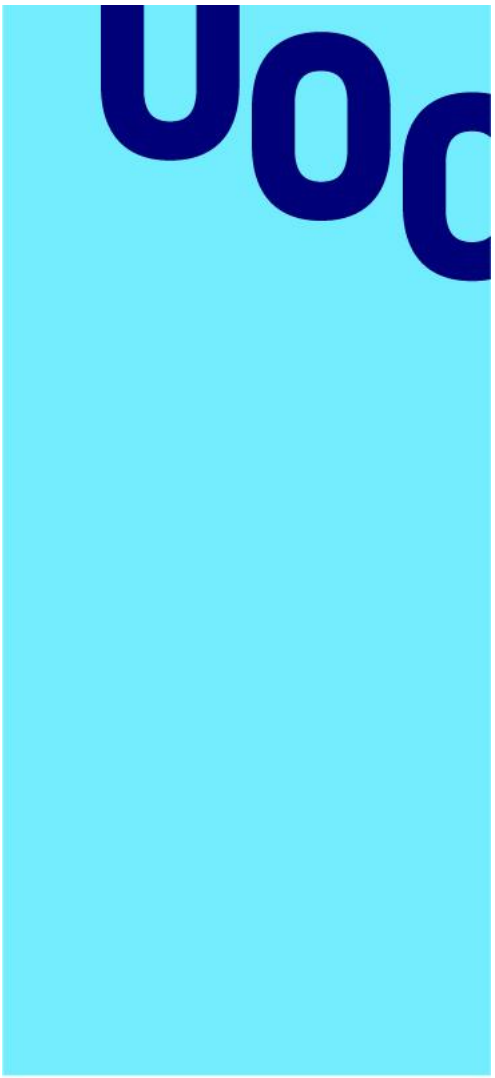
Maria Lucas Gascón - PAC2

Monitoring report

Development of a Web App for Skin Alteration Classification Using Machine Learning



Universitat Oberta de Catalunya

Index

1. Project identification

Project's title:	*Development of a Web App for Skin Alteration Classification Using Machine Learning*
Author's name:	*Maria Lucas Gascón*
Consultant's name:	*Romina Astrid Rebrij*
PRA's name:	*Carles Ventura Royo*
Delivery date:	*01/2024*
Program:	*Master's degree in Bioinformatics and Biostatistics UOC-UB*
Final project's area:	*Statistical Bioinformatics and Machine Learning*
Language:	*English*

This monitoring report has been meticulously formulated and documented on the 17th of November 2023.

2. Project's advancements

This section aims to elucidate the advancements made in the project during the period between October 17 and November 20. Specifically, it encompasses the first stage of the project, during which diverse algorithms were meticulously implemented, trained, and subsequently evaluated.

This segment will also elucidate the problems and obstacles encountered during the project, along with a comprehensive exploration of the chosen methodology employed to overcome these challenges.

2.1 Achievement of objectives

Main objective: Develop a web-based platform equipped with a machine learning algorithm capable of accurately classifying various skin alterations based on user-submitted images. ✘

The primary objective remains pending, as the development of the web-based platform is currently underway. Progress stands at approximately 50% completion, given the successful training of an algorithm capable of accurately classifying various skin alterations.

Secondary objectives:

1. Implement and evaluate a CNN machine learning model in Python to classify a diverse range of skin alterations. ✔
2. Optimize the model to achieve a minimum precision rate of 85%. ✔
3. Develop an interactive and user-friendly web application, integrating the model for skin alteration classification. ✘

Three distinct models underwent rigorous training and evaluation on the designated dataset. Firstly, a Convolutional Neural Network (CNN), constructed from the ground up, was meticulously developed, evaluated, and subsequently optimized, attaining a commendable accuracy rate of 73%. Concurrently, two additional models employing transfer learning methodologies were subjected to testing. An optimization process was applied to a DenseNet121 model, culminating in an impressive accuracy rate of 87%, while a ResNet50 model achieved an accuracy of 81%.

The exceptional performance of the DenseNet121 model, surpassing the predefined precision threshold of 85%, substantiates the successful accomplishment of both the first and second project objectives.

