# A comparative study of convolutional neural networks for classification of pigmented skin lesions

Natalia Camillo do Carmo, João Fernando Mari
*Instituto de Ciências Exatas e Tecnológicas*
Universidade Federal de Viçosa
Rio Paranaíba, Brazil
Email: {natalia.camillo, joaof.mari}@ufv.br

***Abstract*—Skin cancer is one of the most common types of cancer in Brazil and its incidence rate has increased in recent years. Melanoma cases are more aggressive compared to non-melanoma skin cancer. Machine learning-based classification algorithms can help dermatologists to diagnose whether skin lesion is melanoma or non-melanoma cancer. We compared four convolutional neural networks architectures (ResNet-50, VGG-16, Inception-v3, and DenseNet-121) using different training strategies and validation methods to classify seven classes of skin lesions. The experiments were executed using the HAM10000 dataset which contains 10,015 images of pigmented skin lesions. We considered the test accuracy to determine the best model for each strategy. DenseNet-121 was the best model when trained with fine-tuning and data augmentation, 90% (k-fold cross-validation). Our results can help to improve the use of machine learning algorithms for classifying pigmented skin lesions.**

***Index Terms*—Skin cancer, machine learning, convolutional neural networks, classification.**

## I. Introduction

The skin can be affected by different types of tumors with different levels of aggressiveness. These tumors are usually classified as melanoma skin cancer or non-melanoma skin cancer (all tumors except the melanoma type). Non-melanoma is the most frequent skin cancer, corresponding to about 30% of all malignant tumors registered in Brazil. Although non-melanoma has a low mortality, if not properly treated it can cause significant mutilations [1]. Melanoma is the least common type among the other types of skin cancer, but it is the most dangerous as it is much more likely to spread to other parts of the body if it is not detected and treated early [2].

Visual inspection of skin lesions consists of comparing the lesion with the surrounding normal tissue and is subject to misdiagnosis. Dermatoscopy is an examination method that uses a dermatoscope to allow a magnified view of the skin's surface while the reflection is reduced or filtered. Although dermatoscopy exam is more effective compared to visual examination, dermatologists still face challenges to improving the diagnosis of skin cancer. Manual inspection of dermatoscopic images by specialists is often complex, prone to failure, time-consuming, and subjective (may produce different results) [3]. Therefore, automated diagnosis using image classification algorithms has become important. Artificial intelligence and machine learning techniques enable the creation of classifiers that can be used as a computer-assisted diagnosis (CAD), supporting dermatologists' decisions [4].

The objective of this work is to improve the available information about using convolution neural networks (CNNs) for the automatic classification of skin lesions. We exploit this problem by comparing the performance of four CNN architectures, the impact of training the networks from Scratch Vs. using fine-tuning as well as the impact of training with data augmentation.

The remaining of the paper is organized as: In Section II we present some related works. The material and methods are described in Section III. In Section IV the results are presented and discussed and we conclude the study in Section V.

## II. Related Works

Kassani and Kassani [5] compare CNN architectures for detecting melanoma in skin lesions. The work compares the architectures: AlexNet, VGG-16, VGG-19, ResNet-50, and Xception. The dataset used was the HAM10000, the same dataset used in our work. The result shows that the ResNet-50 neural network had the best performance among the architectures. Differently, our work explores other training strategies and considers a deeper architecture, DenseNet-121.

Le et al. [6] applies class-weighted and focal loss functions along with transfer learning to increase performance and deal with problems such as class imbalance. Dropout layers were also implemented, which randomly ignore some connections during the training phase. The authors used a pre-trained ResNet-50 model with some modifications, such as using global average pooling (GAP) instead of traditional pooling layer.

Mohamed et al. [7] used the Densenet-121 and MobileNet, both pre-trained with the ImageNet dataset. Dataset down-sampling and data augmentation were applied to handle unbalanced data. MobileNet surpassed the highest accuracy of ISIC 2018 challenge by 4.2 p.p. The work of [7] is used as a reference for some common accuracy and sensitivity averages in the literature.

