Diabetic Foot Ulcers Classification using a fine-tuned CNNs Ensemble

Elineide Santos*, Francisco Santos*, João Manuel Tavares†
Andrea Bianchi‡, and Rodrigo Veras*
*Universidade Federal do Piauí, Teresina, Brasil
†Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
‡ Universidade Federal de Ouro Preto, Minas Gerais, Brasil
Email: {elineide.silva, fsantos, rveras}@ufpi.edu.br,
tavares@fe.up.pt, andrea@ufop.edu.br

***Abstract*—Diabetic Foot Ulcers (DFU) are lesions in the foot region caused by diabetes mellitus. It is essential to define the appropriate treatment in the early stages of the disease once late treatment may result in amputation. This article proposes an ensemble approach composed of five modified traditional convolutional neural networks (CNNs) - VGG-16, VGG-19, Resnet-50, InceptionV3, and Densenet-201 - to classify DFU images. To define the parameters, we fine-tuned the CNNs, evaluated different configurations of fully connected layers, and used batch normalization and dropout operations. The modified CNNs were well suited to the problem; however, we observed that the union of the five CNNs significantly increased the success rates. We performed tests using 8,250 images with different resolution, contrast, color, and texture characteristics and included data augmentation operations to expand the training dataset. A 5-fold cross-validation led to an average accuracy of 95.04%, resulting in a Kappa index greater than 91.85%, considered "Excellent".**

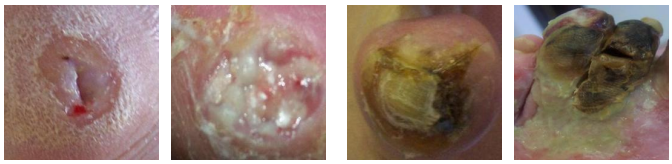
***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Diabetes Mellitus is a chronic disease characterized by high blood glucose levels. An estimated 536.6 million people will live with diabetes (diagnosed or undiagnosed) by 2021 [1]. This number is expected to increase by 46% to reach 783.2 million in 2045. This disease can cause many complications such as blindness, cardiovascular disease, kidney failure, and diabetic foot ulcers [2].

Ulcers result in wounds in the foot region, usually caused by trauma, repetitive mechanical stress, or continuously applied mechanical stress [3]. Diabetic foot ulcers (DFU) need proper treatment, as they can lead to the amputation of infected limbs in an advanced stage. Thus, an early diagnosis can delay the development of the disease and prevent adverse scenarios.

Severe injuries can be classified as infection or ischemia. Infection, as shown in Figure 1b, is recognized by the presence of inflammation or purulence, as well as increased redness around the ulcer. On the other hand, ischemia, Figure 1c, is the inadequate circulation of blood through the lesion, being visually identified by the presence of poor reperfusion in the gangrened foot or toes. In some cases, as in Figure 1d, the ulcer has both ischemia and infection. However, after treatment, the ulcers reach a healing state and resemble healthy skin, as shown in Figure 1a.



(a) Ulcer in Healing (b) Infection (c) Ischemia (d) Infection and Ischemia

Fig. 1. Examples of diabetic foot ulcer images.

Monitoring diabetic foot injuries is usually done by visually inspecting the injured areas and observing the signs and symptoms of diabetes [4]. Thus, the assessment relies on the specialist's subjective criteria. In this context, using a diagnostic assistance system can support the specialist and enable automatic monitoring of injuries.

Recent works have proposed automatic methodologies focusing on using convolutional neural networks (CNNs). However, the authors employed distinct neural networks. Therefore, this work proposes a neural network for classifying diabetic foot injuries. To do this, we evaluated and refined the architecture of seven general-purpose CNNs: VGG-16, VGG-19, InceptionV3, ResNet50, DenseNet201, MobileNet, and EfficientNet.

The seven CNNs were analyzed in five different scenarios: (1) their original architectures, (2) with changing the dense layers, (3) with the addition of dropout layers (DP), (4) with the addition of batch normalization layers (BN), and (5) with the addition of dropout and batch normalization.

In the evaluation, two public datasets with a total of 8,250 images belonging to four classes: none (healthy skin, ulcers in the healing process and ulcers without ischemia or infection), ischemia (ulcers with ischemia only), infection (ulcer only with infection) and both (ulcers with ischemia and infection), were used.

Cnns have achieved an excellent performance individually in the classification of images with distinct characteristics (

Fundação de Amparo à Pesquisa do Estado do Piauí (FAPEPI) and Universidade Federal do Piauí (UFPI)

[5]–[7]). In view of the importance of early diagnosis in the identification of ulcer images, in this work we propose a model composed of a set of Cnns to detect the presence of ulcers in the images and identify their class, thus provide an appropriate and efficient treatment. The set is formed by the modified architectures: VGG-19, VGG-16, Resnet-50, InceptionV3 and Densenet-201.

This article is organized as follows: in Section II recent works and methodologies on the problem under study are presented. Section III presents the proposed method, the image datasets, the applied techniques, and the evaluation metrics adopted in the development work. In Sections IV, the results and their discussion are presented. Finally, conclusions and future work are presented in Section V.

II. Related Works

Several methodologies have been developed in recent years with the aim of offering automated and accurate solutions for the diagnosis of diabetic foot pathologies. The main methodologies evaluated are explained below.

In the work of [8], the authors proposed a methodology of automatic tissue identification using a technique that combines neural networks and Bayesian classifiers. To obtain color and texture patterns, we initially performed the segmentation of 113 images using a region growth technique. However, this procedure is not trivial, since ulcer images present a wide variety of shape, color and texture. Then the characteristics were extracted and then the supervised neural networks were trained to differentiate necrotic tissues from other tissues. The output of this system is used by Bayesian classifiers to classify tissues into five types of tissue: skin, healing, granulation, desquamation, and necrosis. This approach obtained an accuracy of 91.50% in its results.

In [9], the authors proposed a methodology that segmented the injured region in foot ulcer images captured by a standardized box using the Support Vector Machine (SVM) classifier. This box has controlled lighting and distance. The 100 images from the Injuries Center of the University of Massachusetts School of Medicine (Umass) are segmented using the superpixel generation technique called Simple Linear Iterative Clustering (SLIC) [10]. After the segmentation stage, color and texture characteristics that are used in the classifier training are extracted. Therefore, the proposed methodology consists of two stages for the classification, the first stage being composed of a set of binary SVM that perform different tests in the image sets, where the results are collected and used in the next stage. In the second stage, only superpixels classified as injury are classified using a new binary SVM. The resulting images are processed by applying morphological operations followed by detection of connected regions and a reclassification method based on the Conditional Random Field (CRF) [11] for incorrectly marked non-sore regions and fill unmarked wound regions. This method obtained a sensitivity of 73.30% and a specificity of 94.60%.

