PREPARED FOR: DEEP LEARNING COURSE, PROFESSOR TONY HUYNH
PREPARED BY: FEDERICA BRUNI, ALESSIO BORGI, MARIA EMILIA RUSSO
23 DECEMBER 2022

PROJECT PROPOSAL

SKIN ME

TABLE OF CONTENTS

01 WHAT IS SKIN ME?

SkinMe is a project we are working on.
Our aim is to build a system able to detect skin cancer.

Skin cancer is the most common human malignancy and it strikes one in five people by the age of 70.

As for all cancers types also for skin cancer timing is essential. In fact when caught early, skin cancers are highly curable but the probability of survival drops to 14% when it is detected in a later stage.

It is therefore essential to make early detection simple and open to everyone. This is what SkinMe is about: an intelligent dermatology.

Today's dermatology system has 2 main problems:

1. dermatological visits are expensive and with annual frequency which sometimes can be too late (as it was in Federica's experience)
2. monthly suggested self-analysis are complex and not reliable

SkinMe aims to build a system able to make early detection free, simple and open to everyone through a deep learning model able to recognize a skin cancer from a single image.

Our model will help patients in the process of self-analysis as they will simply need to take a picture of their mole and the trained deep learning system will return a diagnosis.

Notice that the model analysis must not replace a visit with a specialist, it is just an additional tool for early detection.

02 PROJECT OVERVIEW

This project's aim is the development of a software solution that automates diagnosis for disease classification from a dataset of dermoscopic skin cancer images.

The approach that was taken was to develop two deep learning models that would tackle this task and comprehensively compare their performance.

To carry out this project we will be implementing two image classifiers that will be able to differentiate a new, unlabeled image of skin cancer into one of the seven classes of skin cancer.
To achieve the classification of the disease, the models used labelled data to learn the relationship between the input data (images) and the labels.

The classifiers that we will implement are the CNN (convolutional neural network), and the LSTM (Long short-term memory RNN).

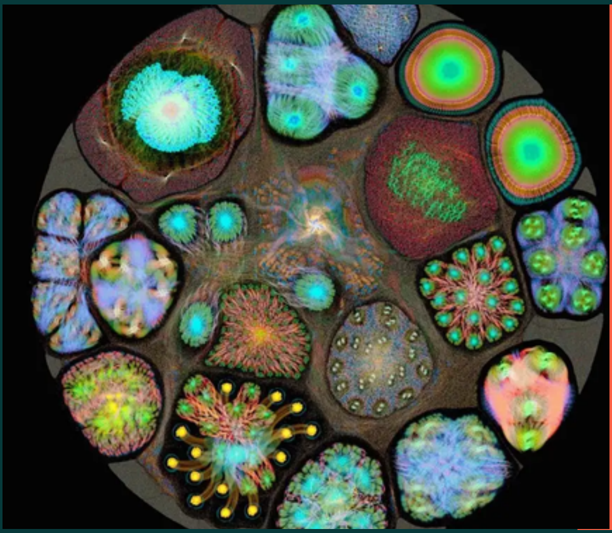
03 THE DATASET

For the dataset we chose the state-of-art in the field: the HAM10000 ("Human Against Machine with 10000 training images") dataset.

Credits and attributions:
https://dataverse.harvard.edu/dataset.xhtmlpersistentId=doi:10.7910/DVN/DBW86T

In the dataset, we have 10015 close-up, high quality, centered, dermatoscopic images of skin lesions. All the images have color, with height 450 pixels and width 600 pixels — this is very convenient since we don't have to worry about standardizing our images to the same size.

