Utilization of Different Models and Convolutional Neural Network Techniques to Skin Lesion Classification

Henrique S. Siqueira[1], **Ademar T. Akabane**[1]

[1]Faculty of Computer Engineering
Pontifical Catholic University of Campinas (PUC-Campinas)
Professor Doutor Euryclides de Jesus Zerbini Street, 1516
Rural Fazenda Santa Cândida Park – 13.087-571 – Campinas – SP – Brazil

henrique.ss2@puccampinas.edu.br, ademar.akabane@puc-campinas.edu.br

Resumo. *Este estudo se propõe a investigação de modelos de redes neurais convolucionais, tais como o ConxNeXt, ResNetV2 e Xception, com a aplicação de técnicas de aumento de dados, segmentação e transferência de aprendizado, visando a classificação de lesões de pele, com ênfase no tipo mais letal: o melanoma. Os resultados obtidos destacam a importância da análise métrica, que deve ser conduzida abrangendo todas as métricas, tanto na perspectiva global quanto na perspectiva específica de cada classe.*

Abstract. *This study aims to investigate convolutional neural network models, such as ConxNeXt, ResNetV2, and Xception, with the application of data augmentation, segmentation, and transfer learning techniques, aiming at classifying skin lesions, with an emphasis on the most lethal type: melanoma. The results obtained emphasize the importance of metric analysis, which should be conducted by covering all metrics, both from a global perspective and from the specific perspective of each class.*

1. Introduction

Melanoma is the most deadly malignant cancer [Codella et al. 2016], having fewer recovery chances in case of late detection, making possible the metastasis manifestation of the disease. An individual with melanoma diagnosis has up to a five-year life expectancy, with a low probability of overcoming this timespan [Esteva et al. 2017] while detecting prematurely the recovery chances get up to 100% [Vasconcelos and Vasconcelos 2017]. However, the diagnosis of this type of cancer can lead to possible mistakes due to subtle differences over skin lesion categories [Romero Lopez et al. 2017]. These differences can be easily covered by the clinical images, in the perspective of illumination can compromise the image quality [Wu et al. 2022], For this reason, the ideal is the utilization of dermoscopy exams leading a clear skin vision and consequently increasing the professional accuracy [Brinker et al. 2019].

Artificial intelligence comes to support doctors in pattern recognition and display the probabilities of possible problems. In this scenario, the AI has the task of knowing and the doctor the task of knowing why the disease has been manifested [Lobo 2017], according to the lifestyle of the patient for example. Nevertheless, the doctor must validate the AI diagnosis, even though it becomes possible to have a fast response to the patient's case.

To make the machine work effectively, a large dataset is required, which is one of the problems, because each cancer type has a different incidence number as well as image examples, leading to an imbalance between skin cancer classes, especially melanoma. Besides that, computational resources, such as processing and memory, are extremely required to perform the model training in an acceptable period [Litjens et al. 2017].

You can access the code for this project on GitHub at the following URL: `https://github.com/h-ssiqueira/SkinLesionAI`

2. Related work

In [Haenssle et al. 2018] work, they compared the performance of the Inception V4, without any techniques, to fifty-eight dermatologists, using two small datasets, one of 100, from ISIC[1], and another with 300 images, from the University of Heidelberg, analyzed separately it shows the result of 95% recall, 63.8% specificity and 86% ROC for the ISIC dataset, the results from the other dataset recall, specificity and ROC are respectively 95%, 80% and 95%. Dermatologists performed better only in terms of specificity in comparison to the ISIC dataset.

In [Ünver and Ayan 2019] contribution, they explored the Yolo V3 model from ISBI and PH2 datasets with about three thousand images in total had trained each dataset separately and achieved 93.39% accuracy, 90.82% sensitivity, and 92.68% specificity for ISBI dataset, in terms of PH2 dataset the results are 92.99% accuracy, 83.63% sensitivity, and 94.02% specificity.

The [Jahanifar et al. 2019] have performed studies over ISIC datasets, totalizing 7473 images, with the models DenseNet169, ResNetV2, ResNet152, and Xception, taking advantage of the data augmentation, transfer learning, and segmentation techniques. The final results showed that DenseNet169 and ResNetV2 achieved the best accuracy with 96.58%, the ResNet152 model made the higher recall with 92% and the Xception model reached 97.30% specificity.

Similarly, [Delazeri and Stevani 2020] used the HAM10000 dataset with 10015 images using the VGG16, ResNet50, Inception-ResNet, and ResNet101 models making use of data augmentation and transfer learning. The results over precision were 84%, 74%, 76%, and 73% respectively to the models previously mentioned.