The union of two neural networks forming a new model was addressed in [8]. Fine-tuning training strategy was applied with the HAM10000 dataset and classic preprocessing techniques, but without increasing the number of images.

Rezvantalab et al. [9] uses the HAM10000 along with another dataset and compared four classic CNN architectures:

Inception-v3, InceptionResNet-v2, ResNet-152 and DenseNet-201. The best architecture was DenseNet-201. Their results surpassed the diagnoses made by dermatologists.

Chaturvedi et al. [10] used the MobileNet CNN architecture to classify the HAM10000 dataset. They also created a Web application integrated with the MobileNet model. The trained model achieved 83.1% accuracy. The models were trained using transfer learning and image preprocessing. Duplicated images (different images of the same lesion) were excluded from the validation set.

## III. Material and Methods

In this section, we describe the image dataset used in the experiments, the CNN architectures, training and validation methods, and data augmentation strategies. Image normalization was applied in all experiments.

### A. Dataset

We used the HAM10000 dataset ("Human Against Machine with 10000 training images")[1]. The HAM10000 is composed of 10,015 dermatoscopic images that were collected over 20 year in different populations and with different acquisition and storage methods. More than 50% of lesions were confirmed by histopathology, while the ground-truth for the remaining lesion were by follow-up examination, expert consensus, or in-vivo confocal microscopy [11].

The dataset metadata contains information about the diagnosis (the classes the stains belong to), the age of the person from whom the stain was collected, biological sex, type of diagnostic procedure, and the part of the body where the stain was.

Each image in the HAM10000 dataset belongs to one of the seven classes, which correspond to relevant pigmented lesions: Actinic keratoses and intraepithelial carcinoma/ Bowen's disease (akiec); basal cell carcinoma (bcc); benign keratosis-like lesions (bkl); dermatofibroma (df); melanoma (mel); melanocytic nevi (nv) and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas, and hemorrhage) (vasc). Figure 1 illustrates one sample from each one of the classes.

[1]Available at: https://bityli.com/FlYo6Y

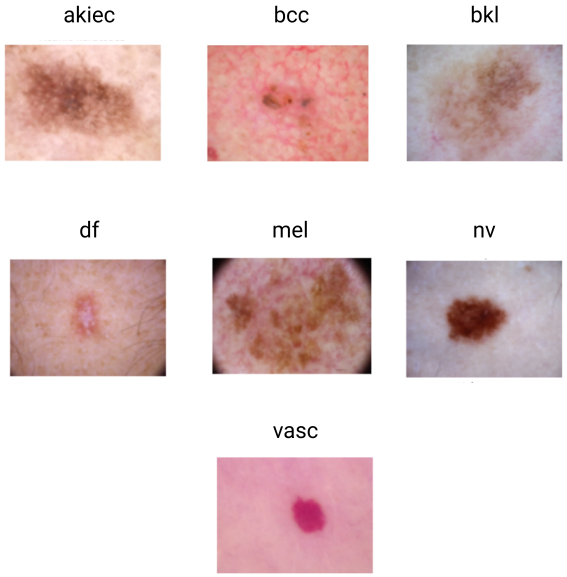


Fig. 1: One sample from each class of pigmented lesion in the HAM10000 dataset.

There is an imbalance in the classes, as seen in Figure 2, which is handled with the data augmentation method.

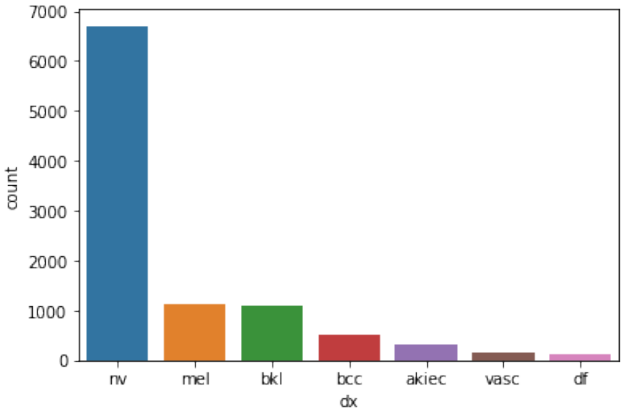


Fig. 2: Unbalance of classes in the HAM10000 dataset.