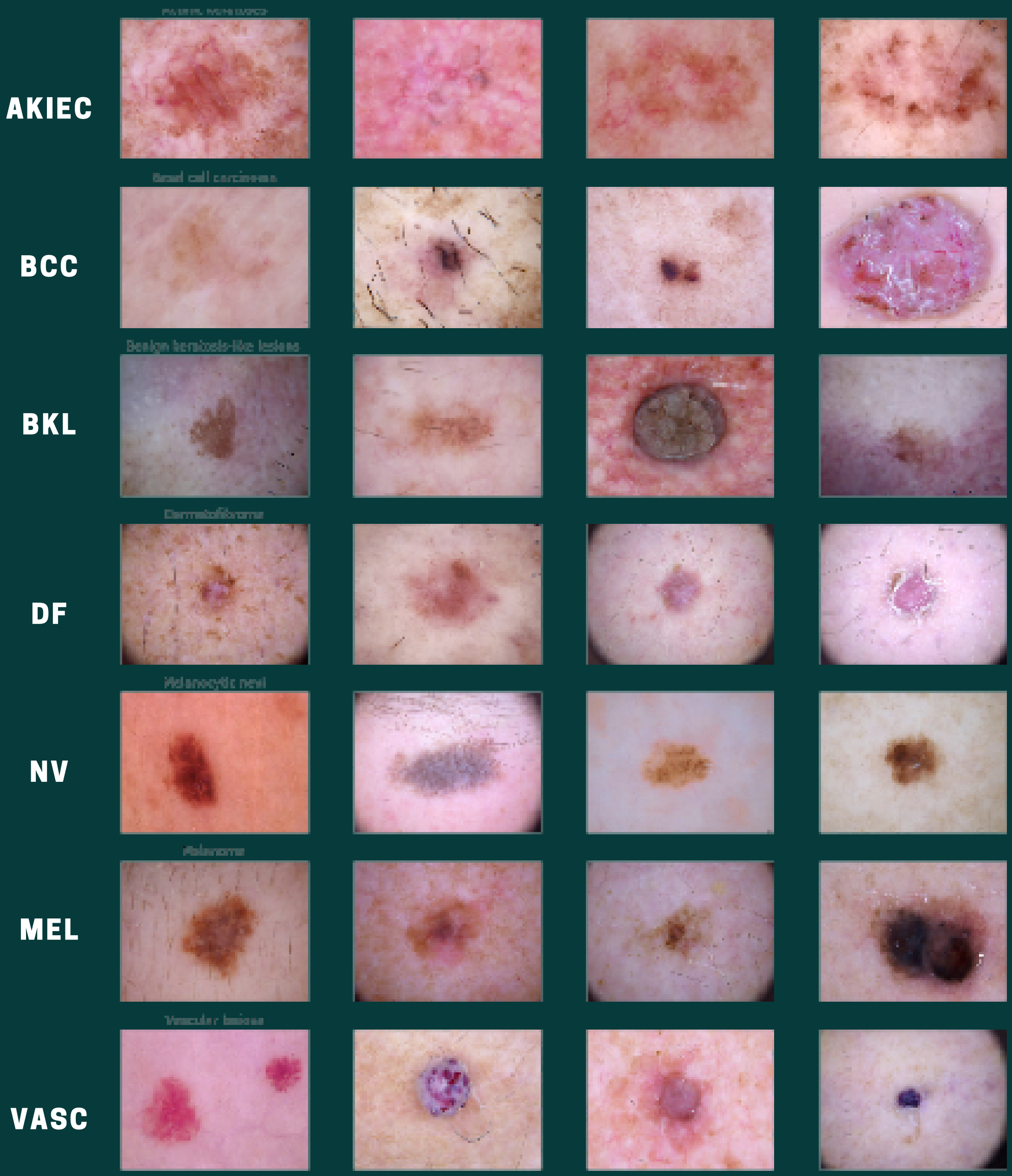
The dataset also consists of metadata for each of the images of the patients.
The metadata variables available to us are:
- lesion_id: this refers to the patient associated with the skin lesion
- image_id: this refers to the image name in the HAM10000 data folders
- dx: this refers to the diagnosis
- age: this refers to the age of the patient associated with the skin lesion image
- sex: this refers to the sex of the patient associated with the skin lesion image
- localization: this refers to the location of the skin lesion on the body



It has 7 different classes of skin cancer:
1. Melanocytic nevi
2. Melanoma
3. Benign keratosis-like lesions
4. Basal cell carcinoma
5. Actinic keratoses
6. Vascular lesions
7. Dermatofibroma

03 THE DATASET

IMAGES OF EACH CANCER TYPE



04

DATA ANALYSIS

To begin our investigation we carried out some exploratory data analysis on the metadata of the images.
From the metadata we were able to extract many information about the patients and the disease.

