A Multiclass Skin Lesion classification approach using Transfer learning based convolutional Neural Network

2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII) | 978-1-6654-4126-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICBSII51839.2021.9445175

Cauvery K
Department of Computer Science and Engineering,
Manipal Institute of Technology,
MAHE, Manipal
cauk.84@gmail.com

P C Siddalingaswamy *
Department of Computer Science and Engineering,
Manipal Institute of Technology,
MAHE, Manipal
*pcs.swamy@manipal.edu

Sameena Pathan
Department of Computer Science and Engineering,
Manipal Institute of Technology,
MAHE, Manipal
sameena.pathan.k@gmail.com

Noel D'souza
Department of Electrical & Electronics Engineering,
Manipal Institute of Technology,
MAHE, Manipal
dsouzanoel2197@gmail.com

***Abstract*— The rapid rise in skin diseases over the past decade has been a growing concern worldwide. Early detection, correct categorization, and accurate identification can result in the successful treatment of melanoma, thereby decreasing the morbidity and mortality rate. Thus, there is a significant need for a system that is capable of identifying skin diseases and precisely classifying them. The proposed work aims to develop a multi class classification system using transfer learning-based convolutional neural networks (CNN). In particular, the proposed solution classifies the dermoscopic images to 8 different categories namely Melanoma (MEL), Basal Cell Carcinoma (BCC), Actinic Keratosis (AK), Benign Keratosis (BKL), Dermatofibroma (DF), Vascular lesions (VASC) and Squamous Cell Carcinoma (SCC). Four state-of-art pre-trained models are used for this task. A functional model-based network is leveraged to embed these sub-models in a larger multi-headed neural network. This will allow the embedded model to be treated as a single large model. An ensemble approach, termed as blending, is employed to combine the predictions efficiently made by the sub-models. Additionally, a robust cropping strategy is implemented to deal with the uncropped images and their impact on the performance of the classifiers is investigated. The impact of applying blending technique to ensemble the pre-trained CNNs are investigated against the performance of the individual classifier. The proposed work is carried out on International Skin Imaging Collaboration (ISIC) 2019 dataset. In this work, the solution for task 1 of the challenge is presented and we obtained balanced multi-class accuracy of 81.2% on the dataset compiled from the original dataset.**

Keywords— Balanced Multi-class Accuracy (BMA), Blending, Convolutional Neural Network, Dermoscopic images, Functional model, Skin lesions

I. INTRODUCTION

Skin lesions can be referred to as any abnormal growth or appearance on the skin when compared to the skin around it. It can be categorized into many different classes and subclasses [1]. Most of these skin lesions are benign though some, such as certain moles and actinic keratosis, can be pre-cursor to skin cancer. Skin cancers – including melanoma and non-melanoma comprising of Basal Cell Carcinoma (BCC) and Squamous Cell Carcinoma (SCC) – are one of the most fatal forms of malignancy occurring in humans, with melanoma contributing for the majority of skin-related deaths worldwide. However, these skin cancers are easily curable if detected in the early stages which have led to massive investment in the field of skin lesion analysis to develop automated tools for detection and classification of skin lesions [2]. Previously, conventional computer-aided (CAD) systems relied on the extraction of hand-crafted image features from the lesion area and its border to be fed to a traditional classifier which often involves extensive pre-processing and manual segmentation [3]. Recently, the rise in the application of deep learning in the medical field has led to the development of numerous promising classification methodologies. Specifically, convolutional neural networks (CNNs) [4] have achieved results as precise as certified dermatologists working on the same task and may also be capable of outperforming them [5]. In deep learning-based classification, the feature extraction and classification are both learned and performed as a single unit.