The successful culmination of Phase 1 of the project has been attained well in advance of the stipulated deadline date of November 19, 2023.

However, it is imperative to note that the realization of the third project objective remains pending and is anticipated to be addressed during Phase 2 of the project. This subsequent phase is projected to conclude on December 22, 2023, signifying the targeted completion timeframe for the objective.

2.2 Necessary changes

The project faced significant challenges attributable to limitations in available computational resources. Coupled with the substantial class imbalance within the dataset, a decision was made to mitigate this issue by eliminating classes with fewer images. Specifically, only images pertaining to nevus, melanoma, and basal cell carcinoma (BCC) were retained. This strategic measure was deemed necessary as achieving optimal model performance proved unattainable given the original class imbalance.

The images and associated metadata related to seborrheic keratosis, actinic keratosis, squamous cell carcinoma, solar lentigo, dermatofibroma, and vascular lesion were removed from consideration. The code responsible for eliminating these classes can be located in the GitHub repository under the filename "reducing_dataset.py". A visual representation of the data distribution before and

after this exclusion is presented in the following table (**Table 1**). The code responsible for extracting class information can be located in the GitHub repository under the filename "exploredata.py".

Diagnosis	Sample Size	Percentage	New sample size	New percentage
Nevus	4206	33,88	2850	33,55
Melanoma	2857	23,01	2857	33,47
Basal cell carcinoma	2809	22,62	2809	32,98
Seborrheic keratosis	929	7,48	0	0
Actinic keratosis	737	5,93	0	0
Squamous cell carcinoma	431	3,47	0	0
Solar lentigo	209	1,68	0	0
Dermatofibroma	124	0,99	0	0
Vascular lesion	111	0,89	0	0

Table 1: Class distribution. This table provides an overview of the diagnosis distribution within the BCN20000 dataset. Rows highlighted in red were removed to mitigate class imbalance and alleviate computational resource constraints. The creation of this table was facilitated by self-made code, and the necessary scripts for its reproduction can be found in the GitHub repository under the filenames reducing_dataset.py and exploredata.py.

Compounding the computational constraints was the inability to leverage GPU acceleration, as TensorFlow exclusively incorporates NVIDIA GPU acceleration, rendering the utilization of AMD GPUs unfeasible. Consequently, model training required extensive processing time, ranging from 12 to 24 hours at 100% CPU usage. Efforts to utilize the AMD graphics card proved unsuccessful, prompting the acquisition of a Google Colab Pro subscription. This subscription offered an enhanced computational environment with 52GB of RAM (as opposed to the previous 16GB) and featured an A1000 NVIDIA Graphics card, expediting the model training process.

The conundrum of limited VRAM prompted the creation of a subset consisting of 1000 images, facilitating a GridSearch to optimize hyperparameters. Following this optimization phase, the model with the most favorable hyperparameters was trained on the CPU of the local machine.

It is essential to note that the hyperparameters determined in this process, while deemed optimal within the constraints of the subset of 1000 images, may not necessarily represent the globally optimal parameters. The subset's random nature introduces a level of uncertainty in the hyperparameter optimization process.

3. Tasks carried out

As previously indicated, Phase 1 of the project concluded with relative success. This denotes that the tasks originally slated for completion within this phase were predominantly achieved; however, unforeseen complications gave rise to supplementary tasks. The subsequent section provides a comprehensive breakdown of the accomplished tasks as well as those that remain pending.

3.1 Programmed tasks

Main tasks:

1. **Definition of the work plan**: Define the project's scope, objectives, methodology and expected outcomes. Create a project charter outlining the project's purpose and goals as well as a calendar with milestones and dates. ✔
2. **Data Collection and Preparation**: Curate a comprehensive and diverse dataset of skin alteration images, ensuring that it represents various skin conditions and ages to train and validate the machine learning model. Split the data into training, validation, and test sets. Preprocess the images, normalizing pixel values to the range, and augmenting the data if needed. ✔
3. **Algorithm Implementation**: Develop a machine learning algorithm in Python capable of accurately classifying a diverse range of skin alterations. ✔
4. **Web Application Development**: Design and develop a user-friendly web application using the Django framework to enable users to upload skin images for classification. ✘

The work plan was formulated and promptly reviewed by the tutor. However, certain modifications suggested by the tutor, primarily concerning the project's introduction, were regrettably not executed within the initially established timeframe. This delay was attributed to unforeseen tasks that emerged during the algorithm optimization process. Nonetheless, a comprehensive revision was successfully conducted before concluding Phase 1 and the redaction of this report.