In the study of [Soenksen et al. 2021], they utilized the Xception and VGG16 models with transfer learning and geometric transformation as data augmentation techniques. Using a 34 thousand image dataset the models performed an AUC of 97% and 85.8%, a sensitivity of 90.3% and 83.7%, and a specificity of 89.9% and 87% for VGG16 and Xception models respectively.

3. Metodology

The following sections will explain briefly the metrics extracted from model training, techniques applied to improve the results, model architecture aspects, the dataset characteristics, and how the project was organized to perform the training for each model.

[1] https://challenge.isic-archive.com/data/

3.1. Metrics

Accuracy is the overall model performance, which represents the correct model prediction over true positives and true negative labels. **Precision** symbolizes the true positive predictions, isolating the false predicted labels. **Sensibility** or **recall** or also the true positive rate stands for the correct true predictions by the model. In contrast, **specificity** or true negative rate illustrates the correct false predictions by the model [Géron 2019]. These metrics can be noticed in the Figure 1.

The **F1 Score** is the harmonic mean of precision and sensitivity, this metric highlights the true positive against false negative and positive predictions, the formula is displayed in Equation 1.

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

The **ROC AUC** metric stands for receiving operating characteristic area under the curve. The ROC curve is the graph true positive rate (recall) versus the false negative rate (1 - specificity). The AUC is the area under the ROC curve which represents how capable the model is of distinguishing the classes in the evaluation.

Loss represents the sum of predicted errors, this metric over epochs penalizes the model making the optimizer adjust parameters to get a better evaluation. All training used the categorical cross-entropy loss, represented in Equation 2.

$$\text{Loss} = -\sum(\text{YTrue} \times log(\text{YPredicted})) \tag{2}$$

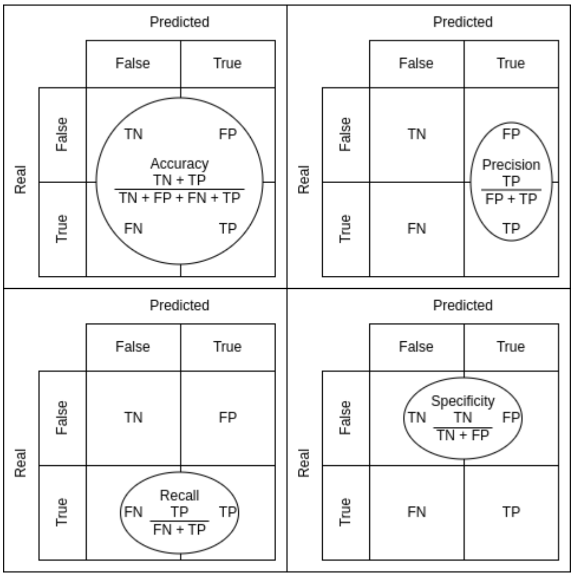
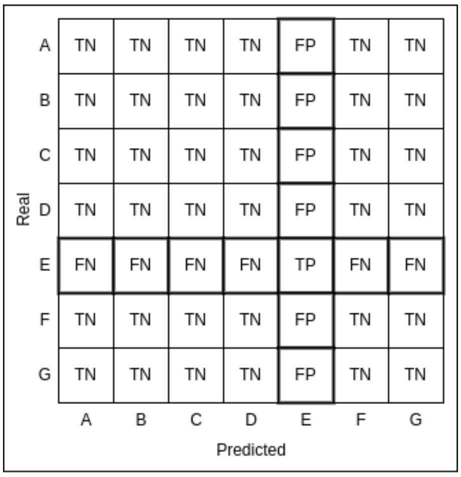


Figure 1. Metrics in the confusion matrix perspective.

Train and validation metrics can highlight possible overfittings, that is the model has a high variance between these two sets, this indicates a perfect modeling performance in only training data, which causes the model to fail when classifying new examples. In this case, a signal to prevent overfitting is observing the graphs after training, the loss between training and validation data needs to decrease over the epochs as the accuracies need to increase [Shorten and Khoshgoftaar 2019].

In the multi-class confusion matrix, each line can be analyzed separately. Each horizontal line represents the class recall, each vertical line represents the class precision, and the group of horizontal lines containing the true negative fields represents the specificity value of the class, the example is notable in Figure 2. The goal is to have the primary diagonal with the highest values, the perfect model training would have the multiclass confusion matrix as an identity matrix.



Real

	A	B	C	D	E	F	G
A	TN	TN	TN	TN	FP	TN	TN
B	TN	TN	TN	TN	FP	TN	TN
C	TN	TN	TN	TN	FP	TN	TN
D	TN	TN	TN	TN	FP	TN	TN
E	FN	FN	FN	FN	TP	FN	FN
F	TN	TN	TN	TN	FP	TN	TN
G	TN	TN	TN	TN	FP	TN	TN

Predicted

Figure 2. Label examples in the multiclass confusion matrix.

