A Transfer Learning and Explainable Solution to Detect mpox from Smartphones images

Mattia Giovanni Campana[a,*], Marco Colussi[b], Franca Delmastro[a], Sergio Mascetti[b], Elena Pagani[b]

[a]*Institute for Informatics and Telematics of the National Research Council of Italy (IIT-CNR), Pisa, Italy*
[b]*Università degli Studi di Milano, Computer Science Department, Milan, Italy*

Abstract

Monkeypox (mpox) virus has become a "public health emergency of international concern" in the last few months, as declared by the World Health Organization, especially for low-income countries. A symptom of mpox infection is the appearance of rashes and skin eruptions, which can lead people to seek medical advice. A technology that might help perform a preliminary screening based on the aspect of skin lesions is the use of Machine Learning for image classification. However, to make this technology suitable on a large scale, it should be usable directly on people mobile devices, with a possible notification to a remote medical expert.

In this work, we investigate the adoption of Deep Learning to detect mpox from skin lesion images derived from smartphone cameras. The proposal leverages Transfer Learning to cope with the scarce availability of mpox image datasets. As a first step, a homogenous, unpolluted, dataset was produced by manual selection and preprocessing of available image data, publicly released for research purposes. Subsequently, we compared multiple Convolutional Neural Networks (CNNs) using a rigorous 10-fold stratified cross-validation approach and we conducted an analysis to evaluate the models' fairness toward different skin tones. The best models have been

*Corresponding author

Email addresses: m.campana@iit.cnr.it (Mattia Giovanni Campana), marco.colussi@unimi.it (Marco Colussi), f.delmastro@iit.cnr.it (Franca Delmastro), sergio.mascetti@unimi.it (Sergio Mascetti), elena.pagani@unimi.it (Elena Pagani)

then optimized through quantization for use on mobile devices; measures of classification quality, memory footprint, and processing times validated the feasibility of our proposal. The most favorable outcomes have been achieved by MobileNetV3Large, attaining an F-1 score of 0.928 in the binary task and 0.879 in the multi-class task. Furthermore, the application of quantization led to a reduction in the model size to less than one-third, while simultaneously decreasing the inference time from 0.016 to 0.014 seconds, with only a marginal loss of 0.004 in F-1 score. Additionally, the use of eXplainable AI has been investigated as a suitable instrument to both technically and clinically validate classification outcomes.

Keywords:
Deep Learning, m-health, mpox, monkeypox, transfer learning, mobile optimization

---

1. Introduction

While the whole world is still dealing with the coronavirus disease (COVID-19) and its mutations [1], the recent outbreaks of mpox[1] virus (formerly known as Monkeypox) in different western countries have raised serious concern among public health authorities [2]. The mpox is a zoonotic disease caused by an orthopoxvirus, and it is closely related with variola (i.e., the smallpox virus), cowpox, and vaccinia viruses [3]. Although it was first isolated in 1958 from laboratory monkeys, its original hosts also included squirrels, rats, and dormice [4].

Since the first human case reported in 1970 in the Democratic Republic of Congo, the spread of mpox has been always limited to Central and West Africa, infecting new hosts through close body contact, respiratory droplets, or animal bites, becoming an endemic disease in those regions. The incubation period ranges from 5 to 21 days, and the actual disease is characterized by generic symptoms such as fever, intense headache and muscle pain, while the most specific sign of mpox is related to the appearance of skin rashes and eruptions that usually begin within 1–3 days of the appearance of fever and tend to be more concentrated on the face and extremities rather than on the trunk [5].

[1]In the rest of the paper we will use Mpox with capital letter when referring to the detection class, while mpox when referring to the virus.

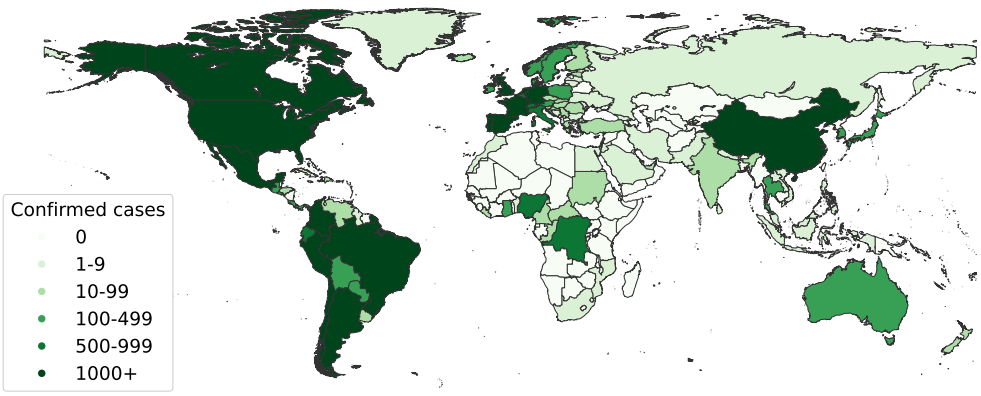


Figure 1: Geographical distribution of the recent mpox outbreak [6].

Since the middle of 2022, a continuously increasing number of cases and sustained chains of transmissions have been reported in regions without direct or immediate epidemiological links to endemic areas, including countries in Europe, North America, and Australia. On 19 September 2023, the World Health Organisation (WHO) reported a total of 90,465 laboratory confirmed cases and 663 probable cases across 115 countries [6], as shown in Figure 1. Even though mpox is usually not fatal, according to the Centers for Disease Control and Prevention (CDC), people with severely weakened immune systems, children under 1 year old, subjects with a history of eczema, and pregnant or breastfeeding women may more likely get seriously ill or even die [7].

Such rapid and widespread dissemination of the virus has raised several worries in the medical community, highlighting the need for proactive countermeasures in order to prevent another global pandemic [5]. In this regard, recent studies have emphasized how *mobile-health systems* (*m-health*), along with Artificial Intelligence (AI), can represent a game changer in containing the spread of a virus [8, 9]. In fact, using the plethora of sensors embedded in modern mobile devices and their increasingly advanced computational capabilities, smartphones and wearables can be used as low-cost, pervasive, and non-invasive tools to support the early diagnosis of new cases. For example, Rong et al. developed a smartphone-based fluorescent lateral flow immunoassay for the detection of Zika virus [10], Brangel et al. proposed the use of a mobile application to read immunochromatographic strips to detect

antibodies against Ebola [11], while more recent works used Deep Learning (DL) models to detect COVID-19 digital biomarkers in respiratory sounds collected by smartphone microphones [12, 13]

In this work, we propose a DL-based m-health solution to detect mpox from skin lesion images captured by personal smartphones. The considered use case is the following: the user takes a close picture of a skin region that the application uses to automatically detect mpox. Technically, we use Transfer Learning [14] to adapt state-of-the-art Convolutional Neural Networks (*CNNs*) models [15] to automatically identify visual features of mpox skin rashes, distinguishing the typical symptoms of the virus from skin lesions produced by other pathologies that can be easily confused also by expert eyes, including Chickenpox and Acne, at different severity levels.

Compared with previous works, this paper addresses three issues. First, the elaboration of available skin lesion images to make them homogeneous with respect to skin section focus and measure, to generate a new homogeneous dataset. In fact, existing datasets include highly heterogeneous images (*e.g.*, images of a group of people or of entire parts of body) that are unsuitable for the considered problem.

Second, the design of a mpox detection system able to run autonomously on personal mobile devices at least to provide a preliminary warning to common users, and that relies on cloud components only for model training and interaction support with a medical expert. To this end, we *optimize* the final DL model to reduce by 4× the memory footprint of our system, without negatively affecting its classification performance.

Third, the integration of *eXplainable AI* (*XAI*) methods [16] to validate the system performance in recognizing the disease from skin lesion pictures and further define a clinical validation process involving medical experts. According to the literature, XAI techniques greatly improve the general understanding of deep neural networks [17], increasing the trust in the overall system by both medical personnel and final users, thus fostering widespread adoption of such digital solutions. In fact, the target of our proposal is twofold: on the one hand, medical experts can take advantage of such a tool to speed up the diagnosis of new cases, while, on the other hand, final users can autonomously perform a preliminary screening of suspicious skin lesions that must be further investigated by their personal physicians or dermatologists.

In summary, we can highlight our contributions as follows:

- We adopted Transfer Learning and fine-tuned 5 state-of-the-art Deep Learning models to detect mpox from skin lesion images.
- We performed an extensive evaluation of the considered solutions through a series of experiments involving the use of a 10-fold cross-validation technique, and provided an additional analysis aimed at verifying the absence of possible bias towards specific skin tones.
- We optimize the best model to be able to perform all the data processing and classification directly on mobile devices, compatibly with the typical memory constraints of commercial smartphones.
- We use XAI techniques to validate our model's predictions.
- We publicly release all the materials produced in this work, including a curated selection of data called *Mpox Close Skin Images* (*MCSI*) that is composed of 400 skin images already pre-processed in order to show homogeneous characteristics, which are also perfectly balanced over 4 different classes: Mpox, Chickenpox, Acne, and Healthy.

The remainder of the paper is organized as follows. Section 2 presents the related work regarding the use of Deep Learning in medical image analysis, including preliminary works recently proposed in the literature for the automatic detection of mpox through image processing. In Section 3, we describe in detail our mpox detection system for mobile devices. Section 4 outlines the experimental setup we adopted to evaluate the classification performance of the considered DL models, and discusses the obtained results. In Sections 5 and 6 we detail the use of XAI techniques and the mobile-oriented optimization. Lastly, in Section 7 we draw our conclusions and present some directions for future work.

2. Related Work

This section first briefly introduces the state of the art in the field of CNN for medical image analysis and in particular in the field of dermatology. Then, it analyzes the existing datasets of mpox skin lesions and the mpox classification techniques.

2.1. Convolutional Neural Networks in medical image analysis

Among the different deep neural networks, Convolutional Neural Networks (CNNs) represent one of the most effective architectures for applications dealing with image data [18]. CNNs can automatically extract relevant features from raw input images by using a series of convolutional, nonlinear, and pooling layers. Thanks to this characteristic, CNNs made impressive achievements in many computer vision tasks, including image classification, object detection, image segmentation, and face recognition [15].

In recent years, CNNs have also achieved remarkable results in healthcare applications and, in particular, computer-aided diagnosis. For example, Majumdara et al. [19] proposed an ensemble of 3 pre-trained CNN models, namely, GoogleNet [20], VGG11 [21], and MobileNetV3Small [22] for the detection of breast cancer in histopathological images, obtaining a classification accuracy of 99.16% and 96.95% with two benchmark datasets. Kumar et al. [23] present a custom CNN model to detect malaria parasites in blood cell images. Despite the small size of the proposed network (i.e., only 4 convolutional and pooling layers, followed by 2 fully connected layers for classification), the authors were able to obtain an accuracy score of 96.62%. Other solutions use CNN as shared feature extractors in multitask models to classify images and, at the same time, localize specific elements of the medical image [24]. In the last two years, CNN models have also been adopted in different technological solutions aiming at containing the spread of the COVID-19 pandemic, including: systems to monitor social distancing and the use of face masks in public places [25, 26]; to automatically analyze blood samples [27], chest X-Ray and Computerized Tomography (CT) images [28]; and to fast screening the population by analyzing respiratory sounds collected from mobile devices, represented as spectrogram images [12, 29].

Dermatology is another field of application where the use of CNNs is increasingly investigated [30]. Among others, Shetty et al. compare the performance of several shallow classifiers (e.g., Random Forest and Support Vector Machines) with a custom CNN in classifying dermoscopic images of different types of skin lesions for skin cancer detection [31]. The paper shows that, using a dataset of 700 images and data augmentation techniques, CNN overcome the best classifiers by 7%, scoring an overall accuracy of 95.18%. Roy et al. [32] explore several segmentation approaches to detect different skin diseases (e.g., candidiasis and cellulitis). Finally, Kassem et al. [33] reports 94.92% accuracy by using transfer learning and a pre-trained GoogleNet to detect melanoma among 8 different classes of skin lesions.

Table 1: Mpox datasets

Name	Images	Classes	Available
MSLD [34]	228	2	Yes [2]
MDS22 [35]	161	4	No
MSID [36]	770	4	Yes [3]
RMSD [37]	2056	2	Yes [4]
DM [38]	117	2	Yes [5]
MPXV [39]	139, 198	2	No
MCSI (ours)	400	4	Yes

2.2. Datasets of skin lesions for mpox detection

Since one of the most common symptoms of mpox is the appearance of skin rashes and lesions, the analysis of skin images is a promising solution for the early detection of this novel global outbreak of the virus. Thus, an annotated dataset of images is required to train the model.