The process of data collection and preparation proceeded smoothly, without encountering major impediments. Notably, an adjustment was made to address computational constraints by resizing images from 1024x1024 to 256x256. While this transformation was anticipated, the unexpected challenge arose in executing this task in Python due to high RAM usage. As a workaround, the resizing process was performed in Linux using bash commands. Data augmentation was deemed unviable due to computational constraints.

The implementation of algorithms met the expected performance criteria within the designated time frame. As mentioned earlier, certain adjustments were necessary to tackle class imbalance and achieve the desired model performance. The code for compiling and training the models can be located in the GitHub repository under the filenames "CNN.py", "DenseNet.py" and "ResNet.py".

Conclusively, the development of the web application is still pending and is anticipated to be concluded during Phase 2 of the project, aligning with the project calendar deadline of December 22, 2023. This timeline remains consistent with the planned schedule.

Secondary tasks:

1. **Algorithm Evaluation**: Evaluate the performance of the machine learning algorithm by conducting rigorous testing, including metrics such as accuracy, sensitivity, specificity, and precision, to ensure its effectiveness in skin condition classification. ✔
2. **Algorithm Optimization:** Investigate the potential of utilizing a pre-trained model such as VGG16, ResNet50, or MobileNet. Identify the most effective pre-processing technique to enhance the model's performance. ✔
3. **Accessibility and Scalability**: Ensure that the web application is accessible to anyone with an internet connection, and design it to be scalable for potential future enhancements and improvements. ✘
4. **Educational Content Integration**: Incorporate educational content within the web application to provide users with information about various skin conditions, fostering awareness and encouraging proactive skin health practices. ✘
5. **User Experience Testing**: Conduct user experience testing to refine the web application's interface and functionality, ensuring ease of use and accessibility for a wide range of users. ✘
6. **Data Security and Privacy**: Implement robust data security and privacy measures to protect user-submitted images and information, complying with relevant regulations and standards. ✘
7. **Deployment and Maintenance**: Deploy the web application to a reliable hosting environment and establish a plan for ongoing maintenance and updates to ensure continued functionality and accuracy. ✘

The evaluation of algorithms transpired seamlessly, enabling the extraction of metrics without encountering any impediments. Several graphs were meticulously generated to effectively convey the information within the report.

The implementation and optimization of the three algorithms were successfully executed, notwithstanding the need for certain workarounds to address computational resource constraints. Notably, two pre-trained models were subjected to data-transfer techniques, yielding commendable results.

It is imperative to note that pending tasks pertain to Phase 2 of the project and are thus slated for completion in the subsequent phase.

3.2 Non-programmed tasks

1. **Reducing dataset:** To rectify class imbalance and accommodate limited computational resources, a reduction in the dataset was imperative, involving the elimination of minority classes. Additionally, a test subset comprising 1000 images was randomly selected to facilitate hyperparameter optimization. The code for selecting the test subset can be located in the GitHub repository under the filename "create_test_images.py".
2. **GridSearch for optimization**: To expedite model training, a GridSearch optimization approach was employed, involving the simultaneous testing of multiple fixed hyperparameters on a smaller subset of 1000 images. This computational intensive process was conducted in a Google Colab Pro environment, harnessing the benefits of GPU acceleration. The code for performing GridSearch optimization can be located in the GitHub repository under the filenames "GridSearch_CNN.py", "GridSearch_DenseNet.py" and "GridSearch_ResNet.py".
3. **Exploration of Segmentation Models**: A thorough investigation into various segmentation models was conducted to assess the potential benefits of their implementation on overall model performance. However, the complexity of this task surpassed initial expectations, rendering timely completion unattainable. Subsequently, considering the already robust performance of the DenseNet model, the incorporation of segmentation models was deemed unnecessary and, consequently, discarded.

