Heliyon 10 (2024) e28058

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Classification of skin blemishes with cell phone images using deep learning techniques

José Antonio Rangel-Ramos [a], Francisco Luna-Perejón [b], Anton Civit [b,c], Manuel Domínguez-Morales [b,c,*]

[a] *Universidad de Sevilla, ETS Ingeniería Informática, Avda. Reina Mercedes s/n, Seville, 41012, Spain*

[b] *Computer Architecture and Technology Dept. (Universidad de Sevilla), ETS Ingeniería Informática, Avda. Reina Mercedes s/n, Seville, 41012, Spain*

[c] *Computer Science Research Institute (Universidad de Sevilla), Avda. Reina Mercedes s/n, Seville, 41012, Spain*

ARTICLE INFO

Keywords:
Skin
Deep learning
Convolutional neural network
Artificial intelligence

ABSTRACT

Skin blemishes can be caused by multiple events or diseases and, in some cases, it is difficult to distinguish where they come from. Therefore, there may be cases with a dangerous origin that go unnoticed or the opposite case (which can lead to overcrowding of health services). To avoid this, the use of artificial intelligence-based classifiers using images taken with mobile devices is proposed; this would help in the initial screening process and provide some information to the patient prior to their final diagnosis. To this end, this work proposes an optimization mechanism based on two phases in which a global search for the best classifiers (from among more than 150 combinations) is carried out, and, in the second phase, the best candidates are subjected to a phase of evaluation of the robustness of the system by applying the cross-validation technique. The results obtained reach 99.95% accuracy for the best case and 99.75% AUC. Comparing the developed classifier with previous works, an improvement in terms of classification rate is appreciated, as well as in the reduction of the classifier complexity, which allows our classifier to be integrated in a specific purpose system with few computational resources.

1. Introduction

Various studies point to an increase in global temperature and therefore in the incidence of sun rays on the skin [22]. This fact, in addition to other multiple diseases, causes the proliferation of spots on the skin, which, in most cases, is not easy to distinguish the cause of its occurrence. Because of that, it can lead to situations such as not giving these spots any importance (assuming that they are simple moles or age spots), or going to the health emergency services for any spot.

Regarding this last situation, the World Health Organization (WHO) estimates that by 2030 there will be a global shortage of around 14 million health professionals [29]. This shortage of health workers can lead to overburdened health systems, resulting in increased waiting times for care, a reduction in the quality of health services, and ultimately the collapse of the health system [13].

* Corresponding author at: Computer Architecture and Technology Dept. (Universidad de Sevilla), ETS Ingeniería Informática, Avda. Reina Mercedes s/n, Seville, 41012, Spain.

E-mail addresses: rangel.joseant@gmail.com (J.A. Rangel-Ramos), fluna1@us.es (F. Luna-Perejón), civit@us.es (A. Civit), mjdominguez@us.es (M. Domínguez-Morales).

https://doi.org/10.1016/j.heliyon.2024.e28058
Received 15 February 2024; Received in revised form 21 February 2024; Accepted 11 March 2024

Available online 29 March 2024

Table 1
Dataset distribution.

Class	ID	Abbreviation	Number of images	%
Melanocytic nevi	0	nv	6,620	66.9
Melanoma	1	mel	1,087	11.0
Benign keratosis-like lesions	2	bkl	1,086	11.0
Basal cell carcinoma	3	bcc	509	5.1
Actinic keratoses	4	akiec	327	3.3
Vascular lesions	5	vasc	140	1.5
Dermatofibroma	6	df	115	1.2
TOTAL		9,884	100	

And, on the other hand, the misuse of healthcare is a problem that affects many health systems around the world. The request for emergency healthcare for banal cases is one of the main problems associated with it: many people go to emergency rooms for minor problems that could be treated in primary care or even self-care at home. This not only saturates emergency departments, but also contributes to increased waiting times and reduced quality of care provided to patients who really need urgent care [1,16].

In the case of skin diseases, both cases are worrying: a patient does not give a blemish the importance it deserves and that a patient cannot get prompt attention for a serious case. Therefore, it is important to promote the use of semi-automated screening systems to help reduce the workload of medical professionals and emergency departments.

At this point, the application of Artificial Intelligence (AI) techniques is of great importance to design classifier systems capable of extracting features from images and differentiating between those that show some kind of disease and those that represent a healthy patient [12,4,17,3].

According to previous statistical studies on cancer diagnosis, such as the one conducted by Mark Priebe and Markin [20], in general, in medical image-guided diagnosis, the average percentage of discrepancies in diagnostic reports is 12%; therefore, any assisted diagnosis system that exceeds 88% accuracy would theoretically have a higher accuracy rate than the pathologist. However, the main objective of these systems is not to replace the pathologist but to serve as a tool to help reduce the pathologist's workload, always taking into account a final intervention by the pathologist to validate the results.

Therefore, after explaining this problem, the aim of this work is to design and evaluate a tool to classify skin blemishes using images taken with a mobile device. For this purpose, a dataset will be used that distinguishes between 7 types of skin blemishes (from harmless moles to melanoma), and AI techniques will be applied to obtain the best possible classifier.

