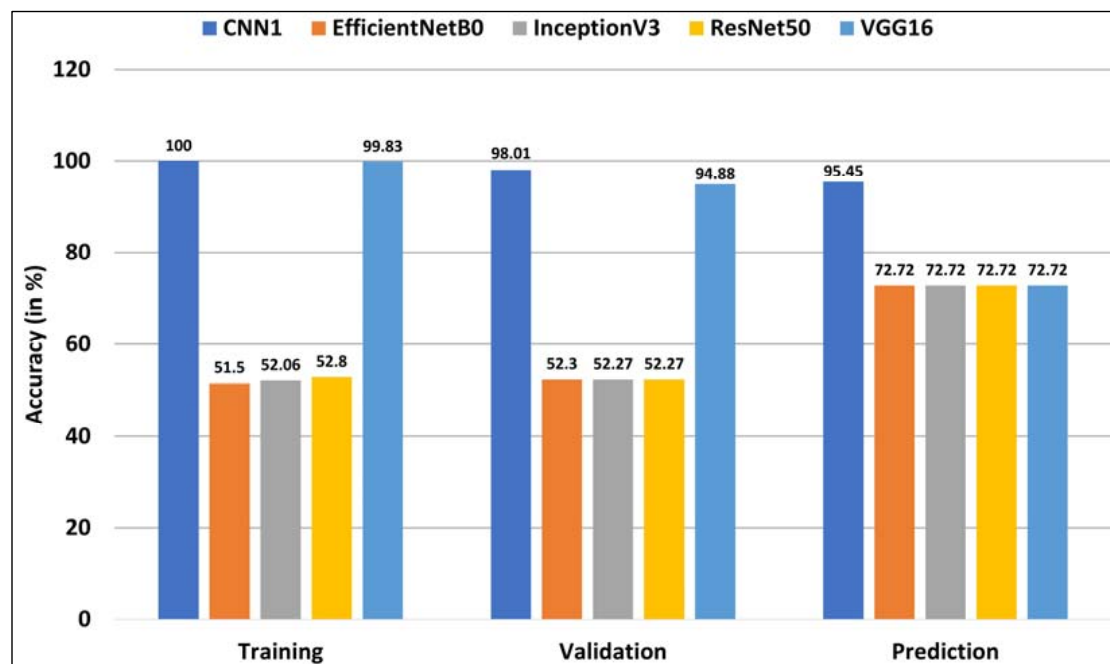


Figure 6. Comparison of the results obtained by the best-performing CNN trained from scratch (labeled as CNN1) and the pre-trained networks including EfficientNetB0, InceptionV3, ResNet50, and VGG16 on the training, validation, and test sets.

5. Discussion

5.1. Key Findings and Proposed Network

This study demonstrates that the convolution layers successfully identified the key parameters in the CVR maps, primarily the spatial distribution and intensity of reduced CVR, and the classifiers were able to discriminate between healthy control subjects and SOD patients. Based on the quantitative results obtained from the sensitivity analysis and the ablation experiments, it is evident that using a dropout layer after the convolution layer addressed overfitting and improved the model performance for the CVR dataset considered. Additionally, using a batch normalization layer further improved the overall model performance.

Based on the results obtained from the hyperparameter fine-tuning of customized CNNs that were trained from scratch, and using the pre-trained networks, a customized double-stacked CNN (Model 14 in Table 3) is proposed as the best-suited model for the CVR dataset used. The proposed network architecture is shown in Figure 7. The network architecture, weights, and biases were saved, and the model can be readily used to classify new CVR maps in the same mosaic configuration used in this study.