3.2. Techniques

The main techniques used in this study are data augmentation, segmentation, and transfer learning. Further information about each is presented in upcoming sections.

3.2.1. Data augmentation

The data augmentation technique has the objective of augmenting images to balance classes and extract more information from them. The generated images do not change the inherent meaning of the original image [Pham et al. 2018]. Geometric transformations such as zoom, color modifications, rotations, crop, blur, and bright are options to apply the data augmentation in the dataset.

Another method used in data augmentation is the Generative Adversarial Networks (GANs), which consists of two adversarial models.

1. The generator has the task of generating realistic new images based on the existent dataset in a precise way.
2. The discriminator evaluates the generated examples from the generator classifying if the image is actually from the original dataset or it was generated.

The generator training has the objective of maximizing the probability that the discriminator makes an incorrect evaluation [Chen et al. 2022]. It guarantees the asymptotic consistency with the Markov chain principle, where the probabilistic distribution of the next state depends uniquely on the actual state and does not include the previous states [Goodfellow et al. 2014]. Both agents must be trained together, none of them can overcome the other, that is, the two trainings are synchronized to avoid overfitting [de Oliveira 2021].

However, it is important to highlight that this technique needs to be applied only in the training set, promoting a data oversampling, due to validation and test set referring to simulating unique images. Also, applying this technique excessively can lead to an overfitting [Shorten and Khoshgoftaar 2019].

3.2.2. Segmentation

Segmentation technique consists of highlighting a certain group of pixels of an image, which contains similar characteristics [Ferreira 2018], in other words, it isolates the lesion region from the healthy skin allowing the model to extract information only from the lesion area [Moorthy and Gandhi 2022] [Codella et al. 2016]. However, issues can happen when applying segmentation due to low image quality [Kawahara et al. 2016]. An example of a segmentation mask is displayed in Figure 3.

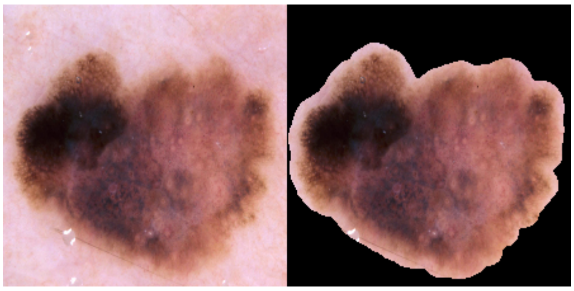


Figure 3. Example of segmenting a lesion image.

3.2.3. Transfer Learning

The transfer learning technique refers to loading pre-trained parameter weights from a generic and larger dataset, using them as initial parameters in the same model to perform specific classifications where the scope, including dataset size, is smaller than the original [Suganyadevi et al. 2022] [Shorten and Khoshgoftaar 2019]. The knowledge obtained by the previous training leads to a better performance over the actual learning phase, due to weight adjustments during the training, that is, the layers are not frozen to preserve the original parameters values transferred from the last train [Menegola et al. 2016]. In this study, transfer learning was used to fine-tune the existing model with medical images, adjusting the weights to achieve an improvement over the metrics.

3.3. Model Architectures

Three convolutional neural network models were selected for this project, and each of them is described in the next sections.

3.3.1. ConvNeXt

The ConvNext model was proposed in 2022 to modernize the ConvNet model family by merging aspects of residual networks and vision transformers [Dosovitskiy et al. 2021]. The model has 295 layers of which 36 convolutional layers and around 89 million parameters [Liu et al. 2022]. The model is designed to accept image input of 224 x 224 pixels.

3.3.2. ResNetV2

The ResNetV2 model was proposed in 2016 to upgrade the first version of the ResNet model, which won the 2015 Imagenet challenge [He et al. 2015]. It has 190 layers in which 53 convolutional layers are divided into 16 residual blocks and has around 23.5 million parameters [He et al. 2016]. Residual networks use shortcut connections in the blocks to skip layers. This approach does not modify the computational complexity nor the parameter numbers but performs identity mappings by preserving the input shape [He et al. 2015]. The model is designed to accept image input of 224 x 224 pixels.

3.3.3. Xception

The Xception model was proposed in 2016 to maximize the performance of the Inception V3. It uses the separable convolutional layers, which perform spatial convolutions for each channel of an input, every convolution component is independent from the other. The model has 132 layers in which 40 convolutional layers are divided into 14 blocks and has around 22.8 million parameters [Chollet 2017]. The model is designed to accept image input of 299 x 299 pixels.