[12] binary classification: healthy and ulcer. Used the cross validation technique 10-fold. 1,679 labeled images. Proposed a network called Dfunet. Dfunet combines two types of convolutional layers, i.e., traditional convolution layers at the beginning of the network that use a single convolutional filter followed by parallel convolutional layers, that use several convolutional layers to extract various resources from the same input. Dfunet is divided into three main sections: the boot layers inspired by Googlenet, parallel convolution layers to discriminate the DFU more effectively than the previous network layers, and finally, both layers fully connected and an output sorter based on softmax. Used natural data increase. Our proposed Dfunet has higher performance measurements in Sensitivity, with a score of 0.934, F-Measure with 0.939 and AUC with 0.962, Accuracy of 0.92.

[13] proposed a deep convolutional network called DFU-Qutnet, which classifies images in normal or abnormal skin with low computational cost. This architecture is able to increase the width of the network without the need to enlarge its depth, in this way, occurs the increase of the learning of the data to carry out the classification. The authors evaluated two scenarios, the first consists in the use of DFU-Qutnet as a classifier, while the second consists of the extraction of the characteristics using the pre-trained DFU-Qutnet and the SVM and K-Nearest Neighbors (KNN) classifiers in the prediction stage. Using DFU-Qutnet+SVM, accuracy was 95.40%, recall 93.60% and F1-score 94.50% in the classification of 754 foot images of patients with diabetic foot ulcer and healthy skin from the Nasiriyah Hospital Diabetic Center in southern Iraq. For comparison purposes, the pre-trained networks Googlenet, VGG-16 and Alexnet were adjusted and re-trained for this task, however, these networks did not surpass the proposed network.

[14] performed the classification of images in $Ischemia \times Nonischemia$ and $Infection \times Non-infection$. Initially, the adopted methodology performs a natural increase of data with the purpose of improving the performance of the injury identification algorithm, since diabetic foot images occupy a very small region in relation to the total. After defining the region of interest, color information (RGB and CIELAB) and texture (Local Binary Pattern - LBP, Histogram of Oriented Gradien - HOG) were extracted from the 1,459 images adopted. Also, superpixels were generated applying the SLIC [10] technique to segment the images and facilitate the extraction of characteristics. These images are from the feet of patients at Lancashire Teaching Hospitals. Another detail was the Ensemble network model, which combines the neural networks Inceptionv3, Resnet50 and Inceptionresnetv2 with the SVM classifier [9] to make the predictions. The results obtained by this approach were 90.00% accuracy in the classification of images in ischemia and 73.00% in the classification in images of infection for the adopted image base.

[15] proposed an architecture that classifies and locates different types of DFU images as ischemia (normal and ischemic images) and infection (normal and infection images). First, classification is performed using a neural network with 16 convolutional layers in combinations with the classifiers: Naive

Bayes, KNN, Softmax, Essemble and Decision Tree. After this classification, the input images are passed to the Yolov2-DFU model which is designed by the Yolov2 combination and a 172-layer random network for the location of the abnormal region. Images of patients from Lancashire Teaching Hospitals were used. The database contains 15,762 images, including images resulting from natural data enhancement. For the classification of $ischemia \times noischemia$, the best accuracy of 97.90% was with the Naive Bayes classifier and for $infection \times noinfection$, 99.60% with decision tree.

[16], for the binary classification of diabetic foot ulcers and normal skin in 1,679 patches of images obtained from Lancashire Teaching Hospitals, proposed a convolutional network called DFU-Spnet. This network consists of three stacked parallel convolution layers of kernel size $1 \times 1, 3 \times 3 and 5 \times 5$. The previous layer of the first stacked parallel block is a single convolution layer of 7×7 followed by a transition layer. The transition layer consists of batch normalization and activation of Leakyrelu and helps standardize inputs to accelerate training. The concatenated output of each stacked parallel convolution block is passed by a sequence of 1×1 convolution, transition and Maxpooling layer 2×2. Finally, the output of stacked parallel convolution blocks passes through a flatten layer and a fully connected 32-unit layer and then a dropout layer. The output layer consists of a sigmoid activation function to obtain a binary predicted value of class 0 (Normal) or 1 (Abnormal). DFU-Spnet using the SGD optimizer obtained 96.40% accuracy, F1-Score 95.40% and AUC 0.974.

[17] performed binary image classification and classification into three classes: diabetic ulcers, venous and surgical ulcers. Two image classifiers were used: one being an Alexnet architecture pre-trained with adjusted weights using the data set itself and the other, a classifier who applies the sliding window technique on the entrance wound image to extract nine sub-regions of equal size together with the classification step of patch to predict the type of wound. For each input image, each classifier generates rating scores for all classes. The type of wound for the entire image is then predicted by the majority vote on the prediction label of the sub-regions that were detected as wounds. Then, the characteristic vector feeds a four-layer Perceptron Multilayer (MLP) classifier with two hidden layers that have eight and seven neurons, respectively. The number of nodes in the input and output layers is determined based on the type of classification problem. The output of the MLP classifier is the wound type of the incoming image. A set of 400 images collected during two years at the Advancing the Zenith of Healthcare Wound and Vascular Center (AZH), Milwaukee, Wisconsin, United States, were used. 5-fold cross-validation was used, with a maximum accuracy of 91.90% and a mean of 87.70% for problems of classification of 3 classes.

[18] performed the classification of DFU images of the 2021 composite challenge with 15,683 images, classified into four classes: no infection and no ischemia, presence of ischemia, presence of infection and presence of infection and ischemia in the same ulcer. Cnns and Vision Transformers (Vision Transformers - Vit) were used, a powerful architecture for natural language processing applications. In general, Vit consider images as sequences of small patches similar to words or tokens, so there is no sense of distance within an image. Four architectures were analyzed: Big Image Transfer (Bit) to Resnext50, Efficientnet, Vit and Data Efficient Image Transformers (Data-Efficient Image Transformers - Deit), a refinement of Vit with better pre-training strategies. The best architecture was the Bit-Resnext50, however, the winning solution of the DFUC 2021 challenge was a linear combination of the forecasts extracted from the Bit-Resnext50 and Efficientnet B3 that achieved 62.16% F1-score, 88.55% AUC, 65.22% Recall and 61.40% accuracy.

In the work [19] the authors evaluated a series of networks for the detection of DFU. All the methods evaluated showed promise in the identification of these images. However, in many cases the nets had difficulty distinguishing the ulcers from other regions of the skin. In addition, they analyzed the combination of these networks, the accuracy of 86.58% achieved was higher when compared to the individual networks.

