

A Classifier for Automatic Categorisation of Chronic Venous Insufficiency Images

Talha Karadeniz^{1*}, Gul Tokdemir², H. Hakan Maras³

¹Department of Software Engineering, Cankaya University,
Eskisehir Yolu, 29. km, Ankara, 06790, Turkiye

²Department of Computer Engineering, Cankaya University,
Eskisehir Yolu, 29. km, Ankara, 06790, Turkiye

³Department of Computer Programming, Cankaya University,
Eskisehir Yolu, 29. km, Ankara, 06790, Turkiye

*talhakaradeniz@cankaya.edu.tr; gtokdemir@cankaya.edu.tr; hhmaras@cankaya.edu.tr

Abstract—Chronic venous insufficiency (CVI) is a serious disease characterised by the inability of the veins to effectively return blood from the legs back to the heart. This condition represents a significant public health issue due to its prevalence and impact on quality of life. In this work, we propose a tool to help doctors effectively diagnose CVI. Our research is based on extracting Visual Geometry Group network 16 (VGG-16) features and integrating a new classifier, which exploits mean absolute deviation (MAD) statistics to classify samples. Although simple in its core, it outperforms state-of-the-art method which is known as the CVI-classifier in the literature, and additionally it performs better than the methods such as multi-layer perceptron (MLP), Naive Bayes (NB), and gradient boosting machines (GBM) in the context of VGG-based classification of CVI. We had 0.931 accuracy, 0.888 Kappa score, and 0.916 F1-score on a publicly available CVI dataset which outperforms the state-of-the-art CVI-classifier having 0.909, 0.873, and 0.900 for accuracy, Kappa score, and F1-score, respectively. Additionally, we have shown that our classifier has a generalisation capacity comparable to support vector machines (SVM), by conducting experiments on eight different datasets. In these experiments, it was observed that our classifier took the lead on metrics such as F1-score, Kappa score, and receiver operating characteristic area under the curve (ROC AUC).

Index Terms—Classification algorithms; Decision support systems; Particle swarm optimisation.

I. INTRODUCTION

CVI is a serious condition that affects public health and the automatic classification of this state is very beneficial for doctors working in the area. In this work, we propose a “classical” machine learning method integrated to deep learning (DL)-based feature extraction, yielding comparatively superior results in a public dataset containing 221 images.

In this research, we intend to produce a decision support system to help doctors diagnose CVI. Since such a condition is fatal to humanity, the development of this system can be very beneficial.

Our work is highly inspired (methodologically, not technically) by the study in [1], where a specific classifier is

constructed to successfully categorise CVI images. Our research is based on exploiting descriptive statistics (namely, mean absolute deviation (MAD)) to classify images using Visual Geometry Group network 16 (VGG-16) features. Thus, it can be considered as a “hybrid” model fusing the power of DL with the simplicity and the capacity of a MAD-based approach to categorise samples.

The structure of this article is organised as follows. Initially, a review of literature on CVI classification and related studies is provided. Then, brief explanations of each “classical” classifier tested is presented. Afterwards, our methodology and model are proposed. Following this, the description of the dataset used and the experimental results are given, after which we start a discussion on our classifier’s status.

II. RELATED WORK

In [1], researchers propose a bag of visual words variant approach to automatically categorise the CVI image data. They benefit from a supervised learning of patches to obtain a concept classifier. In their study, their method outperforms the decisions made by professional doctors. This is the work we take as the state-of-the-art since it has outstanding success in classifying CVI images and has a decent methodology which we replicate in our experiments.

The authors in [2] develop a DL-based approach where the thermal imaging data are used. They compare their results with architectures such as VGG-16, EfficientNet-B0, and Resnet-152. While these models produce accuracy scores 0.947, 0.953, and 0.958, respectively, CVINet (the method proposed in [2]) obtains a 0.968 on binary classification, which is promising. Like in [1], their model outperforms the decisions of the clinician. Their dataset consists of 960 images that have binary labels: CVI class and normal class. Their training process involves 25 epochs, batch size 32, and optimiser set to stochastic gradient descent (SGD) with a learning rate 0.0001 and a dropout rate of 0.5.

The authors in [3] develop another DL-based approach where the input images are the same as the input images in [1]. They apply “almost” the same methodology as the authors in [1] except that in [1] a 20-run average is reported,

while in [3] one run is reported with an accuracy of 0.999, which makes their claim on the “outperforming nature” of their approach somehow problematic. Additionally, in their results they list accuracy scores from several works, but the table is not standardized, i.e., input image types, contents, and cardinalities vary. On the other hand, the study is valuable in the aspect that it collects numerous studies in a structured manner, despite their not very healthy style of reporting/comparing accuracy scores. Their model consists of eight convolutional layers with 3×3 filters, nine exponential linear unit (ELU) layers, and nine batch normalisation layers.

In [4], a new tissue classification procedure is presented for the clinical classification of varicose ulcers, leveraging medical image analysis and knowledge engineering methods. The study proposes a comprehensive methodology including preprocessing through image registration and difference image calculation, segmentation, feature extraction, and finally classification using a k-nearest neighbour (KNN) classifier. The feature extraction process describes colour features such as colour correlogram and colour moments, texture features which contain homogeneity, contrast, and correlation, along with shape features such as solidity and eccentricity. These extracted features are classified to discern the various stages of wound healing, demonstrating efficient performance with average sensitivity, specificity, and accuracy rates of 0.952, 0.944, and 0.948, respectively.