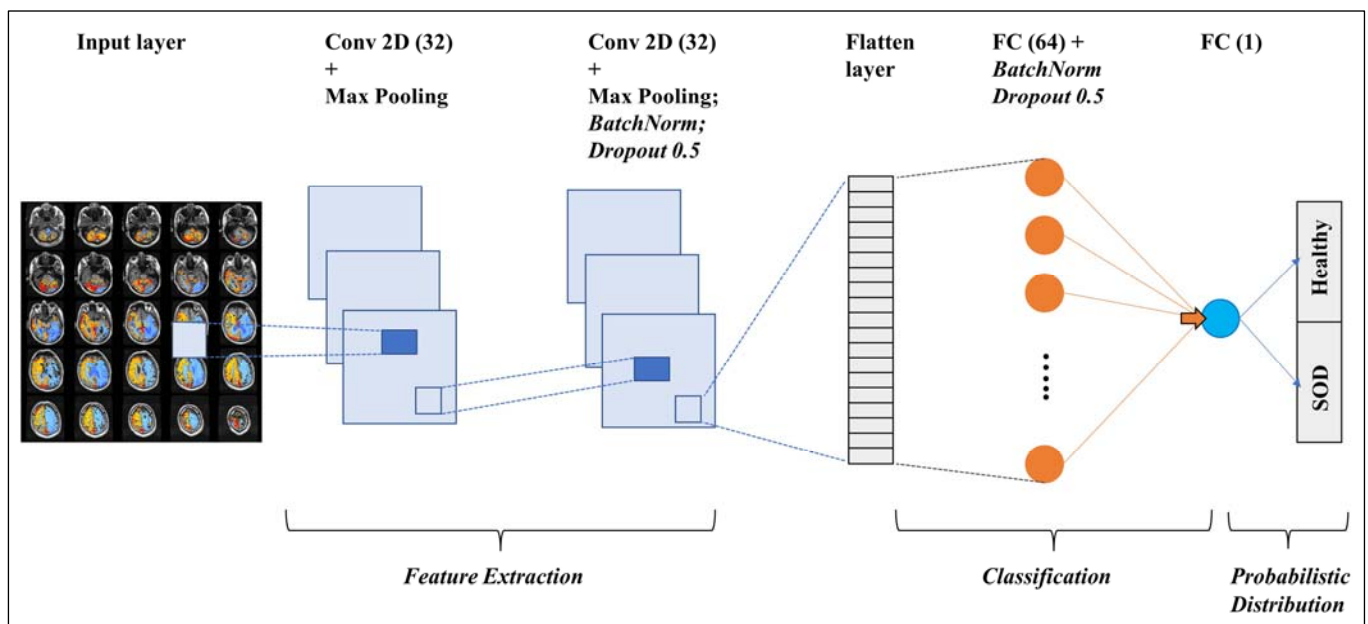


Figure 7. A schematic representation of the proposed CNN. The input layer is a matrix representation of the pixels in the input sample. Conv2D(32) represents a convolution layer with 32 filters, followed by a 2×2 2D max pooling layer. The second Conv2D layer has a batch normalization layer and a dropout layer after the max pooling layer. The features extracted by the convolution layers are fed into a flatten layer, which then goes into the fully connected layer with 64 nodes (represented as FC(64)). Finally, the output layer calculates a probabilistic distribution and classifies the input sample as a healthy subject or a SOD patient.

5.2. Limitations

The proposed CNN incorrectly labeled one sample (shown in Figure 8) in the test set, labeling a SOD patient as healthy. On review of the maps, while this patient has SOD, there is no steal physiology, indicating a normal CVR response. Therefore, although the CVR is “normal” for this patient, based on the objective of this study, this is an incorrect prediction. Such edge cases would require validation from experts.

Additionally, the proposed network is constrained by the color scale of the CVR maps, spacing between the brain slices while creating the mosaic images, and limited to 5×5 mosaic configurations. For other institutions to use this network, the same CVR analysis protocol would be needed.

5.3. Future Work

As stated in Figure 1, the CVR maps used often have additional blue regions due to susceptibility artifacts that mimic steal. Since the key difference between healthy and unhealthy maps is the spatial distribution of reduced CVR, these artifacts may potentially influence the learning process. However, in the samples used in this study, these artifacts consistently appear in the anterior inferior frontal lobes from the paranasal sinuses, and in the temporal lobes from the mastoid air cells. This consistency allowed the CNNs to produce results with high accuracy. Future studies could investigate the impact of these artifacts on the model performance. Additionally, cross-validation should be implemented to validate the stability of the proposed network and its generalization capabilities.

