Melanoma Detection using Convolutional Neural Networks

CS512-Computer Vision

MSc in Computer Science, Illinois Institute of Technology
Due date: 11/13/2024

Antonio Castañares, A20599898

Nicolás Corsini, A20599305

1. Abstract

This project focuses on enhancing melanoma detection using **Convolutional Neural Networks (CNNs)** by replicating and improving the methodology presented in the paper *"Melanoma Detection Using Regular Convolutional Neural Networks"* by Aya Abu Ali and Hasan Al-Marzouqi.

Melanoma, a highly aggressive **skin cancer,** is challenging to diagnose manually due to its visual similarity to benign lesions, leading to significant reliance on automated solutions. Initially, the study replicated the original paper's CNN architecture, achieving comparable results with an **accuracy of 81.6%, high specificity (98%), but low sensitivity (14.9%),** highlighting the model's **inability to detect malignant cases reliably.**

Using a **balanced dataset (HAM10000)** and applying key architectural modifications, such as increasing convolutional blocks, optimizing pooling and kernel parameters, and incorporating data augmentation techniques, we developed a new CNN. Our model **surpassed the original in accuracy (88%), sensitivity, and overall performance, significantly reducing false negatives.**

2. Project Information

We selected the paper **'Melanoma Detection Using Regular Convolutional Neural Networks',** written by Aya Abu Ali and Hasan Al-Marzouqi, members of the Department of Electrical and Computer Engineering of Khalifa University of Science and Technology in Abu Dhabi, United Arab Emirates.

This paper was published in the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA) organized by The American University of Ras Al Khaimah (AURAK) in Ras Al Khaimah, United Arab Emirates. This congress tried to exhibit the latest theoretical and practical developments in the fields of Electrical Engineering, Computing Technologies, their Applications and the related fields.

The study aims to classify melanoma skin lesions as benign or malignant using an **automated CNN model,** reducing reliance on manual dermatologist assessments. We selected this paper for our project because **by choosing this paper, we can:**

- Contribute to **solving a significant healthcare challenge.**
- Work on a **manageable yet impactful deep learning project.**
- **Explore enhancements** to the proposed model, making it a versatile and rewarding project choice.

The authors **did not share their code** when they submitted their project. However the paper describes what they did. These clarifications have been used to develop the code used for the first objective of this project. They tried to classify melanoma skin lesions as benign or malignant using an automated CNN model, previously, images were preprocessed to **remove noise and standardize dimensions**, but **no advanced lesion segmentation or preprocessing was applied.** Furthermore, data augmentation techniques like **resizing and normalization** were employed to enhance the dataset.

Their proposed CNN model achieved an **accuracy of 81.6%,** with high **specificity (98%)** but **low sensitivity (14.9%),** which indicates a significant **imbalance** in the model's performance because the model is excellent at identifying benign cases however the model misses a significant number of malignant lesions, which is a critical issue in medical diagnostics.

Therefore, as for our objectives for this project, first of all, we want to **replicate their experiment.** Secondly, we **want to improve** their results, through the enhancements we intend to implement (e.g., improving sensitivity, adding data augmentation, improving preprocessing or introducing segmentation techniques).

In particular, our second objective is to design a convolutional neural network capable of classifying skin images as either melanoma or benign **with an accuracy exceeding 85%,** surpassing the performance of experienced dermatologists (estimated between 75% and 85%).

3. Problem Statement

Melanoma is a **type of skin cancer** that originates in the melanocytes, the pigment-producing cells responsible for skin coloration. It manifests as pigmented moles or marks on the skin, which can sometimes be mistaken for benign skin lesions. However, melanoma has the potential to **spread to other organs,** making it one of the most aggressive and deadly types of skin cancer. It accounts for approximately **75% of skin cancer-related deaths,** underscoring its severity.

One of the primary causes of melanoma is **excessive exposure to ultraviolet (UV) radiation** from the sun or artificial sources like tanning beds. UV radiation damages the DNA in skin cells, leading to mutations that can trigger cancerous growth. Additionally, UV exposure may alter the immune response, reducing the body's ability to repair this damage.

