Fine-Tuning and Boosting AlexNet for Skin Lesion Classification

Chems Eddine Benaziza

Supervised by Pietro Gori

October 7, 2024

Abstract

In this project, we present an AlexNet-based architecture for classifying skin lesions, which can be deadly if cancerous. The pipeline involves preprocessing steps such as hair removal and physical data augmentation, followed by feeding the augmented images to a AlexNet architecture for classification. Without data augmentation, our model achieved a performance below 50%. However, with our proposed augmentation techniques, we significantly improved the accuracy to **0.69**, resulting in a final ranking of **13/80** on a school Kaggle challenge.

1 Introduction

Skin cancer can be lethal, and among its various forms, malignant melanoma poses a significant threat. Despite representing only about 2% of all skin cancer cases, it accounts for the majority of skin cancer-related deaths. According to the World Health Organization, the incidence of malignant melanoma has been steadily increasing, with approximately 132,000 diagnosed cases reported globally in recent years.

Recognizing the importance of early detection and treatment, this paper focuses on leveraging machine learning techniques, particularly deep learning, to enhance the diagnostic process for melanoma. While deep learning has shown promising results in various domains, we aim to optimize its application specifically for skin cancer detection, with a focus on efficiency and accuracy.

In this study, we investigate the use of a pretrained AlexNet architecture, fine-tuned on an augmented dataset, to improve the performance of melanoma classification. By fine-tuning the model with a relatively small number of training samples, we aim to achieve robust and accurate predictions while minimizing computational resources and training time.

Despite advancements in treatment options, early diagnosis remains critical for improving patient outcomes. The "ABCDE" rule, a well-established criterion for melanoma prognosis, provides visual cues for characterizing skin lesions: Asymmetry, Border irregularity, Color variation, Diameter, and Evolution over time. Early detection of melanoma based on these criteria allows for timely intervention, leading to better prognosis and increased survival rates.

By optimizing deep learning techniques for melanoma detection, we aim to contribute to the ongoing efforts to improve diagnostic accuracy and ultimately enhance patient care in the fight against skin cancer.

Goal: The primary objective of this paper is to demonstrate how the ISIC dataset can be utilized to train a model that accurately classifies skin lesions. This involves identifying the optimal neural network architecture and the appropriate methods for data augmentation.

2 Data Exploration

In this section, we explore the dataset used in our study, comprising both image data and associated metadata. The dataset is divided into a training set and a test set, each accompanied by corresponding CSV files containing metadata information.