3.4. Dataset

The HAM10000 dataset was used in this project, the selection was based on the reliability of the images, as the majority were reviewed by dermatologists to prove the real image classification for each image to avoid possible errors. Those images were collected over twenty years by the Department of Dermatology, Medical University of Vienna, and Skin Cancer Practice of Cliff Rosendahl, in Queensland, Australia [Tschandl et al. 2018]. This dataset contains 10015 images divided into seven skin lesion classes, which cover 95% of all skin lesion types examined in the clinics.

Each image was examined and checked manually to remove possible objects and blurs from the dataset [Tschandl et al. 2018]. However, there are only 7470 unique lesion images as represented in Figure 4. For this reason, just one example of each lesion identifier was used to prevent duplicity and ensure the integrity between image sets.

3.5. Model Training

A configurable notebook template was created for this project, to centralize the main settings for each training made. This template was initially tested with the dataset imported via Google Drive, using only the CPU to train the model, resulting in over 12 hours to conclude a single execution.

Table 1. Lesion classes present in the dataset.

Lesion Name	Lesion Abbreviation
Actinic Keratoses and Intraepithelial Carcinoma	akiec
Basal cell carcinoma	bcc
Benign keratosis	bkl
Dermatofibroma	df
Melanocytic nevi	nv
Melanoma	mel
Vascular lesion	vasc

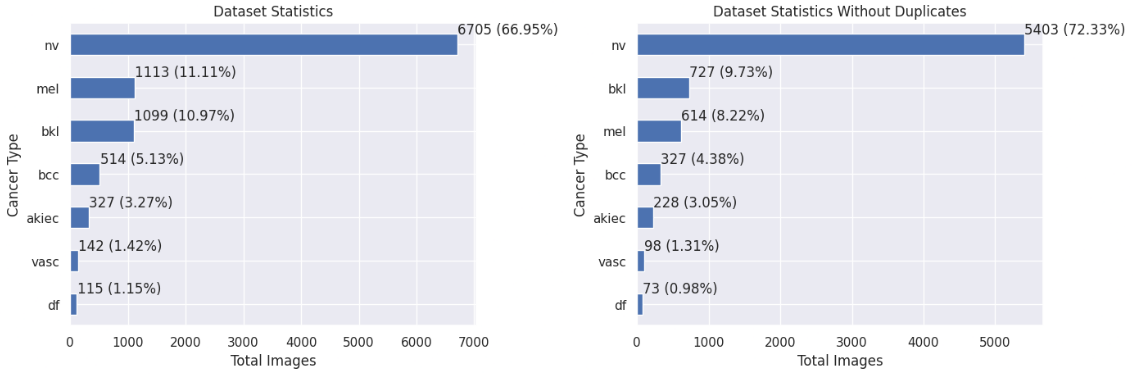


Figure 4. Dataset distribution classes, on the left graph the raw dataset, on the right the removed duplicate images.

Subsequently, the TPU architecture was tested, however, it was not possible to extract the same amount of metrics compared to the GPU. For this reason, the CNN training sessions were executed using GPU. In addition, to optimize time, a Kaggle integration was added in the template to download the dataset at runtime into the environment instead of importing it via the cloud.

In the final version, the template can be dismembered into five sections:

1. Setup - Initial configuration;
2. Load - Loading data into memory;
3. Treatment - Image treatment;
4. Train - Model instantiation and training;
5. Statistics - Extraction of metrics and images from training.

3.5.1. Setup

The setup phase consists of library imports that will be used in runtime execution. A connection is established with Google Drive, to obtain credentials for downloading the dataset, which is also done in this step, and to save the model at the end of execution. The GPU is checked and configured. After downloading the dataset, the configuration variables are declared, which are the techniques and properties to be used in model training.

3.5.2. Load

The load stage is responsible for loading and filtering the dataset data into memory. Duplicate images are removed, with only one image per lesion identifier being considered. The colors and sizes of each image are adjusted to the expected model input.

3.5.3. Treatment

In this stage, the images are divided into three sets: training, validation, and testing with the respective rates: 70%, 10%, and 20%. The exact amount of images for each class is represented in Figure 5.