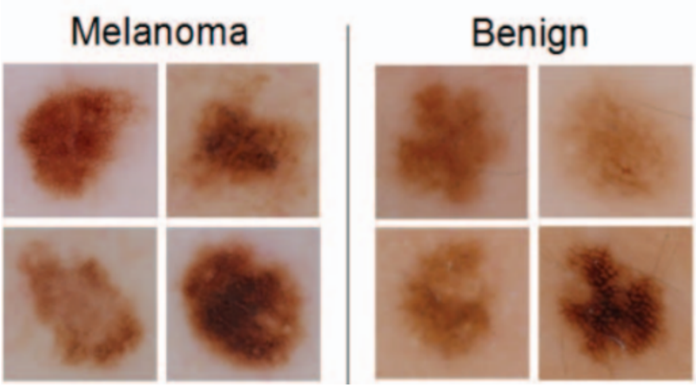


Figure 1: Melanoma and Benign examples

According to the American Cancer Society, **early detection** of melanoma dramatically improves survival rates. Identifying melanoma in its initial stages enables timely treatment, reducing the risk of metastasis and improving outcomes. However, diagnosing melanoma is inherently complex and requires a multifaceted approach.

The clinical process of melanoma diagnosis involves:

- **Medical History Review (Anamnestic Data):** Evaluating the patient's background and risk factors.
- **Pattern Analysis and Recognition:** Comparing lesions and analyzing specific features.
- **Differential Diagnosis:** Distinguishing melanoma from benign moles.

These tasks often rely on dermatologists' expertise, requiring significant training. Even with experienced professionals, accuracy rates are estimated at **75% to 85%**, leaving room for error. This is further complicated by the visual similarity between benign moles and malignant melanoma, as seen in comparative images, where the distinctions are subtle and not easily recognizable.

Given the challenges of manual diagnosis, automated algorithms based on machine learning, particularly Convolutional Neural Networks (CNNs), hold promise. These systems can analyze skin lesion images, classify them as benign or malignant, and significantly assist in early detection. **Automated methods can reduce diagnostic time,** enhance accuracy, and serve as a supportive tool for dermatologists.

4. Dataset

ISBI Challenge 2016 Dataset

Throughout this project, we have worked with **two datasets.** The first dataset is 'ISBI Challenge 2016 Dataset' [2] which consists of **900 images for training and 379 images** for testing. When we started to work with this dataset, one of the first aspects that we realized was this dataset is an **unbalanced dataset,** which means the number of images are not distributed equally between the labels.

An unbalanced dataset can lead to a **biased Convolutional Neural Network** because the model's learning process is heavily influenced by the distribution of data in the training set, mainly over-representation of the majority class.

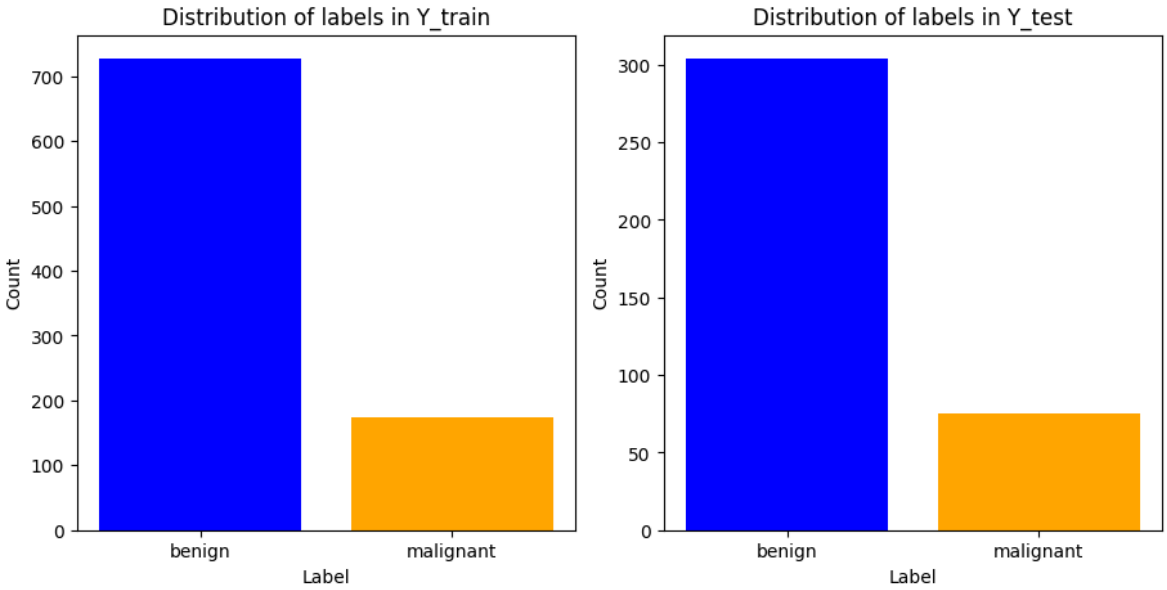


Figure 2: Distribution of the first dataset used.