The existing available datasets include images of skin lesions caused by mpox as well as images in other classes, for example, images of the skin without lesions or with lesions caused by other diseases. The six datasets proposed in the literature are summarized in Table 1.

Ali et al. [34] present the *Monkeypox Skin Lesion Dataset* (*MSLD*)[2] containing 228 skin lesion images collected from different sources on the Internet and divided into two classes: *Mpox* cases and a generic *Others* class, which includes skin lesions caused by other diseases (e.g., Chickenpox and Measles), but also samples without evident lesions. Ahsan et al. [35] provide the *Monkeypox-dataset-2022* (*MDS22*), which includes a total of 161 images of Mpox, Chickenpox, Measles and skin without any lesions labeled as *Healthy*. However, at the time of writing, *MDS22* is no longer accessible. The third dataset, called *Monkeypox Skin Images Dataset* (*MSID*) [36] includes 770 images divided in the same 4 classes as *MDS22*[3]. The fourth dataset, called *Roboflow Monkeypox Skin Dataset* (*RMSD*) [37] includes 2056 images divided into 2 classes as *MSLD*. The positive class includes augmented images

[2] `https://www.kaggle.com/datasets/nafin59/monkeypox-skin-lesion-dataset`

[3] `https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset`

of mpox, while for the negative case, different images were web scraped for different pathologies such as Lyme, Drug Rash, Pityriasis Rosea Rash, and Ring Worm[4]. The Data_monkeypox (DM) [38] is another dataset available on Kaggle [5] and includes a total of 117 pictures collected from web sources. It comprises 45 images labeled as mpox cases, along with 74 images labeled as non-mpox. The latter category encompasses both samples without skin lesions and instances of other pathologies such as scarlet fever and roseola.

Finally, the $MPXV$ dataset introduced by Thieme et al. [39] stands out as the largest dataset employed in the existing literature for the task of mpox detection, featuring a total of 139,129 images. This extensive dataset encompasses 676 images linked to mpox cases, sourced from various outlets including scientific literature, encyclopedia entries, news articles, and social media. A significant portion of these mpox-related images originated from a prospective study conducted in collaboration with the Stanford University Medical Center. Additionally, the dataset includes 138,522 images depicting non-mpox skin lesions, drawn from publicly accessible dermatological repositories and the institutional skin cancer lesion dataset known as Esteva [40]. This last dataset and that derived from the clinical study are not publicly available due to privacy reasons.

Unfortunately, existing public datasets have severe limitations. First, they contain very heterogeneous pictures in terms of both resolution and subjects, including histopathology images, whole-body parts, and full-body images. For example, one image in MSID represents a group of three people (`normal184`), others represent people watching in the mirror (`normal198`) or using skin care product (`normal188`), while others represent full body parts (`chickenpox15` or `monkeypox46`). Similarly, RMSD contains web-scraped images, that in some cases represent monkeys (`images109`, `images224`, `Monkeypox_1`), a collage of both normal and positive samples (`images247`), or images of hospital buildings (`images17`, `images49`). In other cases, images are present in the test folder multiple times with different processing or extracted from different sources (`image9`, `1_MONKEYPOX-BOY`, `image54`). Also, Altun et al. [37] take into account skin rashes that are clearly different from those caused by mpox, like those caused by Lyme disease.

While these datasets could possibly be useful for the training of a model

[4] `https://app.roboflow.com/ds/uHWnw424Sk?key=w8YJKfcD2i`

[5] `https://www.kaggle.com/datasets/ahmadnasayrah/data-monkeypox`

aimed at automatically classifying web-scraped images, we argue that they are unsuitable for our use case in which the user takes a close picture of the skin and manually crops it (if needed) so that it only contains the skin rash and not other visual features. Image heterogeneity may jeopardize the ability of ML models to perform proper classification, due to the presence of "distracting" irrelevant features [41, 42].

Another limitation is that images in some classes are under-represented. For example, MSLD contains only 17 images in the measles class. This is due to the fact that, for some diseases, there are few images that are public and suitable (*e.g.*, with a sufficient resolution). However, dealing with such severe imbalanced data poses a challenge for machine learning models, which typically leads to poor predictive performance (especially for the minority class) due to the lack of an equal distribution of training data samples over the different classes. Although such a problem is usually addressed by oversampling minority class examples or by undersampling the majority class [43], such an approach is unfeasible for small datasets such as those publicly available for mpox. Furthermore, the existing datasets do not include a class of bacterial skin infections (*e.g.*, acne) that, according to the WHO, should be considered in the clinical differential diagnosis of mpox [44].

Unlike existing data sets, in this paper we present the *MCSI* (Mpox Close Skin Images) dataset, which includes 400 homogeneous skin images equally distributed in four classes (`Mpox`, `Chickenpox`, `Acne`, and `Healthy`). *MCSI* has been collected by merging other public datasets, and it only includes close skin pictures, as those produced by users taking photos of their own skin lesions from a short distance. The details related to both data collection and elaboration are described in Section 4.1.

2.3. Classification techniques supporting mpox detection

Three papers propose binary classification techniques trained and evaluated on *MSLD*. In particular, Ali et al. [34] compare the performance of 3 popular CNN architectures, namely, *VGG16* [21], *ResNet50* [45], and *InceptionV3* [46], along with an ensemble of the three with majority voting. The authors note that *ResNet50* is able to score the best accuracy, 82.96%, while the ensemble solution shows a lower performance than the single models. Sahin et al. [47] use transfer learning and fine-tuning for different CNNs, with the aim of finding the best model to implement on a mobile device. The experiments show that *MobileNetvV2* [48] obtains the best accuracy

(91.11%). Finally, Alcalá-Rmz et al. [49] present an alternative solution based on GoogleNet that yields 97.08% accuracy.

Two other recent studies, namely Jaradat et al.[38] and Thieme et al.[39], delves into the application of deep learning (DL) models for the detection of mpox in skin images. However, their focus is exclusively on the binary classification problem, distinguishing between mpox and non-mpox cases. The former study uses the *DM* dataset, yielding an interesting F-1 score of 0.94 when MobileNetV2 is employed as the prediction model. The latter study is based on the more extensive *MPXV* dataset. They report a sensitivity of 0.83 and a specificity of 0.965 using the ResNet34 model. Their study is noteworthy for the model performance evaluation across various skin tones and body regions.

Other three papers present techniques trained on *MDS22* or *MSID* that hence have the opportunity to distinguish among the four classes defined in these datasets. Sitaula and Shahi [50] present a classifier based on an ensemble of Xception and DenseNet-169 with majority voting. In this case, the technique is trained and evaluated on *MDS22* and it achieves an accuracy of 87.13%. The paper by Ahsan et al. [51] also proposes a technique trained and evaluated on *MDS22*. Specifically, it proposes two studies using a pre-trained version of VGG16: the former classifies mpox versus chickenpox, obtaining 83% accuracy on 18 test images; while the latter obtains 78% accuracy when comparing mpox with all other cases. Abdelhamid et al. [52] evaluates optimization algorithms to find the optimal hyperparameters of a deep neural network for image classification in the *MSID* dataset. The solution reports that GoogleNet yields an accuracy of 98.80%.

Finally, [37] proposes a classification technique trained on RMSD. It compares different state-of-the-art models adopting transfer learning in the task of binary classification, obtaining the best results with MobileNetV3 small, with an accuracy of 96.8% and an F-1 score of 0.978.

Unfortunately, the presented works suffer from a main limitation due to the characteristics of the considered datasets. In addition, they also present some issues from the methodological point of view. For instance, [47] and [37] are evaluated with a fixed random split into train-validation-test sets and do not adopt cross-validation. This approach can lead to overfitting the model on specific dataset partitions, potentially overestimating its generalization capabilities and performance when deployed in real application environments, especially when the training is based on a limited dataset [53]. In other cases, the papers do not clearly specify the evaluation methodology (e.g., [52]).

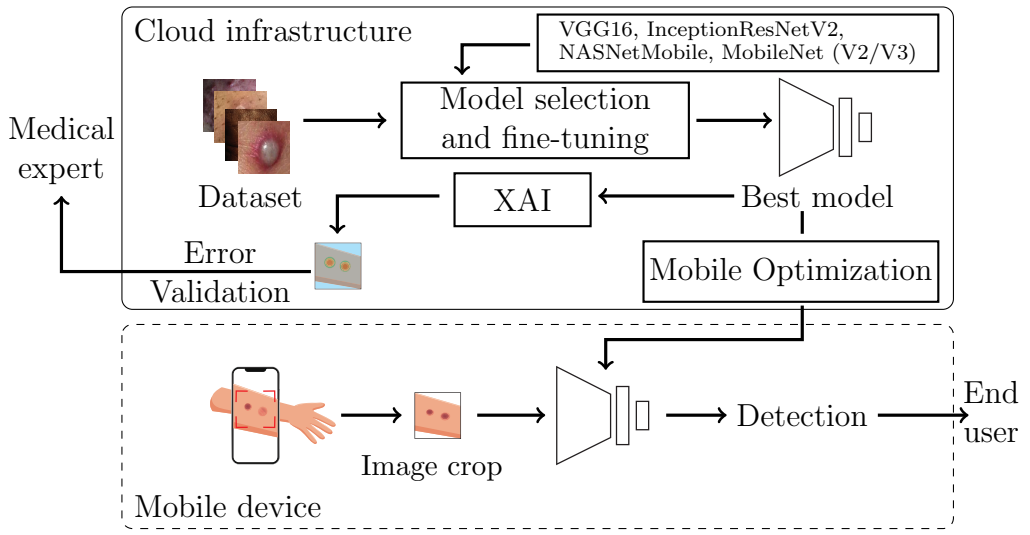


Figure 2: Scheme of the mpox diagnosis infrastructure considered in this work.

Finally, none of the previous papers present solutions optimized for mobile devices.

In this paper, we train the models with a dataset specifically designed for the addressed problem, such as *MCSI*. We provide a thorough and reproducible comparison of several state-of-the-art CNNs, and we investigate the obtained results through the use of Grad-CAM [54], a popular eXplainable-AI technique.

3. Mpox detection system for mobile devices

Figure 2 shows the high-level architecture of the proposed framework to detect mpox from skin lesion images collected from mobile devices. The whole process can be summarized in two main stages. In the first stage, we rely on the Transfer Learning approach to adapt a set of pre-trained CNNs to our application scenario, using MCSI to fine-tune their parameters. The rationale for using existing CNNs is that they have been proven to be effective in addressing classification problems in the medical imaging domain [55]. However, one limitation of the CNNs is that they need to be trained on a large amount of data (e.g., Imagenet[56]) and this is extremely expensive in terms of computational time and resources. We address this limitation by using existing CNNs for which pre-trained weights are available.After the

experimental comparison of the models' performance, we identify the best model for our mpox detection system, which is then optimized for mobile devices. Since the fine-tuning process includes complex and time-consuming operations, it is executed on a remote server.

The second stage involves the use of the optimized best-performing model to identify new mpox cases, performing the whole data processing on user devices: a new picture is firstly acquired from the device camera and then cropped in order to contain the target skin lesion. The resulting image is then used as input to the deep learning model that generates the classification. Moreover, a XAI module is used to both explain and, to some extent, validate the model's prediction, highlighting the most important sections of the input image that led to the model output.

In the following, we describe in detail the main building blocks of the proposed solution.

3.1. Model selection and fine-tuning

The framework relies on transfer learning to adapt a set of pre-trained CNNs to our application scenario, thus reducing the dependence on a large number of training data to build up the target learners [14].

We consider the following 5 CNNs that represent the state-of-the-art on image classification:

- **VGG-16** [21], composed by 5 consecutive blocks of convolutional layers for features extraction, followed by 3 fully-connected layers for classification. Convolutional layers use 3×3 kernels with a stride of 1 and padding of 1 to ensure that each activation map retains the same spatial dimensions as the previous layer. A Rectified Linear Unit (ReLU) activation is performed right after each convolution, and a max pooling operation is used at the end of each block to reduce the spatial dimension. Max pooling layers use 2×2 kernels with a stride of 2 and no padding to ensure that each spatial dimension of the activation map from the previous layer is halved. Finally, two fully-connected layers with 4096 ReLU activated units are used before a final 1000 fully-connected softmax layer.