The authors in [5] propose a varicose ulcer wound classification system consisting of the image preprocessing steps (where the reflection of flashlight is removed), contour segmentation (to separate the wounded areas from the skin), and multidimensional convolutional neural network (for which the segmented images and ground truth data are input). They use Matlab for the implementation and obtain 0.995 accuracy, 0.980 specificity, and 0.956 sensitivity.

The authors in [6] present a method to detect varicose veins by using thermal imaging. Their methodology involves two main phases: first, image acquisition (where a FLIR ONE is used), and second, image processing. In the image processing stage, the thermal image is first converted to greyscale. Then, the segments of the thermal image are extracted. Afterwards, segment areas with higher temperature are detected. Following this, an edge-finding operation is applied via the Roberts method. Following that, morphological structuring is utilised to improve edge detection and then the resulting image is converted to red-green-blue (RGB) to fuse with the original RGB image. Although this work is valuable in the context of varicose vein detection, no evaluation metrics are reported.

The authors in [7] look for association between the recorded skin temperature and the severity of chronic venous diseases (CVD). To accomplish the task, they acquire infrared images from 36 patients. Then they analyse the regions “with respect to seven predefined features”. According to their findings, mild forms of CVD are correlated with local increases in skin temperature. When CVD is more severe, these regions with local increases are 2.0° warmer. The findings indicate that early detection of the disease is possible by infrared thermography.

The authors in [8] consider fuzzy c-means (FCM) for feature extraction. Their pipeline involves steps of image data acquisition, preprocessing, FCM clustering, and the output of

segmented images. The advantage of the work is the application of fuzzy logic and FCM to the domain of varicose vein analysis. The disadvantages are that the work is somehow disorganised (following the steps involved is hard from the presentation), and the detection/classification rate/success is not decently reported. Rather than well-defined metrics, a screen showing “Varicose is observed” is offered.

In the context of VGG - or more specifically DL - and particle swarm optimisation (PSO) combination, the authors in [9] present a “swarm optimised block architecture”, where the blocks are optimised via adaptive acceleration. They obtain 4.78 % error rate in the CIFAR-10 image classification task and succeed to get a 25.42 % error rate in CIFAR-100 classification task. This differs from our work in the sense that for our work, VGG features are directly used and PSO is integrated in the framework to obtain a “classical” classifier.

The authors of [10] present a similar strategy to generate deep neural networks, namely by incorporating PSO to obtain optimal architectures. Based on their report, one can say that on the datasets such as Convex, Rectangles, and MNIST, 7.58 % increase in accuracy and up to 63 % improvement in computational cost are observed.

The authors in [11] present a method to improve fuzzy c-means clustering by the integration of a novel PSO algorithm.

The authors in [12] combine FCM with PSO to create a classifier based on fuzzy systems. The classifier they propose is different from ours in the sense that MAD-statistics and integration through majority voting is not used. On the other hand, their classifier also shows good results in the experiments conducted using the Iris, Ionosphere, and Glass datasets.

The authors in [13] utilise PSO-based FCM segmentation to predict leaf diseases. They use Gaussian mixture model-based background subtraction to preprocess the image and then perform classical feature extraction steps such as edge and “texture features”. They, finally, use a multi kernel parallel SVM (MK-PSVM) to classify the features.

III. METHODOLOGY

The fuzzy c-means (FCM) [14] algorithm is a clustering method that allows one sample to belong to two or more clusters. This is different from the hard clustering methods such as k-means, in which each sample belongs to exactly one cluster. In FCM, each sample has a degree of belonging to clusters and this degree is calculated according to the distance between the sample and the cluster centre. The closer the centre, the higher the degree of belonging, and this belonging is computed in probabilities that sum to one for each sample.

Particle swarm optimisation (PSO) is an evolutionary computational method that simulates the social behaviour of swarms, such as bird flocking or fish schooling, to find optimal solutions to complex problems. In PSO, a population of particles explores the search space by adjusting their trajectories based on their own experience and the experiences of neighbouring particles. Each particle has a potential solution to the optimisation problem and has a position and velocity in the search space. The algorithm iteratively regulates these positions and velocities based on the particles’ historical best positions (personal best) and the global best position found by any particle in the swarm [15].

The influence of the personal and global best guides the swarm towards the most promising areas of the search space, aiming to converge on the optimal solution. One point to note is that, in PSO, the gradient of the objective function is not needed [16].

The VGG-16 architecture is a deep convolutional neural network model. It was proposed by Karen Simonyan and Andrew Zisserman of the VGG at the University of Oxford in their paper [17]. In the computer vision community, it has been widely used for a variety of tasks due to its simple, yet powerful architecture. It has a deep structure that involves 16 layers that have weights, which include 13 convolutional layers and three fully connected layers.

Its architecture details contain the following.

- Input layer: This layer accepts images of dimension 224×224 which has red, green, and blue channels.
- Convolutional layers: The architecture is composed of a series of convolutional layers with small receptive fields of 3×3 , which is the smallest size to capture the notion of left/right, up/down, centre. The stride is fixed to one pixel.
- Activation function: Following each convolution operation, a rectified linear unit (ReLU) activation function is used to introduce nonlinearity into the model. In this way, it can learn more complex patterns.