We tried to apply different approaches to solve this issue, such as **oversampling** (increase the number of minority class samples by duplicating existing sampler generating synthetic ones), **undersampling** (reduce the number of majority class samples to balance the dataset) or assign **higher weights to the minority class** in the loss function to penalize misclassifications of minority samples more heavily.

However, our results closely matched those of the original paper, and we were unable to achieve significant improvements. We decided to switch to a **different dataset due** to the insufficient number of images in the original dataset for effective training.

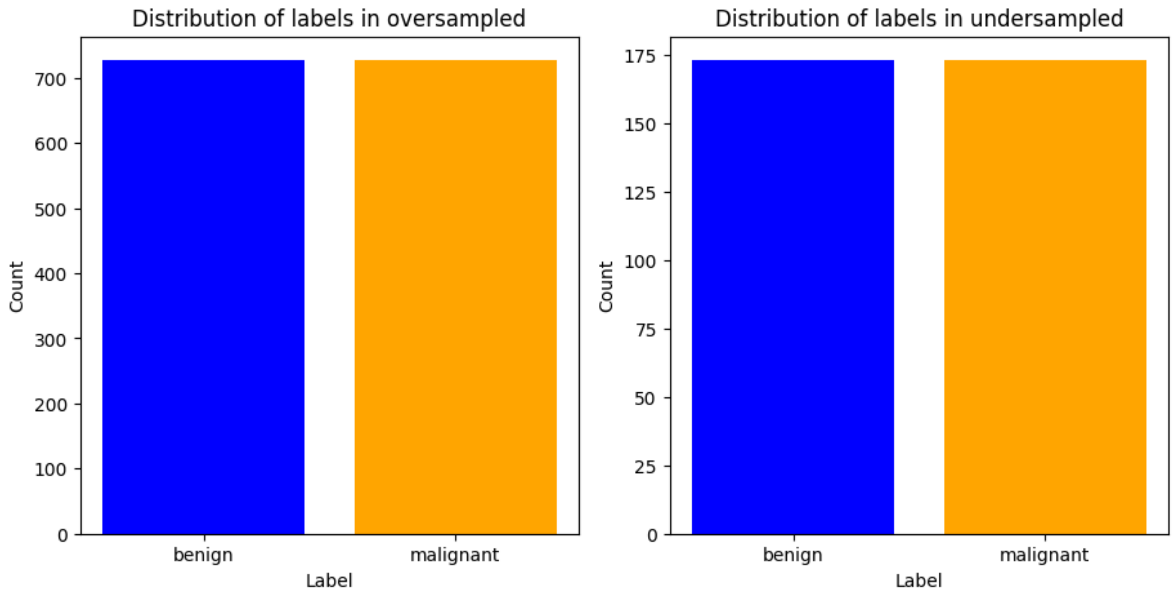


Figure 3: Dataset distribution after oversampled and undersampled.

Note that the label ratio in Y_train is approximately **4.202 benign cases for every malignant** case, while in Y_test, the ratio is approximately **4.053 benign cases for every malignant case.**

Human Against Machine with 10000 training images dataset

Published by National Library of Medicine [3], the dataset consists of **10015 dermatoscopic images** and they were collected over a period of 20 years from two different sites, the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia.

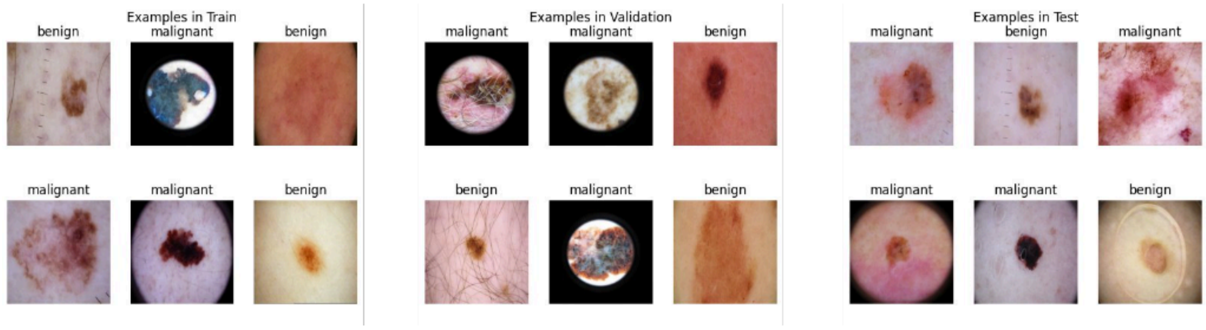


Figure 4: Images of HAM10000 dataset divided into train, validation and test dataset.