- **Inception-Resnet-V2** [57] represents a combination of two popular architectures: GoogleNet [20] and ResNet [45]. While the former is based on the concept of "Network in Network" [58], where a large

number of convolutional kernels constitute a very deep architecture to increase the network's generalization, the latter introduced the idea of directly bypassing the input information to the output, thus changing the direct learning target value into learning the residual value between the input and the output. Inception-Resnet-v2 combines the two concepts, using residual connections instead of filter concatenation, to both accelerate the training and improve the performance.

- **NASNetMobile** [59], a simplified version of Neural Architecture Search Network (NASNet) proposed by GoogleBrain, which is a scalable CNN architecture consisting of basic building blocks, called *cells*, that are optimized using reinforcement learning. A cell consists of only a few operations, including both convolutions and pooling, which are repeated multiple times according to the required capacity of the network. The mobile version consists of 12 cells, with a total of 5.3 million parameters.

- **MobileNetV3** [60], a CNN-based architecture especially tuned to best performing on smartphone CPUs through a hardware-aware Network Architecture Search (NAS), combining a series of building-blocks developed by previous models: the depth-wise separable convolutions as an efficient replacement for traditional convolution layers from MobileNetV1 [61], the linear bottleneck and inverted residual structure introduced by MobileNetV2 [62], and the lightweight attention modules used in MnasNet [63]. The model comes in two flavors - which both are tested in this work - that are **MobileNetV3-Large** and **MobileNetV3-Small**, which are targeted for high and low resource use cases, respectively.

For all the aforementioned architectures, we take into account their instances pre-trained with ImageNet [56], a large-scale dataset of 3.2 million images and 1000 different labels, which is commonly used to train CNNs in the image classification domain [64]. Note that ImageNet does not contain labels related to the specific problem domain considered in this paper. To mitigate this domain shift, we employ Transfer Learning replacing the last fully-connected layers of the network with a novel set of classification layers fine-tuned with MCSI dataset.

We then validate the considered models through the use of a 10-fold cross-validation procedure and *Hyperband*, a broadly used hyperparameter

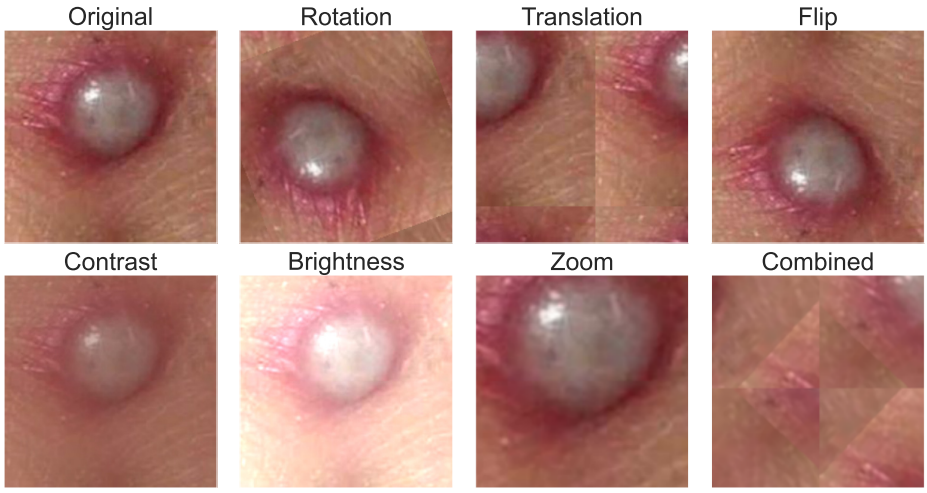


Figure 3: Example of data augmentations used in our experiments.

selection algorithm for deep neural networks, which is able to speed up the random search over the parameter spaces through adaptive resource allocation and early-stopping [65]. In other words, Hyperband uses a combination of small random searches aimed at partitioning the original search space into smaller sub-spaces. Once a search iteration is completed, the most promising sub-spaces (i.e., those that allowed the network to obtain the best results) are further explored until a performance plateau is reached or the iterations budget (i.e., the maximum number of iterations) has been exhausted. In this process, we exclusively fine-tune the final classification layers, which drastically decreases the number of parameters to be trained and, consequently, the amount of data required for the training. Furthermore, to mitigate the risk of model overfitting during the training phase, we employ standard techniques, including *Early Stopping* and *Dropout*.

Furthermore, during the evaluation process, we investigate the feasibility of using data augmentation in our application scenario to possibly improve the performance of the fine-tuned models. Specifically, we employ the 6 standard image augmentation techniques [66] shown in Figure 3: (i) *Rotation*, which changes the image angle, simulating different orientations; (ii) *Translation*, simulating different positions of the skin rash inside a specific picture; (iii) *Flip*, which mirrors the image, thus simulating different type of pimples; (iv-v) *Contrast* and *Brightness*, simulating different settings in the amount and intensity of light; and, finally, (vi) *Zoom*, scaling the image to simu-

late variations in the distance between the skin lesion and the smartphone camera.

Data augmentation is not applied to the test and validation sets to avoid introducing bias in the models' evaluation. We include the parameters that affect the augmentation factors (e.g., rotation angle or zoom level) into the tuning phase to identify the set of values that lead to the best classification performance for our application scenario.

3.2. CNN optimization for mobile devices

Our main goal is the definition of a mpox detection system that can be entirely executed on mobile devices. However, neural networks are both computationally and memory intensive. While modern smartphones are equipped with increasingly powerful hardware (e.g., multicore CPUs and, in some cases, dedicated GPUs) that allows performing the inference phase in just a few milliseconds, neural models' size still represents a challenge, making it difficult to deploy them on embedded systems with limited memory resources.

To cope with this issue, several techniques have been recently proposed to reduce the memory footprint of deep learning models, including *pruning*, where redundant connections among hidden units are removed, or *weight clustering*, which consists in replacing similar weights in a layer with a representative value found by clustering algorithms [67, 68]. *Quantization* is another practical and broadly used technique to optimize deep learning models by simply lowering the operations' precision from 32-bit floats to 16-bit floats or even 8-bit integers. Despite its simplicity, it is generally effective in reducing the overall model's size by $4\times$ at least, with little or no degradation in terms of accuracy [69]. Furthermore, while other approaches must be used during the training phase, quantization can be applied to the final fine-tuned model yield by transfer learning.

3.3. Explaining the model's predictions

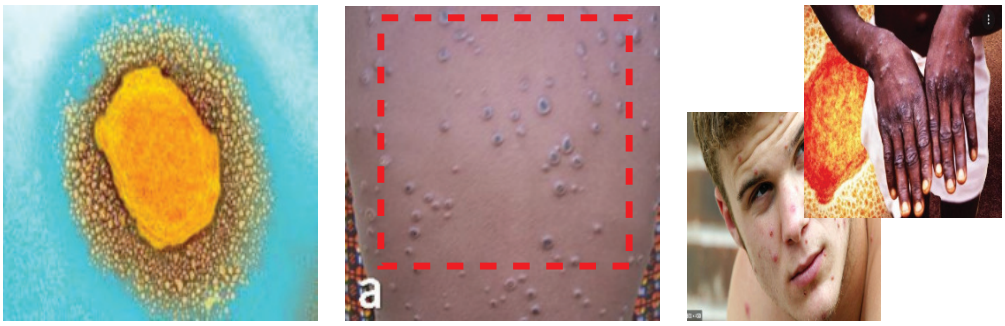
Deep learning models including CNNs are weak in explaining their inference process and final predictions, thus being typically considered as a black-box. This characteristic is not suitable for many real-world applications, and especially for the health sector, in which explainability and transparency are essential not just for researchers and developers to validate their models, but also for the users who can be directly affected by AI decisions.

For this reason, increasing attention has recently been paid to eXplainable AI (XAI) techniques with the aim of making AI models more transparent, understandable, and interpretable, so as to increase trust in their predictions. Different XAI approaches have been recently proposed for deep learning models, based on the characteristics of specific architectures [16]. According to Ibrahim et al. [70], XAI techniques for CNNs can be categorized as *decision models* and *architecture models*. While the former solutions aim at identifying the parts of an image that mostly contributed to the network decision, the latter explore the network internals, analyzing the mechanism of both hidden layers and neurons.

Given its simplicity in both implementation and interpretability, for our mpox detection system, we decided to use Grad-CAM [71] as XAI approach, one of the most popular decision models used in medical imaging [72, 73]. Grad-CAM is defined as an importance attribution feature algorithm that generates a visual explanation for class-discriminative prediction. Specifically, it captures the features that positively influence the prediction of a given class, by computing its gradient and then propagating it back to the last convolutional layer to finally generate a heatmap that visually represents the most relevant part of the input image that has led the model to that prediction. As a preliminary stage, this approach represents a useful tool to validate the ability of the considered fine-tuned deep models in correctly detecting mpox. Then, after a thorough clinical validation performed by experts with a larger amount of data, such a XAI technique might be also implemented on the mobile device of the final user to support the prescreening of suspicious skin lesions.

4. Experimental evaluation

In this section, we present the experimental evaluation performed to identify the best DL model. We first present the *MCSI* dataset. Then, we describe in detail the evaluation protocol and metrics adopted to measure the classification performances of the fine-tuned CNN models. Finally, we discuss the obtained results. The source code and data are publicly available on dedicated Zenodo repositories [74, 75], while the cross-validation folds are completely reproducible by the provided code.



(a) No skin sample (b) Cropped (c) Whole body parts

Figure 4: Examples of the criteria applied during the dataset creation.

4.1. The Mpox Close Skin Images dataset

The Mpox Close Skin Images (*MCSI*) dataset has been created according to three design principles. First, the dataset only includes close skin images with or without skin lesions, as these are representative of the pictures that can be collected by the users in the considered use case. Second, *MCSI* contains images of skin lesions caused by diseases that, according to the WHO, should be considered in the mpox clinical differential diagnosis [44]. In particular, we consider one class for chickenpox rash and one for acne, which is a common skin condition caused by bacterial skin infections. Third, the number of samples is balanced among the different classes, to avoid bias.

Specifically, *MCSI* includes: (1) images of **Mpox** cases collected by Ali et al. [34] by web scraping news portals, publicly available case reports, and websites; (2) pictures of **Chickenpox** lesions available on the Hardin Library for the Health Sciences of the University of Iowa[6], (3) samples of **Acne** at different severity levels, collected by Wu et al. [76] and freely available on Github[7], and (4) samples of skin without evident lesions, named as **Healthy**, available in the dataset collected by Muñoz-Saavedra et al. [77].

In order to create MCSI dataset we followed a two-steps procedure: first, we excluded images where no skin is visible (as in Figure 4a). Then, for the remaining images, we selected the larger square area that contains the skin and no background (see example in Figure 4b). The area is discarded if its sides are less than 224 pixels long. This is due to the fact that some original

[6] `http://hardinmd.lib.uiowa.edu/chickenpox.html`

[7] `https://github.com/xpwu95/LDL`

Figure 5: Sample images from the collected dataset for each of the 4 considered classes: *Mpox*, *Chickenpox*, *Acne*, and *Healthy*.

images contain whole body parts (as in the examples shown in Figure 4c) and hence the selected area can result in having a low resolution.

Currently MCSI dataset labels are derived from those available online and no verification has been conducted by expert medical practitioners. However, we intend to verify the validity of the annotations in MCSI with the collaboration of medical experts as part of our future work.

The resulting dataset comprises a total of 100 images for each of the 4 designated categories. Figure 5 provides a representative selection of images from our dataset, showcasing examples from each category.

4.2. Evaluation protocol and metrics

The evaluation protocol is based on the following: we decided to rely on *10-fold stratified cross-validation* to avoid biasing the results based on specific train/validation/test splits of the dataset. The procedure can be summarized as follows. Firstly, we partition the dataset into 10 folds, ensuring that all the considered classes of images are equally represented in each fold. For each of the 10 cross-validation iterations, one fold is selected as the *test set*, while the remaining 9 represent the *development set* that is further divided into stratified non-overlapping *train* (75%) and *validation* (25%). We apply data augmentation at run-time, only on the training sets. Then, a hyperparameters tuning process (Section 4.3) is used by training models on the train set and testing them on the validation set. The model yielding the best performance is then tested on the test set, providing the performance for that iteration.

We measure the average performance of the fine-tuned models obtained during the 10-fold cross-validation by using the different base models as backbone for features extraction, and a set of fully-connected layers are trained from scratch for classification. We consider the following standard classification metrics: *Accuracy*, which is the percentage of correct predictions; *Sensitivity*, which represents the true positive rate; *Specificity*, that indicates the true negative rate; and *F-1 Score*, which is the harmonic mean of Precision and Sensitivity.

