Universitat de Girona
**Escola Politècnica Superior**

Treball Final de Màster

Estudi: Màster en Ciència de Dades

Títol: Una Plataforma per Classificar Melanomes

Document: Memòria

Alumne: Wilber Eduardo Bermeo Quito

Tutor: Rafael Garcia Campos
Departament: ARQUITECTURA I TECNOLOGIA DE COMPUTADORS
Àrea: ARQUITECTURA I TECNOLOGIA DE COMPUTADORS

Convocatòria (mes/any): Setembre 2022

Universitat de Girona
**Escola Politècnica Superior**

MASTER'S THESIS

# A Platform for Classifying Melanoma

WILBER EDUARDO BERMEO QUITO
September 2022

Master in Data Science

*Advisors:*

DR. RAFAEL GARCIA CAMPOS
Universitat de Girona
Departament of Computer Architecture and Technology

SR. LUIS PLA LLOPIS
Accenture S.L.

# Summary

# Gratitude

I would like to express my sincere gratitude to all those who have contributed to the completion of this thesis. Their unwavering support, guidance, and encouragement have been invaluable throughout this academic journey.

First and foremost, I extend my deepest appreciation to my thesis advisors, Dr. Rafael Garcia Campos and Sr. Luis Pla Llopis, for their exceptional mentorship and insightful feedback. Their expertise and dedication have been instrumental in shaping this research project and improving its quality.

Finally, I am profoundly grateful to my friends who eagerly took the time to read and review this thesis, offering their valuable feedback to enhance its quality and depth. Their willingness to invest their precious time and intellectual energy into providing constructive critiques has been a tremendous source of support and motivation throughout this research journey.

# Contents

# List of Figures

# List of Tables

CHAPTER 1

# Results

In this chapter, we present the results of various machine learning approaches used to train the models discussed in *Chapter* **??**. These results are presented using tables and visualizations to facilitate understanding the benefits and limitations of each decision. The tables and visualizations, provide a concise and comparative summary of the performance metrics, such as accuracy, AUC and recall.

Additionally, we showcase the outcomes of the CAD infrastructure constructed around this thesis.

## 1.1 Metrics

After training each model and inferring the predictions, we saved the metrics of interest for each dataset (train, validate, test). The training information of each training approach is given in *Table* **??**. On the other hand, *Table 1.2* provides a summary of the metrics acquired for each model in the different datasets.

The accuracy metric in *Table 1.2* represents the overall performance of the model in the multiclass classification problem with 8 different labels. However, given the class imbalance, accuracy may not be the most suitable metric to assess the model's performance in this situation. Instead, metrics like AUC (Area Under the Curve) and recall should be considered, as they specifically indicate how well the model is performing in classifying melanoma from the other classes.

The metrics with a gray background in *Table 1.2* are from models that incorporated additional regularization techniques, such as dropout or data augmentation. These models were trained for 40 epochs since regularization tends to slow down the minimization of the objective function compared to the other models that were trained for only 20 epochs.

To facilitate a better comparison between the two approaches, we computed the mean and standard deviation for each training approach.

It is important to mention that the metrics obtained for all models, both on the validation and test sets, were generated using the Test-Time Augmentation technique. As explained in previous chapters, this technique functions as an ensemble, contributing to improved performance during inference.

Models that used a scheduler during the training stage are denoted with a symbol next to their names. For reference, the mapping between the scheduler used and the corresponding symbol is provided in *Table 1.1*.

| **Scheduler Mapping** | |
|---|---|
| $\star$ | Step Learning Rate |
| $*$ | Cosine Annealing Learning Rate |
| $\bullet$ | Cosine Annealing Warm Restarts |

Table 1.1: *Scheduler Mapping. Table by Author*

| | Train AUC | Val AUC | Test AUC | Train Recall | Val Recall | Test Recall | Train Acc | Val Acc | Test Acc |
|---|---|---|---|---|---|---|---|---|---|
| M0 | 0.952 | 0.903 | 0.892 | 0.756 | 0.676 | 0.652 | 0.835 | 0.778 | 0.772 |
| M1 ⋆ | 0.947 | 0.900 | 0.892 | 0.695 | 0.633 | 0.599 | 0.829 | 0.779 | 0.771 |
| M2 ∗ | 0.933 | 0.895 | 0.885 | 0.658 | 0.609 | 0.582 | 0.808 | 0.765 | 0.762 |
| M3 • | 0.935 | 0.896 | 0.886 | 0.663 | 0.605 | 0.589 | 0.811 | 0.767 | 0.764 |
| M4 | 0.886 | 0.877 | 0.858 | 0.478 | 0.475 | 0.446 | 0.757 | 0.750 | 0.741 |
| M5 ⋆ | 0.867 | 0.861 | 0.843 | 0.423 | 0.403 | 0.395 | 0.728 | 0.717 | 0.715 |
| M6 ∗ | 0.874 | 0.868 | 0.848 | 0.451 | 0.440 | 0.418 | 0.738 | 0.728 | 0.722 |
| M7 • | 0.877 | 0.869 | 0.849 | 0.470 | 0.458 | 0.432 | 0.742 | 0.732 | 0.723 |
| Mean | 94.175% | 89.850% | 88.875% | 69.300% | 63.075% | 60.550% | 82.075% | 77.225% | 76.725% |
| SD | 0.921% | 0.370% | 0.377% | 4.509% | 3.260% | 3.178% | 1.327% | 0.727% | 0.499% |
| Mean | 87.600% | 86.875% | 84.950% | 45.550% | 44.400% | 42.275% | 74.125% | 73.175% | 72.525% |
| SD | 0.787% | 0.655% | 0.625% | 2.445% | 3.084% | 2.175% | 1.204% | 1.372% | 1.108% |

Table 1.2: *Model Metrics in Datasets. Table by Author*

## 1.2 Training

In this section, we analyze the performance of the models throughout the training process. Most of the graphics seen in this section were generated using W&B (Weight & Biasses).