### B. Architectures

One of the fundamental factors in determining the performance and efficiency of a CNN is its architecture [12]. How the layers are structured, the elements used in each layer, and how they are designed often affect how quickly and accurately it can perform various tasks [13].

The VGG-16 [14] achieved one of the best performances in the 2014 *ImageNet* Large Scale Visual Recognition Challenge (ILSVRC). The architecture uses filters smaller than 3×3 to better extract image features. Studies have found that using

smaller filters to increase network depth is more effective than increase network width [5].

ResNet [15] uses residual connections to avoid the vanishing gradient effect in very deep networks. Residual connections skip some layers and feed the output of one layer as input to the next layers, which allows this architecture be deeper than other CNN architectures, such as VGG [5], [16]. We used the ResNet-50, with 50 layers, but there are variations such as ResNet-101 with 101 layers [5], [16].

Inception's architecture [17] introduced the Inception modules which computes parallels convolutions with different kernel sizes and pooling layers. The version used in this work is the Inception-v3.

DenseNet [18] concatenates the resulting output from the previous layer with the next one, while ResNet had an additive method to merge its previous layers with the next one. DenseNet concatenates the feature maps while preserving the information, improving feature reuse. In this work, we used the DenseNet-121, with 121 convolutional layers.

### C. Experimental Setups

To perform the experiments, the dataset (described in Section III-A) was treated using two validation methods: holdout and k-fold cross-validation. In the holdout method, the dataset was split in 75% of the images for training, 15% for validation, and the remaining 15% for testing. In the k-fold method, the dataset was split into five non-overlapping folds of equal size, and in each iteration, one fold is used for testing, while the remaining is used for training. The selected number of folds was 5 ($k$ = 5).

In order to improve the classification performance of the models, we performed hyperparameter optimization using random search on the training and validation sets (holdout validation method). As the stochastic gradient descent with momentum (SGDM) optimizer was used, the hyperparameter and respective search spaces were as follows:

- Batch size: {8, 16, 32, 64};
- Learning rate: between 0.0001 and 0.1;
- Momentum: between 0.6 and 0.98;
- Step size: integer values between 5 and 45;
- Weight decay: fixed in 0.1.

The hyperparameter optimization was executed during 30 epochs. The set of parameter values that achieved the best validation accuracy were the same for all CNN architectures: learning rate = 0.0101, momentum = 0.6214, step size = 25 and batch size = 32. These values were used in all training strategies described below.

We designed three training strategies for the experiments. In *strategy 1* the CNN models were trained from scratch (no fine-tuning) and without data-augmentation, in *strategy 2* the models were trained from scratch, but now with data augmentation, and finally, in *strategy 3* the models were trained with fine-tuning and data augmentation. The idea behind this experiment design is to assess the impact of data augmentation and fine-tuning on the model classification performance. *Strategies 1, 2, 3* were applied using the holdout validation method, and *strategy 3* was also applied using the $k$-fold cross-validation method. Figure 3 illustrates the described experimental setup.

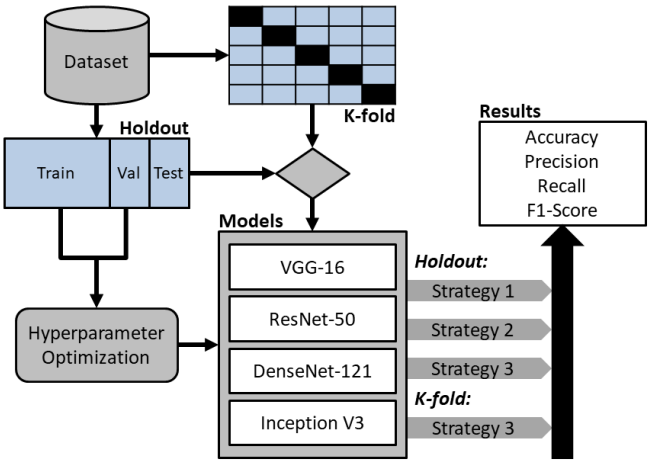


Fig. 3: Workflow of the experimental setup.