[20] performed the binary classification of DFU images in the following scenarios: $normal \times abnormal$, $ischemia \times noischemia$ and $infection \times noinfection$. During the study, we investigated the benefits of using LBP codes (Local Binary Patterns - LBP) as inputs to CNN models in the DFU classification, and designed a CNN architecture with three different inputs: DFU-RGB-Net using the original RGB images; DFU-TEX-Net using LBP-mapped texture-encoding and DFU-RGBTEX-Net images using LBP-mapped texture-encoding and RGB images. The process steps are: (i) extract the texture features using the basic LBP method, and then convert the extracted LBP codes to a 3D space to make the appropriate LBP codes as CNN input, and (ii) train various CNN models in RGB images and separately mapped LBP codes and texture features trained only in DFU classification compared to CNN models trained in RGB images only. It was proposed a CNN with only four convolution layers and two fully connected layers without infill. Each convolution layer consists of a rectifier linear unit (Relu) activation function, followed by a cluster layer. The optimal size of the proposed CNN architecture is found empirically, where the convolution and max-pooling number is increased gradually; subsequently, the number of filters is adjusted gradually and then the best performing network is chosen. The base DFU 2021 was used, being 16,790 images for $normal \times abnormal$ classification including an increase of 10 times and for classification $ischemia \times nonischemia$ 9,870 images and for $infection \times non - infection$ 5,892 images. The results showed that the proposed DFU-RGBTEX-Net performed better than CNN-based methods, with 94.10% accuracy and 98.10% AUC for $normal \times abnormal$ classification, 99.00% accuracy and 99.50% AUC for ischemia classification and 74.20% accuracy and 82.00% AUC for infection classification.

Table I summarizes the found works in terms of year of publication, used classification technique(s), number of im-

ages, number of studied classes, and the performance achieved, which can be as to accuracy (A), sensitivity (S), specificity (E), precision (P) and area under the curve (AUC).

TABLE I
SUMMARY OF THE IDENTIFIED STATE-OF-THE-ART WORKS.

Work	Classification technique(s)	N. of images	N. of classes	Performance(%)
[8]	Neural Networks Bayesian Classifiers	113	5	A: 91.50
[9]	SVM	100	2	S: 73.30 S: 94.60
[12]	DFUNet	1,423	2	A: 92.50
[13]	DFU-QUTNet SVM KNN	754	2	P: 95.40
[14]	InceptionV3 ResNet50 InceptionResNetV2 SVM	1,459	2	A Isc: 90.00 A Inf: 73.00
[15]	Neural Networks Naive bayes Neural Networks Decision tree	15,762	2	A Isc: 97.90 A Inf: 99.60
[16]	DFU_SPNet	1,679	2	A: 96.40
[17]	AlexNet Sliding window MLP	400	3	A max: 91.90 A average: 87.70
[18]	BiT-ResNeXt50	15,683	4	AUC: 88.49 P: 60.53
[19]	YOLOv3 YOLOv5 EfficientDet	2,000	–	P: 86.58
[20]	DFU-RGB-TEX-Net	5,892	2	A: 94.00 A Isq: 99.00 A Inf: 74.20

Analyzing the works indicated in Table I, one can realize that the approaches, in the majority, combine neural networks with data augmentation techniques. Furthermore, there is no standard regarding the number of images, the number of classes, or the choice of evaluation metrics.

III. MATERIALS AND METHODS

This section presents a solution capable of differentiating four patterns of diabetic foot ulcers. We refined seven CNN architectures, evaluated different combinations of fully connected layers and the use of dropout and batch normalization operations. In the following, the proposed solution and the involved techniques, the metrics adopted to assess the solution, and the used image datasets, are described.

A. Proposed Method

Analyzing the achieved results with each CNNs and with the ensembles, we reached the proposed approach shown in Figure 2. The input image goes through a applied pre-processing techniques to size adjustment in the images. The image is classified by five CNNs adapted and trained to perform the classification task. A weighted majority voting ensemble processed the five predicted results, producing the final classification result.

B. Image dataset

In the experiments, we used the public image dataset of diabetic foot ulcers named Diabetic Foot Ulcer (DFU). In the experiments, we used the public image dataset of diabetic foot

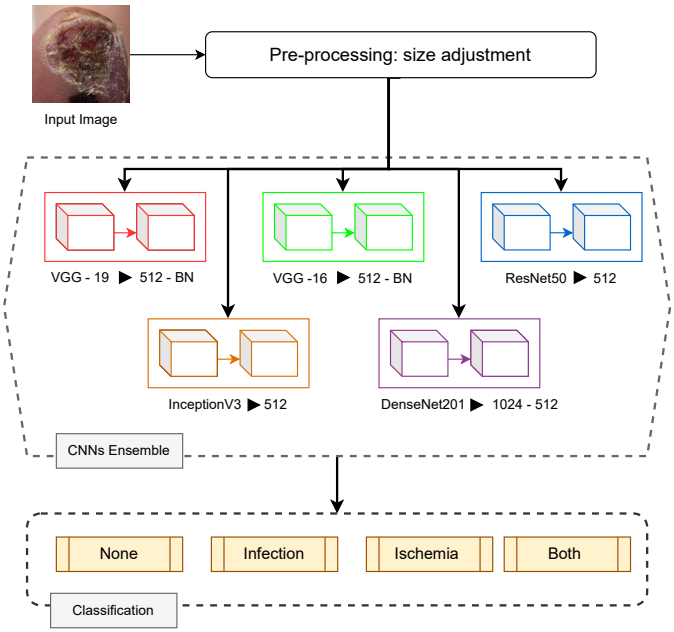
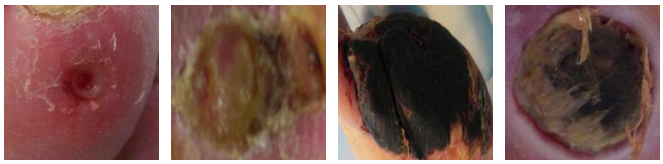


Fig. 2. Proposed methodology: The image input is resized, after which it goes to five models that form an ensemble of CNNs, then, for final classification.

ulcers named Diabetic Foot Ulcer (DFU) in two versions: 2020 and 2021. The images of patients' feet with DFU at Lancashire Teaching Hospitals were captured during five years with three cameras: Kodak DX4530, Nikon D3300, and Nikon COOLPIX P100, after debridement (removal of necrotic and devitalized tissue). The image diagnosis (ground truth) was developed with the help of two specialist physicians. When there was disagreement between these professionals, the most experienced physician decided. The images were captured centered on the lesion area and had different dimensions; the smallest has 34×31 pixels, while the largest has 1103×1127 pixels.

The images were splited into four classes: (1) none, containing images of healthy skin, ulcers in the process of recovery, and ulcers without infection or ischemia (Figure 3a); (2) infection, which contains images of ulcers with infection only (Figure 3b); (3) ischemia, containing images of ulcers with ischemia only (Figure 3c), and (4) both, with ulcers images containing infection and ischemia at the same time (Figure 3d).