Every pair of models in the set $S$, $S = \{\{M0, M4\}, \{M2, M5\}, \{M3, M6\}, \{M4, M8\}\}$, shares the same scheduler if they were used in the training phase. However, each pair differs in batch size, the number of epochs, and regularization techniques.

We are providing this comparison to demonstrate the impact of regularization on the training process. It is evident that models trained for 20 epochs experience overfitting due to the absence of regularization techniques. On the other hand, models trained for 40 epochs do not suffer from overfitting, but their performance is still inferior to models without additional regularization. However, this issue probably can be resolved by increasing the number of training epochs.

### 1.2.1 M0 against M4

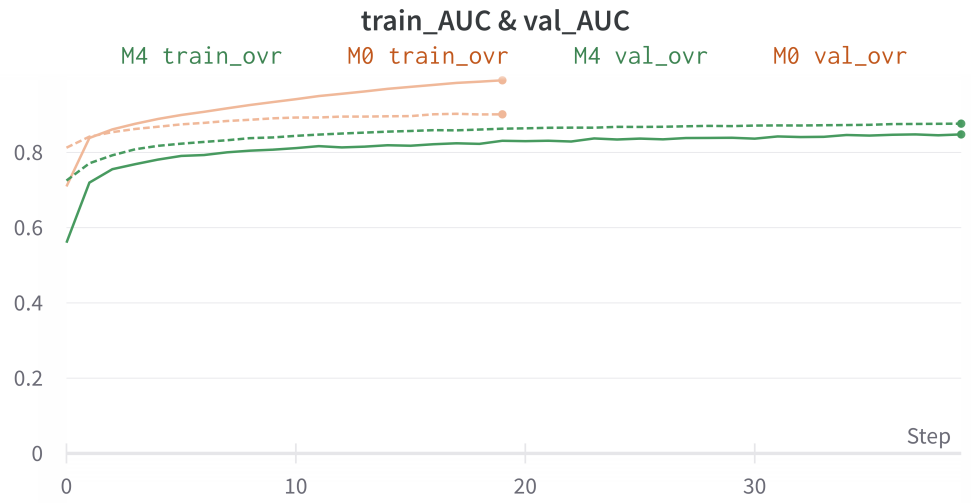


Figure 1.1: *M0 vs. M4, AUC Train and Validation Curves. Illustration by Author*

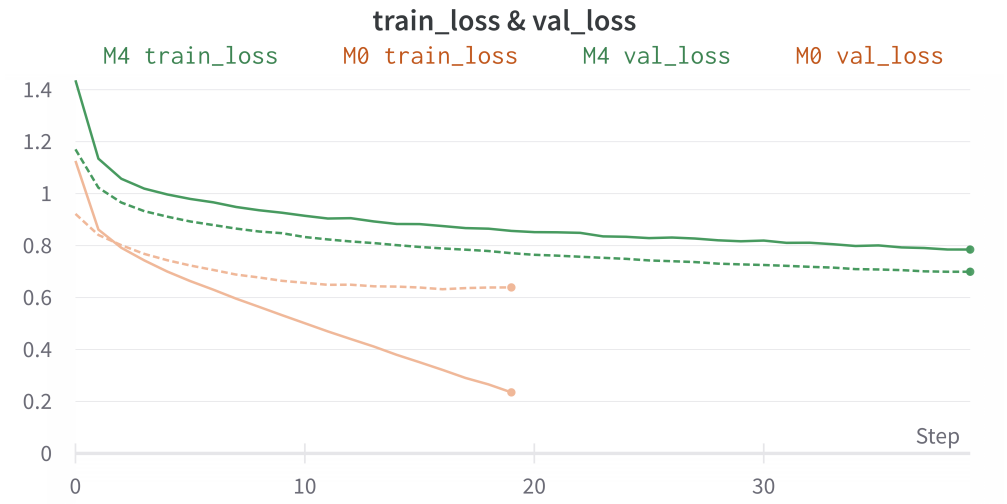


Figure 1.2: *M0 vs. M4, Loss Train and Validation Curves. Illustration by Author*