The remainder of the manuscript is structured as follows: The methods used to develop and test the diagnosis aid system are presented in Section 2. The results obtained after testing the classifier and the discussion comparing the results obtained with previous works are detailed in Section 3. Finally, in Section 4, the final conclusions of this work are detailed.

2. Materials and methods

This section will present the dataset used to train the classifier, the various classifiers trained for this purpose, the process followed to evaluate them individually, and the optimization process followed to obtain the best classifier.

2.1. Dataset

The database known as *Skin Cancer MNIST: HAM10000* has been used for this work [5]. This is a set of skin images that focuses mainly on the creation of neural networks for the detection and classification of skin lesions. These images have been collected by the University Hospital of the Medical University of Vienna and the University of Queensland in Australia. Each image in this database is a high-resolution photograph of a skin lesion accompanied by information about where the lesion is located, the patient's age and gender, and clinical diagnosis. To certify and confirm the diagnosis of the images, more than 50% have undergone histopathology tests. For the remainder, follow-up examination, analysis by an expert panel, or confirmation by confocal microscopy have been used.

This database has a total of almost 10,000 images of skin lesions of 7 different types. The distribution of images for each type is shown in Table 1.

In Fig. 1, a sample image from each class is presented. As can be observed, it is not easy to distinguish between some classes.

This dataset has been used in many research studies with the aim of creating a machine learning algorithm with sufficient capability for the detection and classification of these skin lesions. Therefore, previous works that have used this same dataset will be used to compare the results finally obtained in this work.

As can be seen, the classes provided by this dataset are unbalanced. If the classifier is trained normally, very good overall results may be obtained, with very bad results for the less predominant classes. In the next subsection, the solution provided in this work to solve this problem will be detailed.

2.2. Classifiers

This subsection will detail the classical metrics for evaluating a classifier system, followed by the optimization process performed to obtain a good classifier for the dataset detailed before.

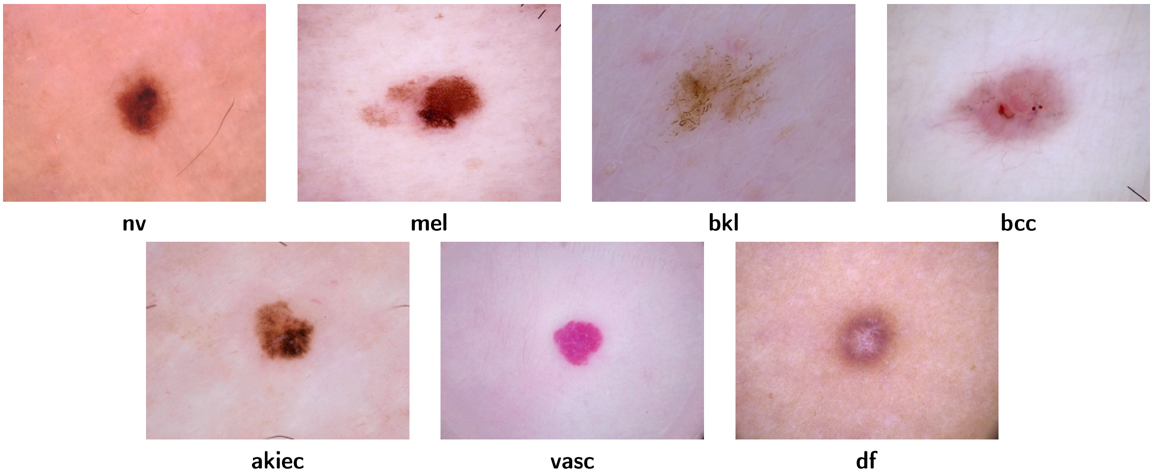

Fig. 1. Samples images from the dataset.

2.2.1. *Evaluation metrics*

To evaluate the effectiveness of the classification results of a classifier, the most common metrics are used: accuracy (most-used metric), sensitivity (also known as recall), precision, and F1$_{score}$ [26]. To this end, the classification results obtained for each class are tagged as *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) or *False Negative* (FN). According to them, the high-level metrics are presented in the following equations:

$$Accuracy = \sum_c \frac{TP_c + TN_c}{TP_c + FP_c + TN_c + FN_c}, c \in classes \tag{1}$$

$$Precision = \sum_c \frac{TP_c}{TP_c + FP_c}, c \in classes \tag{2}$$

$$Sensitivity = \sum_c \frac{TP_c}{TP_c + FN_c}, c \in classes \tag{3}$$

$$Specificity = \sum_c \frac{TN_c}{TN_c + FP_c}, c \in classes \tag{4}$$

$$F1_{score} = 2 * \frac{precision * sensitivity}{precision + sensitivity}. \tag{5}$$

About those metrics:

- Accuracy: all samples classified correctly compared to all samples (see Equation (1))
- Precision: proportion of values classified as *TP* in all cases that have been classified as it (see Equation (2))
- Sensitivity (or Recall): proportion of values classified as *TP* that are correctly classified (see Equation (3))
- Specificity: proportion of values classified as *TN* that are correctly classified (see Equation (4)).
- F1$_{score}$: it considers two of the main metrics (precision and sensitivity), calculating the harmonic mean of both parameters (see Equation (5))

The above metrics are common to all machine / deep learning systems; but there are other commonly used metrics like the ROC curve (Receiver Operating Characteristic) [15], which is of particular interest in diagnostic systems, because it is the visual representation of the True Positives Rate (TPR) versus the False Positives Rate (FPR) as the discrimination threshold is varied. Usually, when the ROC curve is used, the area under the curve (AUC) is used as a value of the system's goodness-of-fit.

