A Study on the Fusion of Pixels and Patient Metadata in CNN-Based Classification of Skin Lesion Images

Fabrizio Nunnari, Chirag Bhuvaneshwara, Abraham Obinwanne Ezema, Daniel Sonntag

A study on the fusion of pixels and patient metadata in CNN-based classification of skin lesion images

Fabrizio Nunnari[0000−0002−1596−4043],
Chirag Bhuvaneshwara[0000−0002−7262−8708],
Abraham Obinwanne Ezema[0000−0002−9671−0925], and
Daniel Sonntag[0000−0002−8857−8709]

German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
{fabrizio.nunnari,chirag.bhuvaneshwara,
abraham_obinwanne.ezema,daniel.sonntag}@dfki.de

Abstract. We present a study on the fusion of pixel data and patient metadata (age, gender, and body location) for improving the classification of skin lesion images. The experiments have been conducted with the ISIC 2019 skin lesion classification challenge data set. Taking two plain convolutional neural networks (CNNs) as a baseline, metadata are merged using either non-neural machine learning methods (tree-based and support vector machines) or shallow neural networks. Results show that shallow neural networks outperform other approaches in all overall evaluation measures. However, despite the increase in the classification accuracy (up to +19.1%), interestingly, the average per-class sensitivity decreases in three out of four cases for CNNs, thus suggesting that using metadata penalizes the prediction accuracy for lower represented classes. A study on the patient metadata shows that age is the most useful metadatum as a decision criterion, followed by body location and gender.

Keywords: Skin lesion classification · Convolutional Neural Network · Machine Learning · Patient Metadata · Data Fusion

1 Introduction

The skin cancer death rate has escalated sharply in the USA, Europe and Australia. However, with proper early detection, the survival rate after surgery (wide excision) increases a lot. For this reason, the research community has put a significant effort in the early detection of skin cancer through the inspection of images. In order to increase their diagnostic accuracy, dermatologists use *dermascopes* (or dermatoscope) for the visual inspection of the skin. A dermascope is typically a cylinder containing a magnifying lens and a light emitter, helping the analysis of the substrate of the skin (see figure 1 for an example).

The use of deep convolutional neural networks (CNN) for the classification of skin lesions has significantly increased in the last years [15,5,2,13]. The breakthrough work of Esteeva et al. [9] being one of the most representative use cases,

Fig. 1. Left: a skin lesion as seen from a normal camera and, right, through a dermatoscope (source: `http://danderm.dk/`). Transparent rulers are often added to give a size reference. In the middle, we see such a dermatoscope with an ergonomic handle (source: Wikipedia).

where a CNN matched the accuracy of expert dermatologists in the diagnosis of skin lesions from image analysis. While this result was achieved through the use of a private dataset, in the Skincare project[1] we work on several extensions [16,20] using image material from the scientific community.

In the context of dermatoscopy, the most popular dataset is provided since 2016 by the Society for Digital Imaging of the Skin, which organizes the ISIC (International Skin Imaging Collaboration[2]) challenge. In all its editions, the ISIC challenge includes a classification task, which increased from the 2 classes of the first edition in 2016 (Nevus vs. Melanoma) to the 8 classes of the 2019 edition[3].

The 2019 challenge is enriched by two elements. In Task 1 (classification using only images) the test set used to evaluate the performances also contains images *not* belonging to any of the 8 classes present in the training set; the so called UNK (unknown) class. In other words, participating machine learning experts can only train models on 8 classes, but must predict 9 classes in the evaluation set. This should resemble actual clinical diagnostic conditions more and hence provide better decision support. In Task 2, participants should use additional patient *metadata* (age, gender, and location of the lesion on the body) to improve the prediction accuracy, and this is the focus of this study.

As pointed by Kawahara et al. [13], the comparison in performance between man vs. machine is often unfair. In most of the literature, machines (either classic Machine Learning algorithms or recent Deep Learning architectures) infer their diagnosis solely from image pixel information, and the comparison with human practitioners is done by providing the same material to both of them. However, in practice, doctors complement visual information with other *metadata*, which is usually collected by medical experts during their daily interactions with patients. Hence, to better match the diagnosis conditions, the data source of the ISIC 2019 challenge, namely the BCN20000 dataset [4], includes information available in clinical routine.

[1] `https://medicalcps.dfki.de/?page_id=1056`
[2] `https://isdis.org/isic-project/`
[3] `https://challenge2019.isic-archive.com/`

Table 1. Participants of ISIC 2019 Task 2, and their scores in the first (images-only) and the second task (images + metadata).

	Task 1		Task 2		
Team	rank	acc.	rank	acc.	gain
DAISYLab	1	0.636	1	0.634	-0.31%
Torus Actions	6	0.563	2	0.597	6.04%
DermaCode	4	0.578	3	0.56	-3.11%
BGU_hackers	11	0.543	4	0.541	-0.37%
BITDeeper	7	0.558	5	0.534	-4.30%
offer_show	14	0.532	6	0.532	0.00%
Tencent	16	0.525	7	0.527	0.38%
VisinVis	21	0.513	8	0.517	0.78%
MGI	31	0.489	9	0.5	2.25%
Le-Health	36	0.469	10	0.488	4.05%
MMU-VCLab	26	0.502	11	0.481	-4.18%
SY1	38	0.464	12	0.47	1.29%
IML-DFKI	40	0.445	13	0.445	0.00%
Panetta's Vision	39	0.461	14	0.431	-6.51%
KDIS	44	0.429	15	0.417	-2.80%
mvlab-skin	59	0.258	16	0.324	25.58%

It has been observed that the use of metadata leads to higher accuracy [12,17,23]. However, out of the 64 teams participating to the ISIC 2019 challenge, only 16 participants followed up with a submission on the images+metadata task. As can be seen in table 1, the challenge obtains unexpected surprising results: when introducing metadata into the predictors, out of 16 teams, only seven increased their performance, with a relative increase between less than 1% to about 6%, with only one team able to boost performance of 25%, but starting from a relatively low initial score. Two teams did not achieve any performance increase. More surprisingly, seven teams decreased their accuracy. These results suggest that the integration of metadata in CNN-based architectures is still not a well-known practice. Additionally, it is worth noticing that, from the point of view of evaluating the usefulness of metadata, the reported scores are biased by the presence of the extra UNK class in the test set, and by the need to handle missing metadata for some samples.