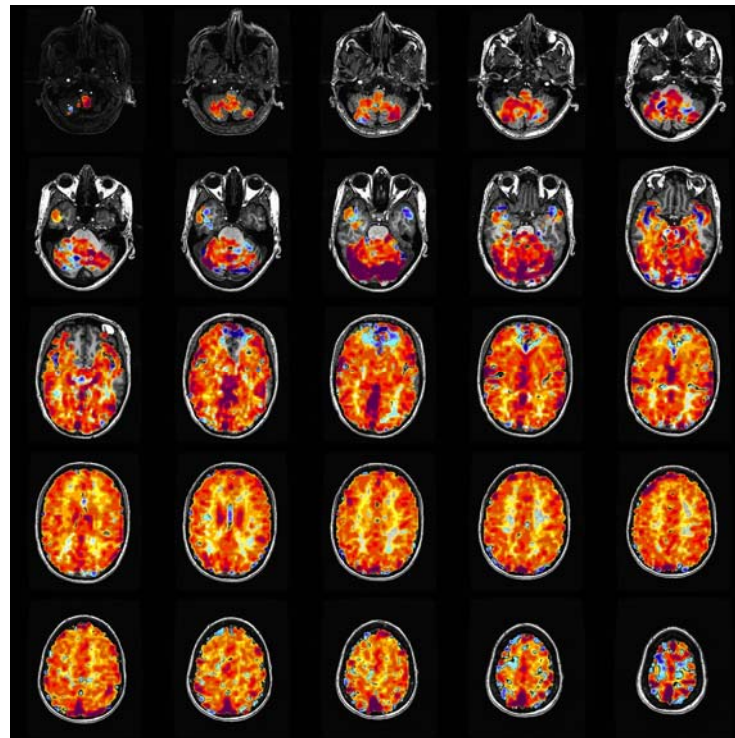


Figure 8. The input sample from the test set that was incorrectly labeled by the proposed CNN. As observed, there is no sign of steal physiology, indicated by the prominent red and yellow across the brain. The blue color is due to susceptibility artifacts.

For furthering this research, it is important to collect more data. The proposed network can be retrained with more samples to improve performance and extend generalization capabilities. In addition to SOD, there are other cerebrovascular diseases such as Alzheimer's disease, Moyamoya disease, sickle cell disease, and small vessel disease, that can be highlighted using CVR studies [7,65]. For this multi-classification study, the data samples were insufficient to train the networks, but as clinical studies continue, CNNs can be implemented for screening these diseases.

The transfer learning experiments, except for VGG16, yielded poor results with the CVR dataset used. The scarcity of training samples, the large number of parameters, and the complexity of the pre-trained networks could be the reasons. VGG16 is simpler and shallower as compared to others and therefore produced significantly better results. In fact, the proposed customized network uses a similar stacking of convolution layers as in the VGG architecture. Several studies have found immense success with transfer learning and there is scope to further the experimentations conducted in this research, including dropping more layers to reduce the learning parameters, changing the classifiers, and re-training them with additional samples as the dataset increases, to improve the results. Another implementation would be to employ the pre-trained networks as a feature extractor followed by a gradient-boosting tree classifier.

The scope of this study did not include investigating the generalization capabilities of the proposed network on a different set of CVR maps, obtained using a different vasoactive stimulus, for example, breath-hold, fixed inspired CO₂, and chemical stimuli. As discussed earlier, research has shown that such transfer learning experiments with closely related datasets can achieve better performance than training a new network from scratch.

Finally, the use of CVR images in the Montreal Neurological Institute (MNI) coordinate space can be investigated, which would standardize the research, furthering the generalization capabilities of Artificial-Intelligence-driven CVR analyses.

6. Conclusions

The objective of this research was to further the current state of the application of deep learning in medical imaging by developing a clinical decision support system that can facilitate the screening of steno-occlusive disease (SOD) patients using cerebrovascular reactivity (CVR) maps. The CVR maps were obtained from clinical breath-control studies using prospective end-tidal targeting to administer a controlled $P_{ET}CO_2$ stimulus, accomplished using the RespirAct device. Image augmentation techniques were used to increase the number of input samples.