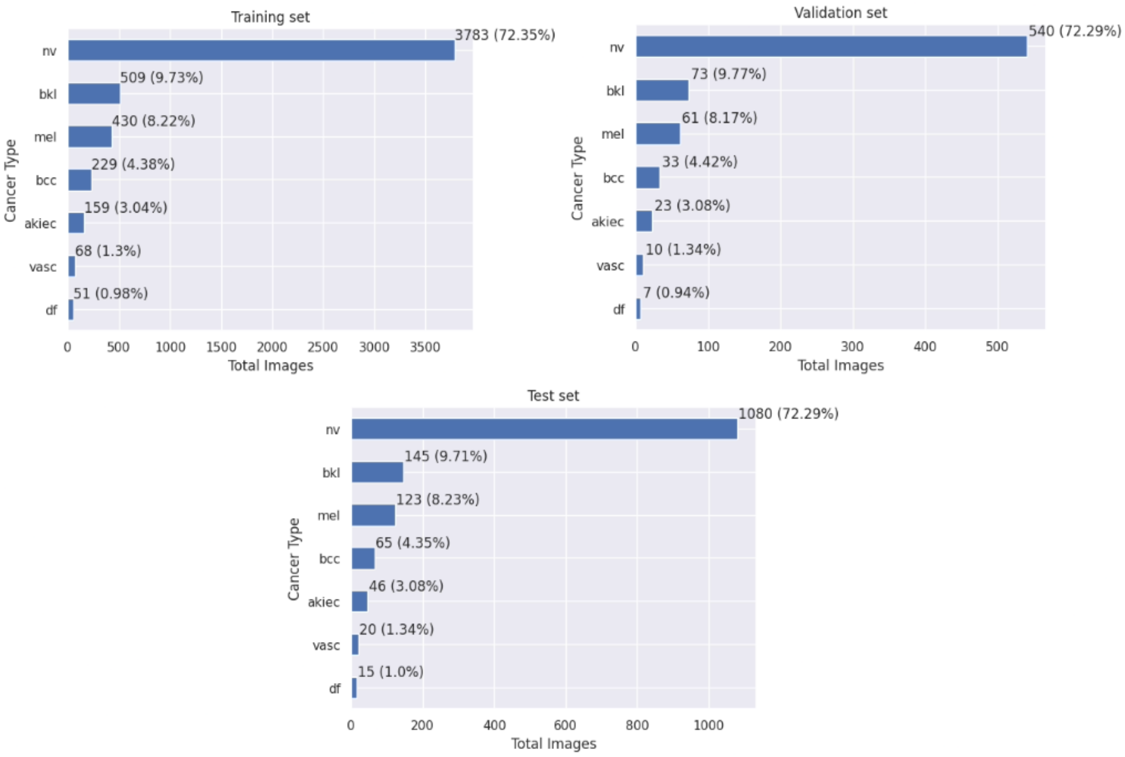


Figure 5. Dataset distribution between training, validation, and test sets.

The techniques such as data augmentation, applied in only training sets, segmentation, and normalization were applied to sets and, in the end, transformed the data into binary class matrices. Data augmentation, in this case, tried to balance classes by using Equation 3.

$$\text{Class Batch Size} = \text{round}\left(\frac{\text{class with most images}}{\text{current class}} - 1\right) \tag{3}$$

With the class batch size, the training set was augmented using the current training images by applying geometric and color transformations.

The GANs technique was tested as an alternative to data augmentation. This technique was applied only in the melanoma class, augmenting about nine times of original

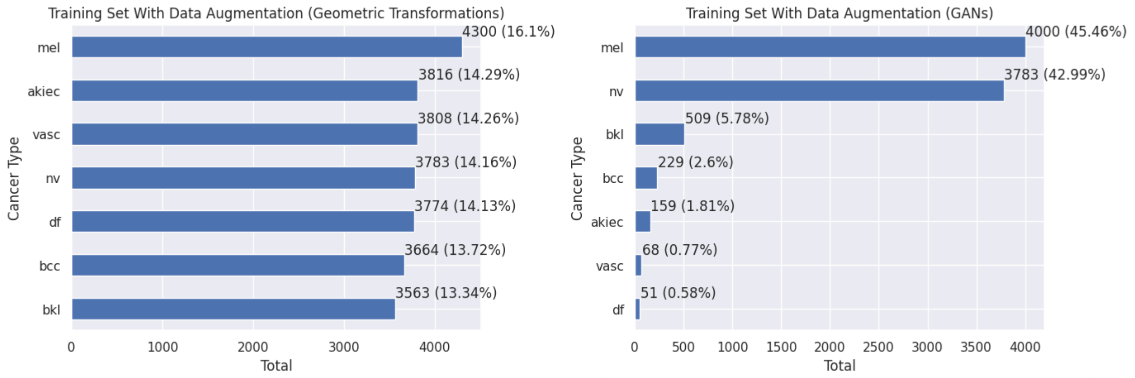


Figure 6. Dataset distribution after data augmentation, on the left geometric transformations and on the right the GANs image generated.

class size. To obtain GANs models, three tests were made, where in each was obtained two generator models. Whereas, to generate images for data augmentation were considered two out of six models, and both model outputs were most likely the real melanoma images. The final distribution of images is denoted in Figure 6.