01 — Frequency of disease classes

We found out that skin cancer of class nv is the most common one. We noticed that there is a high class inbalance in the dataset which we will have to fix.

02 — Frequency of age groups

The age group most at risk is that of ages between 40 and 48.

03 — Frequency of disease localizations

The most frequent locations of the body that skin cancer appeared was the back and the lower extermity.

04 — Frequency of gender

Males are slightly more affected.

05
DATA PREPARATION

The next step of our work was to pre-process the data.

Firstly we divided the data into train and test sample.
The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. This is exactly what we need in order to be able to compare CNN and LSTM.
We adopted the classic 80-20 ratio.

The next step in the data preparation was to find a way to overcome the class imbalance we found in the data analysis process.
Our first strategy was the random oversampling. Random oversampling increases the size of the training data set through repetition of the original examples. The result is that all disease classes have 5000 examples.
However it does not cause any increase in the variety of training examples.

At the end we chose SMOTE.
SMOTE creates new (artificial) training examples based on the original training examples. For instance, if it sees two examples (of the same class) near each other, it creates a third artificial one, bang in the middle of the original two.
Oversampling using SMOTE not only increases the size of the training data set, it also increases the variety.

The SMOTE algorithm is parameterized with k_neighbors (the number of nearest neighbors it will consider) and the number of new points you wish to create. Each step of the algorithm will:
1 - Randomly select a minority point.
2 - Randomly select any of its k nearest neighbors belonging to the same class.
3 - Randomly specify a lambda value in the range [0, 1].
4 - Generate and place a new point on the vector between the two points, located lambda percent of the way from the original point.

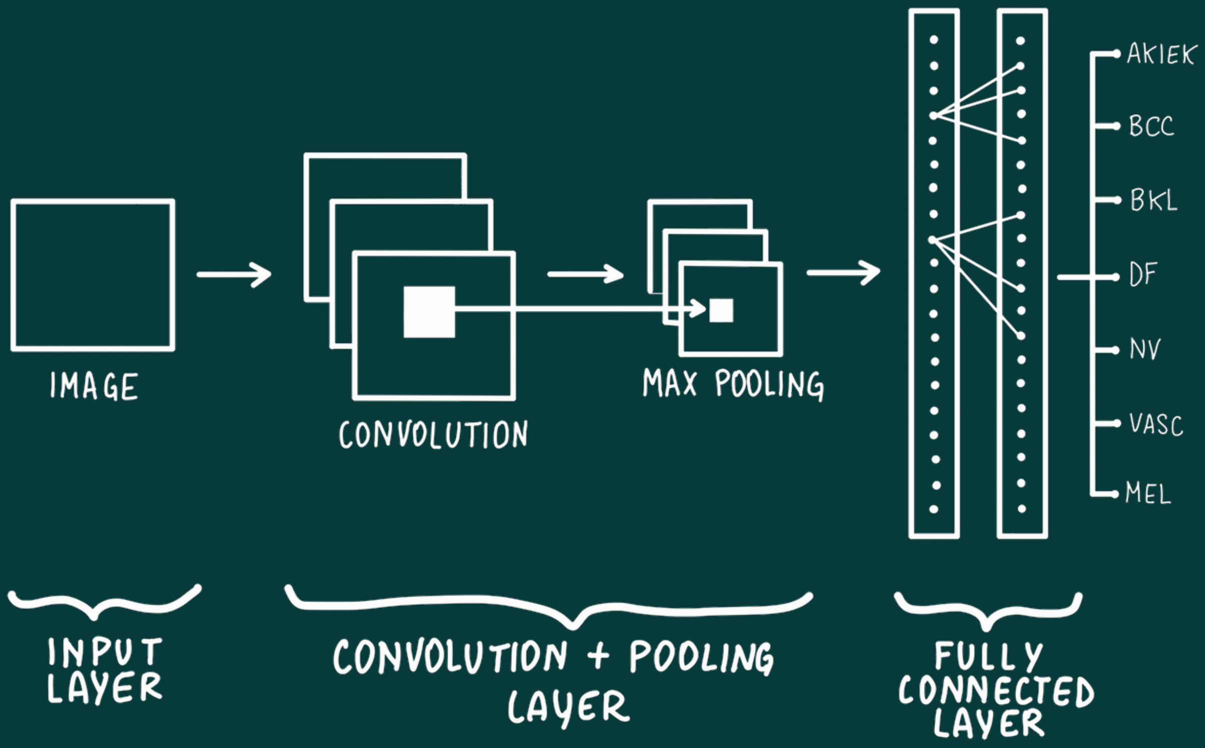
06 CNN

The next step of our work was to implement the classifiers and fit them to the data. We started working with the CNN classifier.