An empirical evaluation-based network optimization strategy was implemented for the customized convolutional neural networks (CNNs), and transfer learning was investigated by importing and modifying some of the popular pre-trained networks. Based on the experiments, a customized shallow CNN with two convolution layers and one hidden fully connected layer is proposed as the most optimal classifier. Results conclude that the proposed network successfully identifies the key features in the dataset considered to discriminate between healthy control subjects and patients with SOD. The generalization capabilities of this network are consistent with expert clinical readings, with only one incorrect prediction out of the 22 samples in the test set.

While the proposed network is not production-ready, it can be used as a research tool to facilitate clinical decision-making and support CVR research advancements.

Author Contributions: Y.D. conceived and conducted the experiments. M.B.K. supervised the research plan and results. A.E.P., D.J.M., E.S.S., H.T.L., J.D., J.A.F., J.P. and O.S. conducted clinical studies to collect the data used in this research. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Thornhill Research Inc., Toronto, ON, Canada.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the University Health Network Research Ethics Board (Study 13-7168, 20 January 2014).

Informed Consent Statement: The studies involving human participants were reviewed and approved by University Health Network. The patients/participants provided their written informed consent to participate in this study.

Data Availability Statement: The datasets and programs used and developed in this article are not readily available because of confidentiality agreements. Anonymized data and programs will be shared by request from any qualified investigator for purposes such as replicating procedures and results presented in the article provided that data transfer is in agreement with the University Health Network and Health Canada legislation on general data protection regulation. Requests to access the datasets should be directed to Olivia Sobczyk, olivia.sobczyk@uhn.ca.

Acknowledgments: This research was carried out in the Maglev Microrobotics Laboratory at the University of Waterloo, Waterloo, Canada. Special thanks to Kevin Morwood, Software Engineering Manager, Thornhill Medical, for sharing his invaluable insights and for his guidance in software-engineering-related activities throughout this work.

Conflicts of Interest: J.A.F. and D.J.M. contributed to the development of the automated end-tidal targeting device, RespirAct™ (Thornhill Research Inc., TRI) used in this study and have equity in the company. O.S. and J.D. receive salary support from TRI. TRI provided no other support for the study. All other authors have no disclosures to report.

References

1. Cerebrovascular Disease | Michigan Medicine. Available online: <https://www.uofmhealth.org/conditions-treatments/brain-neurological-conditions/cerebrovascular> (accessed on 5 January 2023).
2. Feigin, V.L.; Brainin, M.; Norrving, B.; Martins, S.; Sacco, R.L.; Hacke, W.; Fisher, M.; Pandian, J.; Lindsay, P. World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. *Int. J. Stroke* **2022**, *17*, 18–29. [CrossRef] [PubMed]