The segmentation based on the region was applied in images with Gaussian blur to extract the lesion mask and merge it with the original image to have only the lesion highlighted.

There were two possibilities of normalization in the template: a min-max scaler and a standard scaler, where the first sets the image pixels in a range of [0,1] and the second removes the mean of image pixels, scaling to unit variance.

3.5.4. Train

In this phase, the selected model is loaded and configurations, such as transfer learning parameters weights and the classifier with a flattened and a fully connected layer to predict one of seven possible classes, are set up. The optimizer settings include the metrics to be collected and displayed during epochs. The model is compiled with these configurations. For the training parameters, two static configurations were used for each training, besides the class weighting, which promotes a balance between classes: Number of epochs (10) and batch size (8).

The fixed number of epochs was selected due to low-performance improvement after the tenth epoch. About batch size, the value eight fits the memory usage in the Google Colab[2] environment.

3.5.5. Statistics

The last step is responsible for extracting metric values, graphs, and confusion matrices from general notebook results and for each class. In the end, the trained model and normalized are saved into Google Drive.

[2] https://colab.google

A script was created to extract CNN notebook configurations and generate output data, specifically from the statistic phase. The extraction was made using regular expressions to match template patterns and retrieve their values. The data was stored in a simple relational database. The database was responsible for supporting data analysis which attended to the goal of generating graphs over training result values.

4. Results

Twenty-two training configurations were realized, where six used geometric transformations, eight utilized GANs specifically as data augmentation, eighteen pieces of training made use of transfer learning, and five applied segmentation. For each training configuration, three notebooks were created, one for each model. However, in this study, only sixty-five trainings were executed, because one configuration (10) did not have the Xception model notebook due to exceeding the 83.5 GB RAM limit. For each notebook, seven class metrics were created, totaling 455 registers.

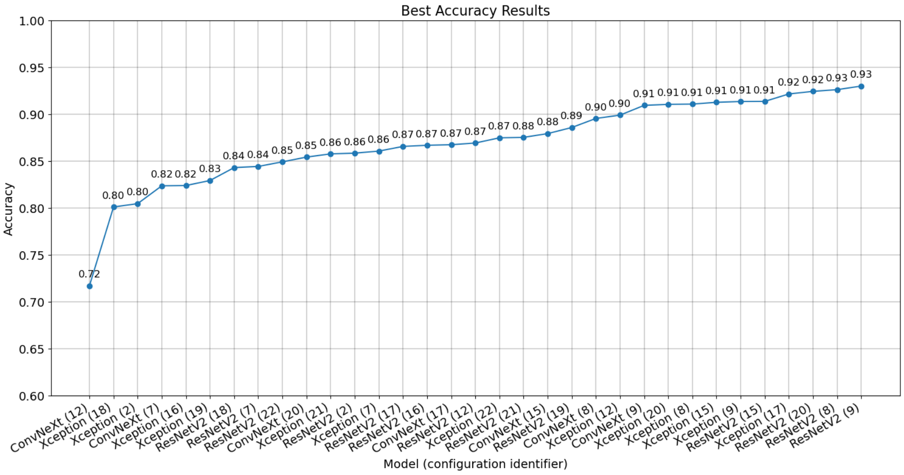


Figure 7. Graph representing best model training accuracies.

About ConvNeXt model training, the configuration identifiers 8 and 9 had the highest accuracies according to the graph. The major and unique difference between these configurations is the usage of class balance. Notebook general metrics from configuration 9 only overcome the other in terms of accuracy, while configuration 8 has all other metrics with better performance. Regarding melanoma statistics, configuration 9 presents unbalanced metric values and only has precision and specificity with higher values than configuration 8, while this last one shows a better balance between metrics. The configuration identifier 8 specifically had the best performance over all models with all seven classes having 50% or higher value in all metrics. For these reasons, configuration eight was selected to have the best performance with the ConvNeXt model.

The ResNetV2 model achieved four configurations with high accuracy, two using GANs and two using geometric transformations as data augmentation. Configurations 15 and 20 differ from each other with respect to the GAN model version used in the generation of new melanoma samples. Similarly to ConvNeXt configurations 8 and 9 present higher results, in a brief comparison, considering general notebook metrics, configuration 9 presents better statistics compared to the other. Despite the high loss, both configurations present similar general notebook metrics, however, in terms of melanoma class,

Table 2. Best configurations details. *GANs usage as data augmentation.

Configuration Identifier	Data Augmentation	Segmentation	Transfer Learning	Balance Classes
8	True	False	True	False
9	True	False	True	True
12	False	False	True	True
15*	True	False	True	False
17*	True	True	True	False
20*	True	False	True	False

configuration 8 demonstrates better and balanced values in contrast to configuration 9. Melanoma class is the most important class to predict the true positive images, for this reason, the best configuration for the ResNetV2 model is the configuration 8.

