A Platform for Classifying Melanoma

Wilber Eduardo Bermeo Quito

Master in Data Science

Higher Polytechnic School

University of Girona

August 8, 2023

Summery

Introduction

Motivations

- Enhance AI[1] knowledge.
- Automation as way to democratize access to research and AI solutions.
- CAD[2] system are promising path towards medical automation.

[1] Artificial Intelligence.
[2] Computer-Aided Diagnosis.

Objectives

- Gain expertise in deep learning theory and its real-world applications.
- Explore and study the optimal approach for utilizing the distribution of dermoscopy images from the dataset during the training process.
- Propose and train deep learning models using transfer learning on ISIC[3] Challenge melanoma images.
- Create an easy deployment CAD infrastructure running in Docker, with the trained models, a user-friendly web UI[4] and a HTTP[5] API[6].

[3] Skin Imaging Collaboration.
[4] User Interface.
[5] Hypertext Transfer Protocol.
[6] Application Programming Interface.

Domain

Problem

Detection of Melanoma Skin Cancer

- Melanoma exhibits a high mortality rate.
- Dermoscopy procedures are utilized for melanoma detection.
- Dermoscopy images are examined by professionals to study cutaneous lesions.
- Several studies have shown that melanoma task classification using CAD systems achieve comparable or superior results to dermatologists.

Solution

CAD Training and Deployment Pipeline

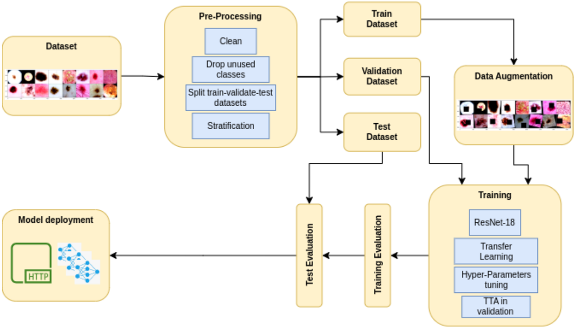


Figure 1: *CAD Infrastructure Pipeline.*

Micro-Service Architecture to Infer Images

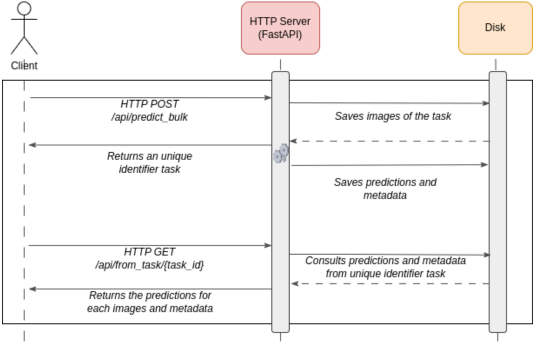


Figure 2: *Inferring Images Through the Background Task Mechanism.*

Concerns

Ethical Concern

- The solution employs "black box" models, lacking explain-ability.
- The thesis presents a CAD tool designed to aid human decision-making rather than being an autonomous decision-making system.

Regulatory Framework

- When dealing with medical images, obtaining signed consent is necessary for data publication.
- Recent research collaborations prioritize data sharing through de-identification methods to tackle these challenges.
- The thesis made use of the ISIC Archive database, which serves as a publicly accessible resource.

Data

Origin Data

- The data originates from the ISIC Archive.
- It includes images from the years 2019 and 2020.
- The images are available in three different resolutions: 512x512, 768x768, and 1024x1024 pixels.
- The dataset contains more than eight distinct classes.

Used Data

- Resolution selected: 512x512 pixels.
- The used dataset comprises 31,265 distinct image samples.
- Eight classes were selected to work with.
- Imbalanced dataset.

Classes Distribution in the Dataset

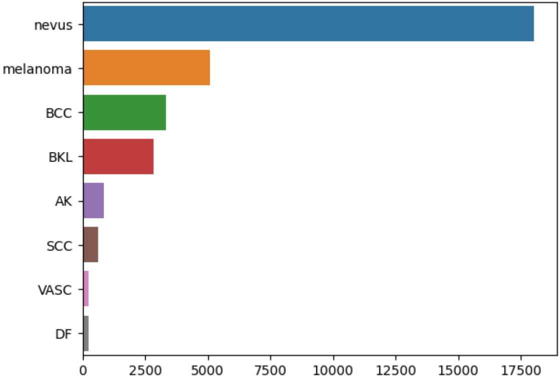


Figure 3: *Data Distribution.*

Train, Validation and Test Sets

- The dataset was stratified to ensure an equal distribution of classes in each subset.
- The training set was created using 80% of the dataset, the validation set using 10%, and the test set using the remaining 10%.