Furthermore, we reshaped and normalized the images and we also encoded the labels to one-hot vectors so that they can be fitted to the model.

We started working with a very simple model and then we refined it until we reached a optimal result.

The model we chose is



06 CNN

We used the Keras Sequential API, where we added one layer at a time, starting from the input.

The first is the convolutional (Conv2D) layer.
We choosed to set 12 filers for the first layer and increase them in the following Conv2D layers starting from 32, 64, 128 up to 256. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter.
We chose ReLu as activation function for the convolutional layer.

The second important layer is the pooling (MaxPool2D) layer.
These are used to reduce computational cost and overfitting. We choose to set the pooling size to 2x2.

Combining convolutional and pooling layers, CNN are able to combine local features and learn more global features of the image.

The Flatten layer is use to convert the final feature maps into a one single 1D vector.
This flattening step is needed so that we can use a fully connected layers.
It combines all the found local features of the previous convolutional layers.

Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their wieghts to zero) for each training sample. This drops randomly a portion of the network and forces the network to learn features in a distributed way.
When defining the Dropout layers, we specify 0.2, meaning that 20% of the layers will be dropped.
This technique also improves generalization and reduces the overfitting.

06 CNN

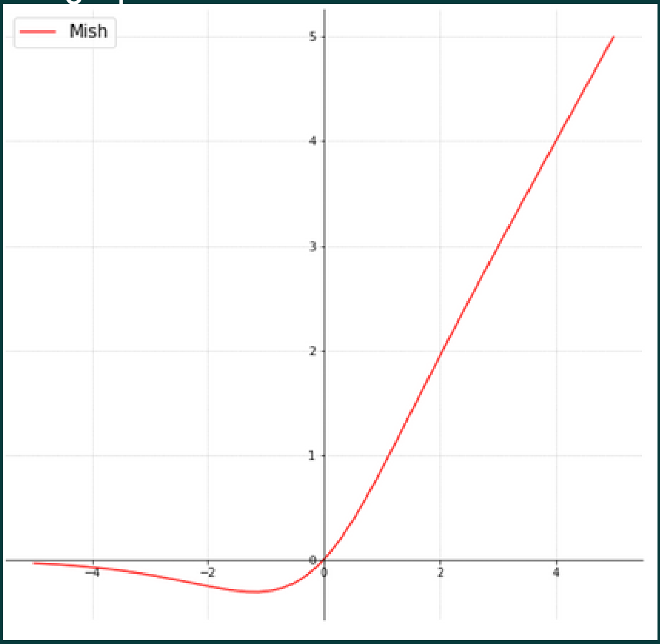
In the end we used five fully-connected (Dense) layers.
In the first four dense layers we used Mish as activation function while in the last layer we used softmax to output distribution of probability of each class of skin cancer.

Mish is a a new (2019) deep learning activation function that shows improvements over both Swish (+.494%) and ReLU (+ 1.671%) on final accuracy.

Source: https://arxiv.org/vc/arxiv/papers/1908/1908.08681v2.pdf

Mish is a non-monotonic neural activation function developed by Cornell University.
It can be defined as: $f(x) = x \cdot tanh(softplus(x))$
where, $softplus(x) = \ln(1 + ex)$ is the softplus activation function.

The graph of Mish is



07

LSTM

For the LSTM we did the exact same process for preparing my data with some minor alterations.
In fact LSTMs expects our data to be in a specific format, usually a 3D array. Thus we needed to reshape the images in a different way than we did for CNN. In fact CNN is working with 4D arrays while LSTM with 3D arrays.

As for CNN we used the Keras Sequential API, where we added one layer at a time, starting from the input. We started with a simple model and built up from there looking at the results.