We perform the whole process for two different classification settings: binary and multiclass. In the former, we evaluate the models' ability to identify mpox cases without distinguishing the other classes, which are merged into a single "other" class. Since in this setting the training data are unbalanced, we replace the standard F-1 Score with its micro average in order to avoid biasing the results towards the majority class (i.e., "other"). By contrast, in the latter setting, the models learn to distinguish all the four classes available in MCSI.

Furthermore, we conduct a statistical analysis to determine the level of significance in the obtained classification results in terms of accuracy, thereby identifying the most effective model(s) for our specific application scenario. Initially, we examine the outcomes of the two classification tasks without employing data augmentation. We conduct this analysis by using *Repeated Measures Analysis of Variance (ANOVA-RM)*, a statistical method used to assess significant differences among the means of three or more dependent groups. We chose this method because our models were evaluated on the

same data folds, making the results dependent on each other. Moreover, even though ANOVA is generally robust to slight deviations from normality assumptions (especially with small sample sizes), we use the *Shapiro-Wilk* test to assess the distribution characteristics of the results. This evaluation aimed to confirm that the models' results can be approximated by a normal distribution. Since ANOVA-RM only indicates the presence or absence of a significant difference, without specifying the specific groups that differ from each other, we subsequently employ the *Tukey's Honest Significant Difference (HSD)* test, which allows us to determine the significance of performance differences between each pair of models, providing a more detailed understanding of the disparities.

Next, we perform a statistical assessment to evaluate the impact of data augmentation on each model, by employing the following procedure. The initial step involves using the Shapiro-Wilk test to determine whether the performance of the model, both with and without augmentation, follows a normal distribution. If both distributions pass the test (i.e., $p > 0.05$), we proceed to assess their homoscedasticity using *Bartlett's* test, which determines if the distributions have equal variances. However, if either distribution failed the Shapiro-Wilk test, indicating non-normality, we utilize the non-parametric *Wilcoxon's rank-sum* test as an alternative to the two-sample t-test. Finally, if the distributions exhibited homoscedasticity, we employ the standard *Independent t-test* to evaluate their statistical significance; otherwise, we use the *Corrected Independent t-test* (also known as *Welch's test*) instead.

4.3. Hyperparameters tuning

Actual performances of deep neural networks depend on several hyperparameters that must be tuned in order to find the best configuration for every application scenarios. We adopted Hyperband for fine-tuning the model and data augmentation parameters. Considering the model's parameters, we tune the *learning rate* (`LR` in the range $[1e-6, 0.001]$) and the *number of classification layers* (`N_layers` among values $\{1, 2, 3\}$). Then, for each classification layer, we tune the *number of hidden neurons* (`Dense` among the values $\{256, 512, 1024, 2048, 4096\}$) and the *dropout* rate (`Dropout` in the range $[0, 0.5]$).

Regarding the data augmentation, we explore two different types of parameters' spaces: continuous and discrete. The former is defined within $[0, 0.5]$ and governs the application of `Rotation`, `Zoom`, `Contrast`, `Brightness`,

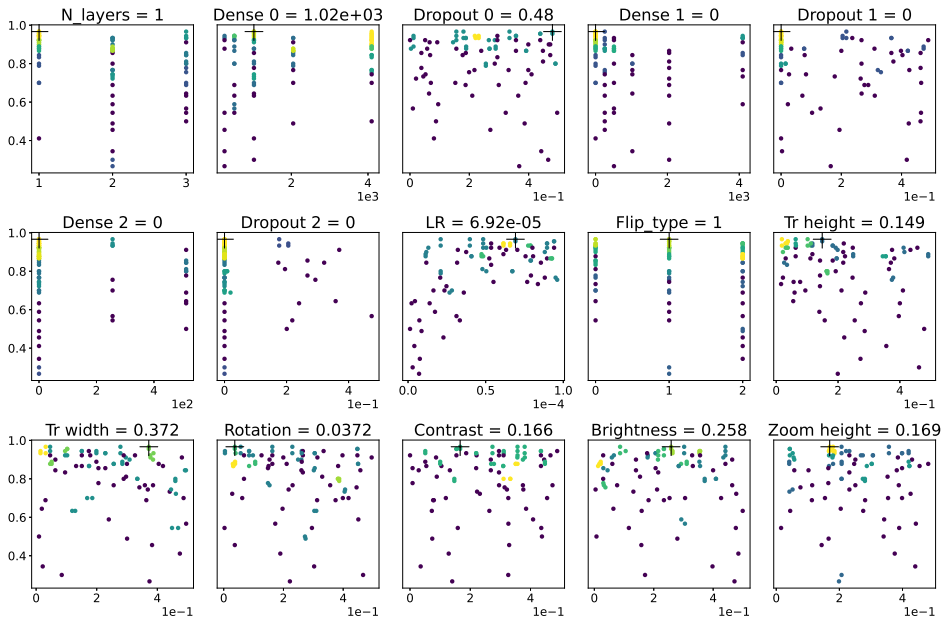


Figure 6: Explored parameters for MobileNetV3Large with augmentation (on fold 0)

`Translation` (both horizontally, `Tr-width`, and vertically, `Tr-height`), indicating the percentage in which each operation is applied on the original image. For example, the value 0.2 for `Rotation`, represents a random rotation of the image between $[-20\%, +20\%]$). The latter controls the application of `Flip type`, which may be applied in three different modalities: *Vertical* (0), *Horizontal* (1), and the combination of the two (2).

Figure 6 shows an example of the parameters space explored by Hyperband during the fine-tuning of MobileNetV3Large with data augmentation. The X-axis indicates the exploration space for a given parameter and can include a finite set of values (*e.g.*, the `N_layers`) or can be continuous in a given interval (*e.g.*, `Dropout`). Instead, Y-axis indicates the accuracy levels. In order to ease the visualization, the density of points is shown with colors (with the *viridis* color map): a single point is shown in purple while multiple overlapping points are shown in yellow. Finally, the cross symbol (+) highlights the combination of parameters that produced the best results, which is also reported on the sub-plot titles. Note that the parameters `Dense` and `Dropout` refer to the corresponding classification layer. So, for example, `Dense 1` represents the number of hidden neurons in classification layer 1.

Table 2: Binary classification performance of the considered base models, with and without data augmentation in the training phase. The performance is reported as mean and standard deviation over the 10-folds of the cross-validation.

Base model	Augmentation	Accuracy	Sensitivity	Specificity	F-1 Score
VGG16	✗	.898 (±.059)	.833 (±.106)	.710 (±.223)	.897 (±.057)
	✓	.890 (±.028)	.835 (±.027)	.730 (±.067)	.890 (±.028)
InceptionResNetV2	✗	.732 (±.051)	.568 (±.063)	.240 (±.196)	.734 (±.052)
	✓	.728 (±.068)	.544 (±.109)	.180 (±.244)	.728 (±.068)
NASNetMobile	✗	.811 (±.038)	.726 (±.061)	.550 (±.151)	.812 (±.037)
	✓	.835 (±.044)	.727 (±.080)	.510 (±.173)	.835 (±.044)
MobileNetV3Small	✗	**.930(±.041)**	.877 (±.067)	**.780(±.123)**	**.929(±.040)**
	✓	.921 (±.043)	.872 (±.062)	**.780(±.114)**	.919 (±.040)
MobileNetV3Large	✗	.930 (±.042)	.861 (±.086)	.730 (±.177)	.928 (±.040)
	✓	**.930(±.039)**	**.878(±.071)**	**.780(±.140)**	.928 (±.037)

Hence, if a classification layer does not exist (as in the case of layer 2 when `N_layers` is 2) the corresponding `Dense` and `Dropout` parameters have a value of zero.

4.4. Mpox detection performances

In this section, we present in detail the results obtained by fine-tuning the considered CNN architectures in both binary and multiclass classification settings, with and without data augmentation. We also present an analysis of their ability to correctly represent image data samples in the latent features space, thus providing additional support to the standard evaluation metrics.

4.4.1. Binary classification task

Table 2 summarizes the binary classification results of the fine-tuned models, both with and without data augmentation; the results are expressed in terms of mean and standard deviations of the considered evaluation metrics, calculated over the 10-folds of the cross-validation.

Most of the considered base models are able to reach an accuracy level above 80%. InceptionResNetV2 performs worst, thus clearly indicating that such an architecture is not able to detect mpox skin rashes from lesions produced by other pathologies. This is even clearer by observing the confusion matrix in Figure 7, noting that the model incorrectly classifies 76% of the overall Mpox samples with the original training data and 82% with data augmentation.

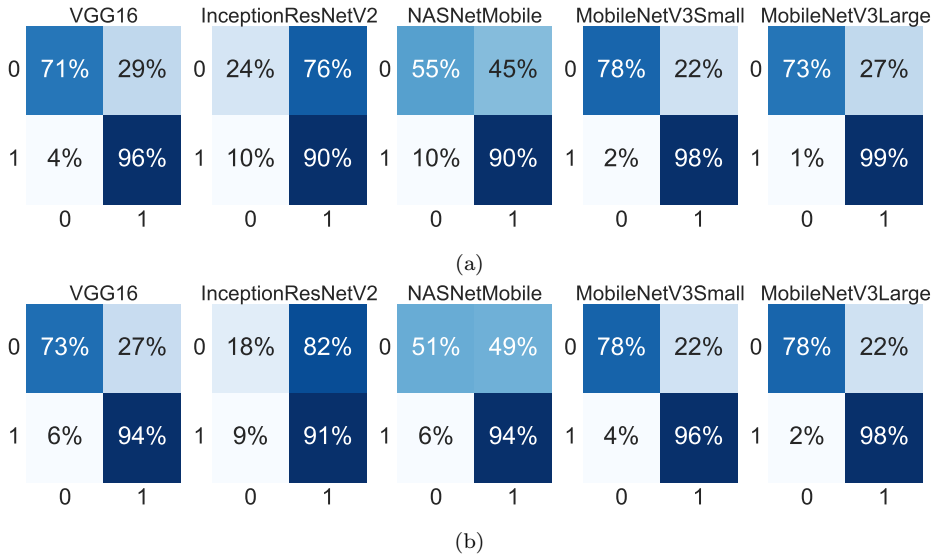


Figure 7: Confusion matrices related to the binary classification task with original training data (a) and by employing data augmentation (b). Label 0 refers to `Mpox` samples, while label 1 indicates the generic class `Others`.

NASNetMobile obtains better results than InceptionResNetV2, but its specificity score is still too low, and its misclassification rate is particularly high to be considered a valid candidate for our system. On the other hand, VGG16 performs better than the previous models. In this case, we can also note a small improvement introduced by using data augmentation, reducing the percentage of incorrectly classified mpox samples from 29% to 27%.

The two variants of MobileNetV3 obtain the best results, reaching in both cases an average accuracy level of 0.93 and with comparable results for all the considered metrics. MobileNetV3Small is able to reach the maximum value also in terms of F-1 score, overcoming by approximately 10% the performance of the larger model. In terms of misclassification rate without data augmentation, MobileNetV3Small improves MobileNetV3Large by 5%, while the larger model performs slightly better in classifying data samples labeled `Others`. On the other hand, in this case, data augmentation seems to introduce more confusion in the model predictions. In fact, while it allows MobileNetV3Large to improve its `Mpox` detection rate, at the same time, it increases the misclassification of `Others` samples for both models, reaching an error rate of 4% and 2% for MobileNetV3Small and MobileNetV3Large, respectively.

Despite MobileNetV3 achieving the highest classification score, the statistical analysis does not reveal significant differences in accuracy compared to VGG16, with a probability of $p = 0.609$. On the contrary, the analysis confirms that InceptionResNetV2 is the least performing model, exhibiting lower performance compared to the other architectures. It shows a decrease of -16.5% compared to VGG16 ($p = 0.0$), a decrease of -8% compared to NASNetMobile ($p = 0.004$), and a decrease of -19.5% compared to the two MobileNetV3 alternatives ($p = 0.0$).

Finally, regarding the utilization of data augmentation, the statistical analysis verifies that employing this technique does not significantly impact the average performance of the models, obtaining probabilities considerably higher than the significance threshold of 0.05 for all the architectures. Specifically, we observe a probability of $p = 0.625$ for VGG16, $p = 0.857$ for InceptionResNetV2, $p = 0.226$ for NASNetMobile, $p = 0.602$ for MobileNetV3Small, and no difference at all for MobileNetV3Large, obtaining a probability of $p = 1.0$.