Therefore, the classifier systems developed in this work will be evaluated according to all the metrics detailed in this subsection. Moreover, the results obtained for the classification system will be compared with the results obtained in previous works.

2.2.2. *Optimization process*

In order to obtain the best classifier for the system, an optimization process based on two steps is presented: In the first step, a global search is performed including multiple trainings with different variations of the hyperparameters for each classifier; and in the second step, the robustness of the best models is tested applying cross-validation technique. After this process, the best classifier is compared with previous works. The full process diagram can be seen in Fig. 2.

2.2.2.1. Grid search. The first step before starting the training process is to divide the dataset randomly in two subsets: train and test. The training subset represents 80% of the original data set, and the testing subset contains the remaining 20%.

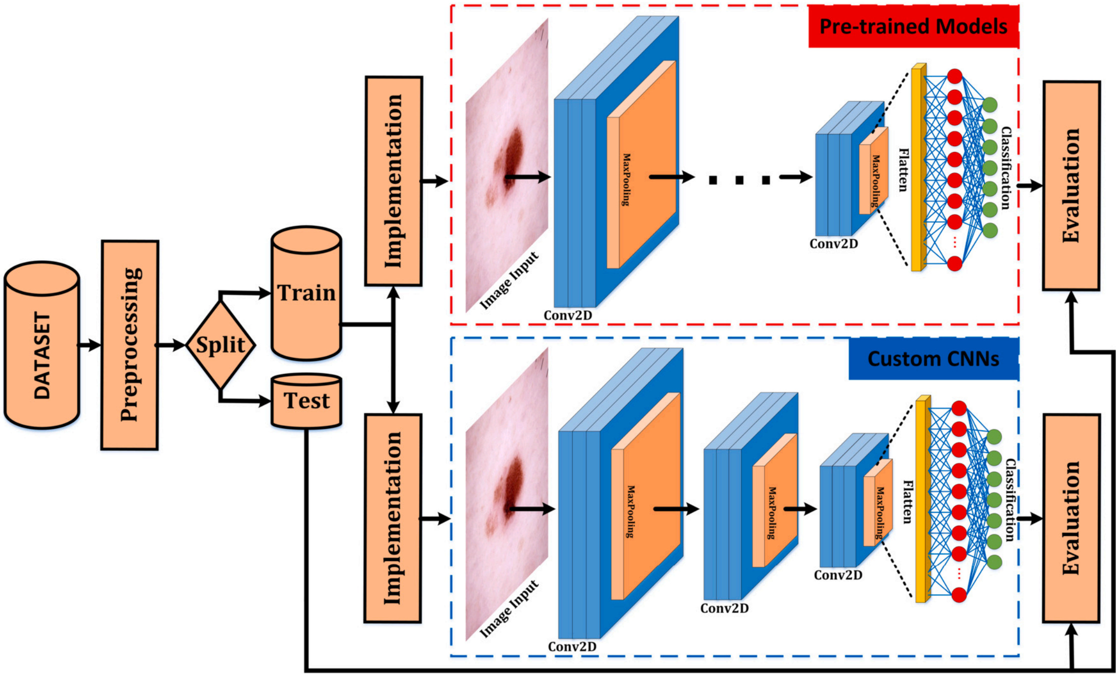

Fig. 2. Graphical Abstract.

Table 2
Hyperparameters' values.

Parameter	Values
Batch size	8, 16, 24, 32
Learning rate	0.001, 0.0001, 0.00001

After this division, a global search is performed, which means that we are going to train several combinations of architectures and hyperparameters in order to find the best combinations. The main criterion for selecting the best candidates after the global search process is the Accuracy value of the Test subset.

In addition, there is a big issue regarding the dataset unbalance. To avoid this problem, there are usually two mechanisms to solve it: "data augmentation" and "weighted classes" [28]. In the case of "data augmentation", new artificial images are generated from the existing ones for the classes with fewer data by performing visual transformations (rotations, zoom, etc.). However, when there is a very large imbalance, this causes that the amount of artificial data can exceed the amount of real data (and that can be a bias for the classifier). For this reason, the "weighted classes" technique has been used: in this technique, a numerical weight value is given to each class during the training process. Thus, if one class is unbalanced and has fewer data than another, the training can be balanced by providing a weight to that class that is inversely proportional to the percentage of data it has.

It is known that deep networks need huge amounts of data, and several augmentation methods have been applied in the literature to increase the reliability and robustness of the methods [11]. Therefore, the proposed approach can be tested with the increased number of images as a future work.

The hyperparameters taken into account in this study are *batch size* and *learning rate*. Table 2 details the values for each hyperparameter used in each architecture.