- Pooling layers: There are five max-pooling layers in the network, which is after some of the convolutional layers. These are used to shrink the dimensionality of the feature maps. Hence, shrinking the number of parameters and computation in the network. The pooling size is 2×2 pixels.

In measuring accuracy, Kappa score, and macro-F1-score, the method given in [1] is strictly followed. 20-runs of measurements are executed and the average is reported for the scores. The train-test split is done with a 1/3 and 2/3 ratio, the latter representing the amount of data separated for training.

Our overall algorithm can be summarised as follows (please see Fig. 1 and Fig. 2):

- Extract VGG-16 features from all images;
- Use proposed binary classifier as a basis to train a one-vs-one multiclass classifier;
- Train and test using this multiclass classifier.

The proposed binary classifier runs as follows:

Fitting phase:

- Get the training features and labels;
- Separate the features as X^+ and as X^- depending on the class labels (which means that X^+ is the training samples

with positive class while X^- contains the samples with negative class);

- Fit fuzzy c-means to each of the positive and negative sets, namely to X^+ and X^- ;
- Get the cluster centres of each fit to X_1 and X_0 , respectively, meaning that X_1, X_0 have the cluster centres of FCM fitted to X^+ and X^- , respectively;
- Optimise the coefficients of the absolute value distances to clusters with PSO with respect to the average of accuracy, the F1-score, and the Kappa score. Make predictions during this optimisation according to the following “Prediction” method.

Prediction phase:

- Get the input sample x ;
- Store the *weighted* (weights are determined/tested by the PSO) mean absolute deviation array of X_1 about the input sample in mad_{tmp} and take the exponential of each element in the array. Record the resulting array in mad_1 . That is, we measure weighted mean absolute deviation of each component of X_1 about the sample x into mad_{tmp} and take the exponential of each item in mad_{tmp} to find mad_1 . Do the same for negative class and measure exponential-MAD array of X_0 about the input sample into mad_0 ;
- Initialise s_0 to 0. Initialise s_1 to 0. We hold the total vote for each class in these variables;
- For $i = 1$ to $\text{size}(mad_0)$:
 - if $mad_1[i] < mad_0[i]$, add $mad_0[i] - mad_1[i]$ to s_1 ,
 - else add $mad_1[i] - mad_0[i]$ to s_0 ;
- If $s_1 > s_0$, return 1 (sample belongs to the positive class);
- Else return 0.

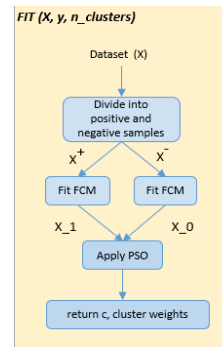


Fig. 1. Fitting Phase.

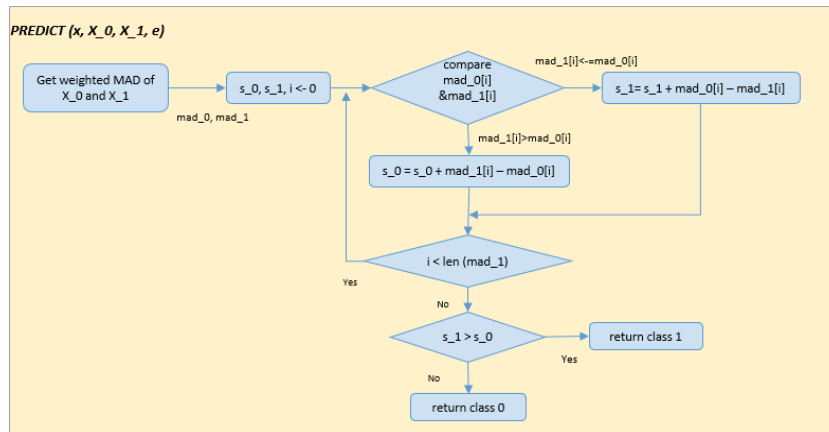


Fig. 2. Predicting phase.

One-vs-one multiclass support in sklearn [18] also needs a *predict_proba* method which must return the probability of the classes. For this, we simply return the pair $(\frac{s_0}{s_0+s_1}, \frac{s_1}{s_0+s_1})$. Recall that s_0 and s_1 are the sums of MAD-differences for class 0 and 1, respectively.

One can summarise the method as follows: we characterise the positive and negative classes with the cluster centres of FCMs. We measure the MAD of cluster centres about the input sample and take an exponential of each dimension. We compare MAD-exponential scores and decide to favour the class which has less MAD-exponential (i.e., the class which deviates less from the test sample). MADs are measured in a weighted manner by which we mean each cluster centre has its own weight, and these are determined by the PSO with objective function set to the average of training accuracy, F1-score and Cohen Kappa score.

It can also be noted that the *find_weights_pso* function given in the pseudocode is not very straightforward (it is kept simple in the pseudocode for the aim of not distracting the reader). In the implementation, an *obj_func* method is given as a parameter to a GlobalBestPSO object. *obj_func* then calls *measure_acc* several times. *measure_acc* measures the mean of accuracy, F1-score, and Cohen Kappa score for a particle (set of coefficients). It does this by calling the internalised *predict*, which is *__predict0*. *__predict0* calls the internalised *__get_mad*, which is *__get_mad0*. In summary, application of the PSO in the algorithm is not very straightforward. You can see the call graph of the classifier from the Fig. 3.