We also conducted leave-one-out cross-validation on the best-performing model, namely MobileNetV3Large, for the binary task with and without augmentation. For this experiment, we used the same hyperparameters as in the best-performing folder after hyperparameter tuning. The results show slightly improved performance (i.e., micro F-1 Score of 0.94 and 0.93 without and with augmentation, respectively) that are due to the larger training set used in this specific evaluation approach.

4.4.2. Multiclass classification task

Table 3 summarizes the multiclass classification results of the fine-tuned models, again with and without data augmentation, over the 10-fold cross-validation. It is worth knowing that the specificity in the multiclass setting is the average of the specificity for each class. More specifically, for a given class C, we calculate the specificity of the model based on the one-vs-all approach, thus as the binary problem of distinguishing between samples belonging to C (positive samples) and samples in all other classes (negative samples). Specificity is calculated as true negative, the number of negative cases that are correctly identified as negative, divided by true negatives plus false positives, which is the number of negative cases that are incorrectly identified as positive.

Similarly to the binary results, InceptionResNetV2 and NASNetMobile show the worst performances, clearly indicating their inability to recognize

Table 3: Multiclass classification performance of the considered base models, with and without data augmentation in the training phase. The performance is reported as mean and standard deviation over the 10-folds of the cross-validation.

Base model	Augmentation	Accuracy	Sensitivity	Specificity	F-1 Score
VGG16	✗	.779 (±.052)	.779 (±.054)	.927 (±.018)	.777 (±.057)
	✓	.745 (±.059)	.744 (±.059)	.915 (±.020)	.738 (±.062)
InceptionResNetV2	✗	.396 (±.087)	.398 (±.088)	.780 (±.023)	.388 (±.084)
	✓	.301 (±.057)	.301 (±.067)	.767 (±.023)	.252 (±.078)
NASNetMobile	✗	.464 (±.073)	.464 (±.073)	.822 (±.025)	.461 (±.076)
	✓	.504 (±.103)	.505 (±.104)	.835 (±.034)	.499 (±.106)
MobileNetV3Small	✗	.846 (±.062)	.847 (±.061)	.948 (±.020)	.843 (±.065)
	✓	.859 (±.054)	.860 (±.052)	.954 (±.017)	.860 (±.049)
MobileNetV3Large	✗	**.882(±.057)**	**.881(±.055)**	**.960(±.019)**	**.879(±.058)**
	✓	.866 (±.088)	.866 (±.080)	.956 (±.029)	.863 (±.086)

the different pathologies in the images. Moreover, data augmentation further reduces the performance of InceptionResnetV2, reducing its F-1 score to 0.252, while it boosts the F-1 score of NASNetMobile to 0.499. In Figure 8 we can note in detail how these two models wrongly classify each class and, in particular, how InceptionResNetV2 tends to classify every sample as `Acne` (i.e., class 0). In contrast, VGG16 yields better results, although, similarly to InceptionResnetV2, data augmentation slightly decreases its performance.

The MobileNetV3 variants achieve the best results also in the multiclass setting. MobileNetV3Small yields slightly lower performance: -3.6% in accuracy, -3.4% and -1.2% for sensitivity and specificity, and -3.6% in terms of F-1 score. On the other hand, it benefits more from data augmentation, improving its F-1 score from 0.843 to 0.860. Quite the opposite happens for MobileNetV3Large; in fact, with data augmentation, all its indexes drop. Nevertheless, the confusion matrices clearly show how both of the MobileNetV3 variants are able to successfully identify samples in the `Mpox`, `Acne`, and `Healthy` classes (almost 98% of accuracy, both for augmented and non-augmented models), while `Chickenpox` represents the hardest class, where MobileNetV3Small scores an accuracy of 79% by augmenting the training data, and the larger variant reaches 80% and 81%, respectively with and without data augmentation.

Statistical analysis generally confirms the classification results obtained in our study. Indeed, there were no significant differences found between InceptionResNetV2 and NASNetMobile ($p = 0.188$), which both perform

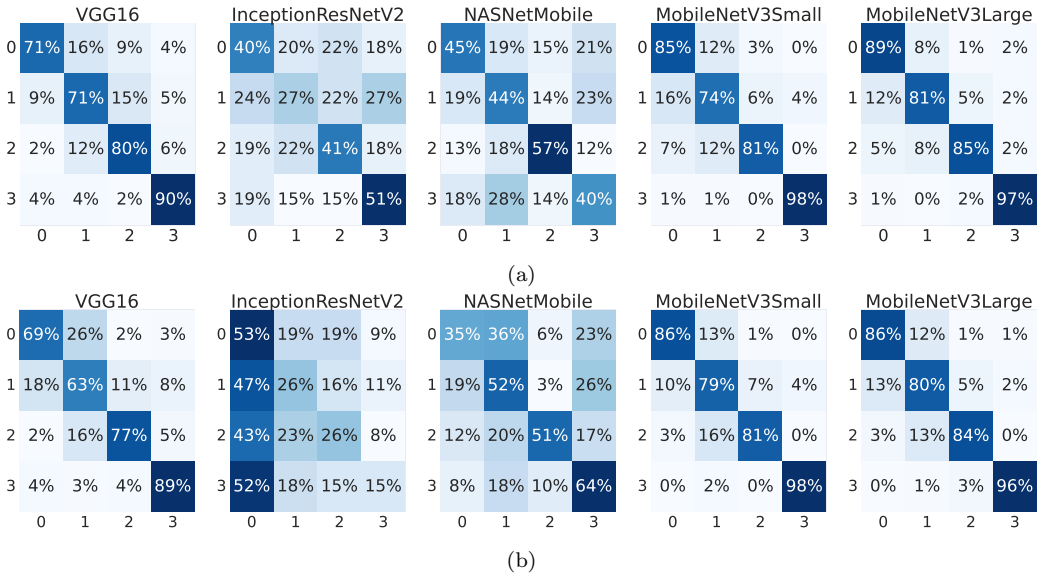


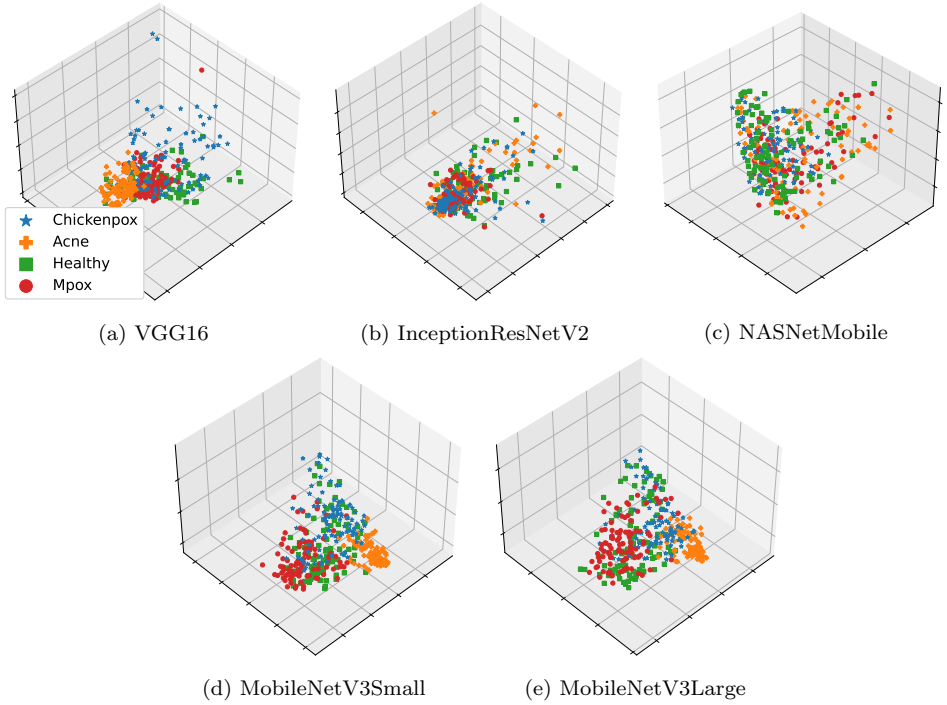
Figure 8: Confusion matrices related to the multiclass classification setting with original training data (a) and with data augmentation (b). Label 0 refers to Acne samples, label 1 indicates Chickenpox, label 2 indicates *Mpox*, while label 3 indicates the normal class.

worse than the other considered models. Furthermore, the two variations of MobileNetV3 exhibited a very high probability of $p = 0.776$, suggesting that there were no significant differences between them.

In contrast to the binary classification problem, in the multiclass setting, a noticeable difference can be observed between MobileNetV3Large and VGG16 ($p = 0.0154$), while MobileNetV3Small and VGG16 are similar with a probability of 0.2189. This difference can be attributed to the fact that in the two-sample tests among the three models, the performance of MobileNetV3Small fell between the other two. Indeed, on average, it showed a slight decrease of 3.6% in accuracy compared to its larger variant, while performing better than VGG16 by 6.5%.

Finally, in the case of data augmentation, most of the models did not show statistically significant differences. The probabilities observed were $p = 0.190$ for VGG16, $p = 0.330$ for NASNetMobile, $p = 0.551$ for MobileNetV3Small, and $p = 0.734$ for MobileNetV3Large. Only InceptionResNetV2 showed a probability below the threshold at $p = 0.012$, confirming the largest drop in performance of 6.5% in terms of accuracy.

To sum up, we can consider both the MobileNetV3 variants as the best



(a) VGG16 (b) InceptionResNetV2 (c) NASNetMobile

(d) MobileNetV3Small (e) MobileNetV3Large

Figure 9: 3-D representation of the dataset based on the deep embeddings learned by each model.

choice to detect mpox from skin lesion images, while the larger model is preferable to accurately distinguish mpox from similar diseases. Moreover, based on the statistical analysis, we can also note that data augmentation does not lead to significant performance improvements, highlighting the need for a larger amount of original training data, as well as a further investigation of more sophisticated approaches of image data augmentation.

Similarly to the binary setting, we conducted a leave-one-out cross-validation for the multiclass classification task. In this case, the results show similar or slightly improved performance (i.e., F-1 Score of 0.90 and 0.85 without and with augmentation, respectively).

4.4.3. Deep embeddings analysis

The obtained results are also supported by the analysis of the deep features (*i.e.*, embeddings) extracted by the different CNNs. Figure 9 shows how each model represents the different classes of data samples in the deep latent space, by using Principal Component Analysis (PCA) as data dimensionality

algorithm to project the embeddings onto a 3-dimensional plane.

As we can note, for both InceptionResNetV2 and NASNetMobile, it is particularly difficult to distinguish the 4 data clusters: while in the data space modeled by the former CNN, the data points are mainly concentrated in a single blob, in the latter they are distributed on a V-shaped hyperplane, where data of different classes are overlapped to each other. By contrast, the data space modeled by VGG16 makes it easier to distinguish the different classes, even though data points belonging to `Healthy` are still considerably mixed with both `Acne` and `Mpox` samples. The best deep representations are given by the two MobileNetV3 variants, where the considered classes are well-separated. In addition, it is worth noting a lower data dispersion in the MobileNetV3Small embeddings space, thus facilitating the separation of the 4 clusters and, consequently, better classification performances.

4.4.4. Skin Tone-Based Classification Fairness

It is reasonable to posit that diversity in skin tones may influence the predictive performance of DL models. Consequently, we undertook an additional investigation to assess the models' accuracy in the context of varying skin types.

Since MCSI dataset does not include information regarding the skin tone, we relied on the well-known *Fitzpatrick scale* [78] to classify the available data samples based on the skin pigment. This scale, originally devised within the dermatology field, classifies human skin color into six distinct categories, predicated on the skin's response to ultraviolet (UV) light exposure. The categories range from *Type I*, representing the palest skin that is prone to sunburning and resistant to tanning, to *Type VI*, characterizing deeply pigmented, dark brown skin that does not sunburn easily.

For the purpose of our analysis, we opted to adopt a methodology akin to that employed by Tadesse et al. [79] for categorizing the images into two distinct groups: light and dark skin tones. Specifically, researchers grouped the first four levels of the Fitzpatrick scale under the designation of *Light skin*. Conversely, the fifth and sixth levels were categorized as *Dark skin* tones.

A common approach to annotating images with Fitzpatrick labels is estimating skin tone via *Individual Typology Angle* (*ITA*), which is calculated based on statistical features of image pixels and is negatively correlated with the melanin index [80]. Following the same approach used in [81], we firstly calculated the ITA value of each data sample by using the open-source *Derm-*

Table 4: Average classification accuracy (and standard deviation) for the two types of skin tones in binary and multiclass settings.