Regarding network architectures, 4 types of pre-trained models have been tested: VGG16 [25], ResNet [14], MobileNet [24] and EfficientNet [27]. In addition, 10 customized architectures have been defined. These custom architectures are detailed in Table 3.

As can be observed in Table 3, all custom architectures are composed of combinations of *convolution layers* (apply 3x3 convolutions to the images) and *max-polling layers* (reduce images' sizes), ended with a *flatten layer* (transform from 2D data to 1D data) and a *dense layer* (output layer).

In addition, in architectures with more than one convolutional layer, *batch-normalization layers* (standardize the inputs to a layer) and additional *dense layers* are introduced before the output layer. Finally, every custom architecture has two versions (*a* and *b*), with version *a* not including *dropout layers* (randomly removes a different percentage of connections between layers of neurons at each training epoch), and version *b* including them.

In total, 168 training sessions are held. And, from the results obtained, the four classifiers with the best results are extracted. These four final classifiers will be subjected to the second phase of the optimization process, which will be described below.

Table 3
Custom Architectures detailed layer by layer.

ID	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9	Layer 10
1a	CO MP FL	DS(7)								
1b	CO MP FL DP(0.2)	DS(7)								
2a	CO BN MP	CO BN MP FL	DS(4096)	DS(7)						

2b	CO
	BN
	MP
		CO
		BN
		MP
		FL
			DS(4096)
			DP(0.2)
				DS(7)						

3a	CO BN MP	CO BN MP	CO BN MP FL	DS(256)	DS(4096)	DS(7)				

3b	CO
	BN
	MP
		CO
		BN
		MP
			CO
			BN
			MP
			FL
				DS(256)
				DP(0.2)
					DS(4096)
					DP(0.3)
						DS(7)				

4a	CO BN MP	CO BN MP	CO BN MP	CO BN MP	CO BN MP FL	DS(256)	DS(4096)	DS(7)		

4b	CO
	BN
	MP
		CO
		BN
		MP
			CO
			BN
			MP
				CO
				BN
				MP
					CO
					BN
					MP
					FL
						DS(256)
						DP(0.2)
							DS(4096)
							DP(0.3)
								DS(7)		

5a	CO BN MP	CO BN MP	CO BN MP	CO BN MP	CO BN MP	CO BN MP	CO BN MP FL	DS(256)	DS(4096)	DS(7)

5b	CO
	BN
	MP
		CO
		BN
		MP
			CO
			BN
			MP
				CO
				BN
				MP
					CO
					BN
					MP
						CO
						BN
						MP
							CO
							BN
							MP
							FL
								DS(256)
								DP(0.2)
									DS(4096)
									DP(0.3)
										DS(7)

CO: 2D Convolution
MP: Max-Polling
BN: Batch-Normalization
FL: Flatten
DS(num): Dense Layer
num: number of neurons
DP(per): Dropout Layer
per: percentage

2.2.2.2. Cross-validation. Secondly, classifiers with the best results from the previous phase are evaluated with additional robustness tests.

One of the most common mechanisms for this purpose is *cross-validation*, which consists of repeating the training with different divisions of the training and test subsets, to determine how the model will behave with an unknown test dataset not used before [2]. This method is widely used to evaluate the results and to guarantee the independence of the results with respect to the division of the data used in the training and test sets.

Among the various implementations of this technique, this work will apply the *K-fold cross-validation* variant, in which the dataset is divided into K groups, where one of them is used for testing and the next *K-1* as training. This process is performed a total of K times, mixing the groups in each interaction so that the same groups are not used. This process is detailed in Fig. 3. For this work, the dataset has been divided in 5 folds, so we are working with a 5-fold cross-validation.

After applying this technique, five accuracy results would be obtained for each candidate (one per split). With these results, the average and standard deviation would be obtained for each classifier, metrics that will help determine the best classifier obtained in this work.

And finally, the results of all the metrics described in the previous subsection will be shown for this classifier. These would be the metrics used for the comparison with the previous works.

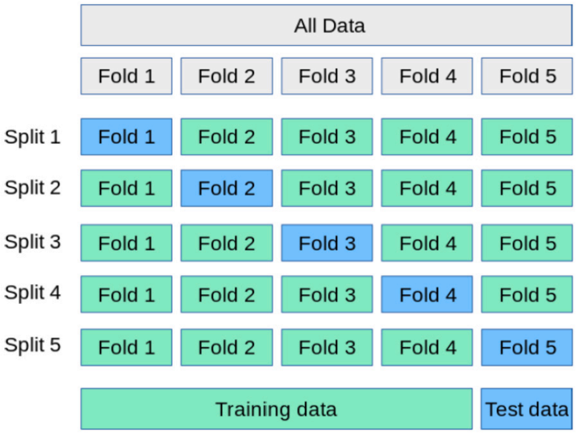

Fig. 3. K-fold cross validation representation.

Table 4
Grid search results.