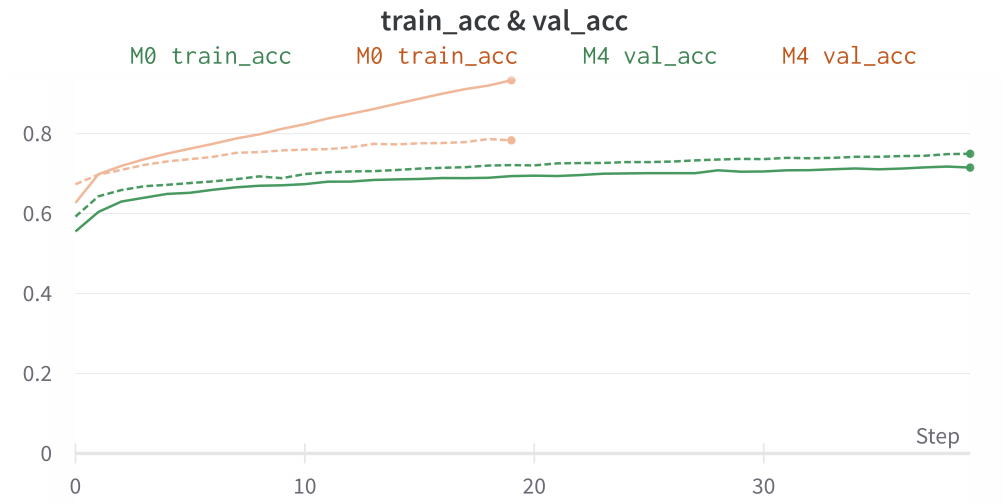


Figure 1.3: *M0 vs. M4, Acc Train and Validation Curves. Illustration by Author*

### 1.2.2 M1 against M5

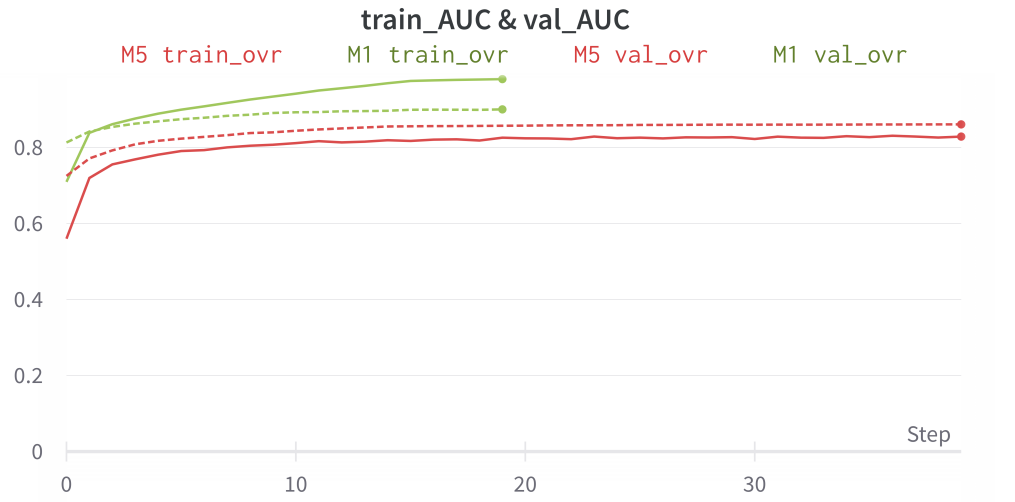


Figure 1.4: *M1 vs. M5, AUC Train and Validation Curves. Illustration by Author*


Figure 1.5: *M1 vs. M5, Loss Train and Validation Curves. Illustration by Author*