In medical image analysis, transfer learning using pre-trained models has been incorporated extensively by various medical fields. Some of the recent related works using deep learning mentioned below incorporate this strategy. Lee *et al.,* [6], provides a very good example of combining image segmentation using U-Net and DenseNet models and classification using the ensemble of classifiers to provide classification accuracy close to practiced professionals. Their work was done as a part of ISIC-2018 Challenge and claimed to have got a balanced accuracy of 78.9% for task3. Gessert *et al.,* [7] have used ensemble strategy to select the best subset of models that provide good balanced accuracy. Their work was done as a part of ISIC-2019 Challenge and have obtained a balanced accuracy score of 63.6% and 63.4% for task1 and task2 respectively. Frangi *et al.,* [8] presented another novel idea of incorporating dual CNNs (ResNet-50) simultaneously and sending the concatenated output to a synergic network which helps to reduce the intra-class variations and inter class similarity problem. Their work aimed at identifying melanoma and nevus skin images.

From the literature survey, it is identified that early and accurate diagnosis of skin cancer can dramatically reduce the

978-1-6654-4126-1/21/$31.00 ©2021 IEEE

mortality rate as most of these skin diseases are curable in their nascent stages. It is also identified that diagnostic analysis of dermatologists can be highly subjective because there are high inter-class similarity and intra-class variations as well as low contrast of the skin lesions making it difficult for dermatologists to precisely identify the skin lesions. We can also acknowledge the importance of the role played by deep learning in the classification of skin lesions and how the diagnostic accuracy has improved without being affected by human subjectivity. Therefore, in this work, we will be focusing on addressing the issues of classifying skin lesions with deep learning using the concepts of transfer learning and ensemble by blending technology.

II. Methodology

A. Dataset

The dermoscopic images used in this work are provided by the ISIC organization. The main dataset is a combination of BCN20000 [9], HAM10000 [10], and MSK [11] datasets with a total of 25,331 dermoscopic images acquired at multiple sites and with different pre-processing methods applied beforehand. It contains images of the class's melanoma (MEL), melanocytic nevus (NV), basal cell carcinoma (BCC), actinic keratosis (AK), benign keratosis (BKL), dermatofibroma (DF), vascular lesion (VASC) and squamous cell carcinoma (SCC). The image distribution across lesion classes are depicted in Table I. These images are in 24-bit (8 bits per channel) JPEG format. Dimensions of these images are inconsistent ranging from 600 × 450 to 1024 × 1024. The HAM10000 dataset contains images that centered and cropped around the lesion. Histogram corrections have been applied to some of these images. The images are of size 600 × 450. BCN20000 contains images for size 1024 × 1024.

TABLE I. ORIGINAL TRAIN DATASET STATISTICS

Diagnostic Category	Count (25,331)
Melanoma (MEL)	4522
Melanocytic nevus (NV)	12875
Basal Cell Carcinoma (BCC)	3323
Actinic keratosis (AK)	867
Benign Keratosis (solar lentigo/seborrheic keratosis/lichen planus-like keratosis) (BKL)	2624
Dermatofibroma (DF)	239
Vascular lesion (VASC)	253
Squamous Cell Carcinoma (SCC)	628

In this work, we have considered only a subset of images from the main dataset due to the issues we encountered during training. Two datasets, having 4999 and 7900 images, are compiled from the original dataset. Datasets are chosen in a way that they maintain the class imbalance present in the original set as much as possible. Note that the original dataset did not have any images of unknown (UKN) class, hence that class is not included in the current dataset. An additional dataset containing 4242 images are compiled as test dataset for testing purpose. The original test dataset provided by the ISIC Organization is not used in our work. As such, the final prediction is performed for 8 categories.