			Accuracy					Accuracy	
Architecture	Batch	Learning rate	*Train*	*Test*	**Architecture**	**Batch**	**Learning rate**	***Train***	***Test***
1a	8	0.001	0.9126	0.6798	**1b**	8	0.001	0.7721	0.7036
	8	0.0001	0.98	0.65		8	0.0001	0.9607	0.6707
	16	0.001	0.7316	0.653		16	0.001	0.7958	0.6596
	24	0.00001	0.9263	0.7248		24	0.00001	0.7789	0.6930
	32	0.0001	0.9311	0.7051		32	0.0001	0.8090	0.6606
2a	8	0.001	0.9719	0.7527	**2b**	8	0.001	0.9479	0.7350
	8	0.0001	0.9903	0.7527		8	0.0001	0.9833	0.7491
	16	0.001	0.9941	0.7405		16	0.001	0.9791	0.7178
	24	0.00001	0.9986	0.7314		24	0.00001	0.9977	0.7552
	32	0.0001	0.9774	0.7243		32	0.0001	0.9842	0.7162
3a	8	0.001	0.98	0.7638	**3b**	8	0.001	0.9435	0.7572
	8	0.0001	0.9865	0.7602		8	0.0001	0.9851	0.7587
	16	0.001	0.9817	0.7805		16	0.001	0.9851	0.7506
	24	0.00001	0.9949	0.7162		24	0.00001	0.9825	0.7430
	32	0.0001	0.9966	0.7825		32	0.0001	0.9918	0.7891
4a	8	0.001	0.9870	0.7481	**4b**	8	0.001	0.6698	0.6697
	8	0.0001	0.9819	0.7663		8	0.0001	0.9736	0.7289
	16	0.001	0.9887	0.6889		16	0.001	0.976	0.7279
	24	0.00001	0.9936	0.7486		24	0.00001	0.9874	0.7542
	32	0.0001	0.9922	0.6581		32	0.0001	0.9743	0.7587
5a	8	0.001	0.7587	0.6616	**5b**	8	0.001	0.7111	0.6697
	8	0.0001	0.9738	0.7339		8	0.0001	0.9751	0.7365
	16	0.001	0.9795	0.7309		16	0.001	0.9664	0.7557
	24	0.00001	0.9863	0.7198		24	0.00001	0.9825	0.7436
	32	0.0001	0.9211	0.741		32	0.0001	0.9469	0.7309
VGG16	8	0.0001	1.0	0.7425	**ResNet**	8	0.0001	0.9919	0.7521
	16	0.001	0.6798	0.6525		16	0.001	0.9989	0.7218
	24	0.00001	1.0	0.7754		24	0.00001	0.7142	0.7142
	32	0.0001	0.9999	0.7641		32	0.0001	0.9954	0.7016
MobileNet	8	0.0001	0.9822	0.7208	**EfficientNet**	8	0.0001	0.9879	0.7041
	16	0.001	0.9944	0.737		16	0.001	0.9896	0.7031
	24	0.00001	0.9999	0.7511		24	0.00001	0.9994	0.7122
	32	0.0001	0.9999	0.7441		32	0.0001	0.9699	0.6990

3. Results and discussion

This section presents the results obtained from each of the phases of the optimization process described above. Finally, these results will be compared with those obtained in previous works, including a discussion of them.

First, the accuracy results for both training and the test are detailed in Table 4 for a 100-epoch training. In this table, not all the 168 results are detailed, only a subset of 66 training (5 for each custom architecture and 4 for each pre-trained model).

Table 5
Cross-validation accuracy results in % with average value (Avr.) and standard deviation (SD).

	Fold1	Fold2	Fold3	Fold4	Fold5	Avr.	SD
C1	77.85	98.73	99.09	99.34	99.95	94.99	9.59
C2	75.86	91.40	76.77	86.50	91.40	84.39	7.64
C3	74.29	90.84	98.58	91.90	98.28	90.78	9.88
C4	78.34	90.34	88.27	98.23	96.31	90.30	7.84

Table 6
Metrics results for each class.

Class	Acc	Pre	Sen	Spe	$F1_{score}$
nv	99.94	99.92	100	99.84	99.96
mel	100	100	100	100	100
bkl	99.94	100	99.54	100	99.77
bcc	100	100	100	100	100
akiec	100	100	100	100	100
vasc	100	100	100	100	100
df	100	100	100	100	100

Although the results obtained do not exceed 79% for the test subset, it is important to remember that training was carried out with only 100 epochs due to the large amount of time devoted to this phase. Therefore, after visualizing all the results, the best result has an accuracy of 78.91%. And, therefore, the best performing networks are those that have an accuracy as close as possible to this value.

Taking into account the test accuracy value, the four classifiers that obtain the best score (and considered the four selected candidates) are:

- **C1:** pre-trained VGG16 with batch size 24 and learning rate 0.00001, obtaining 77.54% accuracy.
- **C2:** custom architecture 3a with batch size 16 and learning rate 0.001, obtaining 78.05% accuracy.
- **C3:** custom architecture 3a with batch size 32 and learning rate 0.0001, obtaining 78.25% accuracy.
- **C4:** custom architecture 3b with batch size 32 and learning rate 0.0001, obtaining 78.91% accuracy.

These selected models are indicated in Table 4 in red.