To the best of our knowledge, there is no prior work of systematically comparing the performance of pixel-only skin lesion classification with pixel+metadata conditions. Hence, in this paper we present a post-challenge study focusing on the use of metadata to improve the performance of skin lesion classification in the ISIC 2019 dataset. We compare several fusion techniques, some of them used by participants of the challenge, and measure their relative increase in performance for each of the available metadata. The comparison is performed while using two different CNN baseline architectures.

The paper is structured as follows: section 2 gives more details about the ISIC challenge and presents an analysis of the training material. Section 3 de-

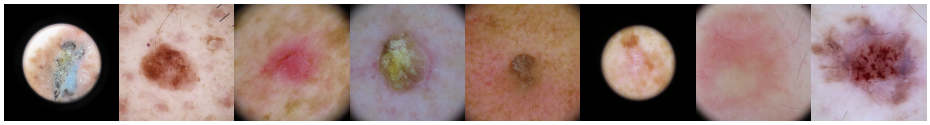

Fig. 2. A sample for each of the eight classes in the ISIC 2019 dataset. From left to right: Melanoma, Melanocytic nevus, Basal cell carcinoma, Actinic keratosis, Benign keratosis, Dermatofibroma, Vascular lesion, and Squamous cell carcinoma.

scribes related work on the use of metadata for skin lesion diagnosis and our ISIC evaluation. Section 4 describes the methodology used in our experiments. Section 5 presents the results of our tests. Section 6 summarizes the results of the experiments, and finally section 7 concludes.

2 The ISIC 2019 challenge

The ISIC 2019 dataset (courtesy of [22,7,3], License (CC-BY-NC) `https://creativecommons.org/licenses/by-nc/4.0/`) provides ground truth for training (25331 samples), while the test set ground truth remains undisclosed. Hence, for our tests, we used a split of the ISIC 2019 training set. Figure 2 shows a sample for each class.

Image *metadata* consist of patient gender (male/female), age, and the general location of the skin lesion on the body. Not all of the images are associated with full metadata info. Since the management of missing metadata would increase the complexity of the deep learning architecture (as in [12]), for this work, we used only the subset of 22480 images for which all metadata are present. As can be seen in table 2, the dataset is strongly unbalanced; this issue has been addressed by applying weightings to the loss function used to train the CNN models (see section 4.1).

Table 2. Class distribution in our ISIC 2019 training subset.

Class	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	Tot
Count	4346	10632	3245	845	2333	235	222	622	22480
Pct	17.8%	50.8%	13.1%	3.4%	10.4%	1.0%	1.0%	2.5%	100%

Age (figure 3, left) is subdivided into groups with bin size 5. The mean value is 54, and the minimum and maximum age bins are 0 (less than 5-year old children) and 85, respectively. Gender (figure 3, right) denotes 11950 male and 10530 female patients. The location of the skin lesion in the body has 8 possible options (Figure 3, down). The samples are not evenly distributed, as categories lateral torso, oral/genital, and palms/soles have only about a hundred samples each, while the others have about 2300.

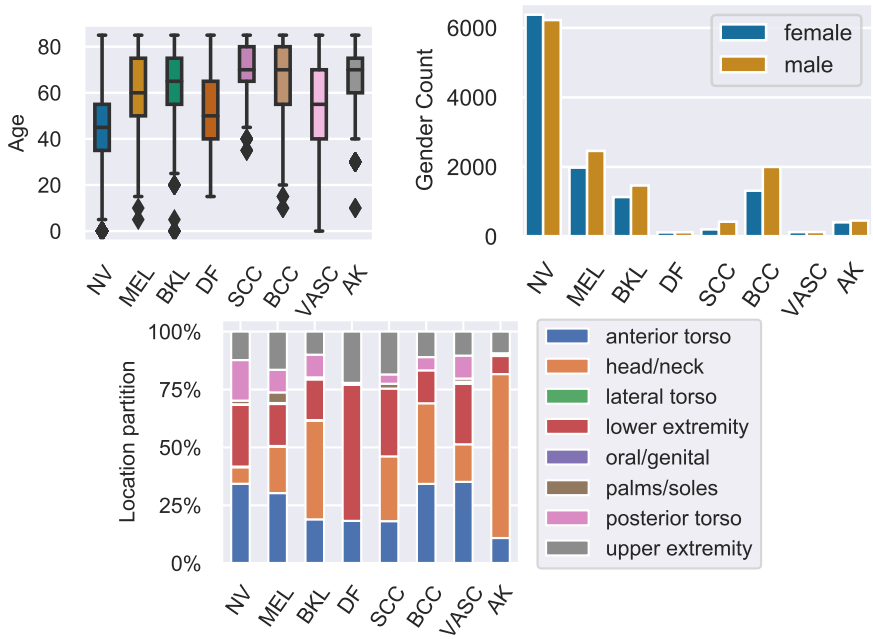

Fig. 3. Metadata distribution divided by class for age (left), gender (right), and body location (down).