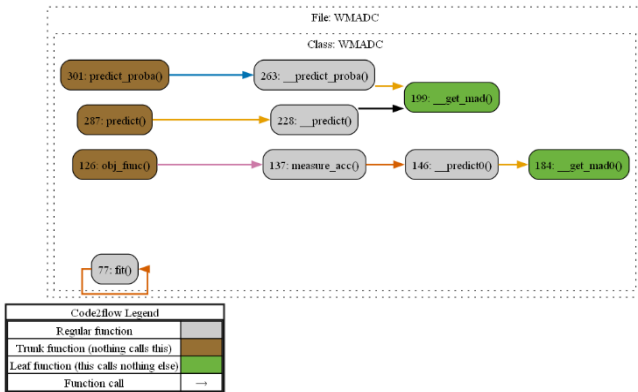


Fig. 3. Call graph of WMADC.

The novelty of the method lies in combining the majority vote scheme with the idea of “MAD about the input sample” (Fig. 4). Normally, MAD is calculated about a mean or another descriptive statistic. However, in our classifier, it is calculated about the input sample (test vector). Accumulating the differences (corresponds to the “ $s_1 < -s_1 + mad_0[i] - mad_1[i]$ ” part of the pseudocode) and integrating the PSO to find the weights of the clusters can be considered as the other novel aspects of the work.

For the default parameters experiment, a train-test split of 1:2 is performed, and all measurements of metrics are done in this setting.

For hyperparameter tuning, a three-fold cross-validation grid search is applied to select the optimal parameters for the modelling. After this, again a train-test split of 1:2 is performed to measure the final outcomes.

Algorithm Proposed Method: Weighted Mean Absolute Deviation Classifier (WMADC)

```

1: procedure FIT( $X, y, n\_clusters$ ) ▷ Training dataset ( $X$ ), class labels ( $y$ ) and
   number of clusters
2:    $X^+ \leftarrow get\_samples(X, y, 1)$  ▷ Get positive samples
3:    $X^- \leftarrow get\_samples(X, y, 0)$  ▷ Get negative samples
4:    $X_{+1} \leftarrow FCM(X^+, n\_clusters)$  ▷ Fit a FCM and find positive cluster centers
5:    $X_{-1} \leftarrow FCM(X^-, n\_clusters)$  ▷ Fit a FCM and find negative cluster centers
6:    $c \leftarrow find\_weights\_pso(X, y, X_{+1}, X_{-1})$  ▷ Apply PSO to find weights of
   clusters for the weighted calculation of Mean Absolute Deviation
7:   return ( $X_{+1}, X_{-1}, c$ )
8: end procedure
9: procedure PREDICT( $x, X_{+1}, X_{-1}, c$ ) ▷  $X_{+1}, X_{-1}$  and  $c$  are results of the FIT
   procedure,  $x$  is the test vector
10:   $mad_{+1} \leftarrow get\_mad(X_{+1}, x, c)$  ▷ Get weighted MAD of  $X_{+1}$  about the sample  $x$ 
11:   $mad_{-1} \leftarrow get\_mad(X_{-1}, x, c)$  ▷ Get weighted MAD of  $X_{-1}$  about the sample  $x$ 
12:   $s_{+1} \leftarrow 0$ 
13:   $s_{-1} \leftarrow 0$ 
14:   $i \leftarrow 0$ 
15:  while  $i < len(mad_{+1})$  do
16:    if  $mad_{+1}[i] < mad_{-1}[i]$  then
17:       $s_{+1} \leftarrow s_{+1} + mad_{+1}[i] - mad_{-1}[i]$  ▷ Accumulate the differences
18:    else
19:       $s_{-1} \leftarrow s_{-1} + mad_{-1}[i] - mad_{+1}[i]$ 
20:    end if
21:     $i \leftarrow i + 1$ 
22:  end while
23:  if  $s_{+1} > s_{-1}$  then
24:     $ret\_val \leftarrow 1$ 
25:  else
26:     $ret\_val \leftarrow 0$ 
27:  end if
28:  return  $ret\_val$ 
29: end procedure

```

Fig. 4. Pseudocode for the algorithm.

IV. EXPERIMENTS

For this research, the dataset used in [1] is retrieved from <http://isyslab.info/CVI/CVI-img-datasets.zip>, a total of 211 images with three classes: mild, moderate, and severe.

We have also tested the generalisation ability of our classifier on eight datasets obtained from UCI Machine Learning repository and compared it with nine classifiers available from sklearn.

Eight datasets are:

- Breast cancer;
- Heart failure;
- Fertility;
- Parkinson’s disease;
- Haberman’s survival;
- Breast cancer Coimbra;
- Blood Transfusion Service Center;
- SPECTF Heart.

The compared classifiers are:

- Decision tree (DT);
- Gaussian Naïve Bayes (GaussianNB);
- Support vector machines (SVC);
- Multi-layer perceptron (MLP);
- Logistic regression (LR);
- Passive aggressive classifier (PAC);
- Perceptron (PCT);
- Ridge classifier (RC);
- SGD classifier (SGD).

The programming platform is Colab Pro. The PSO implementation used is pyswarms, and the FCM implementation is fuzzy-c-means.