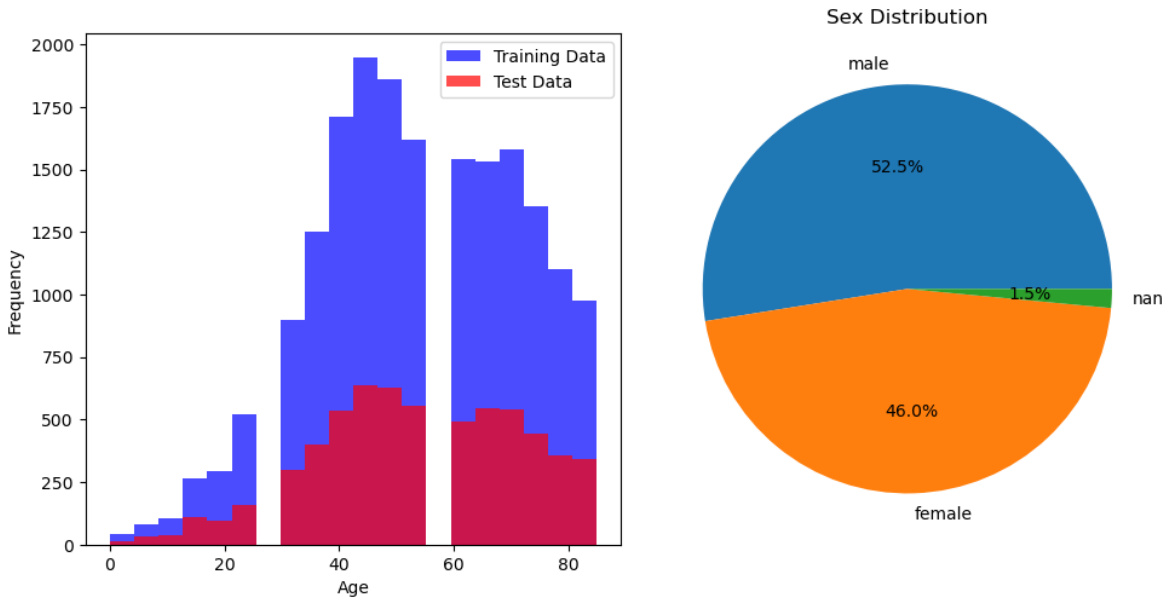

Figure 1: Left: Age distribution - Right : Sex distribution

2.1 Image Data

The dataset consists of two main subsets:

- **Training Set:** This set comprises a total of 18,998 images used for training our machine learning model.
- **Test Set:** The test set contains 6,333 images used for evaluating the performance of our trained model.

2.2 Metadata Files

The metadata files provide additional information about each image, including patient demographics and lesion characteristics. There are two CSV files:

- **metadataTrain.csv:** This file contains metadata information and class labels for all images in the training set. The columns include:
  - **ID:** Unique identifier for each image.
  - **CLASS:** Class label of the image (integer between 1 and 8).
  - **Age:** Age of the patient, when available.
  - **Sex:** Gender of the patient, when available.
  - **Anatomical Location:** Location of the lesion on the body, when available.
- **metadataTest.csv:** This file contains metadata information for all images in the test set, with columns similar to those in **metadataTrain.csv**, excluding the class labels.

Exploring these metadata files provides valuable insights into the demographic distribution of the patients, the anatomical locations of the lesions, and other relevant factors that may influence the classification task.

2.3 Class Distribution

3 Solution & Implementation

In this section we will discuss step after step our pipeline, step by step :

Class	Count
1	Melanoma (3391)
2	Melanocytic nevus (9657)
3	Basal cell carcinoma (2492)
4	Actinic keratosis (650)
5	Benign keratosis (1968)
6	Dermatofibroma (179)
7	Vascular lesion (190)
8	Squamous cell carcinoma (471)

Table 1: Distribution of Images Across Classes

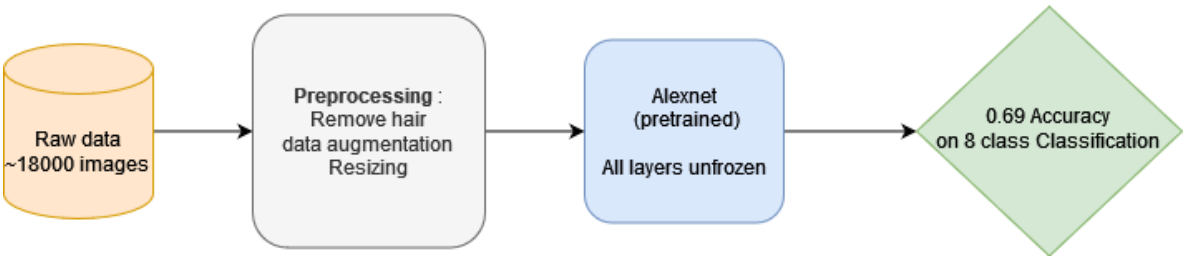

Figure 2: Our pipeline

3.1 Hair Removal

To avoid noise in feature detection, hair removal is an essential preprocessing step for dermatoscopic images. The technique used in this work is based on removing dark hair occlusions in the LUV[1] color space. The method involves morphological closing using a spherical structuring element, followed by hysteresis thresholding to generate masks in the L channel.

The function `occlusion_removal` takes an input image and applies the aforementioned steps to remove hair occlusions. The resulting image has reduced noise, which can significantly improve feature detection performance. The specific details of the implementation can be found in the code.