3. Martinez-Rodriguez, J.E.; Munteis, E.; Gomis, M.; Rodríguez-Campello, A.; Jimenez-Conde, J.; Cuadrado-Godia, E.; Roquer, J.; Ois, A. Stenoocclusive arterial disease and early neurological deterioration in acute ischemic stroke. *Cerebrovasc. Dis.* **2008**, *25*, 151–156.
4. Lee, P.H.; Oh, S.H.; Bang, O.Y.; Joo, I.S.; Huh, K. Isolated middle cerebral artery disease: Clinical and neuroradiological features depending on the pathogenesis. *J. Neurol. Neurosurg. Psychiatry* **2004**, *75*, 727–732. [CrossRef] [PubMed]
5. Sebök, M.; van Niftrik, C.; Germans, M.; Katan, M.; Kulcsar, Z.; Luft, A.; Regli, L.; Fierstra, J. Recurrent stroke in symptomatic steno-occlusive disease: Identifying patients at high-risk using impaired BOLD cerebrovascular reactivity. *Brain Spine* **2022**, *2*, 101215. [CrossRef]
6. Gaillard, F. BOLD Imaging | Radiology Reference Article | Radiopaedia.org. Radiopaedia. Retrieved 4 November 2022. Available online: <https://radiopaedia.org/articles/bold-imaging> (accessed on 3 April 2023).
7. Sleight, E.; Stringer, M.S.; Marshall, I.; Wardlaw, J.M.; Thruppelton, M.J. Cerebrovascular reactivity measurement using magnetic resonance imaging: A systematic review. *Front. Physiol.* **2021**, *12*, 643468. [CrossRef]
8. Leoni, R.F.; Mazzetto-Betti, K.C.; Silva, A.C.; Dos Santos, A.C.; de Araujo, D.B.; Leite, J.P.; Pontes-Neto, O.M. Assessing cerebrovascular reactivity in carotid steno-occlusive disease using MRI bold and asl techniques. *Radiol. Res. Pract.* **2012**, *2012*, 268483. [CrossRef]
9. Fisher, J.A.; Venkatraghavan, L.; Mikulis, D.J. Magnetic resonance imaging-based cerebrovascular reactivity and hemodynamic reserve. *Stroke* **2018**, *49*, 2011–2018. [CrossRef]
10. Vorstrup, S.; Brun, B.; Lassen, N.A. Evaluation of the cerebral vasodilatory capacity by the acetazolamide test before EC-IC bypass surgery in patients with occlusion of the internal carotid artery. *Stroke* **1986**, *17*, 1291–1298. [CrossRef]
11. Slessarev, M.; Han, J.; Mardimae, A.; Prisman, E.; Preiss, D.; Volgyesi, G.; Ansel, C.; Duffin, J.; Fisher, J.A. Prospective targeting and control of endtidal CO₂ and O₂ concentrations. *J. Physiol.* **2007**, *581*, 1207–1219. [CrossRef]
12. RespirAct®RA-MRTM, Thornhill Medical Canada. Available online: <https://thornhillmedical.ca/research/respiract-ra-mr/> (accessed on 3 April 2023).
13. Sobczyk, O.; Sayin, E.S.; Sam, K.; Poubanc, J.; Duffin, J.; Fisher, J.A.; Mikulis, D.J. The Reproducibility of Cerebrovascular Reactivity Across MRI Scanners. *Front. Physiol.* **2021**, *12*, 668662. [CrossRef]
14. Fisher, J.A.; Mikulis, D.J. Cerebrovascular reactivity: Purpose, optimizing methods, and limitations to interpretation—A personal 20-year odyssey of (re)searching. *Front. Physiol.* **2021**, *12*, 621651. [CrossRef]
15. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Available online: <http://www.deeplearningbook.org> (accessed on 7 August 2023).
16. Build with AI | DeepAI. Available online: <https://deepai.org/machine-learning-glossary-andterms/neural-network> (accessed on 3 April 2023).
17. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
18. Wu, H.; Gu, X. Towards dropout training for convolutional neural networks. *Neural Netw.* **2015**, *71*, 1–10. [CrossRef] [PubMed]
19. Activation Function. DeepAI. 27 September 2020. Available online: <https://deepai.org/machine-learning-glossary-and-terms/activation-function> (accessed on 22 June 2023).
20. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *arXiv* **2018**, arXiv:1811.03378.
21. Nirthika, R.; Manivannan, S.; Ramanan, A.; Wang, R. Pooling in convolutional neural networks for medical image analysis: A survey and an empirical study. *Neural Comput. Appl.* **2022**, *34*, 5321–5347. [CrossRef]
22. Kim, H.E.; Cosa-Linan, A.; Santhanam, N.; Jannesari, M.; Maros, M.E.; Ganslandt, T. Transfer learning for medical image classification: A literature review. *BMC Med. Imaging* **2022**, *22*, 1–13. [CrossRef] [PubMed]
23. Shen, D.; Wu, G.; Suk, H.-I. Deep Learning in Medical Image Analysis. *Annu. Rev. Biomed. Eng.* **2017**, *19*, 221–248. [CrossRef]
24. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef]
25. Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep Learning Applications in Medical Image Analysis. *IEEE Access* **2017**, *6*, 9375–9389. [CrossRef]
26. Farooq, A.; Anwar, S.; Awais, M.; Rehman, S. A deep CNN based multiclass classification of Alzheimer’s disease using MRI. In Proceedings of the 2017 IEEE International Conference on Imaging Systems and Techniques (IST), Beijing, China, 18–20 October 2017; pp. 1–6. [CrossRef]
27. Chen, D.Y.T.; Ishii, Y.; Fan, A.P.; Guo, J.; Zhao, M.Y.; Steinberg, G.K.; Zaharchuk, G. Predicting PET Cerebrovascular Reserve with Deep Learning by Using Baseline MRI: A Pilot Investigation of a Drug-Free Brain Stress Test. *Radiology* **2020**, *296*, 627–637. [CrossRef]
28. Hussein, R.; Zhao, M.; Shin, D.; Guo, J.; Chen, K.T.; Armindo, R.D.; Davidzon, G.; Moseley, M.; Zaharchuk, G. Mul-ti-task Deep Learning for Cerebrovascular Disease Classification and MRI-to-PET Translation. *arXiv* **2022**, arXiv:2202.06142.
29. Hashemzahi, R.; Mahdavi, S.J.S.; Kheirabadi, M.; Kamel, S.R. Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE. *Biocybern. Biomed. Eng.* **2020**, *40*, 1225–1232. [CrossRef]
30. Meijs, M.; Meijer, F.J.; Prokop, M.; van Ginneken, B.; Manniesing, R. Image-level detection of arterial occlusions in 4D-CTA of acute stroke patients using deep learning. *Med. Image Anal.* **2020**, *66*, 101810. [CrossRef]