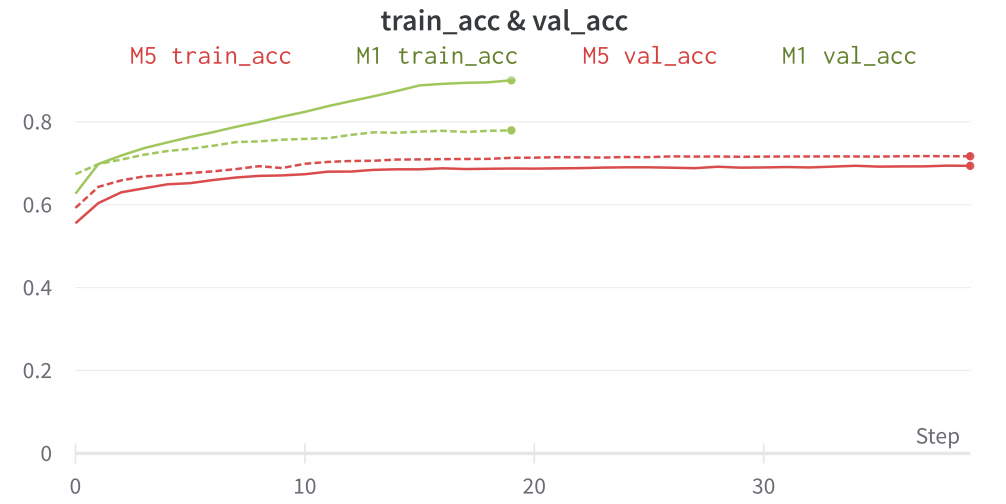


Figure 1.6: *M1 vs. M5, Acc Train and Validation Curves. Illustration by Author*

### 1.2.3 M2 against M6

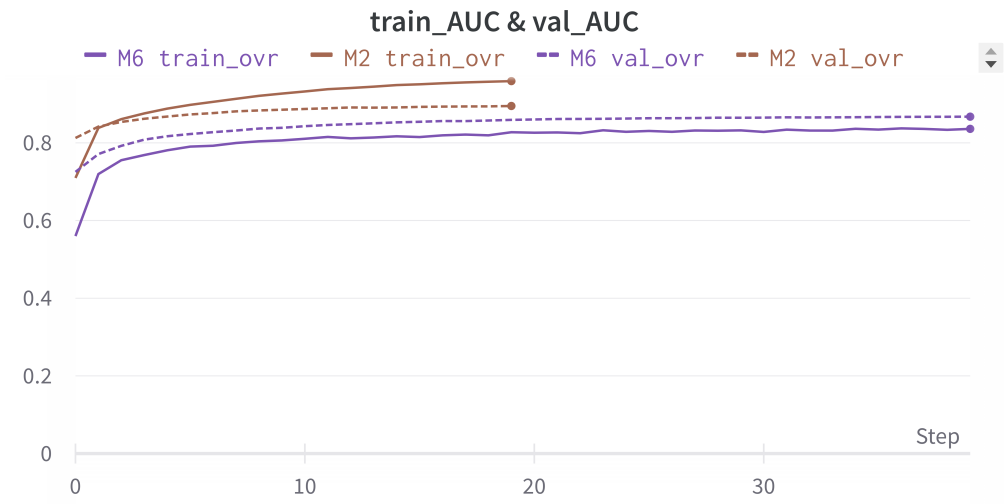


Figure 1.7: *M2 vs. M6, AUC Train and Validation Curves. Illustration by Author*


Figure 1.8: *M2 vs. M6, Loss Train and Validation Curves. Illustration by Author*

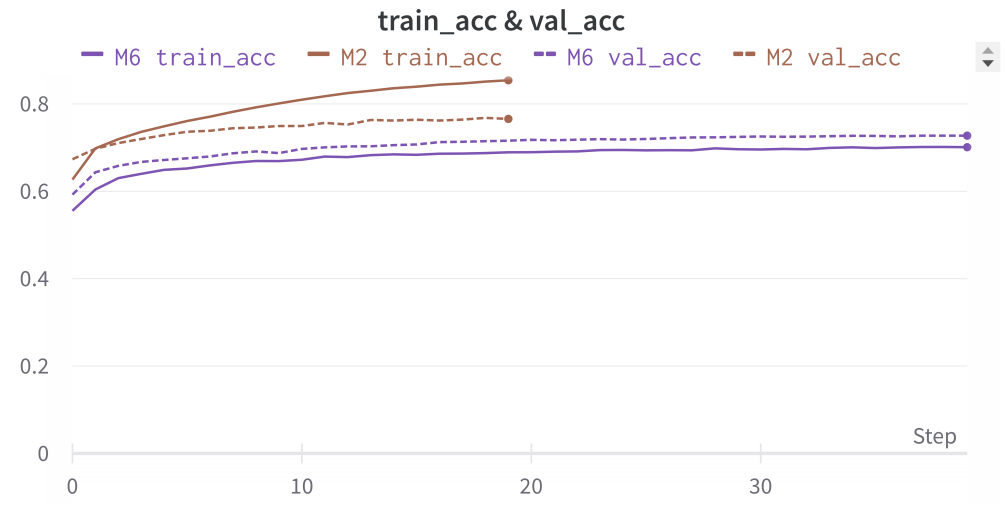


Figure 1.9: *M2 vs. M6, Acc Train and Validation Curves. Illustration by Author*

### 1.2.4 M3 against M7


Figure 1.10: *M3 vs. M7, AUC Train and Validation Curves. Illustration by Author*


Figure 1.11: *M3 vs. M7, Loss Train and Validation Curves. Illustration by Author*

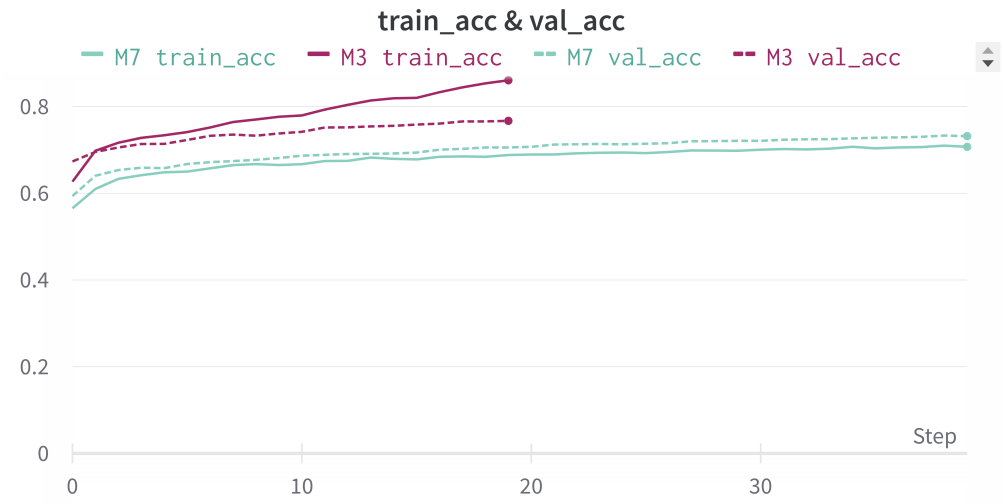


Figure 1.12: *M3 vs. M7, Acc Train and Validation Curves. Illustration by Author*

## 1.3 Testing

In this section, we assess the performance of the models on the test dataset by employing our chosen main evaluation metric, ROC-AUC with One VS Rest (OvR) strategy.