V. RESULTS

The CVI results can be seen from Table I. “Doctors” and “CVI-classifier” results are taken from the work in [1]. Others are the VGG combined with classical machine learning

algorithms tested with default implementation parameters. In our method, $n_clusters$ is taken as 3. For PSO, the maximum iteration is set to 30, the number of particles is set to 11 and the remaining parameters are taken from the default settings. The bounds are taken as $(-5, 5)$ where -5 is the minimum, 5 is the maximum. Before running each classifier, we have applied min-max scaling, and we have selected features through RFECV with estimator set to linear SVM. Note that, except for min-max scaling, this is not done in the generic classification experiments (i.e., no feature selection is performed on generic classification). Test runs results are visualised at Fig. 5–10.

TABLE I. CVI RESULTS.

Method	Accuracy	F1-Score	Kappa
Doctors	0.818	0.805	0.748
CVI-Classifer [1]	0.909	0.900	0.873
VGG-MLP	0.926	0.909	0.881
VGG-NB	0.924	0.875	0.90
VGG-GBM	0.912	0.893	0.855
VGG-RF	0.913	0.895	0.861
VGG-SVM	0.924	0.907	0.879
Proposed Method (PM)	0.931	0.9162	0.888

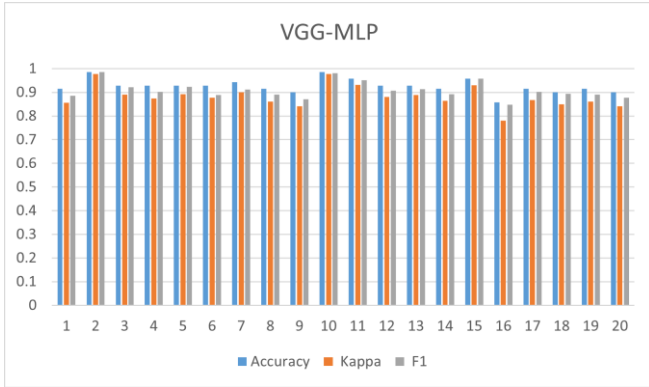


Fig. 5. VGG-MLP test runs.

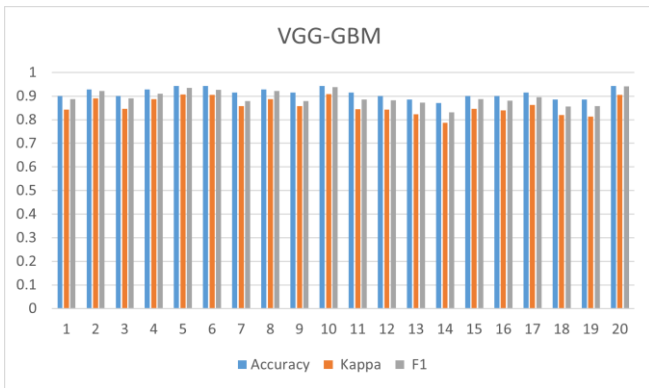


Fig. 6. VGG-GBM test runs.

The first part of the generic classification results can be seen from Tables II–V where F1-score, Kappa score, and receiver operating characteristic area under the curve (ROC AUC) scores are measured, respectively (in the tables, “PM” stands for “Proposed Method”). In these experiments, to be fair, the iteration count and number of particles are set to the same numbers as in CVI. Other classifiers are also used with their default parameters on sklearn.

The second part of the generic classification results

involves hyperparameter tuning. A total of 12 parameter combinations for grid search is used for each classifier. These can be seen from the parameter Tables V–XIV.

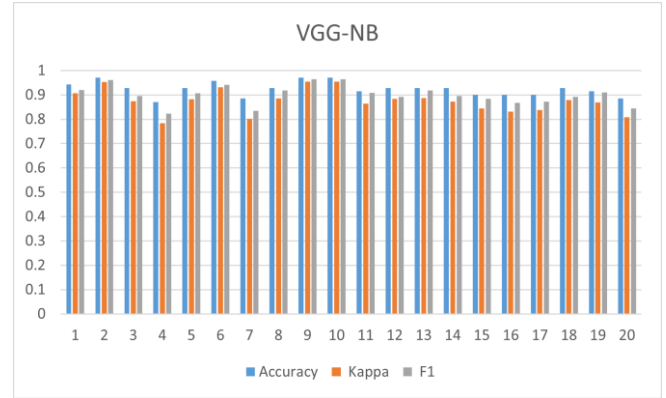


Fig. 7. VGG-NB test runs.

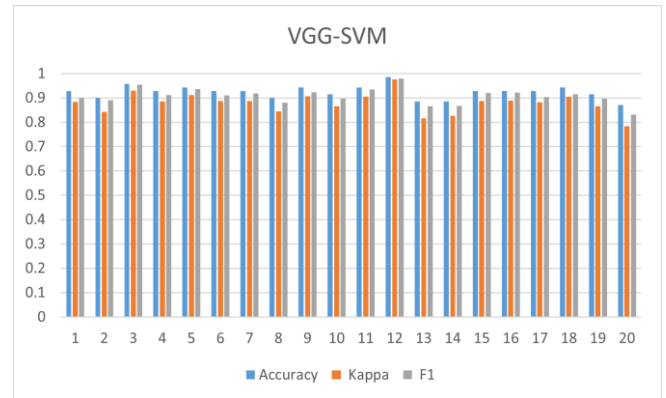


Fig. 8. VGG-SVM test runs.