### D. Data Augmentation

Overfitting usually occurs when there are a small number of training examples. One way to solve this problem is to increase the dataset to have a sufficient number of training samples. Data augmentation is the process of generating more training images from existing training samples through a series of random transformations that produce reliable-looking images. This procedure enables exposing the model to more aspects of the data improving the model generalization [5]. In this work, we applied randomly horizontally and vertically flips and random rotations. In Figure 4 it is shown samples of the augmented images in one single batch.

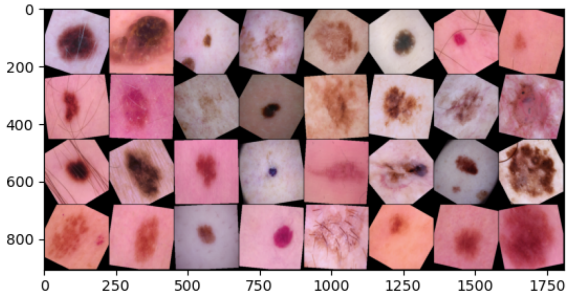


Fig. 4: Augmented images from one training batch.

### E. Fine-tuning

As a transfer learning strategy, fine-tuning consists of using a network that has been previously trained for a classification task using a very large dataset, such as ImageNet [12]. The fine-tuning strategy adopted here consists of not freezing any layers and training on the pre-trained model using our image dataset. In this work, we used fine-tuning to train the four CNN models. The last layer of the pre-trained networks was

replaced by a new softmax layer to fit the number of classes in the present working problem.

### F. Computational Resources

The experiments were developed using Python 3.7, PyTorch 1.8.1, and TorchVision 0.9.1. Scikit-learn 0.24.2 and matplotlib 3.4.2 were also used. The experiments were performed on a machine running Ubuntu 16.04.2 LTS with an Intel i5 3.00 GHZ processor, 32 GB RAM, and a NVIDIA Titan XP GPU with 12 GB.

## IV. Results and Discussion

In this section, we present and discuss the results obtained by applying the strategies described in Section III-C considering the holdout and the k-fold cross-validation methods: *strategy 1*: training the models from scratch (no data augmentation); *strategy 2*: training from scratch with data augmentation and *strategy 3*: fine-tuning with data augmentation.

Table I presents the results of all experiments in terms of test set accuracy for the holdout validation method and in terms of accuracy mean and standard deviation of the five validation folds for the k-fold cross-validation method.

TABLE I: Results of all experiments in terms of accuracy.

| | Holdout | | | Cross-validation |
|---|---|---|---|---|
| | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 3 |
| Model | Test Acc | | | Test Acc |
| VGG-16 | 0.7725 | 0.7698 | 0.8636 | 0.8704 ± 0.0088 |
| ResNet-50 | 0.7904 | 0.7558 | 0.8776 | 0.8929 ± 0.0050 |
| DenseNet-121 | **0.8127** | **0.7904** | 0.8776 | **0.9000** ± 0.0083 |
| Inception-v3 | 0.7734 | 0.7738 | **0.8975** | 0.8993 ± 0.0054 |

Considering the holdout validation method, the DenseNet-121 model performed better when trained with *strategy 1* and *strategy 2*, with 0.8127 and 0.7984 of accuracy, respectively. Surprisingly, the accuracy for all model has decreased when trained with data-augmentation, with exception of Inception-v3 which remain practically unchanged. One possible explanation is the choice of a very naive set of transformations (Section III-D). Other transformations sets or even auto-augmentation methods [19], should be experimented in future works.

Still considering the holdout validation method, Inception-v3 had the best accuracy when training with *strategy 3*. It is possible to observe that each model had its accuracy improved when trained with fine-tuning.

Only the *strategy 3* was applied using the k-fold cross-validation. Cross-validation seeks to avoid biases in the validation process, as it guarantees that each sample participates in the training and validation processes at least once [12], [5]. Standard deviation was calculated to show the dispersion of data within the population relative to the mean. It is possible to observe that all standard deviations are very low, below 0.0088.