These four classifiers will now be subjected to the second phase of the optimization process. Here, a 5-fold cross-validation process will be applied, performing five training runs for each of the classifiers and now allowing for a higher number of epochs. The results obtained are presented in Table 5.

Table 5 shows that the results of the first fold are the most similar to those obtained during the grid search phase, and this is due to the fact that this division is precisely the one carried out for this process.

Moreover, as can be seen, the division that was initially used seems to be the most detrimental in all the training sessions, and that is why the best classifier obtained in phase 1 is not equivalent to the one obtained now.

If the results obtained are analyzed in detail, the classifier with the best accuracy results is C1 (pre-trained VGG16), which obtains an average accuracy of 95%. These results are almost 5% better than those obtained with the second-best classifier. Although the standard deviation results of this classifier are higher than those of two of the other three classifiers analyzed, it is not a substantial difference and is mainly caused by fold 1 (since, removing it, the average accuracy would be 99.28% with a standard deviation of 0.51).

After this detailed explanation of the choice of the best classifier, it can be concluded that C1 will finally be selected. Then, using the best model among those trained with cross-validation, the results of all metrics for each class independently are presented in Table 6.

The results shown in the table above indicate that 5 of the 7 classes have a perfect classification. For the remaining 2, the first class (*nv*) shows a decrease in *specificity*, as a result of having false positives, but a percentage of 100% in *sensitivity*, so it has no false negatives. The opposite is true for the *bkl* class: no false positives (100% *specificity*) but false negatives (99.54% *sensitivity*).

These findings are also reflected in the confusion matrix (see Fig. 4). Furthermore, the result obtained for the unified AUC is 99.75% (see Fig. 5).

As a final point, it is important to mention that intensities in the images are usually inhomogeneous and affect the performance of the automated image analysis methods. Also, the images are noisy. Although several denoising and normalization algorithms have been applied with different types of images to obtain high performance [10], they may cause an increase in computational costs. In the proposed approach, the efficiency has been provided without any denoising. Also, except the batch normalization, there is no intensity normalization step in the proposed method.

Finally, the classifier designed and evaluated in this work will be compared with the results of previous work. For this purpose, a search was carried out with the works that used the same dataset as ours and the five best papers (taking into account their accuracy) were extracted. These works are those published by Rezvantalab et al. [23], Emara et al. [8], Lan et al. [18], and Datta et al. [7].

In the study proposed by Rezvantalab et al. [23], different pre-trained CNN models are tested: Inception-v3, DenseNet and AlexNet; obtaining a global value of 98.8% accuracy for the best case.

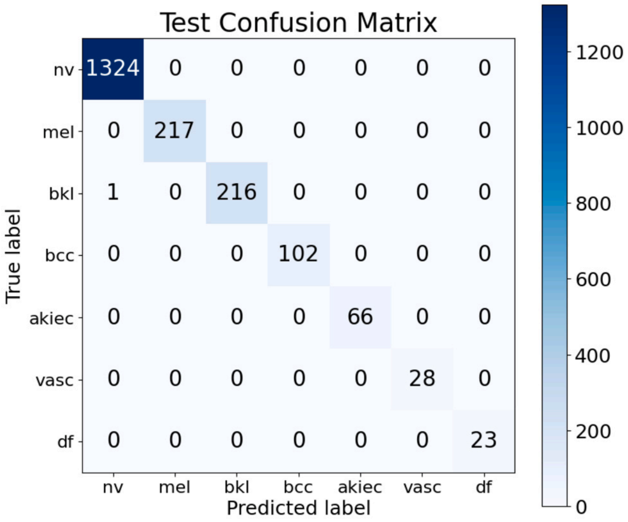

Fig. 4. Confusion matrix.

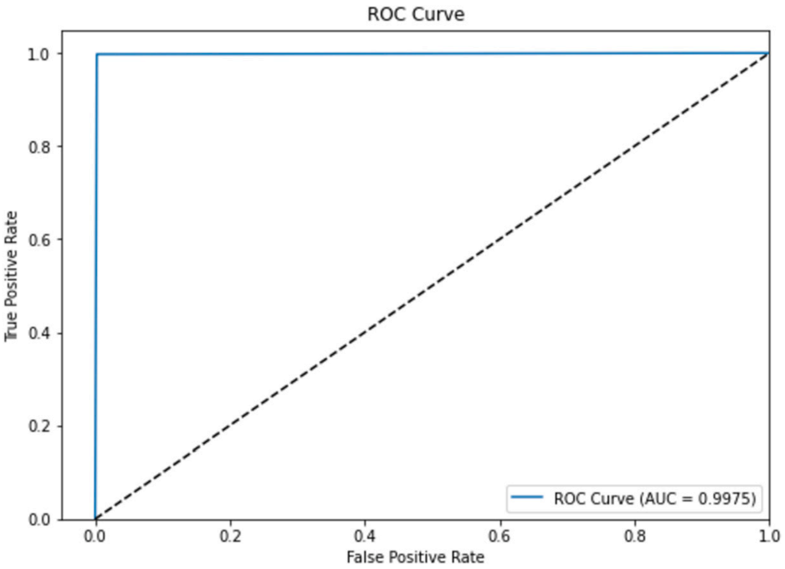

Fig. 5. ROC curve and AUC.