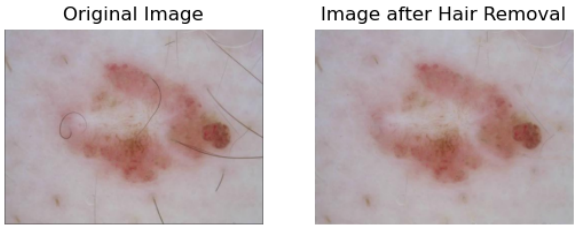

Figure 3: Hair removal in LUV space through a morphological closing

3.2 Data Augmentation

In a small-sized dataset, image augmentation is required to avoid overfitting the training dataset.

Several spatial and pixel-level transformations were selected to improve the prediction accuracy of the model, including transpose, flip, rotate, random brightness and contrast, motion blur, median and Gaussian blur, Gaussian noise, optical and grid distortion, elastic transform, CLAHE, hue-saturation-value modification, shift-scale-rotate, and cutout. The specific details of these transformations can be found in the code.

The augmented images were physically registered in a folder called "aug". Figure 5 augmented image. The augmentation was applied only to the training set while just normalizing the validation

[1]The LUV color space is a color model that represents colors based on their luminance (L) and chromaticity (U and V) components.

and testing dataset. You can see in Figure 3 a sample of the augmented data

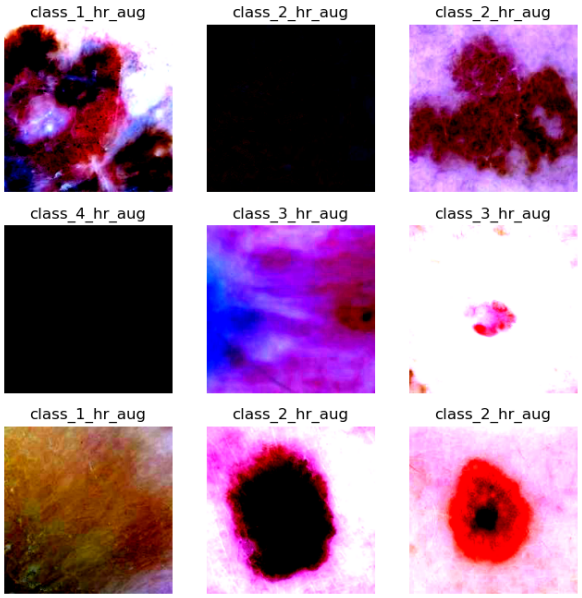

Figure 4: Sample from our Augmented Data

Index	Nombre de lésions
0	Melanoma (3391)
1	Melanocytic nevus (9657)
2	Basal cell carcinoma (2492)
3	Actinic keratosis (1950)
4	Benign keratosis (1968)
5	Dermatofibroma (1253)
6	Vascular lesion (1140)
7	Squamous cell carcinoma (1413)

Figure 5: Augmentation counts 23264 samples

4 Searching for the best CNN

4.1 First Custom CNN

We first tried a custom CNN, with a basic architecture but the trainig up to 30 epochs didn't give promising results, the kaggle score was aroudn **0.11** in kaggle, here is the architecture of this custom CNN

Layer	Output Shape	Parameters
Conv2d	(batch_size, 256, 224, 224)	kernel_size=3, stride=1, padding=1
MaxPool2d + Dropout + ReLU	(batch_size, 256, 112, 112)	kernel_size=2, stride=2, dropout=0.3
Conv2d	(batch_size, 128, 112, 112)	kernel_size=3, stride=1, padding=1
MaxPool2d + Dropout + ReLU	(batch_size, 128, 56, 56)	kernel_size=2, stride=2, dropout=0.3
Conv2d	(batch_size, 64, 56, 56)	kernel_size=3, stride=1, padding=1
MaxPool2d + Dropout + ReLU	(batch_size, 64, 28, 28)	kernel_size=2, stride=2, dropout=0.3
Flatten	(batch_size, 49152)	-
Linear	(batch_size, 32)	-
Linear	(batch_size, 8)	-

Table 2: Architecture of the custom Accuracy ≤ 30

results were not satisfying as the paper was claiming it to have, may be because it was made for the HAM100 dataset

4.2 ResNet - Too Big

We also experimented with ResNet, a popular deep learning architecture known for its ability to train very deep neural networks. However, we found that it was not well-suited for our specific task of skin lesion classification.