Figure 5 shows the evolution of accuracy and loss curves during training DenseNet-121 with fold 4 under *strategy 3*, in which it is possible to observe that overfitting does not occur.

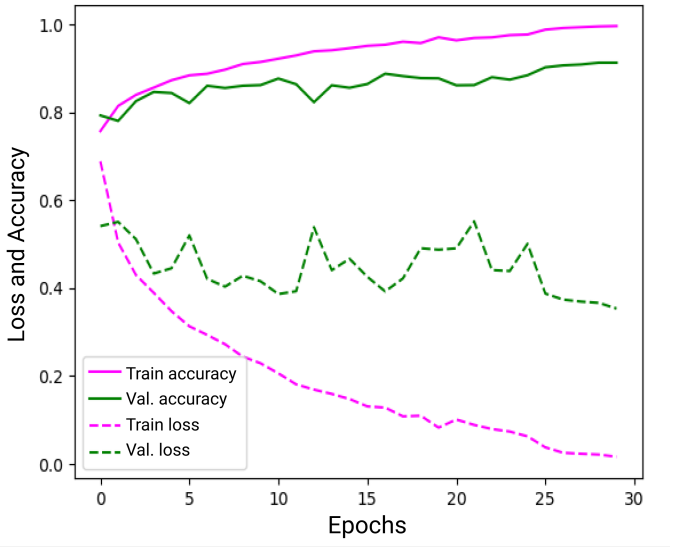


Fig. 5: Accuracy and Loss evolution during the DenseNet-121 model training (cross-validation in fold 4 with *strategy 3*).

Table II shows the confusion matrix of the Inception-v3 when trained with the holdout method. Inception-v3 trained using *strategy 3* had the highest accuracy considering the holdout method.

TABLE II: Inception-v3 confusion matrix using holdout and trained with *strategy 3*.

| Inception-v3, holdout + strategy 3 | | | | | | | |
|---|---|---|---|---|---|---|---|
| | akiec | bcc | bkl | df | mel | nv | vasc |
| akiec | **0.673** | 0 | 0.184 | 0 | 0.102 | 0.020 | 0.020 |
| bcc | 0.039 | **0.766** | 0.065 | 0 | 0.039 | 0.090 | 0 |
| bkl | 0.018 | 0 | **0.860** | 0.006 | 0.048 | 0.067 | 0 |
| df | 0 | 0.055 | 0 | **0.778** | 0.055 | 0.111 | 0 |
| mel | 0 | 0.012 | 0.054 | 0.006 | **0.736** | 0.186 | 0.006 |
| nv | 0 | 0.001 | 0.025 | 0.001 | 0.030 | **0.942** | 0.001 |
| vasc | 0 | 0 | **0** | 0 | 0 | 0 | **1** |

Table III shows the confusion matrix obtained by the DenseNet-121 when trained with k-fold cross-validation, considering the fold 4 for validation. DenseNet-121 was the model with the highest accuracy using k-fold cross-validation, remembering that only strategy 3 was applied using cross-validation.

TABLE III: DenseNet-121 confusion matrix using cross-validation with strategy 3, considering fold 4

| DenseNet-121, cross-validation + strategy 3 | | | | | | | |
|---|---|---|---|---|---|---|---|
| | akiec | bcc | bkl | df | mel | nv | vasc |
| akiec | **0.708** | 0 | 0.123 | 0.015 | 0.092 | 0.062 | 0 |
| bcc | 0.019 | **0.884** | 0.039 | 0 | 0.029 | 0.029 | 0 |
| bkl | 0.009 | 0.009 | **0.864** | 0 | 0.054 | 0.064 | 0 |
| df | 0.044 | 0.130 | 0.044 | **0.652** | 0 | 0.130 | 0 |
| mel | 0.009 | 0 | 0.059 | 0 | **0.739** | 0.194 | 0 |
| nv | 0 | 0.004 | 0.010 | 0.001 | 0.018 | **0.966** | 0 |
| vasc | 0 | 0 | 0 | 0 | 0 | 0.103 | **0.897** |

Figure 6 shows examples of images correctly or incorrectly classified by the Inception-v3 trained with strategy 3 and holdout validation method.