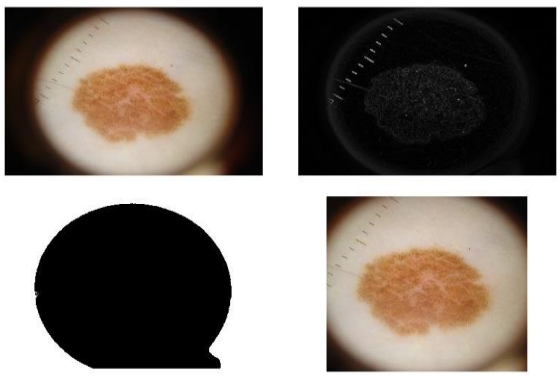

Fig 1. Cropping strategy for uncropped skin images: top left: original image, top right: elevation map, bottom left: watershed segmented image, bottom right: cropped image

As the dataset contains a mixture of cropped and uncropped images, the first step we do is use a cropping strategy to deal with uncropped images as illustrated in Fig. 1. These uncropped images often show large, black mass. Crop algorithm by [12] is partially adapted in this work to identify the lesion region. First, an elevation map is created using the Sobel gradient of the image. Sobel filter allows us to detect the edges in the image. Then the markers are found for the background and the lesion region (foreground) using a very low threshold value (0.3). The threshold value is computed experimentally. Watershed transform is used to then fill regions of the elevation map starting from the markers determined in the previous step. This will give us the segmented lesion. The next step is to find the bounding box for the segmented lesion. To calculate the bounding box, we apply the connected component labeling algorithm. This assigns the same label to all the pixels that are in the same region. In this way, different regions obtained after segmentation are labeled using distinct labels. This also allows us to access the properties associated with each region like area, extent, major and minor axis, and so on. Next, to identify the lesion region among others, we consider the area (the number of pixels of the region) and extent (the ratio of pixels in the region to pixels in the total bounding box) features of each region. The region having the largest area and extent > 0.5 is considered as the target. Otherwise, we consider the three largest regions, and the first one that has an extent > 0.5 is taken as the target lesion region. Once the target region is found, to get the bounding box for that region, we find the centroid (center of mass) and the major and minor axis of that region. Based on these values, the bounding box for cropping the lesion area is calculated. Furthermore, we resize all the images in the dataset to a constant size of 480 pixels (HAM1000 resolution as reference) preserving the aspect ratio.

In this work, we mostly rely on the Inception family of networks that first introduced 1x1 convolution for dimensionality reduction in its Inception block which is capable of performing multi-level feature extraction. The Inception module is based on a pattern recognition network that mimics the animal visual cortex. Inception models are trained on the ImageNet dataset and were selected because of their high-performance accuracies on ImageNet challenge. We have used Inception-v3, Inception-Resnet-v2, and Xception models from the Inception family of models. Another model Densenet was also selected due to its high performance in classification tasks. This also brings variability in our final ensemble. The specification of

these models and the number of trainable parameters is given in the Table II.

TABLE II. CNN ARCHITECTURES SPECIFICATIONS FOR TRANSFER LEARNING

Architecture	Input Size	Trainable Parameters
DenseNet201	224 × 224	19.3 M
Inception-v3	299 × 299	23.63 M
Inception-ResNet-v2	299 × 299	55.87 M
Xception	299 × 299	22.85 M

In this section, we will describe the modifications done on these architectures to perform the classification of skin lesion images into 8 categories. All of these CNN models have been pre-trained on the ImageNet dataset and then on a reduced ISIC-2019 train-split dataset. A few modifications are done to adapt the model to our use cases. The original classifier part of these pre-trained CNNs is replaced by batch-normalization layer followed by dropout layer, then a global average pooling layer, followed by two dense layers of 512 neurons, a second dropout layer, and lastly the classification layer with softmax as activation function to classify the dermoscopic images into 8 categories as shown in Fig 2. The softmax layer gives a prediction score in the range (0.0 to 1.0) distributed over the list of classes. All these models follow the same train pipeline with identical settings. Given a training set $D_{train} = \{x_i, y_i\}_{i=1}^{N}$, where x denotes the input image of dimension d, y its corresponding label, and N is the number of input images, first, the images are cropped before passing them for training.