We selected this dataset to improve our results for two key reasons: first, it offers a significantly **larger number of images,** and second, **it is balanced,** with a similar number of samples for each label.

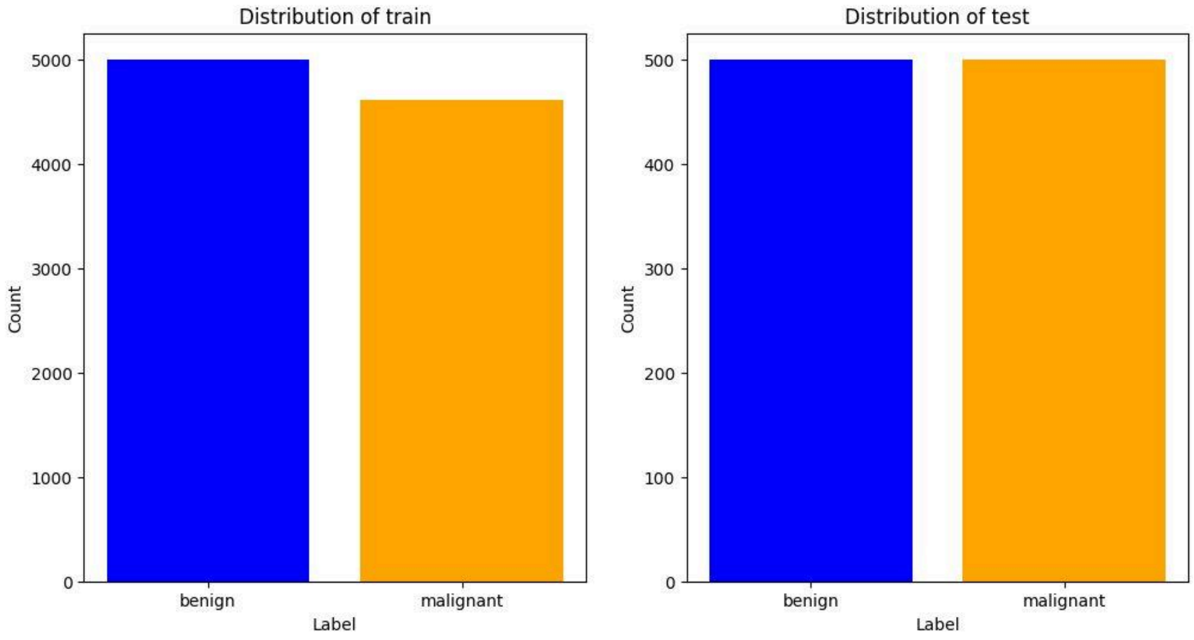


Figure 5: Distribution of the second dataset used.

5. Proposed Solution and Implementation details

As mentioned in the second chapter (Project Information), the primary objective was to replicate the CNN model described in the referenced paper before attempting to improve its performance. The entire implementation was developed in a **Jupyter Notebook**, hosted on **Google Colab** for its computational resources and ease of use.

Using **Python** libraries such as **Keras** and **TensorFlow**, we meticulously designed a CNN architecture based on the detailed specifications provided in the paper. The architecture was constructed using the following guidelines extracted from the paper:

- **Input Size**: The images were resized to 256x256 pixels.
- **Architecture Design**: The network consists of 17 layers, divided into 5 blocks. Each block includes convolutional, ReLU, pooling, and dropout layers, except the last block, which contains the fully connected and softmax layers.
- **Filter Sizes and Kernels**:
  - First three convolutional layers: 5x5 filters.
  - Fourth convolutional layer: 4x4 filters.
  - Fifth convolutional layer: 1x1 filters.
  - The first layer has 32 kernels, while subsequent layers have 64 kernels, except for the final layer, which has 2.
- **Dropout Rates**: Dropout was applied in increasing rates of 0.1, 0.2, 0.3, and 0.5 across the first four blocks.