The ISIC challenge measures the performance of a classifier by "Normalized (or balanced) multi-class accuracy"[4]. It can be computed by considering the diagonal of the confusion matrix. Each element of the diagonal (correctly classified class) is normalized according to the number of samples for that class. The elements of the diagonal are then averaged together, regardless of the number of samples per class, as to give the same importance to each class regardless of their observed frequency. This is equivalent to the average of the per-class sensitivities. In reporting the classification performances, we will use the term *accuracy* as the usual proportion of correctly classified samples, while the term *(average) sensitivity* as equivalent to the ISIC 2019 evaluation metric.

3 Related Work and ISIC 2019 Evaluation

In the realm of skin lesion classification, Kawahara et al. [12] already integrated pixel-based information (macroscopic and dermoscopic images) with human annotation of lesions based on the 7-point checklist [1]. The fusion is performed by concatenating internal CNN features of images [18,8] with 1-hot encoded scores of the 7-point method. The final classification is performed by an additional fully connected layer plus a final softmax. They report an increase of accuracy

[4] `https://challenge2019.isic-archive.com/evaluation.html`

from 0.618 to 0.737 (+19.2%) using Inception V3 as base CNN. In particular, their work addresses the problem of dealing with missing or partial metadata information through a combination of ad-hoc loss functions.

Yap et al. [23] report an increase of the AUC from 0.784 to 0.866 (+10.5%) when merging patient metadata over a RESNET50 baseline. Again, the input is a mixture of macroscopic and dermoscopic images. They used an internal layer of the CNN as image features and concatenated it with the the metadata. The concatenated vector goes through a shallow neural network of 2 dense layers and a final softmax. Their results confirm that internal layers activation of CNNs are useful image feature representations, as it has been seen in other application domains [18,8].

The participants of the ISIC 2019 challenge Task 2 however addressed the problem of metadata fusion using a variety of different approaches.

Among the first five best performers, DAISYLab (1st) fed the metadata to a 2-layer shallow neural network (sNN). The result is concatenated with internal image features and fed to a final dense layer + softmax. Metadata were encoded with 1-hot, but missing metadata were left to 0s and missing age to an arbitrary -5. The final score is slightly worse than using no metadata. Torus Actions (2nd) saw an increase of 6.04% when using metadata, but did not report any detail. Also BGUHackers (4th) and BITDeeper (5th) did not report any detail, but their score worsened.

DermaCode (3rd) used a manually engineered set of rules derived from a visual inspection on metadata analysis. They report having preferred rules since tests using small NNs gave negative results. However, their official score decreased, too, in Task 2 (-3.11%).

Among good performers, MGI (9th) was able to increase their accuracy (+2.25%) by concatenating the final softmax output with a shallow NN of 2x dense layers plus softmax. Le-Health (10th) increased accuracy (+4.05%) following the principle: "To combine image data with meta-data, we first use one-hot encoding to encode meta-data and then concatenate them with the image feature extracted from the layer before the first fully-connected layer". No more details are given. Finally, mvlab-skin used a first shallow NN to reduce the number of features from the convolution output. The result is concatenated with metadata and fed to another sequence of two dense layers + softmax. They achieved a remarkable increase in accuracy of +25.58%, but starting from a low initial accuracy of 0.258.

Given the variety of strategies and baselines it is difficult to objectively state what is the best approach for metadata integration, especially in presence of the UNK class influencing the final scores and the lack of details on how to handle partial metadata information.

In this paper, we replicate existing approaches, we add new ones based on classic ML algorithms, and compare them all over the same baseline CNNs, without the biases of unknown class samples and missing metadata information in the test set.

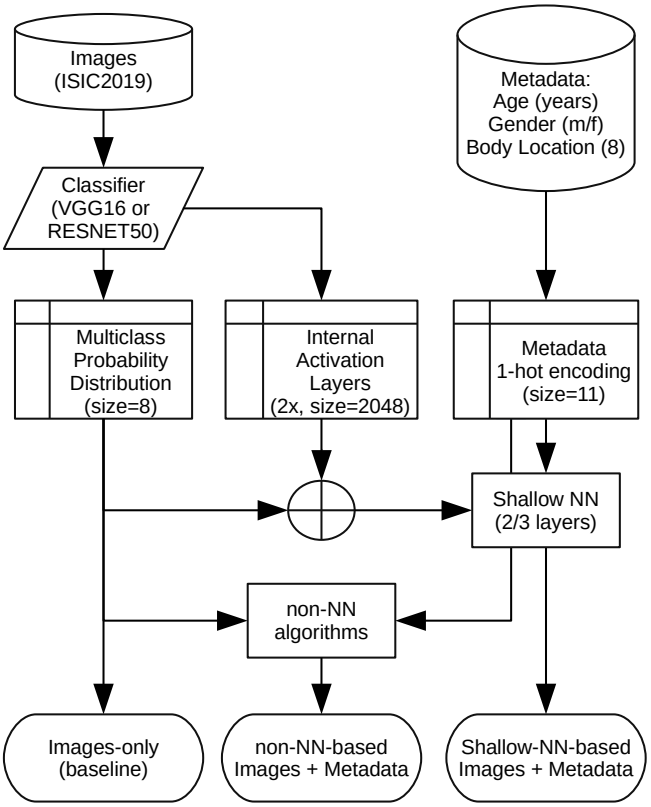

Fig. 4. The data flow of our experiments, concatenating the output of a CNN (either final predictions or internal layer activation values) with metadata to improve prediction.

4 Method

Figure 4 depicts the methods used for integrating metadata together with pixel-based image classification. The pipeline starts with an initial training of a deep CNN for the classification of an image across the eight ISIC 2019 categories (softmax output). We repeated the same procedure for two CNN architectures, VGG16 [19] and RESNET50 [11] (details can be found in section 4.1), to monitor the difference in the relative improvement given by metadata on two baselines. In parallel, metadata are encoded as 1-hot vectors of 11 values (details in section 4.2).