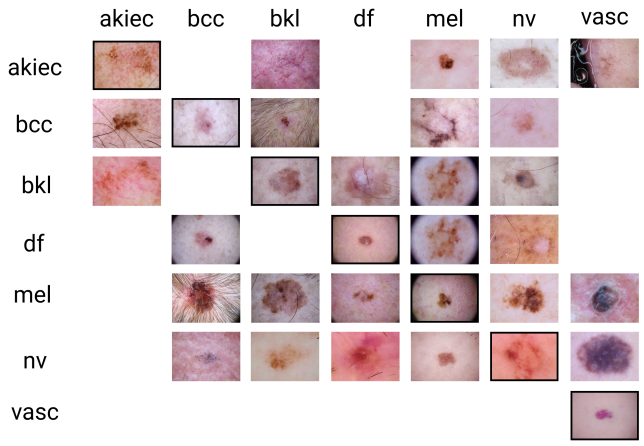


Fig. 6: Inception-v3 confusion matrix when trained with strategy 3 and holdout validation. Showing some correctly and incorrectly classified images.

Figure 7 shows examples of images correctly and incorrectly classified by the DenseNet-121 when trained using k-fold cross-validation.

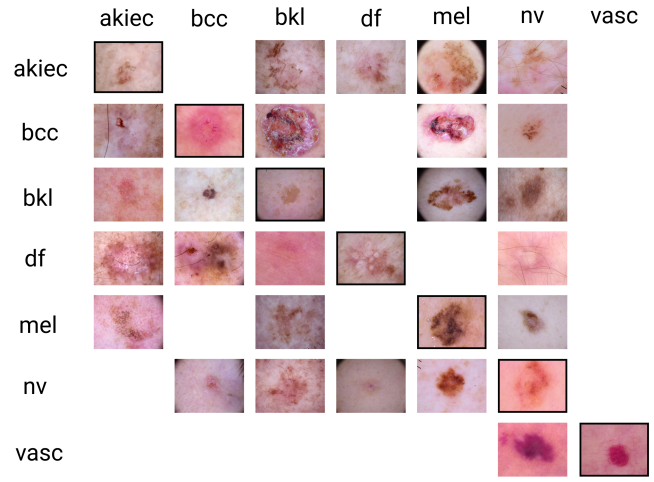


Fig. 7: DenseNet-121 confusion matrix when trained using cross-validation in fold 4. Showing some correctly and incorrectly classified images.

Table IV presents other evaluation metrics computed on the model with better results in terms of accuracy (see Table I) when using the k-fold cross-validation method – the DenseNet-121. We computed the per class precision, recall, and f1-score. The values in the table were calculated as the mean across the 5 folds, and the last row shows the overall metric value computed as the mean across all classes.

Table V presents other evaluation metrics computed on the model with better results in terms of accuracy (see Table I) using the holdout validation method.

TABLE IV: Classification report for DenseNet-121 training using cross-validation method with strategy 3.

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| akiec | 0.8010 | 0.6944 | 0.7401 | 327 |
| bcc | 0.8577 | 0.8424 | 0.8496 | 514 |
| bkl | 0.7941 | 0.8298 | 0.8115 | 1099 |
| df | 0.8281 | 0.7391 | 0.7769 | 115 |
| mel | 0.7741 | 0.7062 | 0.7382 | 1113 |
| nv | 0.9449 | 0.9603 | 0.9525 | 6705 |
| vasc | 0.9453 | 0.9301 | 0.9366 | 142 |
| accuracy | | | 0.9000 | 10015 |
| avg | 0.8493 | 0.8146 | 0.8293 | 10005 |

TABLE V: Classification report for Inception-v3 training using holdout method with strategy 3.

| | **precision** | **recall** | **f1-score** |
|---|---|---|---|
| akiec | 0.8462 | 0.6735 | 0.7500 |
| bcc | 0.9365 | 0.7662 | 0.8429 |
| bkl | 0.7474 | 0.8606 | 0.8000 |
| df | 0.8235 | 0.7778 | 0.8000 |
| mel | 0.7235 | 0.7365 | 0.7300 |
| nv | 0.9480 | 0.9423 | 0.9452 |
| vasc | 0.8750 | 1.0000 | 0.9333 |
| accuracy | | | 0.8916 |
| avg | 0.8429 | 0.8224 | 0.8288 |

Based on confusion matrices (Table II and Table III) and the per class evaluation metrics (Table IV), it is possible to observe that the models could classify each class very well.