4. Calendar revision

As depicted in the figure below (**Figure 1**), all scheduled tasks for Phase 1 of the project were successfully executed. Notably, the coding of the models extended beyond the initially projected time frame due to the computational constraints elucidated earlier. The dates marked in red are approximations, and indeed, there was an overlap of certain tasks, particularly during periods where concurrent work was feasible. This occurrence was especially prevalent during model training, where substantial time intervals were required for code execution, allowing for the utilization of free time for other tasks such as writing and revising project documentation.

Nombre		Fecha de inicio	Fecha de fin
Work plan development CAA1		27/9/23	15/10/23
Exploring and choosing a dataset	✓	27/9/23	2/10/23
Determine the models to be tested	✓	2/10/23	8/10/23
Writing CAA1	✓	9/10/23	15/10/23
Work development CAA2		16/10/23	19/11/23
Explore necessary tools	✓	16/10/23	19/10/23
Pre-processing data	✓	19/10/23	23/10/23
Code the models	✓	24/10/23	~~2/11/23~~ 7/11/23
Study model accuracy	✓	3/11/23	~~5/11/23~~ 14/11/23
Optimize the best model	✓	6/11/23	~~10/11/23~~ 14/11/23
Optimize and document the code	✓	~~11/11/23~~ 14/11/23	~~13/11/23~~ 16/11/23
Writing CAA2	✓	14/11/23	19/11/23
Result's analysis	✓	15/11/23	18/11/23

Figure 1: Phase 1 Calendar. Calendar presented in CAA1 featuring tasks and associated dates. An icon denoting task completion has been incorporated, with actual start and finish dates highlighted in red.

In light of the successful adherence to the schedule during Phase 1, no anticipated modifications to the calendar for subsequent phases are foreseen (**Figure 2**). It is noteworthy that the analysis of results was accomplished within Phase 1, providing an advantageous head start for the development of Phase 2.

Nombre	Fecha de inicio	Fecha de fin
Work development CAA3	20/11/23	22/12/23
~~Result's analysis~~	~~20/11/23~~	~~23/11/23~~
Learning to develop web-app in Django	~~24/11/23~~ 20/11/23	27/11/23
Designing web map	28/11/23	29/11/23
Delevop app	30/11/23	9/12/23
Write educational content	7/12/23	9/12/23
Test web app	10/12/23	14/12/23
Write CAA3	15/12/23	22/12/23
Final report and presentation CAA4	27/12/23	13/1/24
Optimize report	27/12/23	30/12/23
Optimize code presentation and documentation	31/12/23	3/1/24
Making presentation	4/1/24	13/1/24
Public defense CAA5	14/1/24	1/2/24

Figure 2: Phase 2 Calendar. Calendar presented in CAA1 featuring tasks and associated dates. The new initiation date for Phase 2 of the project is highlighted in red.

5. Obtained results

Two noteworthy additions have been incorporated into the main report. Initially, a comprehensive methodology section elucidating all models and code has been appended. Subsequently, upon the completion of all tasks, the results have been included and expounded upon. It is imperative to acknowledge that these sections remain work in progress, as the methodology for web development and corresponding results are yet to be included.

Moreover, a strategic reduction and summarization of the introduction section have been implemented. Certain details previously found in the introduction have been relocated to the state-of-the-art section, which has also undergone a summarization process. Notably, information pertaining to skin alterations beyond melanoma, nevus, and BCC has been omitted, aligning with the project's focused scope.

In a concluding step, a dedicated GitHub repository has been established for the project. A preliminary structure has been designed and implemented, housing all code utilized in algorithm training and data processing. Future additions to the repository will encompass project reports, documentation, and web-related code.

For reference, the repository is accessible at the following link: GitHub Repository
Additionally, the main report can be accessed here: MemoriaMariaLucas