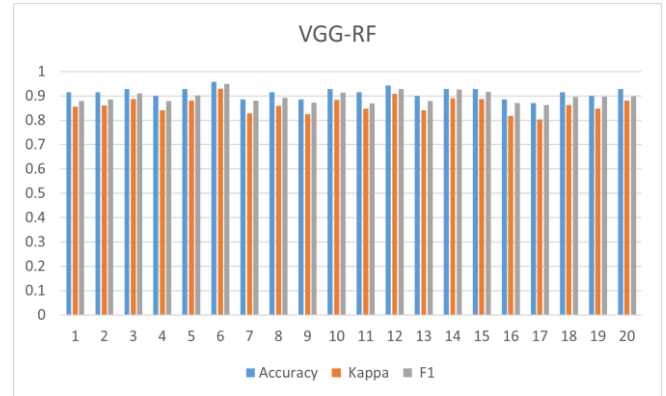


Fig. 9. VGG-RF test runs.

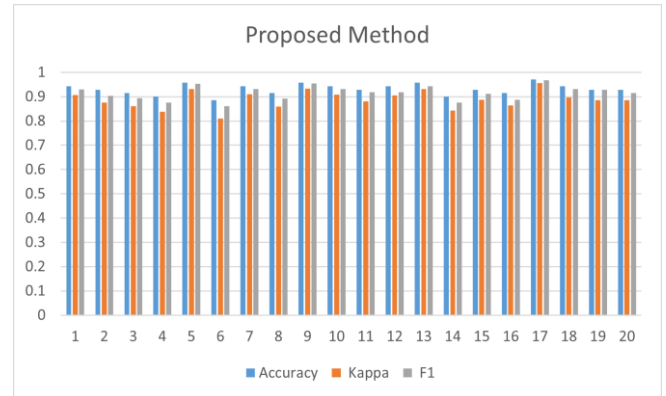


Fig. 10. Proposed method test runs.

TABLE II. F1-SCORE COMPARISON ON EIGHT DIFFERENT DATASETS.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.876	0.883	0.938	0.969	0.937	0.955	0.97	0.929	0.955	0.917
Heart Failure	0.739	0.566	0.51	0.733	0.771	0.49	0.418	0.766	0.639	0.592
Fertility	0	0.206	0	0	0	0	0	0	0	0.363
Parkinson's	0.876	0.75	0.895	0.893	0.884	0.912	0.886	0.884	0.891	0.876
Haberman's Survival	0.431	0.333	0	0	0.083	0.414	0.37	0.083	0.083	0.618
Breast Cancer Coimbra	0.585	0.526	0.75	0.68	0.692	0.413	0.638	0.68	0.666	0.634
Blood Transfusion Service Center	0.389	0.294	0	0.2	0.107	0.229	0.229	0.203	0.2	0.529
SPECTF Heart	0.822	0.732	0.887	0.887	0.898	0.88	0.536	0.88	0.859	0.871
Avg.	0.59	0.536	0.497	0.545	0.546	0.537	0.506	0.553	0.536	0.675

TABLE III. KAPPA COMPARISON ON EIGHT DIFFERENT DATASETS.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.833	0.874	0.942	0.93	0.872	0.875	0.894	0.883	0.92	0.874
Heart Failure	0.5	0.545	0.31	0.443	0.465	0.266	0.485	0.444	0.259	0.402
Fertility	-0.1	-0.047	0	0	0	-0.078	-0.078	0	-0.083	0.016
Parkinson's	0.292	0.319	0.440	0.292	0.202	0.360	0	0.292	0.265	0.232
Haberman's Survival	0.146	0.023	0.041	0	0	0.023	0.023	0.061	0	0.294
Breast Cancer Coimbra	0.434	0.34	0.172	-0.087	-0.194	0.231	0.328	-0.087	0.223	0.231
Blood Transfusion Service Center	0.045	0.151	0.192	0.217	0.144	0.150	0.156	0.144	0.144	0.331
SPECTF Heart	0.234	0.326	0	0.107	0.080	0	0	-0.021	0	0.283
Avg.	0.298	0.316	0.262	0.238	0.196	0.228	0.226	0.214	0.236	0.333

TABLE IV. ROC AUC COMPARISON ON EIGHT DIFFERENT DATASETS.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.912	0.872	0.934	0.925	0.92	0.952	0.958	0.91	0.952	0.934
Heart Failure	0.699	0.623	0.612	0.718	0.697	0.708	0.78	0.738	0.66	0.747
Fertility	0.416	0.5	0.5	0.5	0.5	0.433	0.316	0.5	0.483	0.583
Parkinson's	0.929	0.70	0.733	0.66	0.669	0.5	0.77	0.669	0.76	0.75
Haberman's Survival	0.551	0.563	0.493	0.5	0.538	0.538	0.538	0.531	0.538	0.587
Breast Cancer Coimbra	0.682	0.717	0.831	0.766	0.725	0.634	0.741	0.735	0.828	0.756
Blood Transfusion Service Center	0.560	0.591	0.565	0.581	0.541	0.544	0.614	0.542	0.541	0.614
SPECTF Heart	0.651	0.744	0.559	0.586	0.526	0.592	0.712	0.5	0.579	0.751
Avg.	0.675	0.664	0.653	0.654	0.639	0.613	0.679	0.641	0.668	0.715

TABLE V. DECISION TREE PARAMETERS.

Parameter name	Values
criterion	"gini", "entropy"
Splitter-*	"best", "random"
max_depth	None, 5, 10

TABLE VI. NAIVE BAYES PARAMETERS.