Table 3. Performance Metrics for Melanoma Lesion Type.

Model (Configuration Identifier)	Precision	Recall	F1-Score	Specificity	ROC AUC
ConvNeXt (8)	52.76%	54.47%	53.60%	95.62%	85.24%
ConvNeXt (9)	62.50%	12.20%	20.41%	99.34%	65.50%
ResNetV2 (8)	52.78%	30.89%	38.97%	97.52%	64.50%
ResNetV2 (9)	64.29%	7.32%	13.14%	99.64%	54.25%
ResNetV2 (15)	63.64%	5.69%	10.45%	99.71%	53.11%
ResNetV2 (20)	60.00%	7.32%	13.04%	99.56%	53.44%
Xception (8)	37.50%	19.51%	25.67%	97.08%	59.44%
Xception (9)	45.10%	37.40%	40.89%	95.92%	68.14%
Xception (12)	28.93%	28.46%	28.69%	93.73%	60.89%
Xception (15)	41.94%	21.14%	28.11%	97.37%	60.84%
Xception (17)	40.91%	21.95%	28.57%	97.16%	59.90%
Xception (20)	37.50%	26.83%	31.28%	95.99%	61.93%

The Xception model training had a variety of configurations bringing good overall results, they are displayed in Table 2. Also bringing the unique configuration without the usage of data augmentation achieved good efficiency. The general notebook configuration demonstrates that configuration 17 has better metrics over the others, except by loss, being the unique configuration using all three main techniques. However, in terms of melanoma metrics, configuration 9 contrasts with general metrics, having the specific class metrics with higher values and enhanced performance, except for the specificity metric value. The choice of best configuration was based on the melanoma class performance, in which configuration identifier 9 reached balanced values compared to the other settings.

5. Final Considerations

The transfer learning technique was decisive in reaching good results, every model with accuracy equal or higher than 89% has used the weight parameters from the thousand classes of the Imagenet dataset. To overcome the class unbalance, data augmentation and/or the balance classes techniques were also essential to achieve higher results.

Table 4. General Notebook Metrics.

Model (Configuration Identifier)	Loss	Accuracy	Recall	Precision	AUC
ConvNeXt (8)	13.90	89.54%	84.27%	84.27%	91.03%
ResNetV2 (8)	478.23	92.61%	74.90%	74.90%	85.36%
Xception (8)	691.57	91.07%	69.54%	69.54%	82.25%
ConvNeXt (9)	21.23	90.94%	81.53%	81.53%	89.60%
ResNetV2 (9)	569.49	92.99%	76.17%	76.17%	86.14%
Xception (9)	564.25	91.35%	70.75%	70.75%	82.94%
Xception (12)	787.61	89.89%	66.13%	66.13%	80.31%
ResNetV2 (15)	829.41	91.36%	70.62%	70.62%	82.92%
Xception (15)	772.99	91.26%	70.21%	70.21%	82.66%
Xception (17)	818.76	92.15%	73.43%	73.43%	84.53%
ResNetV2 (20)	774.98	92.43%	73.96%	73.96%	84.81%
Xception (20)	746.36	91.04%	69.95%	69.95%	82.47%

Segmentation, based on the results specifically, does not demonstrate to have high effectivity over top configurations since only one training performed effective results. The reason for this might be the unbalance between lesion classes in dataset, where the extraction of features of low class examples could not have offered sufficient information to distinct a lesion type from other.

This study generated 22.4 GB of data, including models and normalizers as notebook outputs. One aspect that can be inferred from this study is that the accuracy is not the only metric that can demonstrate the model performance, other models had better accuracy than others. However observing the specific class metrics, also other general metrics, was decisive in choosing which model performed better even if having a lower accuracy.

References

Brinker, T. J., Hekler, A., Hauschild, A., Berking, C., Schilling, B., Enk, A. H., Haferkamp, S., Karoglan, A., von Kalle, C., Weichenthal, M., Sattler, E., Schadendorf, D., Gaiser, M. R., Klode, J., and Utikal, J. S. (2019). Comparing artificial intelligence algorithms to 157 german dermatologists: the melanoma classification benchmark. *European Journal of Cancer*, 111:30–37.

Chen, Y., Yang, X.-H., Wei, Z., Heidari, A. A., Zheng, N., Li, Z., Chen, H., Hu, H., Zhou, Q., and Guan, Q. (2022). Generative adversarial networks in medical image augmentation: A review. *Computers in Biology and Medicine*, 144:105382.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions.