The fusion with metadata is then performed in two ways. In the first fusion method (non-NN, section 4.3), the output of the CNN classification (softmax, size 8) is concatenated with the metadata. The resulting feature vector (size 19) is passed to several well-known learning methods (either tree-based or SVM). In the second fusion method, (sNN, section 4.4), we take, from the CNN, either

the softmax output (size 8), or the activation values of the two fully connected layers (size 2048, each) located between the last convolution stage and the final softmax. Each fusion vector is then the concatenation of the softmax output (or the activation values for that matter) and the metadata vector. The concatenated vector is passed through a shallow neural network of two or three layers.

The following sections give more details about the metadata encoding and the training procedures.

4.1 Baseline CNN classifier

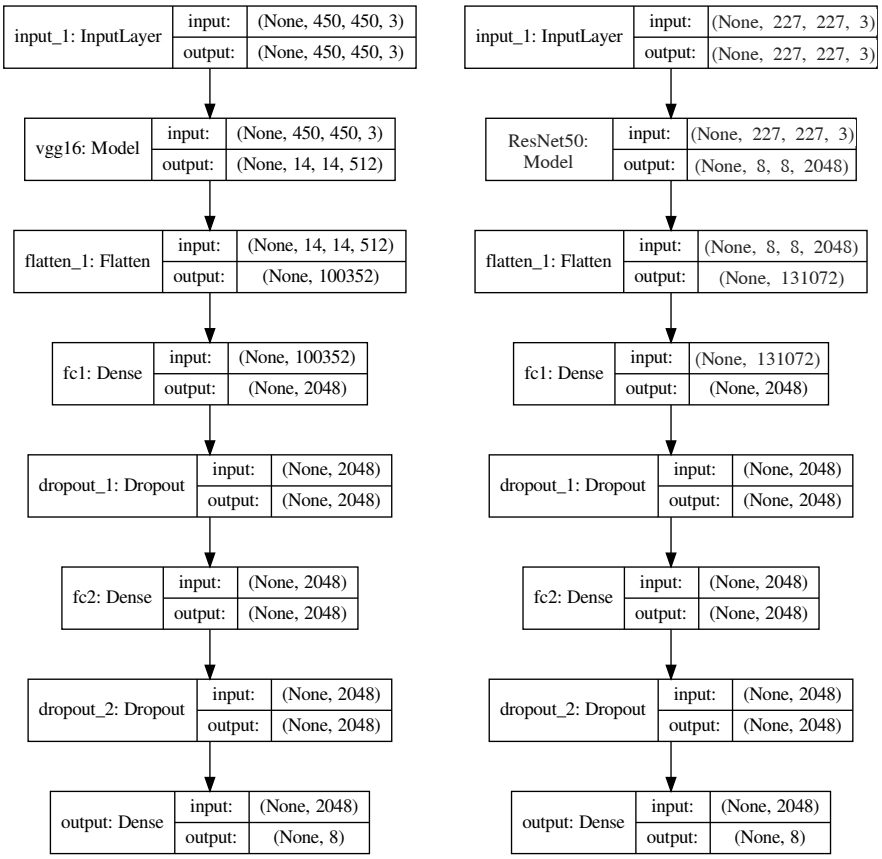

Fig. 5. Architecture of the baseline classifier for VGG16 (left) and RESNET50 (right).

Figure 5 shows the architectures for the VGG16 and RESNET50 baseline classifiers, the main differences being the resolution of the input image and the size of the flattened layer. The architecture was implemented using the Keras

framework (`https://keras.io/`). We follow a transfer learning approach [21] by first initializing the weights of the original VGG16/RESNET50 of a pre-trained model from ImageNet [6] and then substituting the final part of the architecture. The last convolution layer is flattened into a vector of size 100352/131072, followed by 2 fully connected layers of size 2048. Each fully connect layer is followed by a dropout layer with probability 0.5. The network ends with an 8-way softmax output. For VGG16, we set the input resolution to 450x450 pixels, instead of the original 224x224, to improve accuracy by introducing more details on the images.

The 22480 samples of the ISIC2019 subset arew randomly split in 19490 samples for training and 2x1495 samples for validation and testing. The sample selection is performed by ensuring the same between-class proportion on all of the three subsets. This means that in all three sets (train, development, and test), for example the nevus class represents about 50% of the samples, melanoma 20%, and so on for the remaining 6 classes. The same split is used for all further experiments that include metadata information.

The class unbalance is compensated by providing a class weight vector (size 8) as parameter `class_weight` to the method `Sequential.fit()`. The weight vector is used to modulate the computation of the loss for each sample in a training batch, and it is computed by counting the occurrences of each class in the training set and then normalizing the most frequent class to 1.0. For our training set, this translates into a base identity weight (1.0) for class NV and a maximum weight (47,89) for class VASC.

We trained our VGG16/RESNET50 baseline models for 10/30 epochs, batch size 8/32, SGD optimizer, lr=1E-5, with a 48x augmentation factor (each image is flipped and rotated 24 times at 15 degree steps, as in Fujisawa et al. [10]). Training takes about 3/5 days on an NVIDIA RTX 2080Ti GPU.

4.2 Metadata Preprocessing

In our scheme, the extracted high-level image features are continuous, in contrast to the age approximation (discrete), anatomical location (categorical), and gender (categorical) from the metadata. We normalized the age range [0,100] to the range [0,1], and applied one-hot encoding to the categorical metadata. Hence, for each image in the dataset, a corresponding metadata information vector of size eleven is generated. Our choices in representation are influenced by the needs to have a uniform representation (allowing for the encoding of all data into a single 1D feature vector) and to reduce the variation in the different input sources.