(a) None (b) Infection (c) Ischemia (d) Both

Fig. 3. Examples of images of the four classes under study.

Table II indicates the number of images per class of the used DFU 2020 and DFU 2021 datasets.

TABLE II
NUMBER OF IMAGES, PER CLASS, OF DFU 2020 AND DFU 2021 DATASETS.

Class	DFU 2020	DFU 2021	Total
None	1,281	2,552	3,833
Ischemia	26	227	253
Infection	779	2,555	3,334
Both	209	621	830
Total	**2,295**	**5,955**	**8,250**

C. Data augmentation

Usually, CNNs have millions of parameters and need a large amount of data to be trained. Even to refine a small CNN, thousands of images are required. The state-of-the-art methods applied data augmentation techniques to overcome the latter requirement. Data augmentation consists of creating a new set of images using variations of the original images. The increase in data has the main goals of reducing the CNN overfitting and improving the generalization of the trained model [21].

Hence, we used the random data augmentation technique provided by the Keras API. The chosen rotation interval was 40º, while the vertical, horizontal, shear, and zoom translation interval was equal to 0.2. We also used horizontal and vertical flip. The reflection fill technique was applied to replace black pixels resulting from the rotation and translation techniques. Finally, we normalized the image pixels to 0 (zero) and 1 (one). The augmentation resulted in an image dataset 20 times larger than the original one.

D. Evaluated Architectures

We evaluated CNN architectures designed for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [22]. According to Kornblith et al. [23], the better the architecture performs on the ImageNet dataset, the better the transfer to other natural image datasets. Furthermore, another determining factor for the selection of state-of-the-art architectures was the performance obtained in works in the literature, as in Vogado et al. [24]. The evaluated architectures are indicated in Table III, being referred in terms of topological depth of the network, number of parameters, and year of publication.

TABLE III
CHARACTERISTICS OF THE EVALUATED DEEP LEARNING MODELS.

CNN	Topological depth	Number of parameters	Year
VGG-16	23	138,357,544	2014
VGG-19	26	143,667,240	2014
ResNet50	168	25,636,712	2015
InceptionV3	159	23,851,784	2016
DenseNet201	201	20,242,984	2017
MobileNetV2	88	3,538,984	2018
EfficientNetB0	240	5,330,571	2019

The VGG-16 and VGG-19 models are versions of the VGGNet network proposed by [25]. VGG-16 has 16 trainable layers, with 13 convolutional layers divided into five blocks and three fully connected layers. There is a MaxPooling layer between two convolutional blocks, two fully connected layers with 4,096 units, and an output layer with the softmax activation function. VGG-19 has 19 trainable layers, being three more convolutional layers than VGG-16. The VGG-16 and VGG-19 networks have 138 and 143 million parameters, respectively.

The Residual Neural Network proposed by [26] was developed to solve the problem of gradient vanishing that occurs with the addition of many layers in a sequential model. The ResNet architecture is formed by residual blocks that skip the connections between the input of the block itself and the output. Residual maps are easier to optimize and, consequently, avoid the degradation caused by many layers. ResNet was defined with five architectures with different depths: 18, 34, 50, 101, and 152 trainable layers. In this work, ResNet50 was evaluated, which has a topology of 168 layers, being trainable only 50 of which.

InceptionV3 was proposed by [27] as a successor to the GoogLenet and InceptionV2 architectures. With topological depth of 159 layers and 24 million parameters, InceptionV3 has symmetric and asymmetric blocks, convolutional layers, average and MaxPolling, feature concatenation, dropouts, and dense layers. The ability to factor convolutions is one of the main features of InceptionV3. Its primary purpose is to reduce the number of parameters and reduce the cost associated with these operations. Even so, InceptionV3 has a higher computational cost than GoogLenet. This cost is justified by the performance obtained by this architecture, which is more efficient than its predecessors.

The DenseNet network was proposed by [28]. In this model, each layer receives all previous layers as input, while its output is fed to all later layers. Dense connections in DenseNet alleviate the gradient vanishing and exploding problems and make it easy to reuse resources. DenseNet has different versions, depending on the number of layers in the neural network. In this work, DenseNet201 was evaluated, which has a topological depth of 201 layers and about 20 million parameters.

MobileNetV2 [29] was designed for use on mobile devices. This architecture uses an inverted residual structure where the residual connections are in the bottleneck layers. Filtering features in the middle expansion layer uses light depth convolutions.

EfficientNetB0 [30] is a CNN and scaling method that uniformly defines depth, width, and resolution, using a fixed scaling coefficient. It belongs to the EfficientNet family of architectures resulting from studying the model's sizing. It was observed that the balance of depth, width, and resolution of the network could substantially improve its performance.

E. Transfer Learning

The transfer learning technique that is often employed for convolutional networks uses weights that are pre-trained in large datasets, such as the ImageNet Challenge dataset [22]. This procedure decreases the requirement to retrain all parameters of the CNN from scratch [31].

Two approaches are often employed when using pre-trained weights. One approach is to extract features as the activation maps of the pre-trained network layers, defining those as feature vectors to be used as input to shallow classifiers. The other one is to perform fine-tuning by creating a new classification layer. This approach has a higher computational cost than the first one, since it must resume the CNN training with the target dataset, adapting the desired model domain.

According to Izadyyazdanabadi et al. [32], there are two types of fine-tuning: shallow fine-tuning (SFT) and deeply fine-tuning (DFT). SFT consists of freezing layers from the beginning of CNN; usually, the first convolutional layers, which are considered more general and allow representations of shape, texture and color. The top layers are often domain-specific, carrying semantic content from the instance labels. Therefore, SFT provides greater specialization in the later layers while keeping the first ones.

We opted, however, for the DFT. The DFT approach allows training the entire network, adapting even the first layers. Although it has a higher computational cost and requires a larger amount of data, it can benefit applications where the target domain differs from the one used to pre-train the weights. For example, natural photographic images from the ImageNet dataset belong to a distinct domain relative to diabetic foot ulcer images.

F. Dropout and batch normalization

Overfitting and long training time are two fundamental challenges in CNNs. Dropout and batch normalization are two well-recognized strategies to tackle these challenges.

The dropout [33] is a regularization technique used in neural network training. Its main feature is to disable, temporarily, some neurons. This effect provides the equivalent of different training architectures since different neurons will be disabled during the training (in other iterations). The use of dropout tends to reduce CNN complexity and overfitting.

The time for a network to converge depends on initializing the hyperparameters and using small learning rates. Also, a layer depends on previous layers, so small changes in one layer can be amplified as they flow to the subsequent layers. Batch normalization [34] normalizes the input of each layer of the network. Thus, the training time can be reduced as it allows the use of higher learning rates.