The *1.13* provides an overview of the performance of all the models trained in this thesis. As explained in earlier sections, models with superior performance are those without additional regularization, whereas models with higher regularization exhibit better generalization and can achieve similar or even superior performance compared to the non-regularized models if they undergo additional epochs of training.

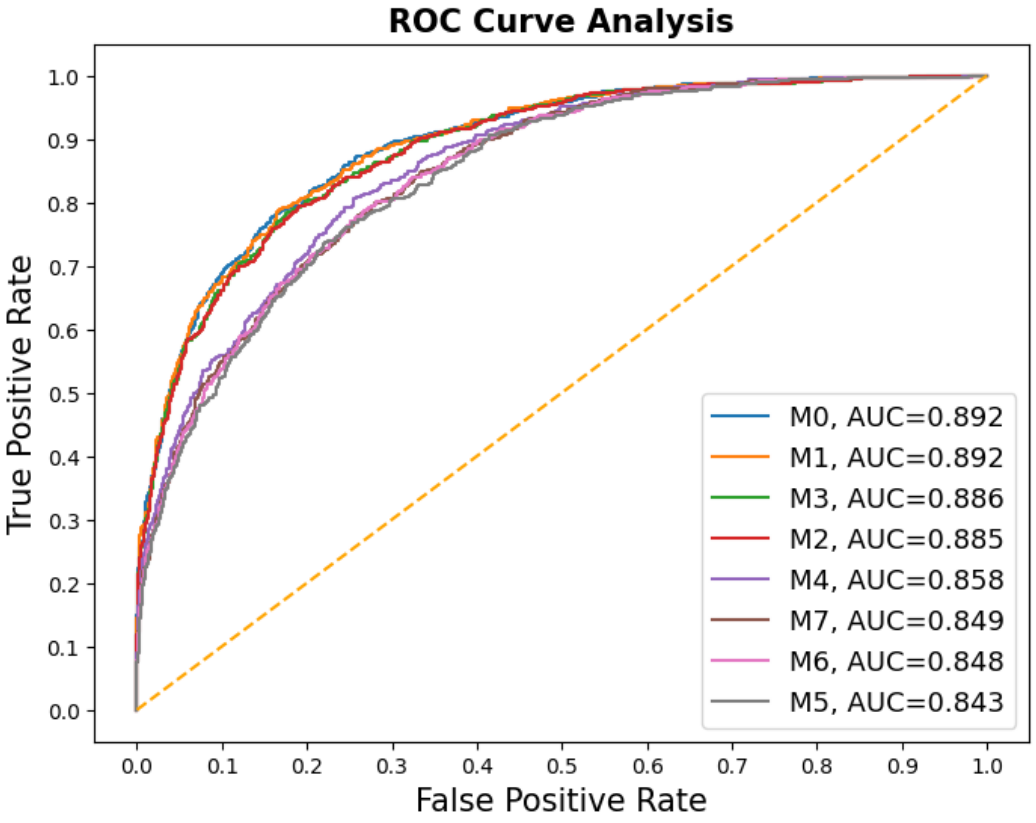


Figure 1.13: *ROC-AUC Results in Test Dataset. Illustration by Author*

## 1.4 API Service

The API service provides various endpoints accessible through the following URL:

```
http://<hosted-url>/docs
```

By opening this URL in a web browser, you can visualize the supported endpoints of the API (refer to *Figure 1.14*). This convenient visualization is made possible by using fastAPI, the framework chosen for building the API, which operates on the OpenAPI scheme.

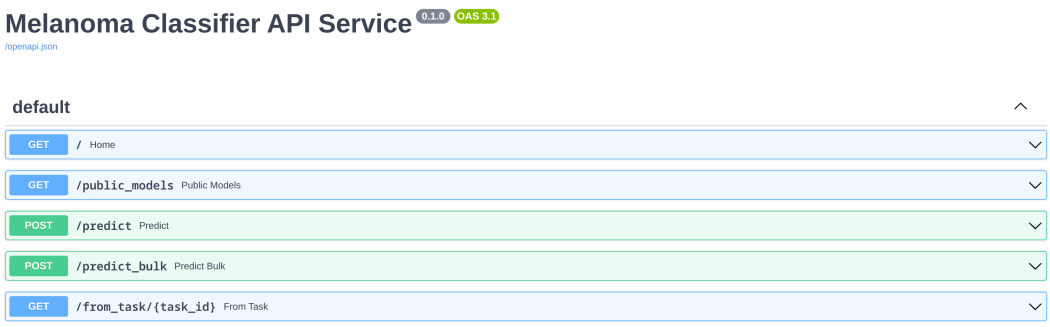


Figure 1.14: *API Service End-Points. Illustration by Author*

Below, we present an explanation of the functionality of each use case and it's end-point along with a brief code summary. However, for a comprehensive understanding of the implementation, we highly recommend referring to the source code attached to the thesis. The complete implementation can be found in the GitHub repository: https://github.com/wilberquito/melanoma.thesis.