4.3 Data Fusion using Classical ML

In this approach, we concatenate the final probability density from our baseline networks with the metadata information.

We experimented with several well established Machine Learning algorithms like Support Vector Machines (SVM), Gradient Boosting and Random Forest from the scikit-learn library (`https://scikit-learn.org/`), and XG Boost

from the xgboost library (`https://xgboost.readthedocs.io`). All of these models were trained with a train, development, and test split of the data with the hyperparameters being tuned on the train data w.r.t. the development data. Hyperparameter exploration was conducted using a grid approach. The final results are reported on the test data set, which was not included in the training nor in the hyperparameter tuning.

SVM (Support Vector Machines) is a supervised discriminative classifier based on generating separating hyperplanes between the different classes. Hyperplanes can be generated using different kernels. We experimented with several hyperparameters and found the best ones to be regularization parameter C = 1000 and kernel coefficient $\gamma = 0.1$ with the Radial Basis Function Kernel.

XGBoost and **Gradient Boosting** are similar, the former works on the second derivative of the formulated loss function and the latter works on the first derivative. XGBoost uses advanced regularization which helps achieve better regularization. In the case of XGBoost, we found the best hyperparameters to be colsample bytree (Subsample ratio of columns when constructing each tree) = 1.0, gamma = 3, learning rate = 0.05, max depth = 6, minimum child weight = 10, number of estimators = 500, subsample = 0.6 . For Gradient Boosting, the best hyperparameters are learning rate = 0.1, maximum depth = 6, maximum features = 2, minimum samples per leaf = 9, number of estimators = 500, subsample = 1.

Finally, we use the ensemble learning method of **Random Forests** which is a decision tree algorithm that trains multiple trees and, for classification problems, picks the class with the highest mode during testing. This ensemble learning method prevents overfitting on training data, which is a common problem for decision trees. In the case of Random Forests model, we found the best hyperparameters to be bootstrap = False, class weight = balanced, maximum depth = 100, maximum features = 1, minimum samples per leaf = 1, minimum samples per split = 2, number of estimators = 500.

To address the problem of class imbalance in the data, we used the same class weights computed for the baseline CNN classifiers. The weights were directly fed into the Random Forests and SVM classifiers. As Gradient Boosting and XG Boost functions only support sample weights, we assigned to each sample the weight of its true class.

Each of these models contain multiple other hyperparameters, which were left at their default settings as provided in the scikit-learn and xgboost libraries. All of the four approaches can be trained in a few minutes.

4.4 Data Fusion using Shallow NN

In this approach, we concatenate different feature vectors from our baseline networks with the metadata information. The concatenated vectors are then forwarded to stacks of uniform Glorot-initiliazed dense layers, terminated by a softmax, to predict disease classes.

As feature vectors, we utilize the activation values of the internal layers (`fc1` and `fc2`) as well as the output softmax layer. We experimented with different