		Binary		Multiclass	
Base model	Augmentation	Light	Dark	Light	Dark
VGG16	✗	.793 (±.118)	.886 (±.100)	.774 (±.058)	.766 (±.158)
	✓	.774 (±.052)	.899 (±.054)	.733 (±.062)	.757 (±.163)
InceptionResNetV2	✗	.586 (±.071)	.551 (±.096)	.413 (±.081)	.321 (±.149)
	✓	.556 (±.114)	.521 (±.106)	.307 (±.071)	.276 (±.118)
NASNetMobile	✗	.673 (±.088)	.785 (±.096)	.474 (±.081)	.428 (±.147)
	✓	.676 (±.105)	.786 (±.086)	.496 (±.085)	.481 (±.159)
MobileNetV3Small	✗	.839 (±.093)	.921 (±.077)	.854 (±.093)	.820 (±.131)
	✓	.851 (±.089)	.883 (±.072)	.850 (±.063)	.857 (±.111)
MobileNetV3Large	✗	.773 (±.106)	.965 (±.059)	.876 (±.061)	.868 (±.086)
	✓	.835 (±.101)	.919 (±.077)	.850 (±.113)	.862 (±.143)

ITA software [8], and then we mapped values greater than 10 as *Light skin*, while the others as *Dark skin*. At the end of this process, the resulting labels are distributed as follows: `Mpox` 57 Light and 43 Dark; `Chickenpox`, 78 Light and 22 Dark; `Acne`, 73 Light and 27 Dark; and finally, `Healthy` 69 Light and 31 Dark.

Based on this distinction between light and dark skin, we evaluated the models' performance (without retraining the models) in both binary and multiclass scenarios, accounting for the two distinct skin types. The summarized results are presented in Table 4, showing the average accuracy values and their corresponding standard deviations.

The statistical analysis (i.e., standard t-test) highlights some significant differences only in the binary classification task, showing better performance in classifying the under-represented class, that is, dark skin samples. Specifically, in the binary classification setting, MobileNetV3Large without data augmentation obtains significance of $p = 0.000881$, while VGG16 and NASNetMobile with data augmentation show significance values of $p = 0.000092$ and $p = 0.025547$, respectively. One plausible explanation for this phenomenon could be the higher contrast between skin tone and skin lesion colors in the case of dark skin samples. This contrast likely aids the DL models in accurately identifying conditions such as mpox and the other considered

[8] `https://github.com/AdamCorbinFAUPhD/derm_ita/tree/master`

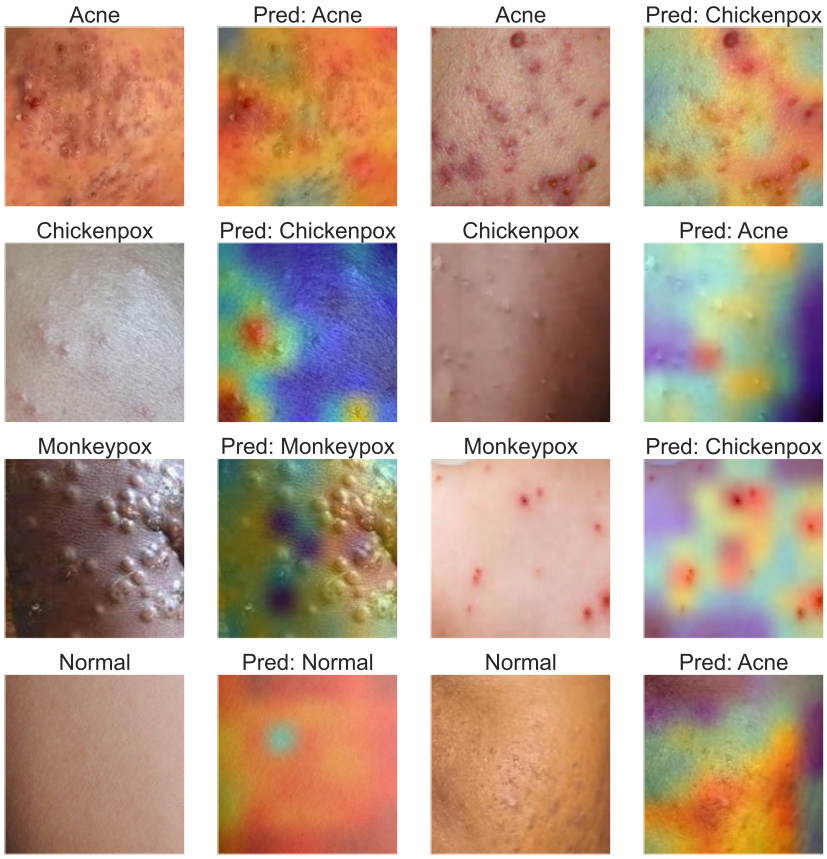


Figure 10: Examples of Grad-CAM results for each class with MobileNetV3Large, first and third columns show the input image, (Correctly and wrongly predicted respectively). Second and fourth columns show Grad-CAM explanations (for correctly and misclassified examples)

pathologies from skin images.

5. Analysis of Grad-CAM indications

Gaining a more profound comprehension of deep learning models, often perceived as "black-boxes", is important in the context of medical applications. Specifically, the field of Explainable Artificial Intelligence (XAI) has emerged with dual objectives: enhancing model interpretation and allowing additional validations of the model results.

One notable XAI technique, Grad-CAM, assumes significance in this pur-

suit by enabling the identification of salient features that drive the model's predictions. Consequently, it serves as a valuable adjunct tool for delving into the rationale underpinning the decisions made by the model.

We decided to apply Grad-CAM to the predictions provided by MobileNetV3Large as one of the best-performing models in both classification tasks. Specifically, in order to understand what features of the input images are considered relevant by the model, in Fig. 10 we reported 8 different examples of explanations, four correctly predicted, along with their class activation maps (first and second columns), and four misclassified samples, with their corresponding maps (third and fourth columns). The ground-truth label and the predicted one are indicated at the top of each image, while the heatmaps have been generated by superimposing the class activation map to the original image. While bluish areas identify less relevant features for the given class, warmer colors (e.g., orange and red) represent the most relevant ones that have led the models to provide the specified prediction.

For example, the first row represents a case of `Acne`. When the model correctly classifies the image, the relevant features are distributed across all scars and pustules, which are typical of a strong presence of acne. However, when the model misclassifies the image, the main focus of the network is on the pimples, neglecting the skin scars, causing the model to classify the image as `Chickenpox`.

Regarding the `Chickenpox` sample, when the model provides a correct prediction, its focus is only on the largest pimples, whereas when the model makes an incorrect prediction, its attention is distributed to minor skin defects in addition to the pimples, classifying the image as `Acne`.

For `Mpox`, the model is capable of correctly identifying the pathology when vesicles and crusts are formed, but it clearly fails in the early stages of the pathology, when pimples have not yet fully developed, providing a wrong prediction (i.e., `Chickenpox` in this case).

Finally, when the model correctly classifies a `Healthy` image, as we can expect, the importance of the feature is evenly distributed throughout the image without focusing on specific elements. On the contrary, when the model misclassifies a healthy sample, it is because it gives great relevance to hair and skin damage, classifying the image as `Acne`.

The model's visual attention analysis shows that MobileNetV3Large effectively identifies reasonable features for each class. The model's misclassifications are justifiable due to the similarity of the different classes, and, despite these errors, the model's overall ability to identify relevant features

Table 5: Model sizes and classification performance with mobile optimization.

Task	Base model	Quant.	Size (MB)	Accuracy	Sensitivity	Specificity	F-1 Score
binary	VGG16	✗	268.44	.894 (±.049)	.841 (±.075)	.740 (±.143)	.851 (±.070)
		✓	67.22	.894 (±.053)	.841 (±.077)	.740 (±.143)	.851 (±.072)
	InceptionResNetV2	✗	350.53	.735 (±.056)	.562 (±.110)	.220 (±.266)	.533 (±.130)
		✓	89.42	.702 (±.113)	.585 (±.105)	.350 (±.310)	.548 (±.127)
	NASNetMobile	✗	336.37	.830 (±.036)	.765 (±.032)	.640 (±.070)	.769 (±.036)
		✓	85.03	.738 (±.060)	.750 (±.065)	.780 (±.123)	.702 (±.058)
	MobileNetV3Small	✗	211.05	.932 (±.043)	.883 (±.070)	.790 (±.129)	.902 (±.064)
		✓	53.01	.915 (±.046)	.851 (±.067)	.730 (±.116)	.875 (±.068)
	MobileNetV3Large	✗	382.93	.928 (±.042)	.884 (±.090)	.800 (±.200)	.891 (±.074)
		✓	96.17	.923 (±.051)	.875 (±.106)	.780 (±.225)	.884 (±.089)
multiclass	VGG16	✗	318.21	.779 (±.053)	.779 (±.054)	.927 (±.018)	.777 (±.057)
		✓	79.63	.782 (±.044)	.782 (±.044)	.927 (±.015)	.779 (±.050)
	InceptionResNetV2	✗	485.12	.398 (±.088)	.398 (±.088)	.799 (±.027)	.388 (±.084)
		✓	122.48	.308 (±.064)	.306 (±.065)	.769 (±.022)	.243 (±.075)
	NASNetMobile	✗	259.23	.470 (±.074)	.470 (±.074)	.822 (±.026)	.467 (±.076)
		✓	65.51	.471 (±.095)	.471 (±.095)	.823 (±.034)	.449 (±.102)
	MobileNetV3Small	✗	225.77	.847 (±.061)	.847 (±.055)	.949 (±.014)	.843 (±.065)
		✓	55.62	.833 (±.066)	.833 (±.066)	.944 (±.020)	.831 (±.066)
	MobileNetV3Large	✗	278.73	.881 (±.055)	.881 (±.055)	.962 (±.018)	.879 (±.058)
		✓	69.98	.880 (±.046)	.879 (±.046)	.961 (±.014)	.875 (±.052)

highlights its potential in our specific use-case scenario, providing more reliability on the model's predictions.

6. Impact of mobile optimization

Table 5 shows the great advantage of using quantization to reduce the memory footprint of the models without requiring their retraining. As we can note, the original size of the DL models trained for mpox detection considerably varies for the different base architectures, ranging between 200 MB and almost 500 MB, which can limit their implementation on several personal mobile devices. On the other hand, by using quantization to lower the operations' precision from 32-bit floats to 16-bit floats, all the models' sizes are reduced by approximately 4 times. For example, the size of VGG16 tuned for binary classification dropped from 268.44 MB to just 67.22 MB, while the size of InceptionResNetV2 for multiple classes (i.e., the most demanding model in terms of memory) has been reduced by 74.75%, limiting its memory footprint from 485.12 MB to 122.48 MB.

Furthermore, it is important to highlight that the impact of quantization on the classification performance of the majority of the examined architectures remains relatively modest, resulting in an average reduction of no more than 1% in accuracy.

However, it is noteworthy that InceptionResNetV2 and NASNetMobile exhibit more pronounced performance penalties due to quantization. Specifically, InceptionResNetV2 experiences a decline of approximately 3% and 9% in accuracy in the binary and multiclass settings, respectively. Meanwhile, NASNetMobile's accuracy registers a noteworthy 10% reduction, albeit exclusively in the binary task. Remarkably, in the multiclass experiments, it performs nearly on par with its non-quantized counterpart. We suspect that this can be attributed to the inherent effect of quantization, which compromises the precision of both weight parameters and activation functions. Consequently, this effect is more pronounced in larger networks, such as InceptionResNetV2 and NASNetMobile. Additionally, it is worth noting that these models already exhibit relatively lower accuracy levels prior to quantization, and when this factor is coupled with quantization, it results in more substantial performance losses compared to the other models.

Besides the memory size and classification performance, we also conduct an empirical evaluation of the models' time complexity. Even though our application scenario does not require real-time predictions, fast computation represents a key requirement when dealing with mobile personal devices like smartphones. Therefore, to perform this type of experiment, we rely on the benchmark tool provided by TensorFlow Lite (TFLite) [9], the Google-released mobile library for deploying models on mobile devices, microcontrollers, and other edge devices. Specifically, we first convert our CNN models to the TFLite format; then, we deploy such models on the TFLite Android benchmark app[10] that executes each model 50 times with synthetic input to collect reliable statistics related to the inference times on a real Android smartphone. Moreover, in order to get insights on the models' performance on different hardware settings, we perform our evaluation on 2 smartphones, by using both CPU (with multithreading) and GPU for the computation: (i) a recent Google Pixel 6a released in 2022, with the latest Android 13 operating system, an Octa-Core CPU (2x2.80 GHz Cortex-X1, 2x2.25 GHz Cortex-A76,

[9] `https://www.tensorflow.org/lite`

[10] `https://www.tensorflow.org/lite/performance/measurement`

Table 6: Average inference times (in seconds) on different mobile devices, by using both CPU (4 threads) and GPU for the computation.