- **Loss Function**: The **cross-entropy loss function** was employed for classification.

By following these guidelines, we successfully recreated a model that follows the original design. The structure of this CNN is visualized in the following image.

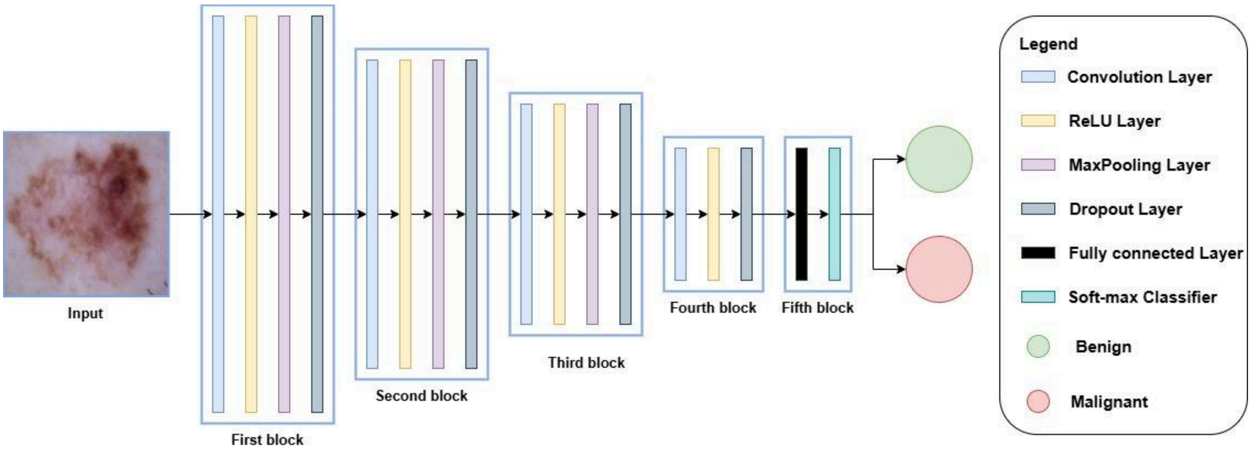


Figure 6: Original CNN.

As it can be seen in the results, this model performed with a decent level of precision when matched with the right dataset. However, we still looked for ways of improving it. Therefore, we built **our own CNN**, based on the paper's proposal, but with some key **changes in the structure**. This can be observed on the following image:

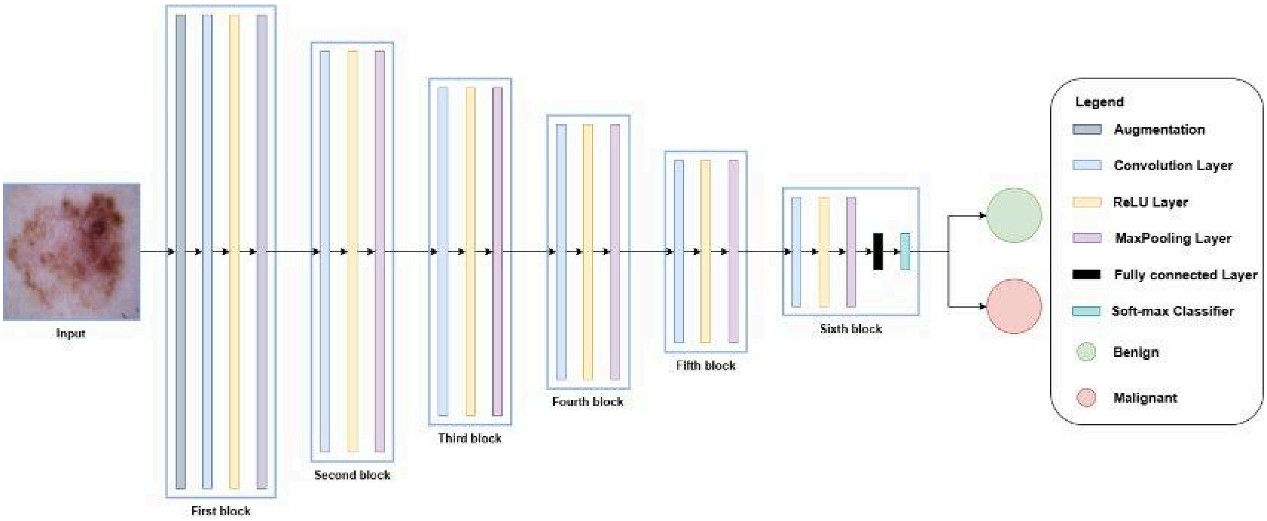


Figure 7: Our new CNN.