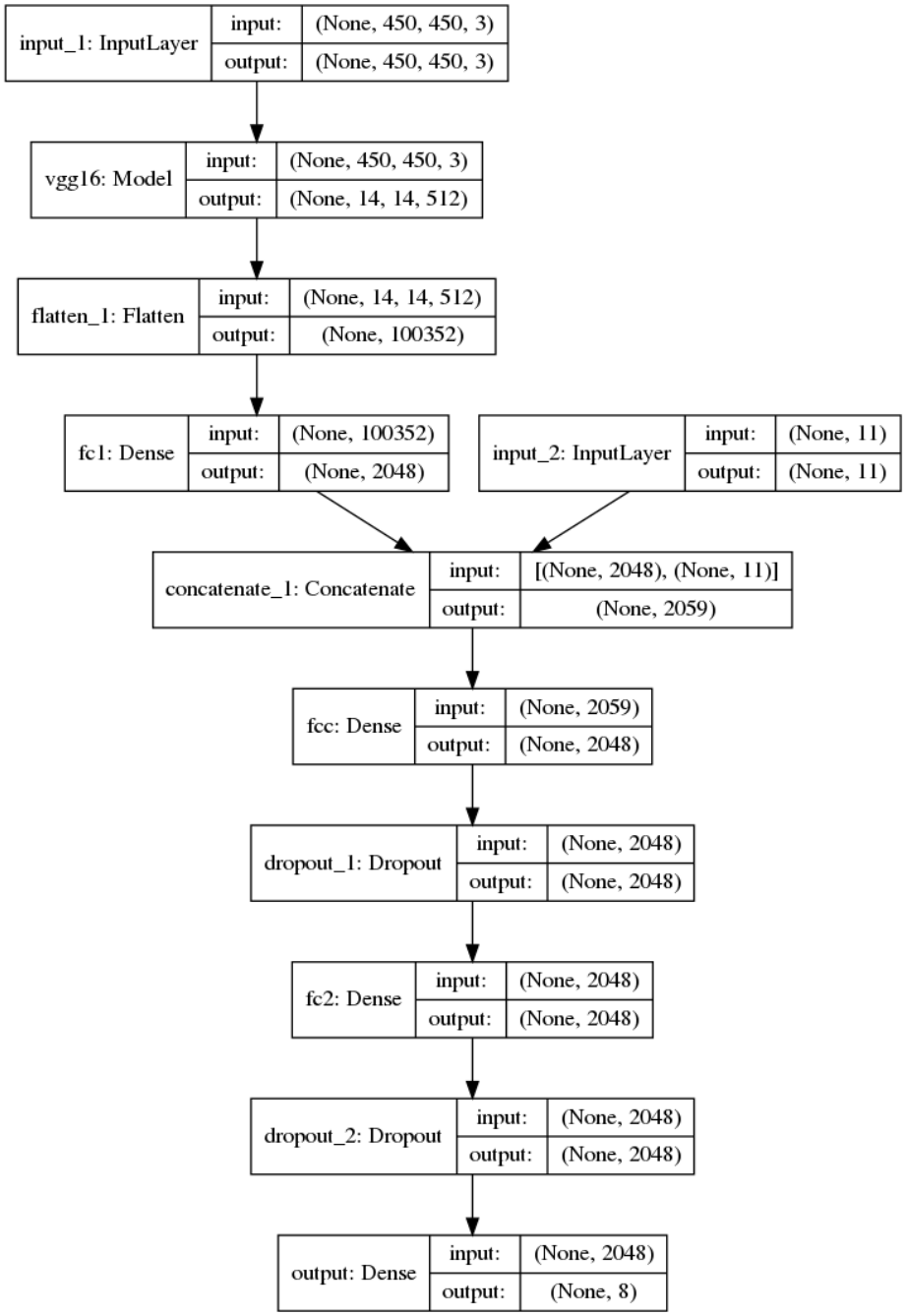

Fig. 6. Architecture for configuration `fc1_fc+doX2`, where fc1, concatenated with metadata, is followed by two blocks of dense and dropout layers. `input_2` is the 1-hot encoded metadata.

combinations of `fc1`, `fc2`, and `prediction` probabilities followed by one or two dense layers (size 2048) coupled with dropout layers (p=0.5). Figure 6 shows an example for the configuration `fc1_fc+doX2`. We tested using a sampling search approach.

When utilising internal layers, after the concatenation layer, a dense layer follows in the network, contrary to a dropout layer from our baseline models. This design decision was made to avoid masking out some metadata input values during Bernoulli sampling, which is peculiar to the inverted dropout method.

Among our experiments, we also considered concatenating the metadata directly to the `flatten_1` layer, just after the convolution (see figure 5), which means concatenating a vector of size 100352 (VGG16) or 131072 (RESNET50) to only 11 elements. Intermediate experimental results showed that this configuration gives no improvements with respect to the baseline. This is likely due to the significant difference between the sizes of the two feature vectors, which leads to the metadata being "obscured" by the high amount of other features. To overcome this issue, strategies that perform size-dependent output normalization do exist [14]. However, this would require the manual tuning of an extra equalization parameter. Hence, we did not further investigate in this direction and left the "flatten+metadata" configuration for future work.

For all configurations, we froze the whole base convolution model and trained the remaining layers for an average of 19 epochs, with lr=1E-4, using adam (default) or SGD optimizers. On average, training one epoch takes 5 minutes.

5 Results

We applied the two fusion methods (cML and sNN) to the two baseline CNNs (VGG16 and RESNET50) in several configurations, where each configuration is either a different ML algorithm or shallow NN architecture. For each of the four combinations between CNN+method, we report i) the performance of each configuration, followed by ii) an analysis on the contribution of each metadata (age, gender, location) in the best performing configuration.

As for the metrics, we report the overall classification accuracy together with class-averaged specificity, sensitivity, and F1 score. The three last scores are computed by considering each class separately, measuring the metric as class-vs-others, and finally averaging the eight results. The so-computed sensitivity is hence equal to the scoring metric used in the ISIC 2019 challenge.

Among all metrics, sensitivity is the most important in practical, medical terms, as it represents the number of lesions correctly predicted in the specific class, for which a correct treatment would follow. However, while trying to obtain the best configuration, we choose accuracy, as it is the metric normally used to optimize the predictors. It is worth noting that the specificity is strongly influenced by the highly unbalanced dataset. A misclassification of a few samples in a lower-represented class doesn't affect global accuracy but can lower the average sensitivity significantly.

In all tables, **bold** font marks the highest value in a metric as well as the configuration with highest accuracy, while *italic* marks the highest value in a metadata group.

VGG16+cML As reported in table 3, top, random forest gave the best accuracy followed by SVM, while gradboost worsened the predictions. In some configurations, average sensitivity even decreases with respect to the baseline. A closer look to the per-class results shows that sensitivity lowers for classes with a low number of samples: BKL, VASC, and SCC.

Table 3, bottom, shows that *age* gives the best accuracy boost and disrupts sensitivity the least, followed by location and gender. The combination age+location gives the best performances.

VGG16+sNN Table 4, top, shows that, in contrast to classical ML, with sNN all the metric performances are increasing with respect to the baseline. Configurations using the image features from the internal layer fc1 gives the best results, and clearly surpasses classical ML methods. The simplest neural network, composed of only 1 dense layer (size 2048) followed by a dropout (p=0.5) performs the best in terms of accuracy (nonetheless, also using 2 dense layers gives similar accuracy).

Table 4, bottom, shows the same pattern as in VGG16+cML: best metadata are in order age, location, and gender. Curiously enough, the combination of only gender and location gives better sensitivity than using all metadata.

RESNET50+cML When switching to RESNET50 as a baseline, integration through classic ML (Table 5) shows nearly the same behavior as for VGG16+cML: random forest and SVM give the best performance but SVM's performance is marginally better in this case, and age, location, and gender improve performance with the same order of efficiency. Again, the combination age+location gives the best performances.

RESNET50+sNN Finally, when using RESNET50+sNN (table 6) the best configuration still uses the input from fc1, followed by two dense layers. It is worth noticing that the configurations using class predictions as input data (pred_fc+...) are able to increase sensitivity over the baseline, even though the overall accuracy isn't as good as when using internal layers. Among metadata, age is still the most useful as a decision criterion, but location and gender switch positions.

6 Discussion

VGG16 represent a baseline with relatively low performance (accuracy=0.66). In VGG16 + rf accuracy increases (+0.1003, +15.0%), but sensitivity is quite the same (0.0131, +2.0%). In VGG16 + sNN both accuracy (+0.1278, +19.1%) and sensitivity (+0.0587, +8.9%) increase.