The second study is the one proposed by Emara et al. [8]. The authors made a modified version of the Inception-v4 architecture to better handle the problem of data imbalance. The authors made a division of 90-10 for the train and test subsets, achieving an accuracy of 94.7% for the test subset and an 86% for the test subset.

The third study [7], presents a Soft-Attention technique to improve accuracy by paying more attention to certain image features during the classification process. To apply this new technique, the authors use an additional layer to generate an attention map showing the most relevant areas of the image during classification, which are used to adjust the weights of each layer. This technique has achieved an accuracy of 93.7%.

Finally, the fourth study considered is the one proposed by Lan et al. [18]. This work presents a capsule network (inspired by the structure of the human visual system) to improve the detection performance of these diseases. The authors include an activation correction layer, Fixcaps. This layer fixes the activation problems that may occur in the previous layers. With this study, a 96.49% accuracy is achieved.

The general summary of these works ordered by year (together with the results of our system) is shown in Table 7.

If we compare our classifier with previous work in this table, we can see that we have an improvement over previous work in both accuracy and AUC. In addition, it is worth noting that the pre-trained VGG16 model has significantly lower computational cost than the models used in previous works. This comparison will be made in detail:

Table 7
Previous works comparison.

				Complexity		
Work	Classifier	Accuracy	AUC	CL	PL	DL
[23]	DenseNet201		98.8%	200	5	1
[8]	Inception-v4	86-94.7%	83.8%	172	19	2
[7]	Inception-ResNet-v2	93.70%	98.4%	235	7	2
[18]	custom Capsule Network	96.49%		37	136	21
This Work (2023)	VGG16	99.95%	99.75%	13	5	3
This Work (2023)	custom CNN (C4)	98.23%	97.19%	3	3	3

***CL:** Convolutional Layers. **PL:** Polling Layers. **DL:** Dense Layers.*

- Work [23]: This work obtains high AUC results (98.8%), but does not present the accuracy value (although it is intuited that it should exceed 99% due to the AUC value). In this case, the AUC result obtained by our classifier is higher (97.75%). Moreover, if we compare the complexity of the CNN used by both works, we can observe that this work uses 15 times more convolutional layers than ours.
- Work [8]: In this second case, the accuracy results shown are 94.7% for the training set and 86% for the test set. As our results are for the test set, the comparison must be made with the 86% value. In that case, our work obtains much better results in both accuracy (86 versus 99.95) and AUC (83.8 versus 99.75). Moreover, looking at the complexity of the CNN, this work is 13 times more convolutional layers.
- Work [7]: For this third case, we have the most complex network of all those compared, with more than 230 convolutional layers (which is 18 times the number of layers used by us). However, this complexity of the network does not translate into a substantial improvement in the results, as its accuracy value is 93.70% (compared to the 99.95% obtained by us). Similarly, the AUC value is lower than ours (98.4 versus 99.75).
- Work [18]: Finally, for this work, it is important to highlight the creation of a partially personalized neural network, which helps to significantly simplify the number of convolutional layers, but due to its feature extraction mechanism, significantly increases the number of polling layers (average and max polling) and dense layers. Still, it has almost three times the number of convolutions used in our classifier. The results of this work are presented only with accuracy, obtaining almost 96.5%, which is lower than the 99.75% obtained by our classifier.

In summary, if we compare these previous works with the best results obtained by our classifier, we outperform the classification rate in all cases, also using a less computationally complex CNN.

However, the results obtained by the best classifier that is not a pre-trained model are also included in the table as the last row. In that case, the accuracy and AUC results are reduced, but still acceptable and better than most of the previous work. Furthermore, with the custom network we further reduce the complexity of the system, which is an aspect to consider if you want to integrate this classifier into a specific purpose system (such as an embedded system).

It is therefore clear that in recent years research has been carried out on the classification of skin spots, obtaining in all cases more than acceptable results, but the systems developed do not take into account the computational cost associated with the classifiers used. This is important for future developments in which it is intended to integrate the classifier into an embedded system with limited computational resources. This aspect has been taken into account in our work, analyzing the results not only with complex pre-trained networks but also with less complex custom networks. In all cases, the results obtained in our work resemble or exceed previous work.

4. Conclusions

This work detailed the need for the use of an initial screening system to classify skin blemishes.

The use of a classifier based on the application of AI techniques on images taken with a mobile device has been proposed.

As a result, an optimization process based on two phases has been presented to obtain a classifier with acceptable results, using a public dataset of skin images, with a first phase of grid search with more than 150 combinations of convolutional neural networks and a second phase of robustness study using the cross-validation technique.

The results obtained with the final classifier reach a maximum accuracy of 99.49% and 99.75% AUC.

These results have been compared with works from recent years using the same dataset. This comparison reveals that the classifier obtained in this work improves the classification results of previous works and, in addition, substantially reduces the complexity of previous systems.