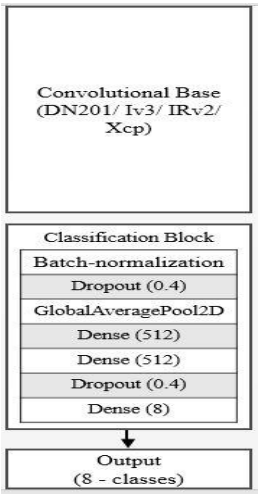

Fig 2. Eight Modified classification block of base-classifiers

The cropped images are then standardized and normalized. Normalizing is done to remove any bias present in the data [13].

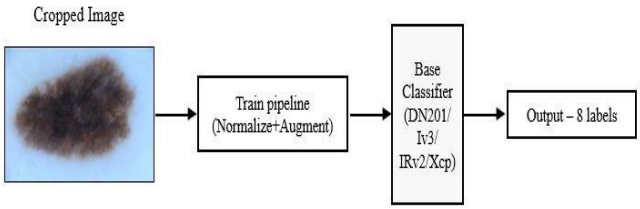

Fig 3. Train pipeline for each base-classifier

The normalized images are then passed to the augmentation pipeline where vertical and horizontal flipping, rotation, zooming, shearing, and width-shift augmentations are applied. In this work, an online-augmentation policy is employed meaning, the augmentations are applied on the fly when the training process takes place. This does not increase the image count rather different versions of augmented images are presented during each pass (epoch) of the training data. These augmented images are then fed to the pre-trained CNN models for training as illustrated in Fig 3.9. The saved models are then loaded and embedded to a meta-classifier and trained using the proposed algorithm. In this work, the blending procedure is implemented to ensemble the predictions done by the base-classifiers. It is a two-level process: level-0 and level-1. The four CNN pre-trained models: DenseNet201, InceptionV3, InceptionResNetV2, and Xception are used as level-0 sub-models to make the initial predictions. These models are trained on the train-split part of the training data and the best versions of each model are saved separately. As neural networks are used as sub-models, we have used a non-linear multi-layer perceptron (MLP) model as a meta-learner (level-1 model) having layers as depicted in Fig. 3, as against the practice of using a linear model like Logistic Regression. Now, this allows sub-networks to be embedded in a larger multi-headed network that then learns how best to combine the predictions from each input sub-model and the ensemble can be treated as a single large model. In our work, each of these sub-models is used as a separate input-head to the meta-learner. All the layers of the loaded models are marked as not trainable so the weights cannot be updated when the blended ensemble model is being trained. The entire blended model is then fitted onto the holdout-split part of the training data. Because the sub-models are not trainable, their weights will not be updated during training and only the weights of the meta-learner will be updated. Thus, the sub-models only predict the holdout-split dataset and the meta-learner will be trained on the predictions made by these sub-models on the holdout-split of the training data. This new ensemble model is used to make predictions on the test dataset. The algorithm I for the blending strategy is given in Table III. The algorithm describes the process to fuse the base classifiers to meta-learner and train the meta-learner using the stacked predictions from the base classifiers. The final predictions are then made on the holdout dataset and test dataset.

TABLE III. ALGORITHM FOR BLENDING

Algorithm **Blending**

Input: training data, $D_{train} = \{x_i, y_i\}_{i=1}^{k}$
level-0 classifiers, $C_{L0}^{1}, \ldots, C_{L0}^{T}$
level-1 classifier, C_{L1}
Output: ensemble classifier E
Split D_{train} into 2 parts, $D_T = \{(x_n, y_n), n = 1, \ldots, N\}$, $D_{HO} = \{(x_m, y_m), m = 1, \ldots, M\}$
Step 1: learn base (level-0) classifiers
for $t \leftarrow 1$ to T do
    learn base classifiers C_{L0}^{t} based on D_T
end for
Step 2: construct new data set from the predictions of D_{HO} and learn meta-classifier, E
embed base classifiers $C_{L0}^{1}, \ldots, C_{L0}^{T}$ in C_{L1} to get ensemble classifier E
for $t \leftarrow 1$ to T do
    for $i \leftarrow 1$ to M do
        $D_E = \{x_i', y_i\}, where\ x_i' = stack\{C_{L0}^{t}(x_i)\}$
    end for
end for
learn E based on D_E
return E