Parameter name	Values
var_smoothing	np.logspace(0, -9, num = 100)

TABLE VII. SVM PARAMETERS.

Parameter name	Values
kernel	"linear", "poly", "rbf", "sigmoid"
C	0.1, 1.0, 10.0

TABLE VIII. MLP PARAMETERS.

Parameter name	Values
activation	"identity", "logistic", "tanh", "relu"
hidden_layer_sizes	(50, 50, 50), (50, 100, 50), (100)

TABLE IX. LOGISTIC REGRESSION PARAMETERS.

Parameter name	Values
fit_intercept	True, False
C	0.001, 0.01, 0.1, 1, 10, 100

TABLE X. PASSIVE AGGRESSIVE CLASSIFIER PARAMETERS.

Parameter name	Values
fit_intercept	True, False
C	0.001, 0.01, 0.1, 1, 10, 100

TABLE XI. PERCEPTRON PARAMETERS.

Parameter name	Values
fit_intercept	True, False
alpha	0.001, 0.01, 0.1, 1, 10, 100

TABLE XII. RIDGE CLASSIFIER PARAMETERS.

Parameter name	Values
fit_intercept	True, False
alpha	0.001, 0.01, 0.1, 1, 10, 100

TABLE XIII. SGD CLASSIFIER PARAMETERS.

Parameter name	Values
fit_intercept	True, False
alpha	0.001, 0.01, 0.1, 1, 10, 100

TABLE XIV. PROPOSED METHOD PARAMETERS.

Parameter name	Values
n_clusters	2, 3, 5
iters	20, 30, 40

TABLE XV. F1-SCORE HYPERPARAMETER TUNING COMPARISON.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.913	0.884	0.942	0.925	0.942	0.934	0.944	0.932	0.94	0.937
Heart Failure	0.571	0.612	0.716	0.714	0.735	0.739	0.702	0.716	0.71	0.684
Fertility	0.0	0.285	0.5	0.25	0.25	0.25	0.25	0.0	0.25	0.4
Parkinson's	0.905	0.779	0.910	0.891	0.901	0.910	0.901	0.9	0.91	0.874
Haberman's Survival	0.26	0.374	0.26	0.0	0.266	0.428	0.205	0.266	0.279	0.514
Breast Cancer Coimbra	0.744	0.628	0.782	0.749	0.816	0.721	0.745	0.733	0.721	0.808
Blood Transfusion Service Center	0.452	0.321	0.386	0.437	0.363	0.035	0.426	0.228	0.126	0.456
SPECTF Heart	0.782	0.765	0.873	0.873	0.866	0.873	0.873	0.873	0.873	0.835
Avg.	0.578	0.581	0.671	0.605	0.642	0.611	0.631	0.581	0.601	0.689

TABLE XVI. KAPPA HYPERPARAMETER TUNING COMPARISON.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.803	0.879	0.976	0.0	0.952	0.976	0.928	0.928	0.976	0.861
Heart Failure	0.63	0.436	0.615	0.608	0.608	0.602	0.555	0.608	0.555	0.371
Fertility	0.063	0.0	0.063	-0.05	0.0	0.0	0.0	0.0	0.0	0.136
Parkinson's	0.48	0.277	0.615	0.621	0.672	0.673	0.362	0.711	0.322	0.621
Haberman's Survival	0.004	0.268	0.013	0.202	0.202	0.05	0.249	0.202	0.092	0.457
Breast Cancer Coimbra	0.023	0.063	0.233	0.131	0.231	0.231	0.221	0.178	0.088	0.39
Blood Transfusion Service Center	0.293	0.172	0.151	0.34	0.142	0.035	0.055	0.122	0.103	0.354
SPECTF Heart	0.233	0.31	0.4	0.0	0.491	0.098	0.0	0.22	0.417	0.189
Avg.	0.316	0.301	0.384	0.231	0.412	0.333	0.296	0.371	0.319	0.422

TABLE XVII. ROC AUC HYPERPARAMETER TUNING COMPARISON.

Dataset	DT	NB	SVC	MLP	LR	PAC	PCT	RC	SGD	PM
Breast Cancer	0.912	0.924	0.979	0.984	0.984	0.98	0.954	0.937	0.984	0.952
Heart Failure	0.704	0.59	0.696	0.666	0.674	0.727	0.734	0.666	0.712	0.734
Fertility	0.466	0.5	0.566	0.716	0.583	0.483	0.70	0.433	0.416	0.649
Parkinson's	0.673	0.817	0.865	0.798	0.5	0.778	0.5	0.673	0.5	0.778
Haberman's Survival	0.644	0.567	0.5	0.537	0.585	0.5	0.5	0.567	0.5	0.71
Breast Cancer Coimbra	0.634	0.653	0.75	0.673	0.807	0.576	0.576	0.73	0.673	0.634
Blood Transfusion Service Center	0.5	0.515	0.5	0.533	0.542	0.526	0.5	0.535	0.514	0.627
SPECTF Heart	0.519	0.69	0.561	0.568	0.568	0.492	0.816	0.5	0.5	0.589
Avg.	0.632	0.657	0.677	0.684	0.655	0.633	0.66	0.63	0.6	0.709

VI. DISCUSSION

In the eight datasets used, the proposed classifier outperformed the previous studies on average F1-score, Kappa score, and ROC AUC. However, this study has also some limitations. First of all, it is (its Computer Vision part is) restricted to only one dataset. On the other hand, it has been shown that the classifier has a certain generalisation power by its status in an experiment series conducted through eight different datasets. On average, F1-score, Kappa score, and ROC AUC take the lead.