Codella, N., Nguyen, Q.-B., Pankanti, S., Gutman, D., Helba, B., Halpern, A., and Smith, J. R. (2016). Deep learning ensembles for melanoma recognition in dermoscopy images.

de Oliveira, C. C. (2021). Utilizando redes adversárias generativas (gans) como agente de apoio à inspiração para artistas. 47f. monografia (Graduação em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2021.

Delazeri, A. V. and Stevani, E. S. (2020). Classificação de câncer de pele usando redes neurais convolucionais: Uma análise do desempenho de classificação em um conjunto de dados desbalanceado.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

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

Ferreira, B. E. S. (2018). Segmentação semântica de lesões de pele utilizando redes neurais convolucionais. 35f. monografia (Graduação em Engenharia de Computação) - Universidade Federal do Maranhão, São Luís, 2018.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2nd edition.

Haenssle, H., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., Kalloo, A., Hassen, A. B. H., Thomas, L., Enk, A., Uhlmann, L., Alt, C., Arenbergerova, M., Bakos, R., Baltzer, A., Bertlich, I., Blum, A., Bokor-Billmann, T., Bowling, J., Braghiroli, N., Braun, R., Buder-Bakhaya, K., Buhl, T., Cabo, H., Cabrijan, L., Cevic, N., Classen, A., Deltgen, D., Fink, C., Georgieva, I., Hakim-Meibodi, L.-E., Hanner, S., Hartmann, F., Hartmann, J., Haus, G., Hoxha, E., Karls, R., Koga, H., Kreusch, J., Lallas, A., Majenka, P., Marghoob, A., Massone, C., Mekokishvili, L., Mestel, D., Meyer, V., Neuberger, A., Nielsen, K., Oliviero, M., Pampena, R., Paoli, J., Pawlik, E., Rao, B., Rendon, A., Russo, T., Sadek, A., Samhaber, K., Schneiderbauer, R., Schweizer, A., Toberer, F., Trennheuser, L., Vlahova, L., Wald, A., Winkler, J., Wölbing, P., and Zalaudek, I. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*, 29(8):1836–1842. Immune-related pathologic response criteria.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks.

Jahanifar, M., Tajeddin, N. Z., Koohbanani, N. A., Gooya, A., and Rajpoot, N. (2019). Segmentation of skin lesions and their attributes using multi-scale convolutional neural networks and domain specific augmentations.

Kawahara, J., BenTaieb, A., and Hamarneh, G. (2016). Deep features to classify skin lesions. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1397–1400.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s.

Lobo, L. C. (2017). Inteligência artificial e medicina. *Revista Brasileira de Educação Médica*, 41(2):185–193.

Menegola, A., Fornaciali, M., Pires, R., Avila, S., and Valle, E. (2016). Towards automated melanoma screening: Exploring transfer learning schemes.

Moorthy, J. and Gandhi, U. D. (2022). A survey on medical image segmentation based on deep learning techniques. *Big Data and Cognitive Computing*, 6(4).

Pham, T.-C., Luong, C.-M., Visani, M., and Hoang, V.-D. (2018). Deep cnn and data augmentation for skin lesion classification. In Nguyen, N. T., Hoang, D. H., Hong, T.-P., Pham, H., and Trawiński, B., editors, *Intelligent Information and Database Systems*, pages 573–582, Cham. Springer International Publishing.

Romero Lopez, A., Giro-i Nieto, X., Burdick, J., and Marques, O. (2017). Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, pages 49–54.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Soenksen, L. R., Kassis, T., Conover, S. T., Marti-Fuster, B., Birkenfeld, J. S., Tucker-Schwartz, J., Naseem, A., Stavert, R. R., Kim, C. C., Senna, M. M., Avilés-Izquierdo, J., Collins, J. J., Barzilay, R., and Gray, M. L. (2021). Using deep learning for dermatologist-level detection of suspicious pigmented skin lesions from wide-field images. *Sci Transl Med*, 13(581):eabb3652.

Suganyadevi, S., Seethalakshmi, V., and Balasamy, K. (2022). A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval*, 11(1):19–38.

Tschandl, P., Rosendahl, C., and Kittler, H. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):180161.

Vasconcelos, C. N. and Vasconcelos, B. N. (2017). Convolutional neural network committees for melanoma classification with classical and expert knowledge based image transforms data augmentation.

Wu, Y., Chen, B., Zeng, A., Pan, D., Wang, R., and Zhao, S. (2022). Skin cancer classification with deep learning: A systematic review. *Frontiers in Oncology*, 12.

Ünver, H. M. and Ayan, E. (2019). Skin lesion segmentation in dermoscopic images with combination of yolo and grabcut algorithm. *Diagnostics (Basel)*, 9(3):72.