### 1.4.1 Consulting The Available Models

We provided a mechanism to consult the available models that currently the API exposes. The supported model are configured in the configuration file of the API.

You can consult the exposed models by requesting:

```
http://<hosted-url>/public_models
```

The response of the API's JSON response should be something similar to this:

```
{
  "models": [
    "M0",
    "M1",
```

```
    "M2",
    "M3",
    "M4",
    "M5",
    "M6",
    "M7",
    "vicorobot.8c_b3_768_512_18ep_best_20_fold0",
    "vicorobot.8c_b3_768_512_18ep_best_fold0",
    "vicorobot.8c_b3_768_512_18ep_final_fold0"
  ]
}
```

### 1.4.2 Predict an Image

To predict an individual image, an endpoint is created. This endpoint expects an HTTP request containing a JPEG image in the body, along with the specified model name for performing the inference.

For example, a valid request to this endpoint might look like:

```
POST http://<hosted-url>/predict?model_id=M1
```

The API will respond with a JSON object containing a unique task identifier:

```
{
  "task_uuid": "721445f5-ff37-4c99-af74-7157925b4a7c"
}
```

### 1.4.3 Predict a Jar of Images

This endpoint is designed to handle requests for predicting multiple images simultaneously. The main distinction is that this endpoint has the capability to run the predictions in separate threads. This prevents the system from becoming unresponsive or stuck while predicting a large number of images, as it allows the processing to occur concurrently and independently of the main system thread.

To utilize this endpoint, you are required to provide two pieces of information in the HTTP request: the model you wish to use for image inference and a list of images attached to the request body. For example, you can use this endpoint as follows:

```
http://<hosted-url>/predict_bulk?model_id=M1
```

The API will respond with a JSON object containing a unique task identifier and the total number of images sended to the API:

```
{
  "task_uuid": "77d5e834-60a1-49b6-a71a-b3472dc21ce5",
  "num_images": 2
}
```

### 1.4.4 Consulting Task Predictions

Lastly, we needed and end-point that given the unique identifier task, we could consult the result of the prediction of the models for all images sended to the API.

You can consult a task prediction as follow:

```
http://<hosted-url>/from_task/77d5e834-60a1-49b6-a71a-b3472dc21ce5
```

A possible JSON response from the API of the consult of the task prediction could be:


```
[
  {
    "name": "ISIC_0052349.jpg",
    "probabilities": {
      "AK": 0.0007466986,
      "BCC": 0.0005002805,
      "BKL": 0.015733117,
      "DF": 0.00086343783,
      "SCC": 0.0007902466,
      "VASC": 0.0017217622,
      "melanoma": 0.017426228,
      "nevus": 0.9622182
    },
    "metadata": {
      "origin": "wilberquito",
      "net_type": "resnet18",
      "img_size": 256,
      "out_dim": 8
    },
    "prediction": {
      "target": false,
      "label": 7,
      "prediction": "nevus"
    }
  },
  {
    "name": "ISIC_1766619.jpg",
    "probabilities": {
      "AK": 0.00016253705,
      "BCC": 0.00011373816,
      "BKL": 0.00052201655,
      "DF": 0.0006251502,
      "SCC": 0.00032183932,
      "VASC": 0.00025595966,
      "melanoma": 0.0012655357,
      "nevus": 0.9967333
    },
    "metadata": {
      "origin": "wilberquito",
      "net_type": "resnet18",
      "img_size": 256,
      "out_dim": 8
    },
    "prediction": {
      "target": false,
```

```
      "label": 7,
      "prediction": "nevus"
    }
  }
]
```


## 1.5 UI Service

## 1.6 Micro-services comunication

The concept of handling long-running tasks in the background is often implemented to improve the responsiveness and scalability of the system. This approach allows the client to initiate a task and receive a unique identifier, which can later be used to query or retrieve the results of that task (*Figure 1.15*).