Firstly, ResNet was significantly slower to train than other architectures we tried, taking around 5 minutes per epoch on the school's GPUs (P100, 32GB). This made it difficult to iterate quickly and experiment with different hyperparameters.

Secondly, the ResNet architecture is quite large and complex, with many layers and parameters. While this makes it effective for classifying a wide variety of images, it also means that a significant portion of the neural network may not be relevant for our specific task. In other words, many of the features learned by ResNet may not be useful for distinguishing between different types of skin lesions.

Due to these factors, we ultimately decided not to pursue ResNet further for this project.

4.3 AlexNet the best trade-off Results

AlexNet is a convolutional neural network (CNN) architecture that was introduced by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012. The architecture used to solve our skin lesion classification consists of the following layers:

Layer	**Filter/Neuron Size**	**Stride**	**Activation Function**
Input	$227 \times 227 \times 3$	-	-
Conv1	$11 \times 11 \times 3$ (96 filters)	4	ReLU, Max Pooling (3×3, stride 2)
Conv2	$5 \times 5 \times 48$ (256 filters)	1	ReLU, Max Pooling (3×3, stride 2)
Conv3	$3 \times 3 \times 256$ (384 filters)	1	ReLU
Conv4	$3 \times 3 \times 192$ (384 filters)	1	ReLU
Conv5	$3 \times 3 \times 192$ (256 filters)	1	ReLU, Max Pooling (3×3, stride 2)
FC1	4096 neurons	-	ReLU
FC2	4096 neurons	-	ReLU
FC3	1000 neurons	-	Softmax
Dense	8 neurons	-	Softmax

Table 3: Architecture of AlexNet with an additional dense layer for 8-class classification

5 Results and Conclusion

5.1 Evaluation Metrics

As ranking metric, we will use the Weighted Categorization Accuracy (WA) which is defined as:

$$WA = \frac{1}{N} \sum_{i=1}^{N} w_i I(y_i = f_i) \tag{1}$$

such that

$$\sum_{i=1}^{N} w_i = N \tag{2}$$

in order to keep the maximum value equal to 1.

Here, y_i are ground truths, f_i are the predicted results, and w_i are the weights of the $i-th$ test image. If we make the hypothesis that we have K groups - or classes - called $G = G_1, \ldots, G_K$ and we associate the same weight w_t to all images of the same group t, we obtain that the weights are equal to:

$$w_t = \frac{N}{k|G_t|} \tag{3}$$

In this way, we can take into account the imbalance nature of the data-set, if present. Please note that the class weights in the test set are equal to: [0.7005531 0.24592265 0.95261733 3.64804147 1.20674543 13.19375 12.5654

Idea: A function with these weights was developed in the code to reproduce a good estimation of the metric on the given dataset. This function provided us with an accurate measure of the performance of our model without having to submit on Kaggle all the time.

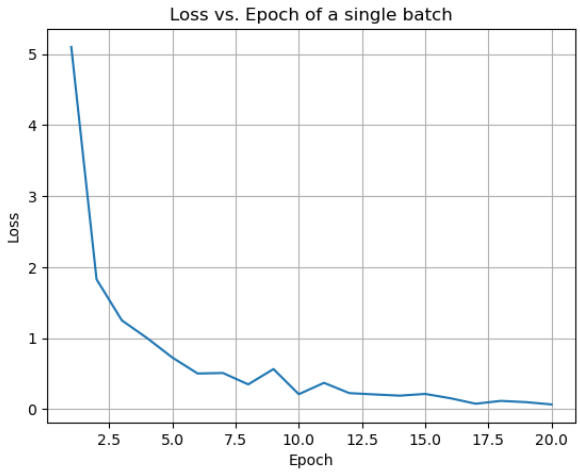

Figure 6: Proof that AlexNet can overfit a single batch

5.2 Performance analysis

5.2.1 Overfitting in a small batch

To test the capacity of our neural network, we intentionally overfit the model on a small batch of training data. We use a batch size of 64 and train the model for 200 epochs using stochastic gradient descent with a learning rate of 0.001. We use the cross-entropy loss function to measure the error between the predicted and true labels. During training, we compute the loss and accuracy after each epoch and print the results every 10 epochs.