G. Ensemble

The ensemble is a technique that existed long before the Deep Learning paradigm emerged [35]. The theory behind this is quite simple and is based on the well-known notion of "wisdom of crowds": instead of relying on just one model for prediction, a set of multiple (pre-trained) models is created. These models' results are then combined in a final classification by constructing some majority votes. The original idea was developed to reduce the classifiers' variance to obtain a better overall performance [36].

We can find in the literature other works that successfully used ensembles in CAD systems [37], [38].

In general, the production of an ensemble involves three main phases: generating base classifiers, selecting ensemble members, and defining the decision mechanism [39]. In addition to analyzing the CNNs separately, we created ensembles by majority voting formed by the best classifiers.

H. Evaluation Metrics

We used the confusion matrix values to evaluate the CNNs. This matrix provides four values: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Based on these values we calculated, the metrics Accuracy (A), Precision (P), Recall (R), F1-score (F) (Equations 1, 2, 3:

$$A = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$P = \frac{TP}{TP + FP}, \tag{2}$$

$$R = \frac{TP}{TP + FN}, \tag{3}$$

$$F = 2 * \frac{P * R}{P + R}. \tag{4}$$

We also computed the kappa index (K), which is frequently recommended as an appropriate exactitude measure, as it can adequately represent the confusion matrix; it takes all elements of the matrix into account, rather than just those on the main diagonal, which occurs when calculating the global classification accuracy. This metric can be calculated as:

$$\mathrm{k} = \frac{observed - expected}{1 - expected}. \tag{5}$$

According to Landis and Koch [40], k assumes values between 0 (zero) and 1 (one). The result is qualified according to the k value as follows: $\mathrm{k} \leq 0.2$: Bad; $0.2 < \mathrm{k} \leq 0.4$: Fair; $0.4 < \mathrm{k} \leq 0.6$: Good; $0.6 < \mathrm{k} \leq 0.8$: Very Good and $\mathrm{k} > 0.8$: Excellent.

The CNNs in their original settings were fine-tuned with input images with 224×224 pixels. Then, we perform an ablation process, changing the fully connected layers of these networks and performing the DFT fine-tuning. Finally, we made changes to the internal architecture of the VGGs, adding layers of dropout, batch normalization, and both. These configurations used input images of 112×112 pixels as input. The training of networks in all configurations was performed with 500 epochs.

We applied the stratified *k*-fold cross-validation technique ($k = 5$), which consists of randomly distributing the dataset instances into *k* mutually exclusive subsets (folds) of approximately equal size, and in the same proportion observed in the original dataset. The CNN is fine-tuned and tested *k* times, and in each round, a different subset is used for testing, and the remaining *k*–1 subsets are used for fine-tuning. A confusion matrix was computed for each fold, and the arithmetic average and standard deviation of the five values achieved from each

studied CNN was taken into account. In addition, K was multiplied by 100 to facilitate the understanding of the tables.

IV. Results and Discussion

A. Individual classifiers

In Section, we show the individual results obtained with the 7 evaluated CNN architectures. The CNNs in their original settings were fine-tuned with input images with 224×224 pixels. Then, we perform an ablation process, changing the fully connected layers of these networks and performing the DFT fine-tuning. Finally, we made changes to the internal architecture of the VGGs, adding layers of dropout, batch normalization, and both. These configurations used input images of 112×112 pixels as input. The training of networks in all configurations was performed with 500 epochs.

We applied the stratified *k*-fold cross-validation technique ($k = 5$), which consists of randomly distributing the dataset instances into *k* mutually exclusive subsets (folds) of approximately equal size, and in the same proportion observed in the original dataset. The CNN is fine-tuned and tested *k* times, and in each round, a different subset is used for testing, and the remaining *k*–1 subsets are used for fine-tuning. A confusion matrix was computed for each fold, and the arithmetic average and standard deviation of the five values achieved from each studied CNN was taken into account. In addition, K was multiplied by 100 to facilitate the understanding of the tables.

1) Results using the original architectures: Initially, we fine-tuned the VGG-16, VGG-19, InceptionV3, ResNet50, DenseNet201, MobileNetV2 and EfficientNetB0 networks in their original configurations.

The results presented in Table IV show that most CNNs in their original configurations achieved Kappa indices between 40% and 60%, that is, considered good. Only InceptionV3 and MobileNetV2 achieved a moderate Kappa index. Overall, DenseNet201 achieved the best results in all used metrics.

TABLE IV
Classification results obtained with the original CNNs (best values in bold).

CNN	A(%)	P(%)	R(%)	F(%)	K(%)
VGG-16	71.95±2.51	63.44±2.68	71.95±2.51	67.23±2.51	50.42±4.60
VGG-19	73.17±0.92	65.05±0.64	73.17±0.92	68.52±0.82	52.71±1.58
ResNet50	74.23±0.59	65.39±0.56	74.23±0.59	69.36±0.54	54.47±1.03
InceptionV3	57.66±6.19	60.95±7.20	57.66±6.19	50.10±8.96	23.27±12.58
DenseNet201	**75.06**±0.71	**66.58**±0.26	**75.06**±0.71	**70.24**±0.58	**56.03**±1.10
MobileNetV2	61.53±0.07	53.87±0.06	61.53±0.07	55.19±0.11	30.87±14.90
EfficientNetB0	46.56±0.10	21.68±0.10	46.56±0.10	29.59±0.10	0±0

2) Results changing FC layers: In this experiment, we preserved the CNNs convolutional layers, inserted a Global Average Pooling layer and then the fully connected layers in two scenarios: (1) a connected layer with the number of neurons assuming the following values: 256, 512, and 1024, and (2) two fully connected layers with configurations of 512-256, 1024-256 and 1024-512 neurons. These configurations led to fewer neurons than original CNNs and, consequently, to a smaller number of weights to be trained.

The results obtained with the best configuration for each CNN are indicated in Table V. When comparing these results with the ones in Table IV, one can notice a significant improvement in performance metrics in all CNNs. In particular, VGG-19 with a dense layer of 512 neurons obtained the best results and a slight standard deviation, indicating CNNs stability in classifying diabetic foot ulcers.

TABLE V
Better results, for each CNN, obtained after changing the fully connected layers (best values in bold).