Similar to the previous CNN, this architecture is divided into blocks with multiple layers; however, several key changes have been introduced, as outlined below:

- **The layers are distributed in 6 blocks.** Increasing the number of blocks allows for deeper feature extraction and finer representation of patterns in the images, improving the model's capacity to learn complex features.
- **Each block includes the following layers: convolutional, ReLU activation, and max pooling.** This design maintains a balance between computational efficiency and effective feature extraction, ensuring the model focuses on core operations without unnecessary complexity.
- **After the convolutional blocks, the network has a fully connected layer and a softmax layer for classification.** This simplifies the transition from feature extraction to classification, with the fully connected layer compressing learned features into meaningful representations for binary classification.
- **All convolutional layers use a 3x3 kernel with stride 1 and padding of 1.** A 3x3 kernel is ideal for capturing fine details while keeping computational cost manageable, and padding preserves spatial dimensions for maintaining image features.
- **Each convolutional layer in the blocks has 32 filters.** Keeping 32 filters ensures consistency and avoids over-parameterization, balancing model complexity and computational efficiency.
- **Max pooling layers use a 2x2 pooling window with a stride of 2.** This provides gradual dimensionality reduction, helping retain spatial information while reducing computational cost.
- **No dropout layers are used in this architecture.** Dropout layers were removed to simplify the architecture, relying instead on data augmentation techniques to prevent overfitting.
- **Data augmentation (random flip and rotation) is applied to improve generalization.** Random transformations during training enhance the model's robustness to variations in input data, improving generalization to unseen examples without adding extra layers or parameters.

These changes can have a significant impact on the model's performance. The full structure of the developed CNN is shown in the following table, including the **parameters** used on each layer. Note that ReLU layers do not appear, as they are included in the convolution layers.

Layer	Parameters	Output Shape
Random Flip	horizontal_and_vertical	(None, 256, 256, 3)
Random Rotation	factor=0,2	(None, 256, 256, 3)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)

2D Convolution	filters=32
	kernel_size=(3,3)
	activation=”relu”
		(None, 254, 254, 32)

Max Pooling 2D	pool_size=(2,2)	(None, 127, 127, 32)
Flatten		(None, 128)

Dense	untis=64
	activation=”relu”
		(None, 64)

Dense	units=2
	activation=”softmax”
		(None, 2)

You can run our project by opening the file **project.ipynb,** which is located in the **src directory.** This Jupyter Notebook contains all the necessary code, explanations, and steps to reproduce our work, including data preprocessing, model implementation, training, and evaluation. Simply navigate to the src folder, open the notebook, and follow the instructions to execute each cell sequentially.

6. Results and discussion.

With the second dataset (**best dataset**), we have compared the obtained results by the two models: the original CNN and our new CNN. Firstly, below are two plots that display the first **model's accuracy** and loss in training and validation, when increasing the **number of epochs**:

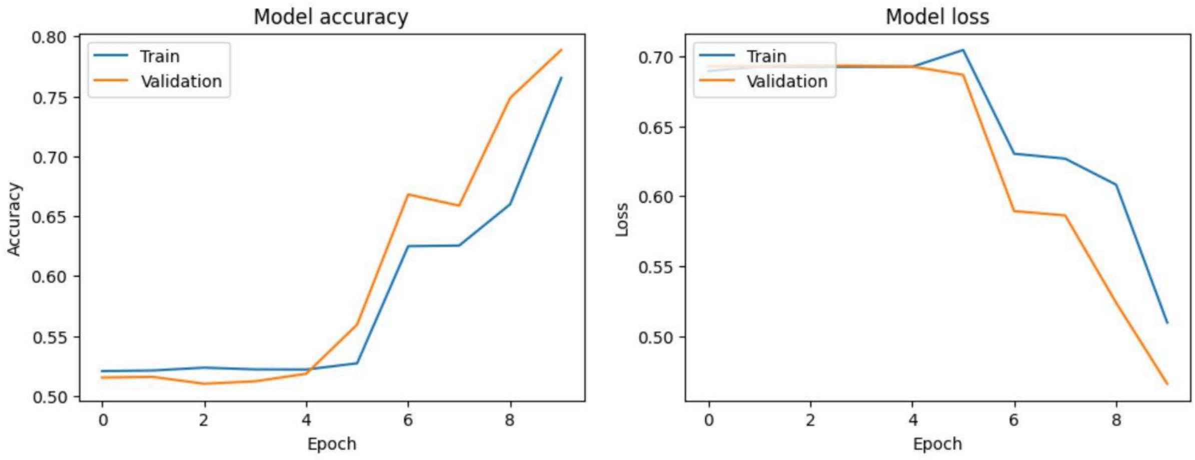


Figure 8: Original model accuracy and loss

This model only obtains optimum accuracy and loss with a **higher number of epochs** (10 epochs). The maximum accuracy is still **under 80%**. Both training and validation metrics are similar, indicating that the model does not suffer from **overfitting**.

Secondly, following plots show the same metrics for the newer model proposed in this paper:

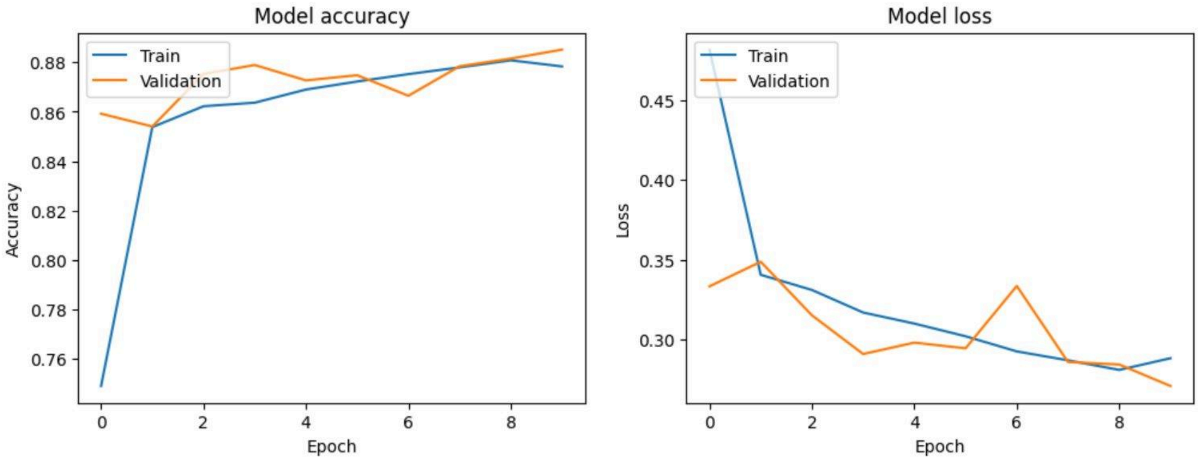


Figure 9: New model accuracy and loss

As it can be seen, the results are **dramatically improved**. With fewer epochs, the new CNN achieves an accuracy **close to 88%**.

Furthermore, it is a good insight to observe a **confusion matrix,** with the purpose of identifying the **false negatives** (undetected melanomas) and **false positives** (incorrectly detected melanomas). For both models, the confusion matrices are:

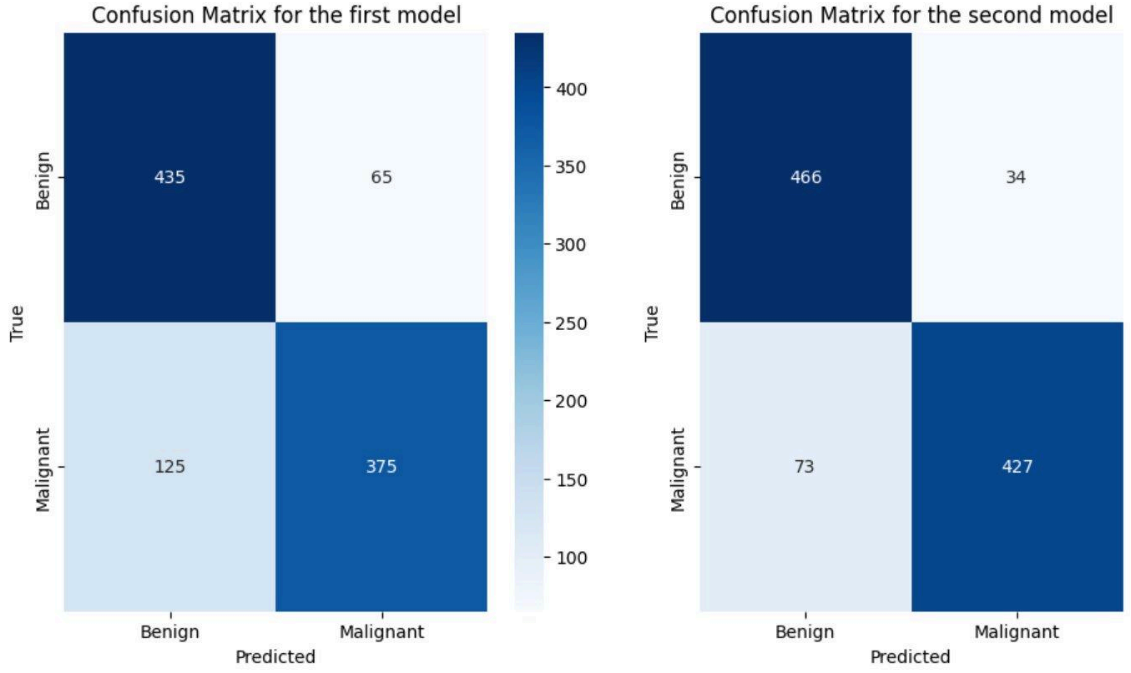
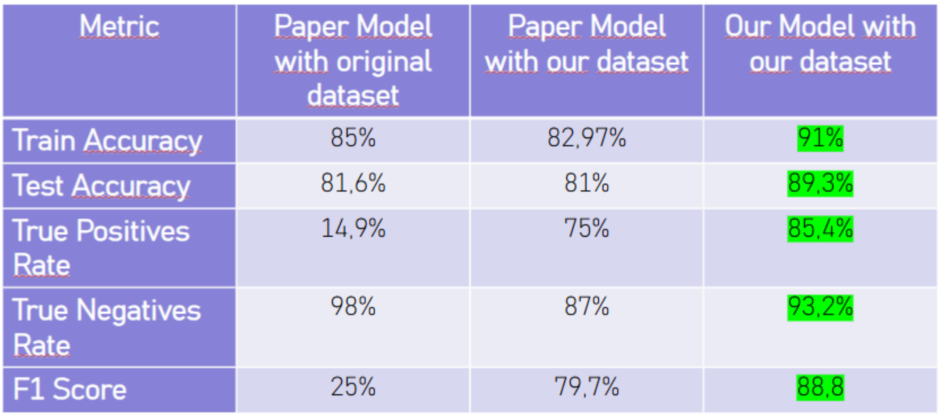


Figure 10: Both model's confusion matrices

Once again, it can be observed that the second model is far superior. When it comes to detecting melanomas, the **worst case scenario** would be a false negative. This would mean that a real melanoma has not been identified. This case corresponds to the bottom left corner of the confusion matrices, in which the second model shows a much smaller number of results (73 vs 125). This reaffirms the **strength of our model** over the first one. Finally, the results can be summarized in the following table:



Metric	Paper Model with original dataset	Paper Model with our dataset	Our Model with our dataset
Train Accuracy	85%	82,97%	91%
Test Accuracy	81,6%	81%	89,3%
True Positives Rate	14,9%	75%	85,4%
True Negatives Rate	98%	87%	93,2%
F1 Score	25%	79,7%	88,8

In conclusion, some valuable insights can be extracted:

- The **original dataset is severely unbalanced** which generates unacceptable results (True Positives Rate of 14,9% and F1 Score of 25%).
- The **original CNN with a balanced dataset** produces acceptable results, but they are still **under our expectations.**
- **Our proposed CNN with the same balanced dataset** achieves **strong results**, meeting our requirements.

7. References

[1] Aya Abu Ali and Hasan Al-Marzouqi, Eds., “Melanoma detection using regular convolutional neural networks.,” Nov. 21, 2017. https://ieeexplore.ieee.org/document/8252041 (accessed Oct. 15, 2024).

[2] “International Skin Imaging Collaboration (ISIC) Dataset 2016.” https://challenge.isic-archive.com/data/ (accessed Nov. 10, 2024).

[3] “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions.” https://pmc.ncbi.nlm.nih.gov/articles/PMC6091241/ (accessed Nov. 10, 2024).