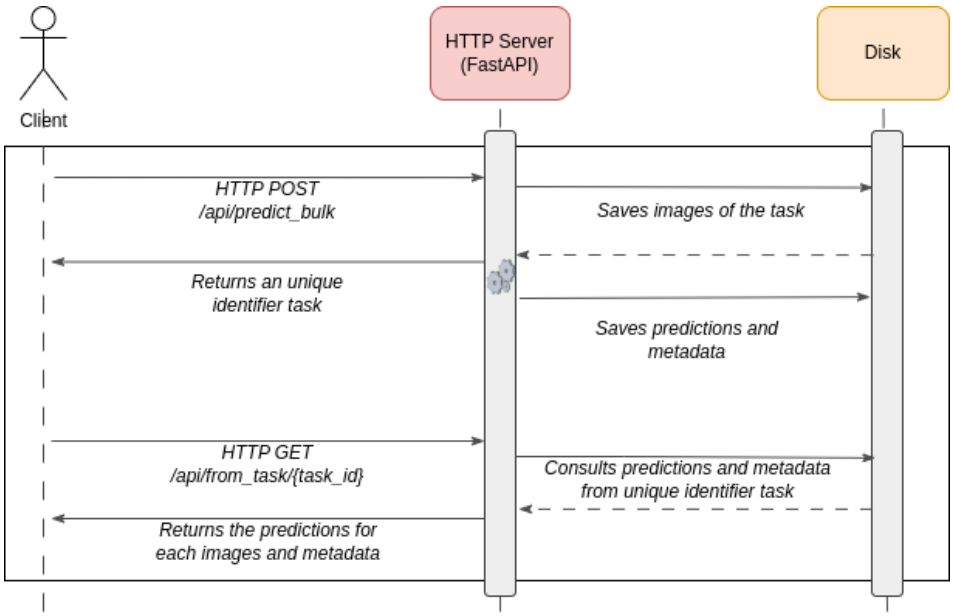


Figure 1.15: *Inferring Images Through the Background Task Mechanism. Notice that in this way we avoid to congest the API thread. Illustration by Author*

When an HTTP task is required, it comes with the identifier of the model that will be used to infer the image or a jar of images. This model is searched for in a volume configuration file located in the API. The configuration file must contain the entry for the model identifier and also include the path to the binary trained model. The trained model is deserialized and used at runtime through the PyTorch API.

Once the path of the model in the operating system is obtained it is as easy as load the searilized model using the PyTorch API as follow:

```
instance_nn = class_nn(out_dim)
checkpoint = torch.load(pytorch_model_path, map_location=device)
model_state_dict = checkpoint['model_state_dict']
instance_nn.load_state_dict(model_state_dict, strict=True)
```

CHAPTER 2

# Conclusions and Future Work

# Appendices

# A Manual Installation

We are pleased to offer you all the necessary tools required to effortlessly set up the CAD infrastructure. Managing the installation of Python and NPM packages is something you need not worry about, as we have already created comprehensive images that encompass all the essential packages, ensuring a seamless execution process.

It's important to note that the tools needed for the installation are primarily focused on project building and remain independent of the services created. Although our example uses Fedora Linux 38 as the operating system, rest assured that all the tools used in this platform are cross-platform compatible, guaranteeing smooth execution on other operating systems as well.

Here is the list of tools needed for the installation:

- Git
- Docker or Podman
- docker-compose
- cURL

Before proceeding, please make sure you have all the required tools listed above. Next, you'll need to become a superuser, and you can do that with the following command:

```
$ sudo su
```

If you opt for Docker, execute the following command to initiate the installation process:

```
$ curl https://raw.githubusercontent.com/wilberquito/melanoma.thesis/main/MAKE.sh | bash
```

If you prefer Podman, you will first need to install the podman docker package to enable Podman to work harmoniously with Docker commands:

```
$ sudo dnf install -y podman-docker
```

Afterward, you can utilize the MAKE.sh script to build the architecture:

```
$ curl https://raw.githubusercontent.com/wilberquito/melanoma.thesis/main/MAKE.sh | bash
```

To confirm that the CAD infrastructure services are operational, you can locate the two containers using the following command:

```
$ docker ps
```

You should see the two containers listed as shown in the image below:

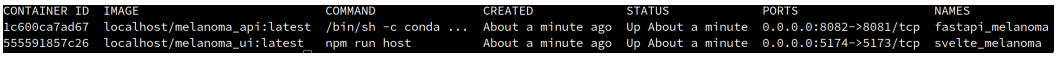


Figure 1: *Container CAD Services. Illustration by Author*

The API is accessible through port 8082. To view the documentation, simply open any web browser and enter the following URL:

```
http://127.0.0.1:8082/docs
```

Similarly, to access the CAD UI, you can use port 5174. Just open your web browser and enter the following URL:

```
http://127.0.0.1:5174
```

Rest assured that with these tools and guidelines, you'll have a smooth and successful installation of the CAD infrastructure.

# Bibliography


[A. Esteva 2017] B. Kuprel A. Esteva and R. Novoa. *Dermatologist-level classification of skin cancer with deep neural networks*, 2017. Available at https://www.nature.com/articles/nature21056. (Not cited.)

[Cancer.Net 2020] Cancer.Net. *Skin cancer: Introduction*, 2020. Available at https://www.cancer.net/es/tipos-de-c. (Not cited.)

[Cancer.Net 2023] Cancer.Net. *Melanoma: Statistics*, 2023. Available at https://www.cancer.net/cancer-types/melanoma/statistics. (Not cited.)

[Docker 2021] Docker. *Docker Documentation*, 2021. Available at https://www.docker.com. (Not cited.)

[Farber 2020] Dana Farber. *Skin cancer, treatment*, 2020. Available at https://www.dana-farber.org/skin-cancer/. (Not cited.)

[FastAPI 2021] FastAPI. *FastAPI Documentation*, 2021. Available at https://fastapi.tiangolo.com/. (Not cited.)

[IBM 2020] IBM. *What is machine learning?*, 2020. Available at https://www.ibm.com/topics/machine-learning. (Not cited.)

[IBM 2021] IBM. *Machine learning, explained*, 2021. Available at https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained. (Not cited.)

[ISIC 2019] ISIC. *ISIC Challenge*, 2019. Available at https://challenge.isic-archive.com/. (Not cited.)