The first is the LSTM layer.
Our LSTM layers were set to have 64 dimensionality of the output space.

The second important layer is the dropout layer.
Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their wieghts to zero) for each training sample. This drops randomly a propotion of the network and forces the network to learn features in a distributed way.
When defining the Dropout layers, we specify 0.2, meaning that 20% of the layers will be dropped.
This technique also improves generalization and reduces the overfitting.

Then we added the Dense layer that specifies the output of 7 unit.
We used softmax to output distribution of probability of each class of skin cancer.

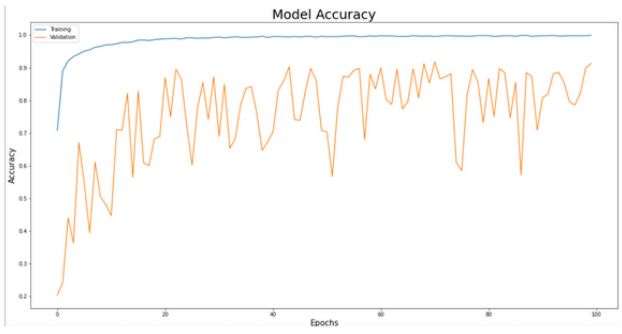
Next, we fit the model to run on 100 epochs with a batch size of 128.

08

RESULTS

CNN

RESULTS

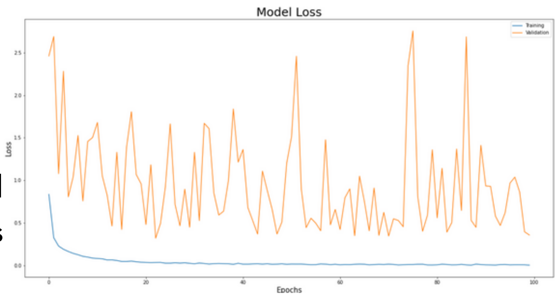


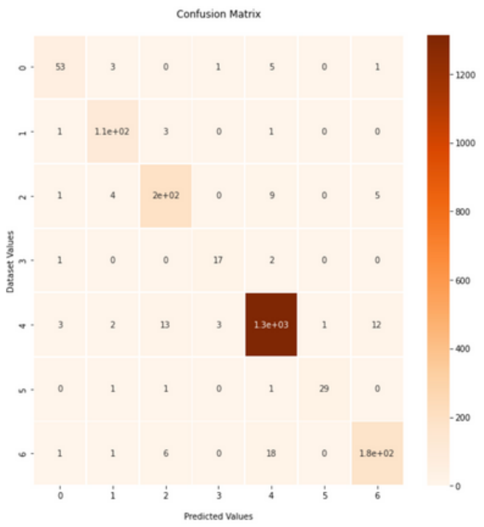
Model accuracy

The accuracy curve is always increasing as the number of epochs increases.
10 epochs is the best trade-off between accuracy and performance, 10 is a low number which means that the model is very effective.

Model loss

The loss curve is always decreasing as the number of epochs increases.
10 epochs is the best trade-off between loss and performance, 10 is a low number which means that the model is very effective.





Confusion matrix

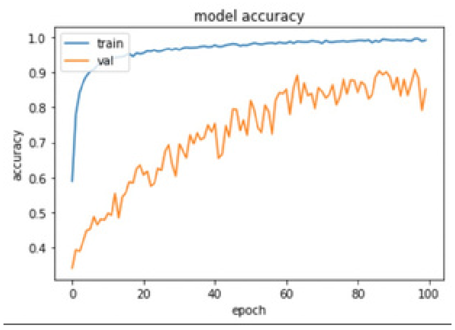
The confusion matrix shows that our model works extremely well with all the skin cancer classes.