31. Hou, X.; Guo, P.; Wang, P.; Liu, P.; Lin, D.D.M.; Fan, H.; Li, Y.; Wei, Z.; Lin, Z.; Jiang, D.; et al. Deep-learning-enabled Brain He-modynamic Mapping Using Resting-state fMRI. *arXiv* **2022**, arXiv:2204.11669.
32. Zhu, G.; Chen, H.; Jiang, B.; Chen, F.; Xie, Y.; Wintermark, M. Application of Deep Learning to Ischemic and Hemorrhagic Stroke Computed Tomography and Magnetic Resonance Imaging. *Semin. Ultrasound CT MRI* **2022**, *43*, 147–152. [[CrossRef](#)] [[PubMed](#)]
33. Taló, M.; Baloglu, U.B.; Yıldırım, Ö.; Acharya, U.R. Application of deep transfer learning for automated brain ab-normality classification using MR images. *Cogn. Syst. Res.* **2019**, *54*, 176–188. [[CrossRef](#)]
34. Maqsood, M.; Nazir, F.; Khan, U.; Aadil, F.; Jamal, H.; Mehmood, I.; Song, O.Y. Transfer learning assisted classification and detection of Alzheimer's disease stages using 3D MRI scans. *Sensors* **2019**, *19*, 2645. [[CrossRef](#)]
35. Yuan, Y.; Qin, W.; Buyyounouski, M.; Ibragimov, B.; Hancock, S.; Han, B.; Xing, L. Prostate cancer classification with multiparametric MRI transfer learning model. *Med. Phys.* **2018**, *46*, 756–765. [[CrossRef](#)]
36. Ghafoorian, M.; Mehrtash, A.; Kapur, T.; Karssemeijer, N.; Marchiori, E.; Pesteie, M.; Guttman, C.R.; de Leeuw, F.-E.; Tempny, C.M.; Van Ginneken, B. Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2017, Proceedings of the 20th International Conference, Quebec City, QC, Canada, 11–13 September 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 516–524.
37. Waddle, S.L.; Juttukonda, M.R.; Lants, S.K.; Davis, L.T.; Chitale, R.; Fusco, M.R.; Jordan, L.C.; Donahue, M.J. Classifying intracranial stenosis disease severity from functional MRI data using machine learning. *J. Cereb. Blood Flow Metab.* **2019**, *40*, 705–719. [[CrossRef](#)]
38. Klöppel, S.; Stonnington, C.M.; Chu, C.; Draganski, B.; Scallan, R.I.; Rohrer, J.D.; Fox, N.C.; Jack, C.R.; Ashburner, J.; Frackowiak, R.S.J. Automatic classification of MR scans in Alzheimer's disease. *Brain* **2008**, *131*, 681–689. [[CrossRef](#)]
39. Zacharaki, E.I.; Wang, S.; Chawla, S.; Yoo, D.S.; Wolf, R.; Melhem, E.R.; Davatzikos, C. Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme. *Magn. Reson. Med.* **2009**, *62*, 1609–1618. [[CrossRef](#)] [[PubMed](#)]
40. Bahadure, N.B.; Ray, A.K.; Thethi, H.P. Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM. *Int. J. Biomed. Imaging* **2017**, *2017*, 9749108. [[CrossRef](#)] [[PubMed](#)]
41. Dasari, Y. Deep Learning-Enabled Cerebrovascular Reactivity Processing Software. UWSpace. 2022. Available online: <http://hdl.handle.net/10012/19002> (accessed on 7 August 2023).
42. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 1–74. [[CrossRef](#)]
43. Machine Learning Glossary. Google Developers. Available online: <https://developers.google.com/machine-learning/glossary> (accessed on 3 April 2023).
44. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2015. Available online: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45166.pdf> (accessed on 22 June 2023).
45. Keras Documentation. Available online: <https://keras.io> (accessed on 26 December 2018).
46. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Fukushima, K. Cognitron: A self-organizing multilayered neural network. *Biol. Cybern.* **1975**, *20*, 121–136. [[CrossRef](#)]
48. Team, K. Keras Documentation: MaxPooling2D Layer. Keras.io. Available online: https://keras.io/api/layers/pooling_layers/max_pooling2d/ (accessed on 3 April 2023).
49. Team, K. Keras Documentation: EarlyStopping. Keras.io. Available online: https://keras.io/api/callbacks/early_stopping/ (accessed on 3 April 2023).
50. What Is Overfitting? | IBM. www.ibm.com. Available online: <https://www.ibm.com/topics/overfitting> (accessed on 3 April 2023).
51. Hinton, G.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
52. Team, K. Keras Documentation: BatchNormalization Layer. Keras.io. Available online: https://keras.io/api/layers/normalization_layers/batch_normalization/ (accessed on 3 April 2023).
53. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]
54. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. May 2019. Available online: <http://arxiv.org/abs/1905.11946> (accessed on 22 June 2023).
55. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.
56. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
57. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015*. Available online: <https://arxiv.org/abs/1409.1556> (accessed on 22 June 2023).
58. Team, K. Keras Documentation: Transfer Learning & Finetuning. Keras.io. Available online: https://keras.io/guides/transfer_learning/ (accessed on 3 April 2023).