5.3 Results of our boosted AlexNet on our physical augmented Data

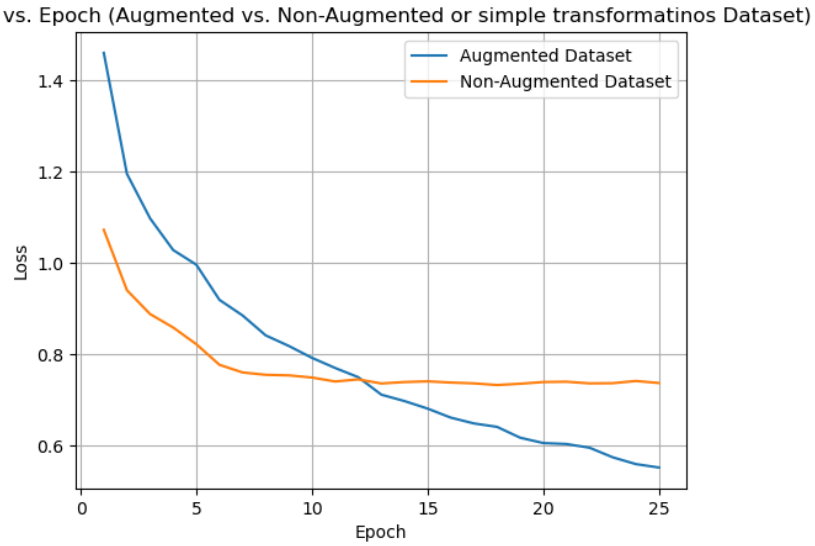

Figure 7: Final loss results on our augmentation and the one with simple transformations

Finally, we can conclude from the previous graphs that physical augmentation with hard changes, in contrast, and distortions significantly improves the performance of our AlexNet model. The augmented dataset consistently achieves lower loss values compared to the non-augmented dataset, demonstrating the effectiveness of data augmentation techniques in enhancing the model's learning capabilities.

Moreover, the results show that AlexNet, even when pretrained on a different image classification task, can be effectively fine-tuned and applied to the skin lesion classification problem. The substantial

improvement in loss and accuracy achieved through fine-tuning and augmentation validates the primary goal of this paper, which is to explore the potential of AlexNet as a powerful classifier for skin lesion images.

6 Difficulties

- Integrating metadata with learning data: One of the main challenges faced during the project was integrating metadata with the learning data. This required careful preprocessing and feature engineering to ensure that the metadata was properly incorporated into the model.
- Time constraints: Due to external commitments, time constraints were a significant challenge. This limited the amount of time that could be dedicated to experimentation and optimization of the model.
- Segmentation on different regions of the picture: Segmenting the skin lesion images into different regions proved to be a difficult task. This was due to the variability in the size and shape of the lesions, as well as the presence of artifacts and noise in the images.
- Unstable dataset due to patches: Another challenge was dealing with an unstable dataset caused by the use of patches. This required careful preprocessing and data augmentation techniques to ensure that the model was robust to variations in the input data.

7 Perspectives

- Future work could involve incorporating metadata and experimenting with alternative deep learning architectures to further improve the model's performance.
- Another promising approach found in the literature is the use of a convolutional neural network in combination with a multi-class SVM classifier, utilizing an appropriate kernel to effectively separate the data. This method could potentially enhance the accuracy of skin lesion classification.

8 Conclusion

In this paper, the authors present an AlexNet-based architecture for skin lesion classification, which is a critical task due to the potential deadliness of skin cancer, particularly malignant melanoma. The pipeline consists of preprocessing steps such as hair removal and physical data augmentation, followed by feeding the augmented images to the AlexNet architecture for classification. The authors found that without data augmentation, the model's performance was below 50%. However, with their proposed augmentation techniques, they significantly improved the accuracy to 0.69. This resulted in a final ranking of 13/80 on a school Kaggle challenge.

9 Code and Libraries

The code for this project was written in Python 3.12 and requires a good conda environment with necessary libraries. You can find the code repository on GitHub at the following link: https://github.com/chemousesi/ima205-skin-lesion-challenge-2025.