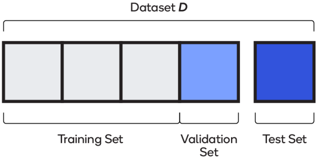


Figure 4: *Holdout Set Scheme. Illustration by Qualcomm*

Data Augmentation

The train dataset (Figure 5),

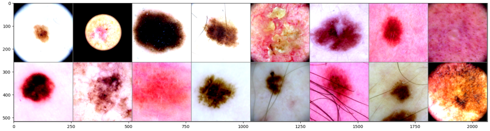


Figure 5: *Random Sample of Images.*

Is mapped into an augmented train dataset (Figure 6).

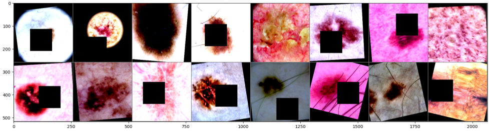


Figure 6: *Augmented Random Sample of Images.*

Modeling

General Modeling Information

- Eight trained models with different ML[7] thecniques.
- Used the ResNet18 pre-trained weights.
- SGD as optimizer.
- Cross-entropy as loss function.
- Model training performance were evaluated with AUC[8] metric.

[7] Machine Learning.
[8] Area Under the Curve.

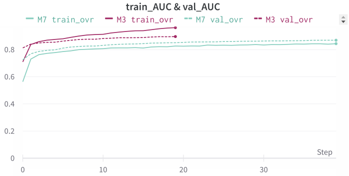
	M0	M1	M2	M3	M4	M5	M6	M7
Model Architecture	R18M	R18M	R18M	R18M	R18DM	R18DM	R18DM	R18DM
Epochs	20	20	20	20	40	40	40	40
Batch Size	400	400	400	400	1024	1024	1024	1024
Scheduler		SLR	CALR	CAWR		SLR	CALR	CAWR
Data Augmentation	No	No	No	No	Yes	Yes	Yes	Yes
Dropout Regularization	No	No	No	No	Yes	Yes	Yes	Yes
GPU	TT4	TT4	TT4	TT4	NA100	NA100	NA100	NA100
Training Time	1h 45m	1h 22m	1h 43m	1h 38m	1d 7h 30m	1d 7h 4m	1d 7h 1m	1d 12h 55m

Table 1: *Training Information For Each Model. Empty spaces represent non-use of that feature.*

	Train AUC	Val AUC	Train Recall	Val Recall	Train Acc	Val Acc
M0	0.952	0.903	0.756	0.676	0.835	0.778
M1 $\star$	0.947	0.900	0.695	0.633	0.829	0.779
M2 $*$	0.933	0.895	0.658	0.609	0.808	0.765
M3 $\bullet$	0.935	0.896	0.663	0.605	0.811	0.767
M4	0.886	0.877	0.478	0.475	0.757	0.750
M5 $\star$	0.867	0.861	0.423	0.403	0.728	0.717
M6 $*$	0.874	0.868	0.451	0.440	0.738	0.728
M7 $\bullet$	0.877	0.849	0.470	0.432	0.742	0.732
Mean	94.175%	89.850%	69.300%	63.075%	82.075%	77.225%
SD	0.921%	0.370%	4.509%	3.260%	1.327%	0.727%
Mean	87.600%	86.875%	45.550%	44.400%	74.125%	73.175%
SD	0.787%	0.655%	2.445%	3.084%	1.204%	1.372%