Task	Base model	Quant.	Google Pixel 6a CPU	Google Pixel 6a GPU	Xiaomi Mi 9T CPU	Xiaomi Mi 9T GPU
binary	VGG16	✗	.429 (±.051)	.031 (±.002)	.606 (±.013)	.245 (±.011)
		✓	.104 (±.013)	.031 (±.002)	.430 (±.021)	.245 (±.011)
	InceptionResNetV2	✗	.134 (±.012)	.057 (±.007)	.515 (±.050)	.188 (±.016)
		✓	.064 (±.005)	.057 (±.007)	.441 (±.039)	.188 (±.016)
	NASNetMobile	✗	.041 (±.014)	.023 (±.003)	.206 (±.051)	.062 (±.029)
		✓	.033 (±.005)	.023 (±.003)	.421 (±.037)	.060 (±.027)
	MobileNetV3Small	✗	.018 (±.004)	**.011** (±**.002**)	.056 (±.014)	.033 (±.010)
		✓	**.011** (±**.002**)	**.011** (±**.002**)	.104 (±.023)	**.032** (±**.010**)
	MobileNetV3Large	✗	.018 (±.010)	.013 (±.004)	.067 (±.028)	**.032** (±**.019**)
		✓	.014 (±.004)	.013 (±.003)	.140 (±.040)	**.032** (±**.020**)
multiclass	VGG16	✗	.423 (±.067)	.031 (±.002)	.612 (±.012)	.249 (±.012)
		✓	.117 (±.084)	.031 (±.002)	.196 (±.007)	.249 (±.012)
	InceptionResNetV2	✗	.139 (±.008)	.059 (±.008)	.243 (±.007)	.192 (±.020)
		✓	.065 (±.004)	.059 (±.008)	.141 (±.016)	.192 (±.020)
	NASNetMobile	✗	.036 (±.009)	.021 (±.002)	.084 (±.022)	.051 (±.021)
		✓	.031 (±.003)	.021 (±.003)	.127 (±.015)	.052 (±.021)
	MobileNetV3Small	✗	.011 (±.007)	.009 (±.003)	.028 (±.016)	**.024** (±**.015**)
		✓	**.008** (±**.003**)	.009 (±.003)	.040 (±.009)	.040 (±.009)
	MobileNetV3Large	✗	.016 (±.005)	.012 (±.001)	.047 (±.014)	.029 (±.007)
		✓	.014 (±.002)	.012 (±.001)	.062 (±.004)	.029 (±.007)

and 4x1.80 GHz Cortex-A55), and the Mali-G78 MP20 GPU; and (ii) an older Xiaomi Mi 9T, released in 2019, with Android 10, an Octa-core CPU (2x2.2 GHz Kryo 470 Gold and 6x1.8 GHz Kryo 470 Silver), and an Adreno 618 GPU.

Table 6 summarizes the average inference times (in seconds) of the considered models in the different hardware settings, both for the binary and multiclass classification tasks, highlighting in bold face the best results for each device and task. It is clear that even the largest models such as VGG16 and InceptionResNetV2 can provide a prediction in less than 0.612 seconds when deployed on modern smartphones. The benefit of using quantization can be mainly observed when the computation is based on CPU, reducing the inference time by 50% at least in some cases (e.g., VGG16 and InceptionResNetV2 with Google Pixel 6a). On the other hand, all models can be executed by the GPU in less than 0.059 seconds on Google Pixel 6a and 0.245

seconds on Xiaomi Mi 9T, thanks to its ability to parallelize all operations that are involved in a deep neural network [82].

Finally, we can also note that the CNN that performs best in terms of classification accuracy, i.e., MobileNetV3 (both Small and Large variants), is also the one with the lowest inference time. In fact, while the larger variant provides a prediction for binary and multiclass classification, respectively, in not more than 0.018 and 0.016 seconds on Google Pixel 6a and not more than 0.140 and 0.062 seconds with Xiaomi Mi 9T, MobileNetV3Small requires only not more than 0.018 and 0.011 seconds on the Google phone and not more than 0.104 and 0.040 seconds on the Xiaomi, thus proving the feasibility of efficiently performing the whole data processing and prediction tasks directly on mobile devices.

7. Conclusions and future work

The paper introduces a novel m-health system for the preliminary screening of mpox infections through pictures of skin rashes and eruptions taken with common smartphone cameras. The system is designed to be entirely executed on mobile devices and is characterized by the use of Transfer Learning to adapt state-of-the-art Convolutional Neural Network (CNN) models for image classification, mobile-oriented optimization of the models through quantization, and the use of Grad-CAM as eXplainable AI (XAI) technique for technical validation.

While the proposed solution cannot replace the expertise of a medical professional, it serves as a preliminary alert system for self-examination in at-home settings, particularly in areas with limited medical assistance and where continuous Internet connectivity is not assured. In addition, such a system can play a pivotal role for supporting the preliminary screening of large populations, alleviating the burden on medical facilities, and limiting the dissemination of the virus, aiding in the prompt identification of emerging outbreaks by detecting new cases as soon as they arise.

The paper also presents the Mpox Close Skin Images (MCSI) dataset, which contains skin images (possibly with lesions), and no other background information. Images have been manually selected and cropped from samples available in other public datasets collected from online resources in uncontrolled environments. Therefore, MCSI contains images that are homogeneous with respect to the image content (skin and lesion) but heterogeneous

with respect to other factors like skin color, lighting conditions, and acquisition camera.

We use MCSI to evaluate the classification performance of the proposed system, using both binary (Mpox vs. All) and multiclass classification tasks with a 10-fold stratified cross-validation approach. The results showed that MobileNetV3Small achieved the best performance in binary classification (0.930 of Accuracy), while MobileNetV3Large was the best model to distinguish the different classes (0.882 of Accuracy). Mobile optimization through quantization allowed us to reduce the models' sizes by 4× without significantly impacting their performance. The models have also been evaluated for their complexity in terms of execution time on commercial smartphones, and they all obtained performances under 1 second to provide the prediction, with quantization further reducing the inference time on CPUs.

Despite achieving promising results, our study has four main limitations. First, the limited number of training data. Second, the lack of other metadata information, that can help evaluate the data heterogeneity with respect to various factors like gender, race, age, and physical conditions. This is clearly relevant for ethical data collection and fair model training. Third, MCSI contains images derived from online resources that were manually selected and cropped by a skilled operator, while in the intended application the images will be self-acquired and possibly cropped by the end-user or a caregiver by following the application instructions. We cannot exclude that self-acquired images will have different properties that can impact the performance of the detection models. The fourth limitation is related to annotations' reliability, in terms of skin lesion type: MCSI derives the annotations from the existing datasets and the source of the annotations is not specified.

A possible solution to address the first three problems above is to release a prototype application implementing the proposed detection system. The application could help remotely collect new images, hence creating a larger dataset to improve the current detection model. Also, the application could easily collect additional user information, like gender and age. Another advantage of this solution is that the images would be collected by the end-users or their caregiver. In order to address the fourth limitation, but also to effectively design the proposed application and clinically validate the related results, it is essential to establish a strict collaboration with medical experts, especially dermatologists and virologists. The collaboration could also provide additional data to further investigate the algorithms performances.

In addition, from a technical point of view, future research includes in-

vestigating the Federated Learning (FL) technique in this use case scenario, which has the potential to improve the m-health system in various ways. First, FL facilitates the collaborative training of the detection model by mobile devices without the need to share users' data with a central server or other devices. Each device gathers data locally, trains a model using it, and then shares only the model updates with a central server, which aggregates and distributes them back to all the devices. This approach can address privacy concerns since sensitive health data remains under the user's control. Secondly, training the model with data from multiple devices can improve the accuracy of the model by incorporating more diverse data. This is particularly crucial for the detection of skin lesion, where the types of lesion and the color of the skin can vary significantly between various populations. Finally, FL may enable real-time updates of the detection model as new data becomes available, thus aiding the system to adapt to data changes and further enhance the model's accuracy over time.

Acknowledgment

This work was produced with the co-funding European Union - Next Generation EU, in the context of The National Recovery and Resilience Plan. The funding derives partially from Investment 1.5 Ecosystems of Innovation, "Project Tuscany Health Ecosystem (THE)", CUP: B83C22003920001 in which the authors M. G. Campana and F. Delmastro are involved, from "Project MUSA – Multilayered Urban Sustainability Action" in the Investment 1.5 Ecosystems of Innovation in which the author S. Mascetti is involved, and from the Research and Innovation Program PE00000014, "SEcurity and RIghts in the CyberSpace (SERICS)", CUP: J33C22002810001, in which the author E. Pagani is involved.

References

[1] E. Callaway, et al., Fast-spreading covid variant can elude immune responses, Nature 589 (7843) (2021) 500–501.

[2] C.-C. Lai, C.-K. Hsu, M.-Y. Yen, P.-I. Lee, W.-C. Ko, P.-R. Hsueh, Monkeypox: An emerging global threat during the covid-19 pandemic, Journal of Microbiology, Immunology and Infection 55 (5) (2022) 787–794. doi:https://doi.org/10.1016/j.jmii.2022.07.004.

[3] E. M. Bunge, B. Hoet, L. Chen, F. Lienert, H. Weidenthaler, L. R. Baer, R. Steffen, The changing epidemiology of human monkeypox—a potential threat? a systematic review, PLOS Neglected Tropical Diseases 16 (2) (2022) 1–20. doi:10.1371/journal.pntd.0010141.

[4] P. v. Magnus, E. K. Andersen, K. B. Petersen, A. Birch-Andersen, A pox-like disease in cynomolgus monkeys, Acta Pathologica Microbiologica Scandinavica 46 (2) (1959) 156–176.

[5] J. G. Rizk, G. Lippi, B. M. Henry, D. N. Forthal, Y. Rizk, Prevention and treatment of monkeypox, Drugs 82 (9) (2022) 957–963. doi:10.1007/s40265-022-01742-y.

[6] World Health Organization (WHO), 2022 mpox (monkeypox) outbreak: Global trends, https://worldhealthorg.shinyapps.io/mpx_global/, accessed: 2023-02-08.

[7] Centers for Disease Control and Prevention (CDC), About mpox, https://www.cdc.gov/poxvirus/monkeypox/about, accessed: 2023-02-08.

[8] A. Asadzadeh, L. R. Kalankesh, A scope of mobile health solutions in covid-19 pandemics, Informatics in Medicine Unlocked 23 (2021) 100558. doi:https://doi.org/10.1016/j.imu.2021.100558.

[9] V. K. Rajendran, P. Bakthavathsalam, P. L. Bergquist, A. Sunna, Smartphone technology facilitates point-of-care nucleic acid diagnosis: a beginner's guide, Critical Reviews in Clinical Laboratory Sciences 58 (2) (2021) 77–100. doi:10.1080/10408363.2020.1781779.

[10] Z. Rong, Q. Wang, N. Sun, X. Jia, K. Wang, R. Xiao, S. Wang, Smartphone-based fluorescent lateral flow immunoassay platform for highly sensitive point-of-care detection of zika virus nonstructural protein 1, Analytica Chimica Acta 1055 (2019) 140–147. doi:https://doi.org/10.1016/j.aca.2018.12.043.

[11] P. Brangel, A. Sobarzo, C. Parolo, B. S. Miller, P. D. Howes, S. Gelkop, J. J. Lutwama, J. M. Dye, R. A. McKendry, L. Lobel, M. M. Stevens, A serological point-of-care test for the detection of igg antibodies against ebola virus in human survivors, ACS Nano 12 (1) (2018) 63–73. doi:10.1021/acsnano.7b07021.

[12] J. Han, T. Xia, D. Spathis, E. Bondareva, C. Brown, J. Chauhan, T. Dang, A. Grammenos, A. Hasthanasombat, A. Floto, P. Cicuta, C. Mascolo, Sounds of covid-19: exploring realistic performance of audio-based digital testing, npj Digital Medicine 5 (1) (2022) 16. doi:10.1038/s41746-021-00553-x.