III. RESULTS AND DISSCUSSION

The dataset is split into two parts, train and holdout (validation) splits. The dataset is split in the ratio of 80:20. The train-split part of the dataset is used to train the level-0 models (DenseNet201, InceptionV3, InceptionResNetV2, and Xception) or sub-models of the blending ensemble model. The validation-split part of the dataset is used by the sub-*models* to make predictions, which are then concatenated to form the train dataset for the level-1 model or meta-model. The final predictions are then made on the holdout-split dataset and test dataset. Since the data is extremely imbalanced, for training we use cross-entropy loss. Each of the class will be set with different weights and is defined in (1),

$$W_i = \frac{N}{C + n_i} \tag{1}$$

where C is the number of categories, n_i is the image count per category and N is the total number of images in the training data set.

A. CNN training Base Models

We leverage on transfer learning to extract features and perform classification. The training of the models is done in 2 steps. In step 1, all the layers in the base models just below the top fully connected (FC) layer are frozen. This allows the base models to behave as initial feature extractors. The training is done for 3 epochs with the learning rate set to 10e-3 and batch size to 64 images. Adam optimizer is used to regulate the learning rate. This step is performed so that the weights of the newly added layers in the classification block does not get randomly initialized. This will also prevent large fluctuations in the gradients' updates when doing fine-tuning. In step 2 of the training process, all the layers of the base models are made trainable and fine-tuned for 50 epochs. The fine-tuning step starts from the 4th epoch onwards and the learning rate is decreased by a factor of 10 and set to 10e-4. Again the other training parameters remain the same with Adam as optimizer and batch size set to 64.

The validation loss is constantly monitored for each epoch during training and the learning rate is set to decrease by a factor of 10 if the validation loss stopped improving after 8 epochs till it reaches 10e-6. To make sure that the model does not over-fit, early stopping is used to monitor the validation loss. Three more checkpoints are used to save the model with the best-balanced accuracy and also the model with minimum validation loss. Finally, the last model after the end of the training was also saved as the latest model. Keras API's with Tensorflow as the backend is used for the above implementation.

Meta-learner training is carried out similarly as the base model training except for the initial 3-epochs training. Before training the meta-learner, all the saved base models with best-balanced accuracy are initially loaded and embedded in a multi-layered perceptron (MLP) model to form a single ensemble model with multiple heads like a hydra. When merging all base-classifiers, the layers of these models are frozen so that their weights do not get updated during the training of the ensemble model.

B. Analysis of the performance of base-models and ensemble model for dataset-1 and dataset-2

Using the configuration described in subsection A, the base models and ensemble model was trained for classification using dataset-1 and dataset-2. The *balance multi-class accuracy and* loss metrics are plotted in the graph at the end of each epoch, as illustrated in Fig 4. The prediction performance of the ensemble model for

dataset-1 and dataset-2 and also the predictions on the test dataset is given in the Fig 5 and Fig 6.