CNN	A(%)	P(%)	R(%)	F(%)	K(%)
VGG-16 512	90.55±0.44	90.60±0.42	90.55±0.44	90.54±0.45	84.47±0.72
VGG-19 512	**91.40**±0.35	**91.45**±0.37	**91.40** ±0.35	**91.40**±0.37	**85.88**±0.59
ResNet50 512	90.24±0.75	90.37±0.70	90.24±0.75	90.25±0.75	83.98±1.22
InceptionV3 512	81.60±1.03	81.97±0.88	81.60±1.03	81.54±1.04	69.56±1.72
DenseNet201 1024-512	90.04±0.64	90.24±0.70	90.04±0.64	90.04±0.65	83.63±1.04
MobileNetV2 512-256	81.43±0.01	81.51±0.01	81.43±0.01	81.33±0.01	69.34±1.25
EfficientNetB0 512	43.12±3.40	22.56±10.00	43.12±3.40	28.79±8.00	16.71±3.34

3) Results with dropout insertion: After each block of dense layers, we inserted a dropout layer in the VGG-16 and VGG-19 CNNs models. We chose to insert only in the VGGs since EfficienteNetB0 already has these layers, while the other networks have many blocks, which requires a study to define where the dropout layers would be included. Although the Kappa values could be considered "excellent", and there has been a gain in training time, the values in Table VI show that the results obtained were lower than the ones obtained without dropout (Table V).

TABLE VI
Classification results obtained with the addition of a dropout layer (best values in bold).

CNN	A(%)	P(%)	R(%)	F(%)	K(%)
VGG-16 + DP	87.89±0.66	87.98±0.60	87.89±0.66	87.89±0.66	80.09±1.15
VGG-19 + DP	**88.86**±0.40	**89.00**±0.39	**88.86**±0.40	**88.87**±0.41	**81.70**±0.65

4) Results with batch normalization insertion: We added batch normalization layers in the VGG-16 and VGG-19 in each convolutional layer block and before the MaxPooling layer. The other CNNs have normalization layers in their original architecture.

As indicated in Table VII, the VGG-16 and VGG-19 networks, with the addition of the normalization layer, obtained accuracy, precision, recall and F1-Score values above 93.00% and Kappa index larger than 89%, which is considered excellent. The normalization layer in each block of VGG-16 and VGG-19 models generated a significant increase in the classification success rate of both CNNs, which are the best results found relatively to the others obtained in this work.

TABLE VII
Classification results obtained with the batch normalization insertion (best values in bold).

CNN	A(%)	P(%)	R(%)	F(%)	K(%)
VGG-16 + BN	93.44±0.26	93.46±0.25	93.44±0.26	93.43±0.26	89.21±0.43
VGG-19 + BN	**93.45**±0.34	**93.56**±0.30	**93.45**±0.34	**93.46**±0.34	**89.24**±0.58

The results of VGG-19 with a fully connected layer with 512 neurons and with the addition of batch normalization layers were the best ones found in this study. Therefore, this configuration is the proposed solution model, and we named it DFU-VGG.

5) Results with dropout and batch normalization insertion: The literature, in general, indicates that the use of dropout and batch normalization in the same architecture causes a decrease in the performance of the results. However, there are circumstances where this combination works well [34]. Thus, we investigated the use of the two operations together.

Batch normalization layers were added in the sequential networks VGG-16 and VGG-19, in each block of convolutional layers and before the MaxPooling layers. Dropout layers were added after MaxPooling layers. The results of this configuration are indicated in Table VIII. From the data in this table, one can realize that the values of the metrics obtained were lower than those of the DFU-VGG (Table VII).

TABLE VIII
CLASSIFICATION RESULTS OBATINED WITH DROPOUT AND BATCH NORMALIZATION INSERTION (BEST VALIUES IN BOLD).

CNN	$A(\%)$	$P(\%)$	$R(\%)$	$F(\%)$	$K(\%)$
VGG-16	92.30±0.37	92.40±0.33	92.30±0.37	92.30±0.36	87.34±0.61
VGG-19	**92.36**±0.56	**92.46**±0.54	**92.36**±0.56	**92.36**±0.56	**87.43**±0.93

B. Ensemble of CNNs

In this section, we detail the ensemble of CNNs evaluated. Initially we selected only the settings of the 7 CNNs that obtained the highest Kappa. The Table IX details the classifiers who make up each committee.

TABLE IX
ENSEMBLES AND ITS CLASSIFIERS MEMBERS.

Acronym	Classifiers
V19V16ResDenIn	VGG-19 + BN, VGG-16 + BN, ResNet50 512, DenseNet201 1024-512, InceptionV3 512
V19V16Res	VGG-19 + BN, VGG-16 + BN, ResNet50 512
V19V16Den	VGG-19 + BN, VGG-16 + BN, DenseNet201 1024-512
V19V16ResDenEf	VGG-19 + BN, VGG-16 + BN, ResNet50 512, DenseNet201 1024-512, EfficientNetB0 512
V19V16ResDenMob	VGG-19 + BN, VGG-16 + BN, ResNet50 512, DenseNet201 1024-512, MobileNetV2 512-256
V19ResDenEf	VGG-19 + BN, VGG-16 + BN, ResNet50 512, DenseNet201 1024-512, EfficientNetB0
V19V16DenMobEf	VGG-19 + BN, VGG-16 + BN, DenseNet201 1024-512, MobileNetV2 512-256, EfficientNetB0 512

Ensemble are combined using simple and weighted majority voting. In simple majority voting we use the average Kappa to select the classifiers; in the case of a tie, we select the one with the highest individual value. In relation to weighted majority voting, weights are assigned to each classifier, the weight being proportional to the mean Kappa value.

The experiments were performed using a significant variety of CNN architectures. This criterion was defined in order to increase the robustness of the method. The Tables X and XI detail the five best results using simple and weighted majority voting, respectively.

Based on the results of the set, we observed that the use of the Ensemble significantly improved the metrics in relation to the individual classifiers. Also, we note the reduction of standard deviation, that is, the sets in this scenario have more stable data.

TABLE X
BETTER RESULTSFOR ENSEMBLE OF CNNS WHITH SIMPLE MAJORITY VOTING.

Acronym	$A(\%)$	$P(\%)$	$R(\%)$	$F(\%)$	$K(\%)$
V19V16Res	**94.86**±0.005	**94.89**±0.004	**94.86**±0.005	**94.86**±0.005	**91.55**±0.007
V19V16Den	94.71±0.005	94.74±0.005	94.71±0.005	94.71±0.005	91.31±0.008
V19V16ResDenEf	94.61±0.004	94.64±0.004	94.61±0.004	94.61±0.004	91.15±0.007
V19V16ResDenIn	94.38±0.004	94.43±0.004	94.38±0.004	94.38±0.004	90.77±0.007
V19V16ResDenMob	94.33±0.005	94.38±0.005	94.33±0.005	94.33±0.005	90.69±0.009

TABLE XI
BETTER RESULTSFOR ENSEMBLE OF CNNS WHITH WEIGHTED MAJORITY VOTING.