08

RESULTS

LSTM

RESULTS

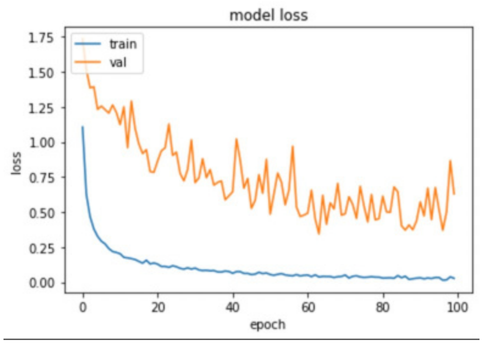


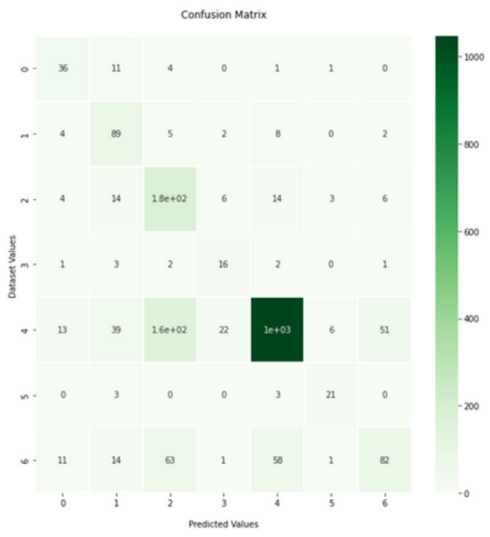
Model accuracy

The accuracy curve is always increasing as the number of epochs increases.
15 epochs is the best trade-off between accuracy and performance, 15 is a low number which means that the model is very effective.

Model loss

The loss curve is always decreasing as the number of epochs increases.
15 epochs is the best trade-off between loss and performance, 15 is a low number which means that the model is very effective.





	0	1	2	3	4	5	6
0	36	11	4	0	1	1	0
1	4	89	5	2	8	0	2
2	4	14	1.8e+02	6	14	3	6
3	1	3	2	16	2	0	1
4	13	39	1.6e+02	22	1e+03	6	51
5	0	3	0	0	3	21	0
6	11	14	63	1	58	1	82

Confusion matrix

The confusion matrix shows that our model works very well with most of the skin cancer classes.
In particular it is good for categories 0,1,2,3,5 while it has room for improvements with classes 4 and 6.

08

RESULTS

Both classifiers were succesfully developed and fitted with image data.

CNN
RESULTS

Total parameters: 504 103
loss: 0.0024
accuracy: 0.997

LSTM
RESULTS

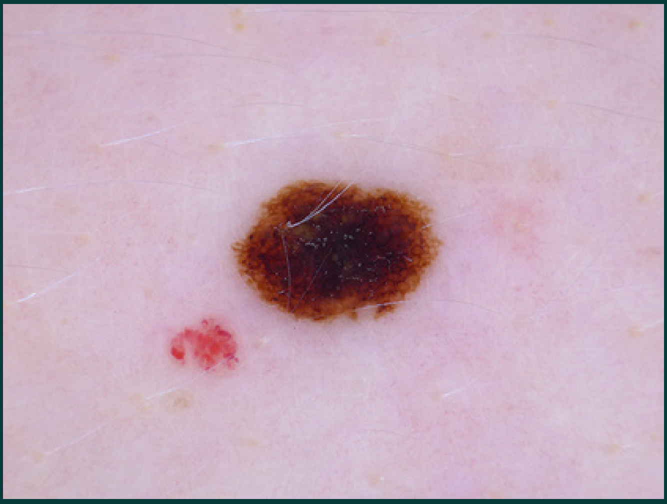
Total parameters: 71 623
loss: 0.0240
accuracy: 0.9926

It was interesting to investigate in depth the different approaches between CNN and LSTM.

We can conclude that even though the CNN reaches a slightly higher accuracy the LSTM does the job with less than 1/7 of the parameters of the CNN!

09 FINAL TEST

Finally we put the picture of Federica's malignant mole to the test.
This is the image:



Our CNN model in less than a second returned ('mel', ' melanoma').
It is the correct diagnosis.

Also our LSTM model diagnosticated correctly Federica's mole!
It took just one second to get the diagnosis. While Federica waited months to see her dermatologist for the annual visit to discover she had a melanoma, our trained model could have helped her discover it in seconds. Deep learning has a huge potential in this field!

We can conclude that we managed to build a system able to detect melanoma and which is really easy to use. We are fascinated by how much we were able to achieve starting from a single image.

We look forward to expand this project to provide for future users a free and easy access to early detection.