[13] M. G. Campana, A. Rovati, F. Delmastro, E. Pagani, L3-net deep audio embeddings to improve covid-19 detection from smartphone data, in: 2022 IEEE International Conference on Smart Computing (SMARTCOMP), 2022, pp. 100–107. doi:10.1109/SMARTCOMP55677.2022.00029.

[14] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109 (1) (2021) 43–76. doi:10.1109/JPROC.2020.3004555.

[15] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: Analysis, applications, and prospects, IEEE Transactions on Neural Networks and Learning Systems 33 (12) (2022) 6999–7019. doi:10.1109/TNNLS.2021.3084827.

[16] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable ai: A brief survey on history, research areas, approaches and challenges, in: J. Tang, M.-Y. Kan, D. Zhao, S. Li, H. Zan (Eds.), Natural Language Processing and Chinese Computing, Springer International Publishing, Cham, 2019, pp. 563–574.

[17] B. H. van der Velden, H. J. Kuijf, K. G. Gilhuijs, M. A. Viergever, Explainable artificial intelligence (xai) in deep learning-based medical image analysis, Medical Image Analysis 79 (2022) 102470. doi:https://doi.org/10.1016/j.media.2022.102470.

[18] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6. doi:10.1109/ICEngTechnol.2017.8308186.

[19] S. Majumdar, P. Pramanik, R. Sarkar, Gamma function based ensemble of cnn models for breast cancer detection in histopathology images, Expert Systems with Applications 213 (2023) 119022. doi:https://doi.org/10.1016/j.eswa.2022.119022.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, H. Adam, Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[23] A. Kumar, S. Sarkar, C. Pradhan, Malaria Disease Detection Using CNN Technique with SGD, RMSprop and ADAM Optimizers, Springer International Publishing, Cham, 2020, pp. 211–230. doi:10.1007/978-3-030-33966-1_11.

[24] M. Colussi, G. Civitarese, D. Ahmetovic, C. Bettini, R. Gualtierotti, F. Peyvandi, S. Mascetti, Ultrasound detection of subquadricipital recess distension, Intelligent Systems with Applications (2023) 200183.

[25] M. A. Ansari, D. K. Singh, Monitoring social distancing through human detection for preventing/reducing covid spread, International Journal of Information Technology 13 (3) (2021) 1255–1264. doi:10.1007/s41870-021-00658-2.

[26] S. Singh, U. Ahuja, M. Kumar, K. Kumar, M. Sachdeva, Face mask detection using yolov3 and faster r-cnn models: Covid-19 environment, Multimedia Tools and Applications 80 (13) (2021) 19753–19768. doi:10.1007/s11042-021-10711-8.

[27] S. A. Tuncer, H. Ayyıldız, M. Kalaycı, T. Tuncer, Scat-net: Covid-19 diagnosis with a cnn model using scattergram images, Computers in Biology and Medicine 135 (2021) 104579. doi:https://doi.org/10.1016/j.compbiomed.2021.104579.

[28] Classification of covid-19 chest x-ray and ct images using a type of dynamic cnn modification method, Computers in Biology and Medicine 134 (2021) 104425. doi:https://doi.org/10.1016/j.compbiomed.2021.104425.

[29] M. G. Campana, F. Delmastro, E. Pagani, Transfer learning for the efficient detection of covid-19 from smartphone audio data, Pervasive and Mobile Computing (2023). doi:https://doi.org/10.1016/j.pmcj.2023.101754.

[30] M. A. Kassem, K. M. Hosny, R. Damaševičius, M. M. Eltoukhy, Machine learning and deep learning methods for skin lesion classification and diagnosis: a systematic review, Diagnostics 11 (8) (2021) 1390.

[31] B. Shetty, R. Fernandes, A. P. Rodrigues, R. Chengoden, S. Bhattacharya, K. Lakshmanna, Skin lesion classification of dermoscopic images using machine learning and convolutional neural network, Scientific Reports 12 (1) (2022) 18134. doi:10.1038/s41598-022-22644-9.

[32] K. Roy, S. S. Chaudhuri, S. Ghosh, S. K. Dutta, P. Chakraborty, R. Sarkar, Skin disease detection based on different segmentation techniques, in: 2019 International Conference on Opto-Electronics and Applied Optics (Optronix), 2019, pp. 1–5. doi:10.1109/OPTRONIX.2019.8862403.

[33] M. A. Kassem, K. M. Hosny, M. M. Fouad, Skin lesions classification into eight classes for isic 2019 using deep convolutional neural network and transfer learning, IEEE Access 8 (2020) 114822–114832. doi:10.1109/ACCESS.2020.3003890.

[34] S. N. Ali, M. Ahmed, J. Paul, T. Jahan, S. Sani, N. Noor, T. Hasan, et al., Monkeypox skin lesion detection using deep learning models: A feasibility study, arXiv preprint arXiv:2207.03342 (2022).

[35] M. M. Ahsan, M. R. Uddin, S. A. Luna, Monkeypox image data collection, arXiv preprint arXiv:2206.01774 (2022).

[36] Monkeypox skin images dataset (msid). doi:10.34740/KAGGLE/DSV/3971903.

[37] M. Altun, H. Gürüler, O. Özkaraca, F. Khan, J. Khan, Y. Lee, Monkeypox detection using cnn with transfer learning, Sensors 23 (4) (2023). doi:10.3390/s23041783.

[38] A. S. Jaradat, R. E. Al Mamlook, N. Almakayeel, N. Alharbe, A. S. Almuflih, A. Nasayreh, H. Gharaibeh, M. Gharaibeh, A. Gharaibeh,

H. Bzizi, Automated monkeypox skin lesion detection using deep learning and transfer learning techniques, International Journal of Environmental Research and Public Health 20 (5) (2023) 4422.

[39] A. H. Thieme, Y. Zheng, G. Machiraju, C. Sadee, M. Mittermaier, M. Gertler, J. L. Salinas, K. Srinivasan, P. Gyawali, F. Carrillo-Perez, et al., A deep-learning algorithm to classify skin lesions from mpox virus infection, Nature medicine 29 (3) (2023) 738–747.

[40] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115–118. doi:10.1038/nature21056.

[41] A. Asilian Bidgoli, S. Rahnamayan, T. Dehkharghanian, A. Grami, H. R. Tizhoosh, Bias reduction in representation of histopathology images using deep feature selection, Scientific Reports 12 (1) (2022) 19994. doi:10.1038/s41598-022-24317-z.

[42] R. B. Parikh, S. Teeple, A. S. Navathe, Addressing Bias in Artificial Intelligence in Health Care, JAMA 322 (24) (2019) 2377–2378. doi:10.1001/jama.2019.18058.

[43] S. Tyagi, S. Mittal, Sampling approaches for imbalanced data classification problem in machine learning, in: P. K. Singh, A. K. Kar, Y. Singh, M. H. Kolekar, S. Tanwar (Eds.), Proceedings of ICRIC 2019, Springer International Publishing, Cham, 2020, pp. 209–221.

[44] World Health Organization (WHO), 2022 mpox (monkeypox) outbreak: Fact sheets, https://www.who.int/news-room/fact-sheets/detail/monkeypox, accessed: 2023-02-08.

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[47] V. H. Sahin, I. Oztel, G. Yolcu Oztel, Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application, Journal of Medical Systems 46 (11) (2022) 1–10.

[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520. doi:10.1109/CVPR.2018.00474.

[49] V. Alcalá-Rmz, K. E. Villagrana-Bañuelos, J. M. Celaya-Padilla, J. I. Galván-Tejada, H. Gamboa-Rosales, C. E. Galván-Tejada, Convolutional neural network for monkeypox detection, in: J. Bravo, S. Ochoa, J. Favela (Eds.), Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022), Springer International Publishing, Cham, 2023, pp. 89–100.

[50] C. Sitaula, T. B. Shahi, Monkeypox virus detection using pre-trained deep learning-based approaches, Journal of Medical Systems 46 (11) (2022) 1–9.

[51] M. M. Ahsan, M. R. Uddin, M. Farjana, A. N. Sakib, K. A. Momin, S. A. Luna, Image data collection and implementation of deep learning-based model in detecting monkeypox disease using modified vgg16, arXiv preprint arXiv:2206.01862 (2022).

[52] A. A. Abdelhamid, E.-S. M. El-Kenawy, N. Khodadadi, S. Mirjalili, D. S. Khafaga, A. H. Alharbi, A. Ibrahim, M. M. Eid, M. Saber, Classification of monkeypox images based on transfer learning and the al-biruni earth radius optimization algorithm, Mathematics 10 (19) (2022) 3614.

[53] A. Vabalas, E. Gowen, E. Poliakoff, A. J. Casson, Machine learning algorithm validation with a limited sample size, PloS one 14 (11) (2019) e0224365.

[54] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[55] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning, IEEE transactions on medical imaging 35 (5) (2016) 1285–1298.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[57] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-first AAAI conference on artificial intelligence, 2017.

[58] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013).

[59] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[60] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314–1324.

[61] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[62] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[63] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le, Mnasnet: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[64] M. Huh, P. Agrawal, A. A. Efros, What makes imagenet good for transfer learning?, arXiv preprint arXiv:1608.08614 (2016).

[65] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, The Journal of Machine Learning Research 18 (1) (2017) 6765–6816.

[66] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of big data 6 (1) (2019) 1–48.

[67] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 28, Curran Associates, Inc., 2015.

[68] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149 (2015).

[69] A. Kwasniewska, M. Szankin, M. Ozga, J. Wolfe, A. Das, A. Zajac, J. Ruminski, P. Rad, Deep learning optimization for edge devices: Analysis of training quantization parameters, in: IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1, 2019, pp. 96–101. doi:10.1109/IECON.2019.8927153.

[70] R. Ibrahim, M. O. Shafiq, Explainable convolutional neural networks: A taxonomy, review, and future directions, ACM Comput. Surv. 55 (10) (feb 2023). doi:10.1145/3563691.

[71] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[72] A. Singh, S. Sengupta, V. Lakshminarayanan, Explainable deep learning models in medical image analysis, Journal of Imaging 6 (6) (2020) 52.

[73] P. Bourdon, O. B. Ahmed, T. Urruty, K. Djemal, C. Fernandez-Maloigne, Explainable ai for medical imaging: Knowledge matters, in: Multi-faceted Deep Learning, Springer, 2021, pp. 267–292.

[74] M. G. Campana, M. Colussi, F. Delmastro, S. Mascetti, E. Pagani, Mpox close skin images (May 2023). doi:10.5281/zenodo.7948350.

[75] M. G. Campana, M. Colussi, F. Delmastro, S. Mascetti, E. Pagani, Mobile mpox detection system supplementary material (May 2023). doi:10.5281/zenodo.7981159.

[76] X. Wu, W. Ni, L. Jie, Y.-K. Lai, S. Cheng, Dongyu, Ming-Ming, J. Yang, Joint acne image grading and counting via label distribution learning, in: IEEE International Conference on Computer Vision, 2019.

[77] L. Muñoz-Saavedra, E. Escobar-Linero, J. Civit-Masot, F. Luna-Perejón, A. Civit, M. Domínguez-Morales, Monkeypox diagnostic-aid system with skin images using convolutional neural networks, Available at SSRN 4186534.

[78] T. B. Fitzpatrick, The Validity and Practicality of Sun-Reactive Skin Types I Through VI, Archives of Dermatology 124 (6) (1988) 869–871. doi:10.1001/archderm.1988.01670060015008.

[79] G. A. Tadesse, C. Cintas, K. R. Varshney, P. Staar, C. Agunwa, S. Speakman, J. Jia, E. E. Bailey, A. Adelekun, J. B. Lipoff, et al., Skin tone analysis for representation in educational materials (star-ed) using machine learning, NPJ Digital Medicine 6 (1) (2023) 151.

[80] M. Wilkes, C. Y. Wright, J. L. du Plessis, A. Reeder, Fitzpatrick Skin Type, Individual Typology Angle, and Melanin Index in an African Population: Steps Toward Universally Applicable Skin Photosensitivity Assessments, JAMA Dermatology 151 (8) (2015) 902–903. doi:10.1001/jamadermatol.2015.0351.

[81] I. UVA, Uva1-induced skin darkening is associated with molecular changes even in highly pigmented skin individuals, Journal of Investigative Dermatology 137 (2017) 1184e1187.

[82] X. Li, G. Zhang, H. H. Huang, Z. Wang, W. Zheng, Performance analysis of gpu-based convolutional neural networks, in: 2016 45th International Conference on Parallel Processing (ICPP), 2016, pp. 67–76. doi:10.1109/ICPP.2016.15.