Acronym	$A(\%)$	$P(\%)$	$R(\%)$	$F(\%)$	$K(\%)$
V19V16ResDenIn	**95.04**±0.004	**95.06**±0.004	**95.04**±0.004	**95.04**±0.004	**91.85**±0.006
V19ResDenEf	95.03±0.004	95.03±0.004	95.03±0.004	95.02±0.004	91.83±0.006
V19V16ResDenMob	95.01±0.004	95.01±0.004	95.01±0.004	95.01±0.004	91.81±0.006
V19V16Den	94.99±0.002	95.01±0.002	94.99±0.002	94.99±0.002	91.77±0.004
V19V16DenMobEf	94.89±0.004	94.91±0.004	94.89±0.004	94.89±0.004	91.61±0.007

Still, it is possible to notice in the Table XI that the formation of ensemble applying a weighted vote between different Cnns presents a high Kappa in comparison to the metrics already obtained by separately and simple voting. Even the lowest result in this table had better performance than all other settings shown.

We achieve the best results using the V19V16ResDenIn set. The accuracy achieved was 93.98% and kappa was 91.85%, indicating a great agreement with the classification performed by a specialist. In addition, the other metrics achieved similar values: Accuracy of 95.04%, recall of 95.04% and F1-score of 95.04. The graph in Figure 4 allows comparing the overall performance of the T5Sd ensemble with their members individually.

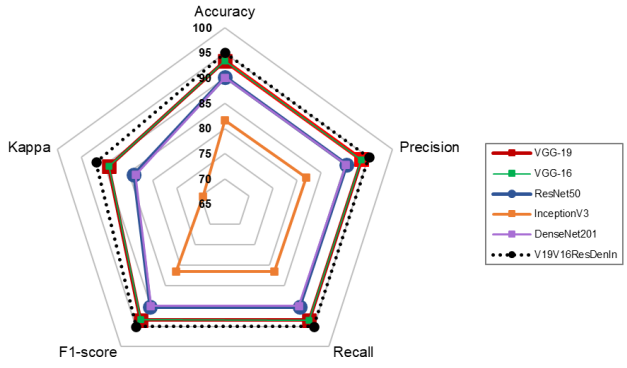
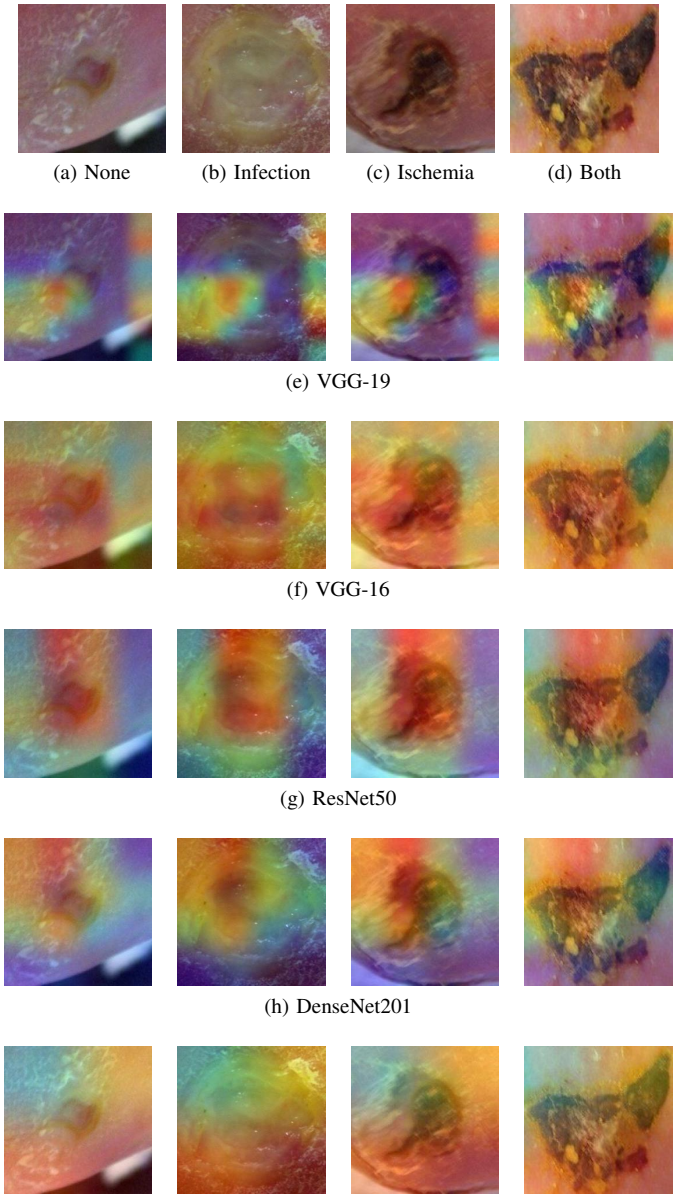


Fig. 4. V19V16ResDenIn ensemble and its members performance comparison.

Figure 5 shows the heat maps with the activation regions that DFU-VGG considered most important during feature extraction and, consequently, classification.



(a) None (b) Infection (c) Ischemia (d) Both

(e) VGG-19

(f) VGG-16

(g) ResNet50

(h) DenseNet201

(i) InceptionV3

Fig. 5. Heat map with activation regions for classes.

In the shown activation maps (Figure 5), blue tones mean low activation and indicate that the correspondent regions are of little importance for the final classification; in contrast, red tones are associated with the regions that contributed most to the final classification.

Most of the networks were concentrated in the areas of the lesion mainly the VGG-19 that practically did not remove the healthy skin areas. In contrast, the VGG-16 and InceptionV3 networks focused their attention on skin areas around the lesions. InceptionV3, in particular, assigned a similar weight in predominate all pixels of the images. Thus, we believe that the quality of the ensemble result is related to the union of the outputs of CNNs that gave emphasis to different characteristics of the same image.

V. Conclusions and Future Work

This work presented a novel CNN architecture and training strategy to classify diabetic foot ulcers, considering four classes. Several architectures, fine-tuning schemes, and parameters were studied to define the proposed model. This allowed us to develop a model for classification that is more accurate and robust than the methods presented in current state-of-the-art works.

The experiments showed that deep fine tuning was more efficient than superficial fine tuning, and that 500 seasons were suitable for training Ncns. The CNN, in their original configurations, were not adequate to the proposed problem, however, the use of the sets of Cnns began to classify the images of diabetic foot ulcers better. In addition, the use of different Cnns provides greater variety and robustness to the results.

The results obtained were promising, but it is believed that they can be improved. Therefore, we intend to conduct experiments with other CNNs to increase the classification accuracy and reduce the percentage of images of the infection class classified as none. Future work may also investigate the use of generative adversarial networks in increasing data availability; notably, these networks can generate heterogeneous images that adequately represent the original distribution. Finally, the evaluation of the computational results by additional experts would be crucial for the routine use of the proposed model.