Table 3. Performances of VGG16 + Classical ML.

Algorithm Comparison

Method	accuracy	specificity	sensitivity	F1
none	0.6676	0.9477	0.6597	0.5597
gradboost	0.5920	0.9342	0.5236	0.4293
rf	**0.7679**	**0.9592**	0.6728	**0.6754**
svm	0.7532	0.9563	0.6543	0.6463
xgboost	0.7338	0.9577	**0.6730**	0.6420

Details for Random Forest

Metadata	accuracy	specificity	sensitivity	F1
none	0.6676	0.9477	0.6597	0.5597
Age	*0.7478*	*0.9562*	*0.6340*	*0.6370*
Gender	0.7090	0.9482	0.5958	0.6016
Location	0.7311	0.9521	0.6280	0.6326
Age+Gender	0.7492	0.9564	0.6416	0.6455
Age+Location	*0.7579*	*0.9573*	*0.6572*	*0.6591*
Gender+Location	0.7324	0.9523	0.6360	0.6415
All	**0.7679**	**0.9592**	**0.6728**	**0.6754**

Table 4. Performances of VGG16 + Shallow NN.

Architecture Comparison

Architecture	accuracy	specificity	sensitivity	F1
none	0.6676	0.9477	0.6597	0.5597
fc1_fc+do	**0.7953**	**0.9646**	0.7184	**0.7087**
fc1_fc+doX2	0.7913	0.9639	**0.7248**	0.7020
fc2_fc+do	0.7833	0.9631	0.7059	0.6916
fc2_fc+doX2	0.7672	0.9616	0.7004	0.6664
pred_fc+bnX2_adam	0.7304	0.9573	0.6913	0.6431
pred_fc+bnX2_sgd	0.7304	0.9572	0.6723	0.6232

Details for fc1_fc+do

Metadata	accuracy	specificity	sensitivity	F1
none	0.6676	0.9477	0.6597	0.5597
Age	*0.7726*	*0.9618*	0.6954	0.6709
Gender	0.7525	0.9574	0.6890	0.6787
Location	0.7579	0.9588	*0.7054*	*0.6802*
Age+Gender	0.7732	0.9613	0.6905	0.6736
Age+Location	*0.7759*	*0.9626*	0.7184	0.6855
Gender+Location	0.7679	0.9613	***0.7259***	*0.6963*
All	**0.7953**	**0.9646**	0.7184	**0.7087**

Table 5. Performance of RESNET50 + Classical ML.

Algorithm Comparison

Method	accuracy	specificity	sensitivity	F1
none	0.7833	0.9645	**0.7849**	0.7569
gradboost	0.7010	0.9501	0.6733	0.6137
rf	0.8100	0.9664	0.7623	0.7728
svm	**0.8127**	**0.9669**	0.7626	**0.7785**
xgboost	0.7926	0.9659	0.7634	0.7652

Details for Random Forest

Metadata	accuracy	specificity	sensitivity	F1
none	0.7833	0.9645	**0.7849**	0.7569
Age	*0.8094*	*0.9662*	*0.7642*	0.7737
Gender	0.8020	0.9644	0.7608	0.7711
Location	0.8027	0.9651	0.7618	*0.7749*
Age+Gender	0.8087	0.9660	0.7622	0.7730
Age+Location	*0.8120*	*0.9667*	0.7629	*0.7782*
Gender+Location	0.8033	0.9651	*0.7631*	0.7751
All	**0.8127**	**0.9669**	0.7626	**0.7785**

Table 6. Performance of RESNET50 + Shallow NN.

Architecture Comparison

Architecture	accuracy	specificity	sensitivity	F1
none	0.7833	0.9645	0.7849	0.7569
fc1_fc+do	0.8194	0.9684	0.7672	0.7806
fc1_fc+doX2	**0.8334**	**0.9700**	0.7718	**0.7908**
fc2_fc+do	0.8194	0.9681	0.7447	0.7691
fc2_fc+doX2	0.8167	0.9677	0.7600	0.7729
pred_fc+bnX2_adam	0.8074	0.9678	0.7868	0.7880
pred_fc+bnX2_sgd	0.8100	0.9686	**0.7904**	0.7859

Details for fc1_fc+doX2

Metadata	accuracy	specificity	sensitivity	F1
none	0.7833	0.9645	**0.7849**	0.7569
Age	*0.8201*	*0.9680*	0.7586	0.7796
Gender	0.8167	0.9665	0.7400	0.7655
Location	0.8147	0.9675	*0.7742*	*0.7846*
Age+Gender	0.8167	0.9665	0.7400	0.7655
Age+Location	0.8221	0.9678	0.7550	0.7744
Gender+Location	*0.8274*	*0.9687*	*0.7611*	***0.7915***
All	**0.8334**	**0.9700**	0.7718	0.7908

When using a baseline with better starting performance (accuracy=0.78), there are only slight improvements in accuracy, and a decrease in sensitivity occurs more often. In RESNET50 + SVM accuracy increases +0,0294 (3,8%), while sensitivity decreases -0.0223 (-2.8%). In RESNET50 + sNN accuracy increases +0,0502 (6,4%), while sensitivity decreases -0,0131 (-1,7%).

In general, it seems that, when applied to poorly performing baselines, metadata help the classification in both accuracy and sensitivity. However, when the CNN are already well performing, the gain in accuracy is marginal and sensitivity generally decreases. This means that metadata can help increase the overall number of correctly classified samples, but compromises the recognition of samples belonging to lower-represented classes. Among the three metadata, age is most useful in increasing accuracy, followed by location and gender.