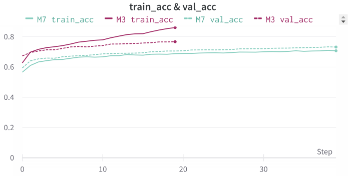
Table 2: *Train & Validaton Metrics.*

M3 vs. M7



(c) AUC Curves

(a) Loss Curves



(b) Accuracy Curves

Figure 7: *M3 vs. M7. Combined AUC, Loss and Accuracy Curves.*

Workflow & Results

Workflow Methodology

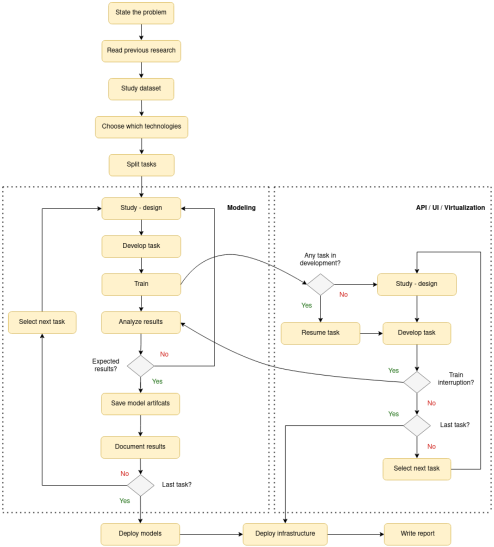


Figure 8: *Activity Diagram Describing the Workflow Methodology.*

Results

Testing Models

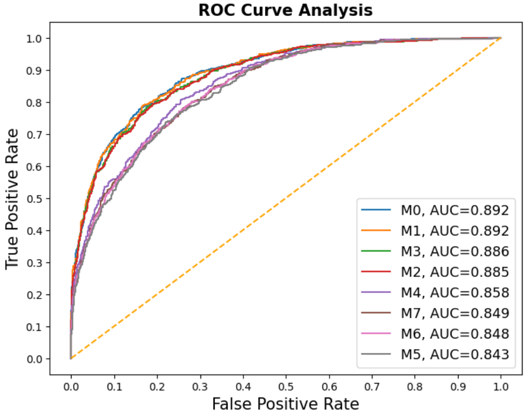


Figure 9: *ROC-AUC Results in Test Dataset.*

API Service

To access the documentation of the API, you can make a request in a web browser using the following URL[9]:

http://<api>/docs

The web browser will display the endpoints of the API.

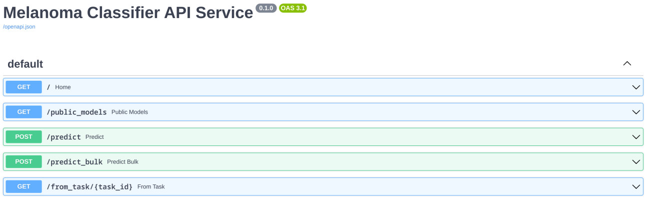


Figure 10: *API Service End-Points.*

[9] Uniform Resource Locator.

Exposed Models

You can consult the exposed models by requesting:

`http://<api>/public_models`

The response of the API's JSON[10] 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"
  ]
}
```


[10] JavaScript Object Notation.

Predict Images

You can consult the exposed models by requesting:

```
http://<api>/predict_bulk?model_id=<model_id>
```

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
}
```

Consult Prediction

You can consult a task prediction as follow:

```
http://<api>/from_task/<task_uuid>
```

A potential JSON response from the API regarding the task prediction:


```
[
  {
    "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
    },
    . . .
  }
  . . .
]
```

UI Service

Accesing the UI service through a web browser using the following URL:

http://<ui>

As a result, a single-page web application with several interactive buttons will appear (Figure 11).

The state of these button depends on the application state.

Figure 11: *Main Interactive buttons of the UI Service.*

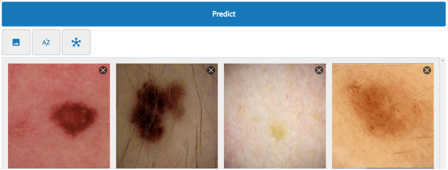


Figure 12: *Dermoscopy Images Loaded in the UI.*

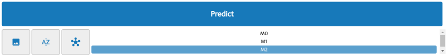


Figure 13: *Selecting Exposed Models by the API.*

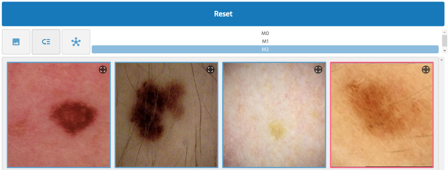


Figure 14: *UI State After Prediction Response.*

Pop-up Extra Information

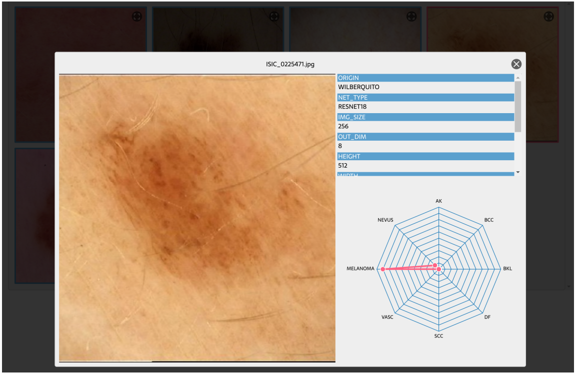


Figure 15: *Extra Prediction Information.*

Open Source CAD Infrastructure

The thesis assets (trained models weights and configurations) can be found in this public GitLab repository:

`https://gitlab.com/wilberquito/open.thesis`

The CAD infrastructure (install guide, install script, experiments and source code) can be found in this public GitHub repository:

`https://github.com/wilberquito/melanoma.thesis`

After following the instruction and installing the required tools, installing the CAD infrastructure should be as simple as running this command in a bash terminal.

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