References

[1] K. Ogurtsova, L. Guariguata, N. C. Barengo, P. L.-D. Ruiz, J. W. Sacre, S. Karuranga, H. Sun, E. J. Boyko, and D. J. Magliano, "Idf diabetes atlas: Global estimates of undiagnosed diabetes in adults for 2021," *Diabetes Research and Clinical Practice*, p. 109118, 2021.

[2] M. Goyal and S. Hassanpour, "A refined deep learning architecture for diabetic foot ulcers detection," *CoRR*, vol. abs/2007.07922, 2020. [Online]. Available: https://arxiv.org/abs/2007.07922

[3] P. R. Cavanagh, B. A. Lipsky, A. W. Bradbury, and G. Botek, "Treatment for diabetic foot ulcers," *The Lancet*, vol. 366, no. 9498, pp. 1725–1735, 2005.

[4] L. O. Solís-Sánchez, J. Ortiz-Rodriguez, R. Castañeda-Miranda, M. Martinez-Blanco, G. Ornelas-Vargas, J. I. Galván-Tejada, C. E. Galván-Tejada, J. M. Celaya-Padilla, and C. L. Castañeda-Miranda, "Identification and evaluation on diabetic foot injury by computer vision," *IEEE International Conference on Industrial Technology (ICIT)*, pp. 758–762, 2016.

[5] M. A. Al-Masni, M. A. Al-Antari, J.-M. Park, G. Gi, T.-Y. Kim, P. Rivera, E. Valarezo, M.-T. Choi, S.-M. Han, and T.-S. Kim, "Simultaneous detection and classification of breast masses in digital mammograms via a deep learning yolo-based cad system," *Computer methods and programs in biomedicine*, vol. 157, pp. 85–94, 2018.

[6] L. H. Vogado, R. M. Veras, F. H. Araujo, R. R. Silva, and K. R. Aires, "Leukemia diagnosis in blood slides using transfer learning in cnns and svm for classification," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 415–422, 2018.

[7] F. Shariaty and M. Mousavi, "Application of cad systems for the automatic detection of lung nodules," *Informatics in Medicine Unlocked*, vol. 15, p. 100173, 2019.

[8] F. Veredas, H. Mesa, and L. Morente, "Binary tissue classification on wound images with neural networks and bayesian classifiers," *IEEE Transactions on Medical Imaging*, vol. 2, no. 29, pp. 410–427, 2010.

[9] L. Wang, P. C. Pedersen, E. Agu, D. M. Strong, and B. Tulu, "Area determination of diabetic foot ulcer images using a cascaded two-stage svm-based classification," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2098–2109, 2016.

[10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[11] X. He, R. S. Zemel, and M. A. Carreira-Perpinán, "Multiscale conditional random fields for image labeling," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[12] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, "Dfunet: Convolutional neural networks for diabetic foot ulcer classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 728–739, 2020.

[13] L. Alzubaidi, M. Fadhel, S. Oleiwi, O. Al-Shamma, and J. Zhang, "Dfu_qutnet: diabetic foot ulcer classification using novel deep convolutional neural network," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15 655–15 677, 2020.

[14] M. Goyal, N. D. Reeves, S. Rajbhandari, N. Ahmad, C. Wang, and M. H. Yap, "Recognition of ischaemia and infection in diabetic foot ulcers: Dataset and techniques," *Computers in Biology and Medicine*, vol. 117, pp. 103 616–103 616, 2020.

[15] J. Amin, M. Sharif, M. A. Anjum, H. U. Khan, M. S. A. Malik, and S. Kadry, "An integrated design for classification and localization of diabetic foot ulcer based on cnn and yolov2-dfu models," *IEEE Access*, vol. 8, pp. 228 586–228 597, 2020.

[16] S. K. Das, P. Roy, and A. K. Mishra, "Dfu_spnet: A stacked parallel convolution layers based cnn to improve diabetic foot ulcer classification," *ICT Express*, 2021.

[17] B. Rostami, D. Anisuzzaman, C. Wang, S. Gopalakrishnan, J. Niezgoda, and Z. Yu, "Multiclass wound image classification using an ensemble deep cnn-based classifier," *Computers in Biology and Medicine*, p. 104536, 2021.

[18] A. Galdran, G. Carneiro, and M. Á. G. Ballester, "Convolutional nets versus vision transformers for diabetic foot ulcer classification," *Lecture Notes in Computer Science*, vol. 13183, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-94907-5_2

[19] M. H. Yap, R. Hachiuma, A. Alavi, R. Brüngel, B. Cassidy, M. Goyal, H. Zhu, J. Rückert, M. Olshansky, X. Huang, H. Saito, S. Hassanpour, C. M. Friedrich, D. B. Ascher, A. Song, H. Kajita, D. Gillespie, N. D. Reeves, J. M. Pappachan, C. O'Shea, and E. Frank, "Deep learning in diabetic foot ulcers detection: A comprehensive evaluation," *Computers in Biology and Medicine*, vol. 135, p. 104596, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482521003905

[20] N. Al-Garaawi, R. Ebsim, A. F. Alharan, and M. H. Yap, "Diabetic foot ulcer classification using mapped binary patterns and convolutional neural networks," *Computers in Biology and Medicine*, vol. 140, p. 105055, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482521008490

[21] J. Wang, L. Perez *et al.*, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, vol. 11, pp. 1–8, 2017.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2661–2671.

[24] L. Vogado, R. Veras, K. Aires, F. Araújo, R. Silva, M. Ponti, and J. M. R. Tavares, "Diagnosis of leukaemia in blood slides based on a fine-tuned and highly generalisable deep learning model," *Sensors*, vol. 21, no. 9, pp. 2989–2989, 2021.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[30] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, Cambridge, MA, USA, 2014, pp. 3320–3328.

[32] M. Izadyyazdanabadi, E. Belykh, M. Mooney, N. Martirosyan, J. Eschbacher, P. Nakaji, M. C. Preul, and Y. Yang, "Convolutional neural networks: ensemble modeling, fine-tuning and unsupervised semantic localization for neurosurgical cle images," *Journal of Visual Communication and Image Representation*, vol. 54, pp. 10–20, 2018.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[34] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, pp. 1–39, 05 2020.

[35] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 708–713, 1979.

[36] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[37] E. Alves, J. B. Souza Filho, and A. L. Kritski, "An ensemble approach for supporting the respiratory isolation of presumed tuberculosis inpatients," *Neurocomputing*, vol. 331, pp. 289–300, 2019.

[38] F. Piccialli, F. Giampaolo, A. Salvi, and S. Cuomo, "A robust ensemble technique in forecasting workload of local healthcare departments," *Neurocomputing*, vol. 444, pp. 69–78, 2021.

[39] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[40] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.