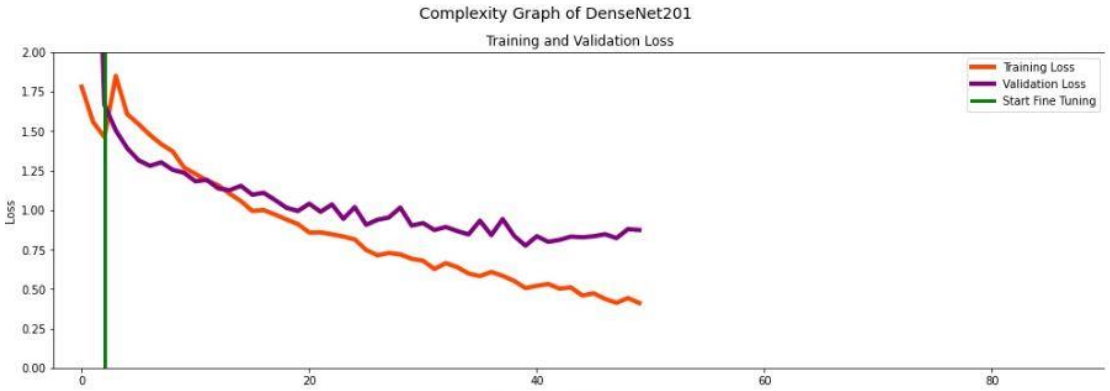

Fig 4 Train and validation loss of DenseNet201 for dataset-1

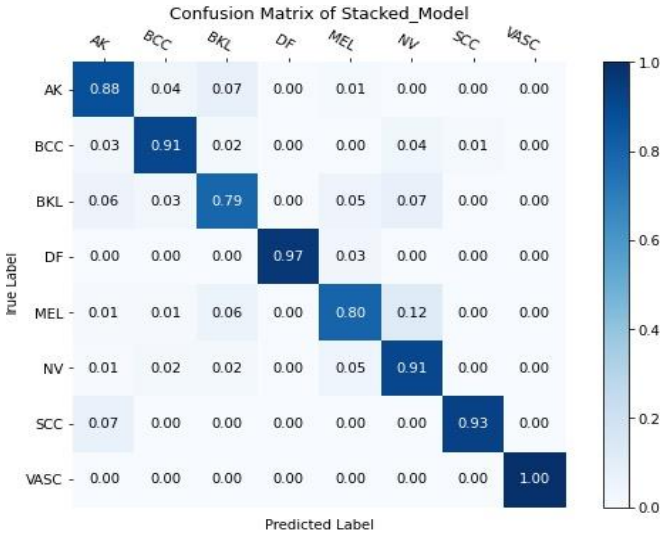

Fig 5. Confusion matrix: Prediction performance of ensemble model on validation-split of dataset-1.

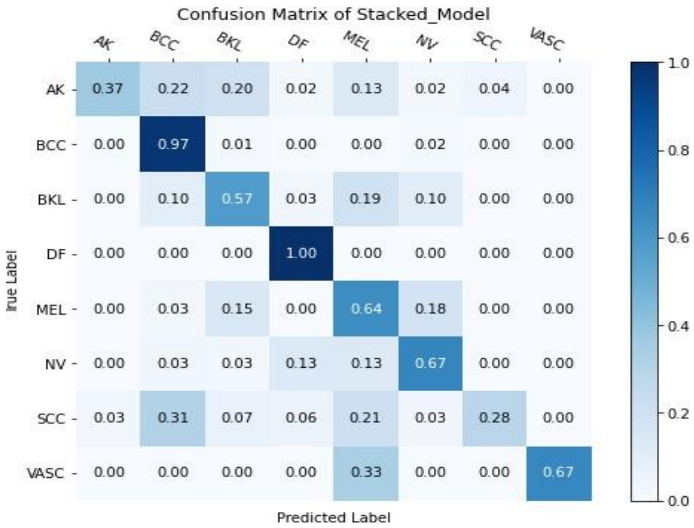

Fig 6. Confusion matrix: Prediction performance of ensemble model on the test dataset-1.

The evaluation results, balance multi-class accuracy, as well as the results of the secondary metrics for the base models and ensemble model on the validation set and test set are given in Table IV and V.

TABLE IV. PERFORMANCE METRICS OF MODELS TRAINED ON DATASET-1

Models	**Performance metrics (%)**				
	BMA	***P***	***Re***	***Sp***	***F1***
DenseNet-V2	**69**	69	59	79	62
Inception-V3	**61**	66	54	78	57
Inception-ResNetV2	**61**	66	54	79	57
Xception	**59**	66	47	75	52
Ensemble model	**64**	**69**	**60**	**78**	**61**