Given these conclusions, we can focus on the contribution of this work. In this regard, it is worth highlighting the importance of reducing the complexity of the classifier system without affecting the accuracy of the classification. This fact opens an important path to being able to integrate these classifiers into an embedded device, allowing in the future to design portable instruments that perform the classification task in real time.

As future work, comparative evaluations with the performances of capsule network-based methods can be performed because a capsule network can preserve spatial relationships of learned features, and have been proposed recently for image classifications [9]. Other future work may be integrating these classification systems into slow-cost embedded systems and study its importance, as has been proposed in previous works [6,21,19].

Funding

This work was supported by the Spanish AEI project MINDROB (PID2019-105556GB-C33 financiado por MICIU/AEI/ 10.13039/501100011033).

CRediT authorship contribution statement

José Antonio Rangel-Ramos: Writing – original draft, Validation, Software, Investigation. **Francisco Luna-Perejón:** Writing – review & editing, Methodology, Conceptualization. **Anton Civit:** Writing – review & editing, Supervision, Funding acquisition. **Manuel Domínguez-Morales:** Writing – review & editing, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Alnasser, et al., Analysis of emergency department use by non-urgent patients and their visit characteristics at an academic center, Int. J. Gen. Med. (2023) 221–232.
[2] D. Berrar, Cross-validation, in: Encyclopedia of Bioinformatics and Computational Biology, vol. 1, 2019, pp. 542–545.
[3] J. Civit-Masot, et al., Dual machine-learning system to aid glaucoma diagnosis using disc and cup feature extraction, IEEE Access 8 (2020) 127519–127529.
[4] J. Civit-Masot, et al., Non-small cell lung cancer diagnosis aid with histopathological images using explainable deep learning techniques, Comput. Methods Programs Biomed. 226 (2022) 107108.
[5] N. Codella, et al., Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (isic), arXiv preprint, arXiv:1902.03368, 2019.
[6] J.M.R. Corral, et al., Energy efficiency in edge tpu vs. embedded gpu for computer-aided medical imaging segmentation and classification, Eng. Appl. Artif. Intell. 127 (2024) 107298.
[7] S.K. Datta, et al., Soft attention improves skin cancer classification performance, in: Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data: 4th International Workshop, iMIMIC 2021, and 1st International Workshop, TDA4MedicalData 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, in: Proceedings, vol. 4, Springer, 2021, pp. 13–23.
[8] T. Emara, et al., A modified inception-v4 for imbalanced skin cancer classification dataset, in: 2019 14th International Conference on Computer Engineering and Systems (ICCES), IEEE, 2019, pp. 28–33.
[9] E. Goceri, Classification of skin cancer using adjustable and fully convolutional capsule layers, Biomed. Signal Process. Control 85 (2023) 104949.
[10] E. Goceri, Evaluation of denoising techniques to remove speckle and Gaussian noise from dermoscopy images, Comput. Biol. Med. 152 (2023) 106474.
[11] E. Goceri, Medical image data augmentation: techniques, comparisons and interpretations, Artif. Intell. Rev. (2023) 1–45.
[12] E. Goceri, Vision transformer based classification of gliomas from histopathological images, Expert Syst. Appl. 241 (2024) 122672.
[13] C. Golz, et al., Preparing students to deal with the consequences of the workforce shortage among health professionals: a qualitative approach, BMC Med. Educ. 22 (2022) 756.
[14] K. He, et al., Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[15] Z.H. Hoo, J. Candlish, D. Teare, What is an ROC curve?, Emerg. Med. J. 34 (6) (2017) 357–359.
[16] Instituto Nacional de Estadística, Encuesta de morbilidad hospitalaria, Online, https://www.ine.es/prensa/emh_2021.pdf. (Accessed 21 June 2023), 2021.
[17] R. Kundu, et al., Pneumonia detection in chest x-ray images using an ensemble of deep learning models, PLoS ONE 16 (2021) e0256630.
[18] Z. Lan, et al., Fixcaps: an improved capsules network for diagnosis of skin cancer, IEEE Access 10 (2022) 76261–76267.
[19] F. Luna-Perejón, et al., Low-power embedded system for gait classification using neural networks, J. Low Power Electron. Appl. 10 (2020) 14.
[20] M. Mark Priebe, R. Markin, Review of anatomic pathology and diagnostic radiology quality assurance tools to reduce diagnostic discordance in cancer, Acta Sci. Cancer Biol. 3 (2019) 04.
[21] L. Muñoz-Saavedra, et al., Designing and evaluating a wearable device for affective state level classification using machine learning techniques, Expert Syst. Appl. 219 (2023) 119577.
[22] E.R. Parker, The influence of climate change on skin cancer incidence–a review of the evidence, Int. J. Women's Dermatol. 7 (2021) 17–27.
[23] A. Rezvantalab, et al., Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms, arXiv preprint, arXiv:1810.10348, 2018.
[24] M. Sandler, et al., Mobilenetv2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[25] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014.
[26] M. Sokolova, et al., A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437.
[27] M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[28] L. Wang, M. Han, X. Li, N. Zhang, H. Cheng, Review of classification methods on unbalanced data sets, IEEE Access 9 (2021) 64606–64628.
[29] World Health Organization, The world health report 2006: working together for health, 2006.