One point which can be criticised is that the datasets are taken from a middle scale level and that the cardinality of training samples are limited to 1000. That is true, and for handling large scale datasets, the algorithm can be edited. Future work can contain theoretical/mathematical analysis of the classifier and a more extensive evaluation of it.

VII. CONCLUSIONS

In this work, we have proposed a new method merging the power of DL with a newly introduced MAD-based classifier. We have measured accuracy, macro F1-score, and Kappa coefficient values, and in all of these, our method outperforms both traditional ML algorithms and state-of-the-art.

Moreover, it also outperforms the manual diagnosis of professional doctors. Additionally, it has been shown that the proposed classifier has a certain generalisation power, which can be understood from the experiment series conducted on eight different datasets.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

- [1] Q. Shi *et al.*, "An automatic classification method on chronic venous insufficiency images", *Scientific Reports*, vol. 8, no. 1, p. 17952, 2018. DOI: 10.1038/s41598-018-36284-5.
- [2] N. Krishnan and P. Muthu, "CVINet: A deep learning based model for the diagnosis of chronic venous insufficiency in lower extremity using infrared thermal images", *International Journal of Imaging Systems and Technology*, vol. 34, no. 2, p. e23004, 2023. DOI: 10.1002/ima.23004.
- [3] M. R. Thanka, E. B. Edwin, R. P. Joy, S. J. Priya, and V. Ebenezer, "Varicose veins chronic venous diseases image classification using multidimensional convolutional neural networks", in *Proc. of 2022 6th International Conference on Devices, Circuits and Systems (ICDCS)*, 2022, pp. 364–368. DOI: 10.1109/ICDCS54290.2022.9780842.
- [4] R. R. Bhavani and G. W. Jiji, "Image registration for varicose ulcer classification using KNN classifier", *International Journal of Computers and Applications*, vol. 40, no. 2, pp. 88–97, 2018. DOI: 10.1080/1206212X.2017.1395108.

- [5] V. Rajathi, R. R. Bhavani, and G. W. Jiji, "Varicose ulcer (C6) wound image tissue classification using multidimensional convolutional neural networks", *The Imaging Science Journal*, vol. 67, no. 7, pp. 374–384, 2019. DOI: 10.1080/13682199.2019.1663083.
- [6] B. Meneses-Claudio, W. Alvarado-Diaz, and A. Roman-Gonzalez, "Detection of suspicions of varicose veins in the legs using thermal imaging", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 5, pp. 431–435, 2019. DOI: 10.14569/IJACSA.2019.0100554.
- [7] S. Dahlmanns, S. Reich-Schupke, F. Schollemann, M. Stücker, S. Leonhardt, and D. Teichmann, "Classification of chronic venous diseases based on skin temperature patterns", *Physiological Measurement*, vol. 42, no. 4, p. 045001, 2021. DOI: 10.1088/1361-6579/abf020.
- [8] S. Mirunalini, C. Jeyalakshmi, and P. Muralikrishnan, "Fuzzy c means based approach for analysis of varicose veins", *Journal of Pharmaceutical Negative Results*, vol. 13, no. 6, pp. 2288–2295, 2022. DOI: 10.47750/pnr.2022.13.S06.298.
- [9] B. Fielding and L. Zhang, "Evolving image classification architectures with enhanced particle swarm optimisation", *IEEE Access*, vol. 6, pp. 68560–68575, 2018. DOI: 10.1109/ACCESS.2018.2880416.
- [10] T. Lawrence *et al.*, "Particle swarm optimization for automatically evolving convolutional neural networks for image classification", *IEEE Access*, vol. 9, pp. 14369–14386, 2021. DOI: 10.1109/ACCESS.2021.3052489.
- [11] Q. Niu and X. Huang, "An improved fuzzy c-means clustering algorithm based on PSO", *Journal of Software*, vol. 6, no. 5, pp. 873–879, 2011. DOI: 10.4304/jsw.6.5.873-879.
- [12] H. Ichihashi, K. Honda, A. Notsu, and K. Ohta, "Fuzzy c-means classifier with particle swarm optimization", in *Proc. of 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, 2008, pp. 207–215. DOI: 10.1109/FUZZY.2008.4630367.
- [13] Pravin Kumar S. K., M. G. Sumithra, and N. Saranya, "Particle Swarm Optimization (PSO) with fuzzy c means (PSO-FCM)-based segmentation and machine learning classifier for leaf diseases prediction", *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, p. e5312, 2021. DOI: 10.1002/cpe.5312.
- [14] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm", *Computers & Geosciences*, vol. 10, nos. 2–3, pp. 191–203, 1984. DOI: 10.1016/0098-3004(84)90020-7.
- [15] A. Tam, "A Gentle Introduction to Particle Swarm Optimization", *Machine Learning Mastery*, October 12, 2021. [Online]. Available: <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/>
- [16] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview", *Soft Computing*, vol. 22, pp. 387–408, 2018. DOI: 10.1007/s00500-016-2474-6.
- [17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", 2014. arXiv: 1409.1556.
- [18] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python", *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) license (<http://creativecommons.org/licenses/by/4.0/>).