[ISIC 2020] SIIM & ISIC. *SIIM-ISIC Melanoma Classification*, 2020. Available at https://www.kaggle.com/competitions/siim-isic-melanoma-classification/code. (Not cited.)

[ISIC 2022] ISIC. *The International Skin Imaging Collaboration*, 2022. Available at https://www.isic-archive.com/. (Not cited.)

[J. Wang 2022] O. Shaik J. Wang R. Turko. *What is a Convolutional Neural Network?*, 2022. Available at https://poloclub.github.io/cnn-explainer/. (Not cited.)

[K. Stacke 2019] J. Unger K. Stacke G. Eilertsen and C. Lundström. *What is Weak AI*, 2019. Available at https://arxiv.org/pdf/1909.11575.pdf. (Not cited.)

[Kaiming He 2013] Xiangyu Zhang Kaiming He and Jian Sun. *Deep Residual Learning for Image Recognition*, 2013. Available at https://arxiv.org/pdf/1512.03385v1.pdf. (Not cited.)

[Korotkov 2012] K. Korotkov and R. García. *Computerized analysis of pigmented skin lesions: A review*, 2012. Available at https://www.sciencedirect.com/science/article/abs/pii/S0933365712001108. (Not cited.)

[L. Davis 2019] S. Shalin L. Davis and A. Tackett. *Current state of melanoma diagnosis and treatment*, 2019. Available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6804807/pdf/kcbt-20-11-1640032.pdf. (Not cited.)

[Marchetti 2020] M. Marchetti. *Computer algorithms show potential for improving dermatologists' accuracy to diagnose cutaneous melanoma: Results of the International Skin Imaging Collaboration 2017*, 2020. Available at https://www.sciencedirect.com/science/article/abs/pii/S0190962219323734. (Not cited.)

[Marghoob 2022] A. Marghoob and N. Jaimes. *Overview of dermoscopy*, 2022. Available at https://www.medilib.ir/uptodate/show/13521. (Not cited.)

[Marr 2021] B. Marr. *What is Weak AI*, 2021. Available at https://bernardmarr.com/what-is-weak-narrow-ai-here-are-8-practical-examples. (Not cited.)

[mrdbourke 2021] mrdbourke. *Learn PyTorch for Deep Learning: Zero to Mastery book*, 2021. Available at https://www.learnpytorch.io/. (Not cited.)

[Oncol 2021] W. Oncol. *Cutaneous Malignant Melanoma: A Review of Early Diagnosis and Management*, 2021. Available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7935621/pdf/wjon-12-007.pdf. (Not cited.)

[Partridge 2013] Dr. Partridge. *The Origins of Skin Cancer and How We Treat it Today*, 2013. Available at https://ohcare.com/the-origins-of-skin-cancer-and-how-we-treat-it-today/. (Not cited.)

[Podman 2021] Podman. *Podman Documentation*, 2021. Available at https://podman.io/. (Not cited.)

[Q. Ha 2020a] B. Liu Q. Ha and F. Liu. *Identifying Melanoma Images using EfficientNet Ensemble: Winning Solution to the SIIM-ISIC Melanoma Classification Challenge*, 2020. Available at https://arxiv.org/pdf/2010.05351.pdf. (Not cited.)

[Q. Ha 2020b] B. Liu Q. Ha and F. Liu. *SIIM-ISIC-Melanoma-Classification-1st-Place-Solution*, 2020. Available at https://github.com/haqishen/SIIM-ISIC-Melanoma-Classification-1st-Place-Solution. (Not cited.)

[Russell 1950] Stuart J. Russell and Peter Norvig. *Artificial Intelligence A Modern Approach.*, 1950. Available at http://repo.darmajaya.ac.id/3800/1/Artificial%20Intelligence%20A%20Modern%20Approach%20%283rd%20Edition%29.pdf%20%28%20PDFDrive%20%29.pdf. (Not cited.)

[Salary.com 2021a] Salary.com. *Hourly wage for Data Scientist*, 2021. Available at https://www.salary.com/research/salary/benchmark/data-scientist-i-hourly-wages. (Not cited.)

[Salary.com 2021b] Salary.com. *Hourly wage for Data Scientist*, 2021. Available at https://www.salary.com/research/salary/posting/computer-and-information-research-scientist-salary. (Not cited.)

[Salary.com 2021c] Salary.com. *Hourly wage for Data Scientist*, 2021. Available at https://www.salary.com/research/salary/listing/data-engineer-hourly-wages. (Not cited.)

[Srivastava 2014] N. Srivastava. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, 2014. Available at https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf. (Not cited.)

[sveltejs 2021] sveltejs. *SvelteKit Documentation*, 2021. Available at https://kit.svelte.dev/docs. (Not cited.)

[Turing 1950] A. Turing. *COMPUTING MACHINERY AND INTELLIGENCE.*, 1950. Available at https://academic.oup.com/mind/article/LIX/236/433/986238. (Not cited.)

[uslaev 2020] Eugene uslaev Alexande and Parinov. *Albumentations: Fast and Flexible Image Augmentations*, 2020. Available at https://www.mdpi.com/2078-2489/11/2/125. (Not cited.)