59. Bengio, Y. Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
60. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. Available online: <https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf> (accessed on 22 June 2023).
61. BN Questions (Old) Issue 1802·Kerasteam/Keras. GitHub. Retrieved 4 April 2023. Available online: <https://github.com/keras-team/keras/issues/1802#issuecomment-187966878> (accessed on 3 April 2023).
62. Keras. ReduceLROnPlateau. Available online: https://keras.io/api/callbacks/reduce_lr_on_plateau/ (accessed on 21 October 2022).
63. Teng, L.; Qiao, Y.; Shafiq, M.; Srivastava, G.; Javed, A.R.; Gadekallu, T.R.; Yin, S. FLPK-BiSeNet: Federated Learning Based on Prior Knowledge and Bilateral Segmentation Network for Image Edge Extraction. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 1529–1542. [[CrossRef](#)]
64. Sharma, S.; Gupta, S.; Gupta, D.; Rashid, J.; Juneja, S.; Kim, J.; Elarabawy, M.M. Performance Evaluation of the Deep Learning Based Convolutional Neural Network Approach for the Recognition of Chest X-Ray Images. *Front. Oncol.* **2022**, *12*, 932496. [[CrossRef](#)]
65. Glodzik, L.; Randall, C.; Rusinek, H.; de Leon, M.J. Cerebrovascular Reactivity to Carbon Dioxide in Alzheimer’s Disease. *J. Alzheimer’s Dis.* **2013**, *35*, 427–440. [[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.