## V. Conclusion

In this work we presented and evaluated methods to classify pigmented skin lesions using four CNN architectures: DenseNet-121, ResNet-50, VGG-16 and Inception-v3. A comparison was made between three training strategies considering holdout and k-fold cross-validation methods.

When we consider the experimented training strategies, training the models using fine-tuning and data augmentation (strategy 3) outperformed the other strategies for all CNN models. When performing a comparative analysis between the CNN models, DenseNet-121 performed better in most scenarios, achieving 90.0% accuracy when trained with fine-tuning and data augmentation using k-fold cross-validation method. In short, DenseNet-121 showed to be the best model for this classification task, but the other evaluated CNN models are able to obtain reasonable results. Taking the best training strategy (fine-tuning and data augmentation), DenseNet-121 (best model) achieved 90.0% and 89.75% accuracy versus 87.04% and 86.36% for VGG-16 (which had the worst results), a difference of 2.96 and 3.36 percentage point, for holdout and k-fold, respectively.

As future work we suggest: a) to consider other hyper-parameters optimization algorithms; b) experiment other image transformations for data-augmentation, including autoaugment; c) use the CNNs as a feature extractor; and d) testing

the proposed strategies on other skin cancer datasets, including datasets with samples of lesions in black skin.

## ACKNOWLEDGMENTS

We gratefully acknowledge the support of NVIDIA Corporation, Brazil with the donation of the TITAN Xp GPU used for this research. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

## REFERENCES

[1] INCA, 2020. [Online]. Available: https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-nao-melanoma

[2] A. C. Society, 2020. [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer.html

[3] M. A. Al-Masni, D.-H. Kim, and T.-S. Kim, "Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification," *Computer Methods and Programs in Biomedicine*, vol. 190, p. 105351, 2020.

[4] R. C. Maron, M. Weichenthal, J. S. Utikal, A. Hekler, C. Berking, A. Hauschild, A. H. Enk, S. Haferkamp, J. Klode, D. Schadendorf *et al.*, "Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks," *European Journal of Cancer*, vol. 119, pp. 57–65, 2019.

[5] S. H. Kassani and P. H. Kassani, "A comparative study of deep learning architectures on melanoma detection," *Tissue and Cell*, vol. 58, pp. 76–83, 2019.

[6] D. N. Le, H. X. Le, L. T. Ngo, and H. T. Ngo, "Transfer learning with class-weighted and focal loss function for automatic skin cancer classification," *arXiv preprint arXiv:2009.05977*, 2020.

[7] E. H. Mohamed and W. H. El-Behaidy, "Enhanced skin lesions classification using deep convolutional networks," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*. Cairo: IEEE, 2019, pp. 180–188.

[8] S. S. Chaturvedi, J. V. Tembhurne, and T. Diwan, "A multi-class skin cancer classification using deep convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–22, 2020.

[9] A. Rezvantalab, H. Safigholi, and S. Karimijeshni, "Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms," *arXiv preprint arXiv:1810.10348*, 2018.

[10] S. S. Chaturvedi, K. Gupta, and P. S. Prasad, "Skin lesion analyser: an efficient seven-way multi-class skin cancer classification using mobilenet," in *International Conference on Advanced Machine Learning Technologies and Applications*. India: Springer, 2020, pp. 165–176.

[11] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Massachusetts: MIT Press, 2016, http://www.deeplearningbook.org.

[13] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Beijing: IEEE, 2016, pp. 770–778.

[16] H. Lei, T. Han, F. Zhou, Z. Yu, J. Qin, A. Elazab, and B. Lei, "A deeply supervised residual network for hep-2 cell classification via cross-modal transfer learning," *Pattern Recognition*, vol. 79, pp. 290–302, 2018.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[19] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.