TABLE V. PERFORMANCE METRICS OF MODELS TRAINED ON DATASET-2

Models	**Performance metrics (%)**				
	BMA	*P*	*Re*	*Sp*	*F1*
DenseNet-V2	**79**	71	62	98	58
Inception-V3	**77**	68	59	97	55
Inception-ResNetV2	**79**	68	59	98	55
Xception	**72**	62	55	98	49
Ensemble model	**81**	**73**	**62**	**98**	**56**

From the above table, we can conclude that the ensemble model performs reasonably better on the unseen test images than the base-models. We also observe that the models learn better as the train data-size increases, giving better performance results. Our ensemble model got the *BMA* of 64.7% when trained on dataset-1 (4,999 images), however, its performance increased with the increased dataset (7,900 images). Note that, dataset-2 is more skewed than dataset-1. One more point to note is that DenseNetV2 performance is

better than other models used as base-models for our ensemble network. Perhaps, the performance of the ensemble network will improve further on increasing the number of base-classifiers.

IV. Conclusion

On comparing the prediction results obtained, it can be summarized that the ensemble model trained using the blending technique proposed in this work performs reasonably better than the individual models. Blending based ensemble technique used in this work is one of the ways to train the ensemble model. Another well-known technique Stacking or Stacked generalization can also be used with k-fold validation. The analysis metrics got for both base classifiers and ensemble model during predictions done on test set indicates that the models perform reasonably well. There is still scope for further improvements by varying (increasing/changing) the base classifiers used. The dataset can be expanded to include all the images from the original dataset for better prediction. The metrics have shown that though the ensemble model behaves reasonably, there is still scope for improvement.

References

[1] Pathan, S., Prabhu, K. G., & Siddalingaswamy, P. C. (2018). Techniques and algorithms for computer aided diagnosis of pigmented skin lesions—A review. *Biomedical Signal Processing and Control*, *39*, 237-262.

[2] Pathan, S., Prabhu, K. G., & Siddalingaswamy, P. C. (2019). Automated detection of melanocytes related pigmented skin lesions: A clinical framework. *Biomedical Signal Processing and Control*, *51*, 59-72.

[3] Pathan, S., Prabhu, K. G., & Siddalingaswamy, P. C. (2018). Hair detection and lesion segmentation in dermoscopic images using domain knowledge. *Medical & biological engineering & computing*, *56*(11), 2051-2065.

[4] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] N. C. B, J. Cai, M. Abedini, and R. Garnavi, "Deep Learning , Sparse Coding , and SVM for Melanoma Recognition in Dermoscopy Images Deep Learning , Sparse Coding , and SVM for Melanoma Recognition in Dermoscopy Images," no. October, 2015.

[6] Y. C. Lee, S.-H. Jung, and H.-H. Won, "WonDerM: Skin Lesion Classification with Fine-tuned Neural Networks," pp. 1–4, 2018.

[7] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, "Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data," pp. 1–10, 2019.

[8] A. F. Frangi, J. A. Schnabel, C. D. C. Alberola-lópez, G. F. Eds, and D. Hutchison, *and Computer Assisted Intervention – MICCAI 2018*. 2018.

[9] M. Combalia *et al.*, "BCN20000: Dermoscopic Lesions in the Wild," pp. 3–5, 2019.

[10] P. Tschandl, C. Rosendahl, and H. Kittler, "Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, pp. 1–9, 2018.

[11] N. C. F. Codella *et al.*, "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)," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2018-April, pp. 168–172, 2018.

[12] B. Montaruli, "Skin Lesions Classification using Computer Vision and Convolutional Neural Networks Image Processing and Artificial Vision Master Degree in Computer Science Engineering Polytechnic University of Bari," vol. 10000, no. 2018, pp. 1–33, 2019.

[13] Convolutional Neural Networks for Visual Recognition, "https://cs231n.github.io/".

[14] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, "Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data," pp. 1–10, 2019.

[15] A. F. Frangi, J. A. Schnabel, C. D. C. Alberola-lópez, G. F. Eds, and D. Hutchison, *and Computer Assisted Intervention – MICCAI 2018*. 2018