Among the fusion techniques, random forests and SVM are the best for non-neural techniques, which can be used to merge only predictions and metadata, and offer a fast computation. However, best results are obtained by merging metadata with the activation values of the first fully connected layer.

7 Conclusions

Motivated by the observation of a lack of active participation, together with the surprisingly negative results in the Task 2 of the ISIC 2019 challenge, we presented a detailed study on the fusion between pixels and metadata for the improvement of the accuracy in the classification of images of skin lesions. In general, our experiments confirm (and quantify) the superiority of shallow neural networks over SVM and tree-based ML algorithms. The experiments suggest that internal CNN activation values are the best option for an integration with metadata.

Concerning the ISIC 2019 challenge, from our overview, it appears that some teams chose suboptimal strategies to merge metadata. But more interestingly, "good" strategies (increasing accuracy) are always associated with a decrease in the average sensitivity, the evaluation metric used in the ISIC challenges.

Trying to explain the unexpected reduction in performance in Task 2 when compared to Task 1, the reason of this behaviour might be the fact that Age, Gender, Location have been added to the dataset (by the designers) because it was known that they correlate with the higher-represented classes in the dataset (Nevus and Melanoma). Possibly, this doesn't hold for lower-represented classes (e.g., VASC) which were only recently added to the ISIC challenge dataset. Further data analysis is needed to confirm this hypothesis.

Another possible explanation for the divergence between accuracy and average sensitivity might be related to the optimization goal (loss function) of our baseline models, which aim at maximizing accuracy. Another set of experiments should be conducted to check if the decrease in sensitivity still emerges when directly training a network to maximize for average sensitivity.

We hope that this work will give an overview of the techniques and solution that are worth pursuing when including metadata to pixel-based classification of skin images and the challenges that might occur.

It is still to be validated how our findings generalize to other medical contexts, different metadata, or to non-medical contexts, where images and metadata pertain to very different domains (e.g., the use of age and nationality in the detection of emotion from facial expressions).

References

1. G. Argenziano, C. Catricalà, M. Ardigo, P. Buccini, P. De Simone, L. Eibenschutz, A. Ferrari, G. Mariani, V. Silipo, I. Sperduti, and I. Zalaudek. Seven-point checklist of dermoscopy revisited: Seven-point checklist of dermoscopy revisited. *British Journal of Dermatology*, 164(4):785–790, April 2011.
2. M. Emre Celebi, Noel Codella, and Allan Halpern. Dermoscopy Image Analysis: Overview and Future Directions. *IEEE Journal of Biomedical and Health Informatics*, 23(2):474–478, March 2019.
3. Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv:1710.05006 [cs]*, October 2017. arXiv: 1710.05006.
4. Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and Josep Malvehy. BCN20000: Dermoscopic Lesions in the Wild. *arXiv:1908.02288 [cs, eess]*, August 2019. arXiv: 1908.02288.
5. Clara Curiel-Lewandrowski, Roberto A. Novoa, Elizabeth Berry, M. Emre Celebi, Noel Codella, Felipe Giuste, David Gutman, Allan Halpern, Sancy Leachman, Yuan Liu, Yun Liu, Ofer Reiter, and Philipp Tschandl. Artificial Intelligence Approach in Melanoma. In David E. Fisher and Boris C. Bastian, editors, *Melanoma*, pages 1–31. Springer New York, New York, NY, 2019.
6. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009. IEEE.
7. Hospital Clínic de Barcelona Department of Dermatology. Bcn_20000 dataset.
8. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China, June 2014. PMLR.
9. Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542:115, January 2017.
10. Y. Fujisawa, Y. Otomo, Y. Ogata, Y. Nakamura, R. Fujita, Y. Ishitsuka, R. Watanabe, N. Okiyama, K. Ohara, and M. Fujimoto. Deep-learning-based, computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. *British Journal of Dermatology*, September 2018.

11. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
12. Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets. *IEEE Journal of Biomedical and Health Informatics*, 23(2):538–546, March 2019.
13. Jeremy Kawahara and Ghassan Hamarneh. Visual Diagnosis of Dermatological Disorders: Human and Machine Performance. *arXiv:1906.01256 [cs]*, June 2019. arXiv: 1906.01256.
14. Taku Komura, Daniel Holden, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics*, 36(4):1–13, July 2017. Siggraph 2017 ; Conference date: 30-07-2017 Through 03-08-2017.
15. Nabin K. Mishra and M. Emre Celebi. An Overview of Melanoma Detection in Dermoscopy Images Using Image Processing and Machine Learning. *arXiv:1601.07843 [cs, stat]*, January 2016. arXiv: 1601.07843.
16. Fabrizio Nunnari and Daniel Sonntag. A CNN toolbox for skin cancer classification. https://arxiv.org/abs/1908.08187, 2019. DFKI Technical Report.
17. Andre G. C. Pacheco and Renato A. Krohling. The impact of patient clinical information on automated skin cancer detection. *arXiv:1909.12912 [cs, eess, stat]*, September 2019. arXiv: 1909.12912.
18. Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.
19. Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, September 2014. arXiv: 1409.1556.
20. Daniel Sonntag, Fabrizio Nunnari, and Hans-Jürgen Profitlich. The Skincare project, an interactive deep learning system for differential diagnosis of malignant skin lesions. Technical Report. https://arxiv.org/abs/2005.09448, 2020. DFKI Technical Report.
21. Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A Survey on Deep Transfer Learning. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.
22. Medical University of Vienna ViDIR Group, Department of Dermatology. Ham10000 dataset. https://doi.org/10.1038/sdata.2018.161.
23. Jordan Yap, William Yolland, and Philipp Tschandl. Multimodal skin lesion classification using deep learning. *Experimental Dermatology*, 27(11):1261–1267, November 2018.