



# Project : IA on the Cloud

Detect skin lesion through Android Application

# Team



Olivier Randavel

---

**Data Scientist**

---

@ olivier.randavel@gmail.com



Hippolyte Mayard

---

**Data Scientist**

---

@ hippolyte.mayard@dauphine.eu

# Skanner App



# Skanner App



Take a picture of your skin and get an advice  
regarding your lesion

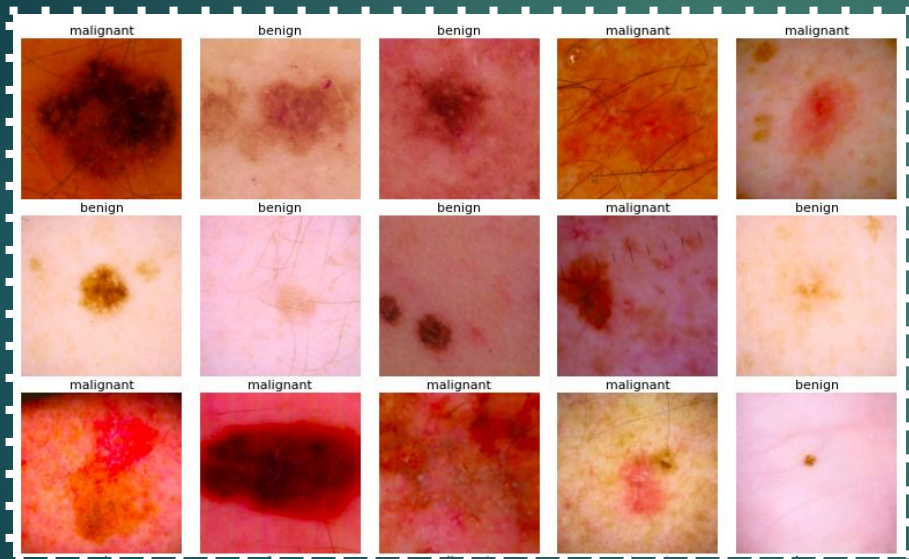


# Model & Architecture



# The data

## ArchitecturSIIM-ISIC Melanoma Classification Kaggle competition



- Over 34 000 skin lesion images
- Highly Imbalanced dataset:
  - 32610 benign lesions
  - 585 malignant lesions

# Data pre-processing

## TensorFlow input pipeline

- Oversampling
  - Data augmentation
- Solution for the imbalanced dataset

Image rotation



Image Zooming



Horizontal flip

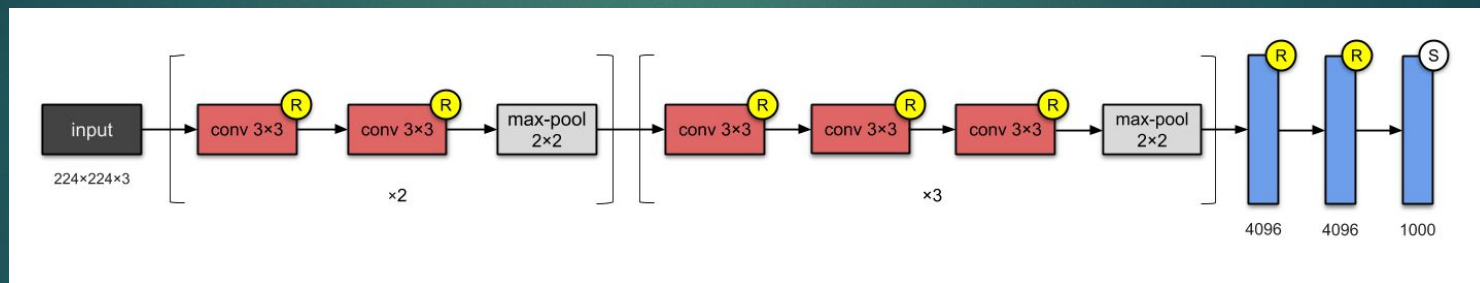


Image Brightness



# Model Architecture

## Transfer Learning



VGG16 architecture



# Model Evaluation

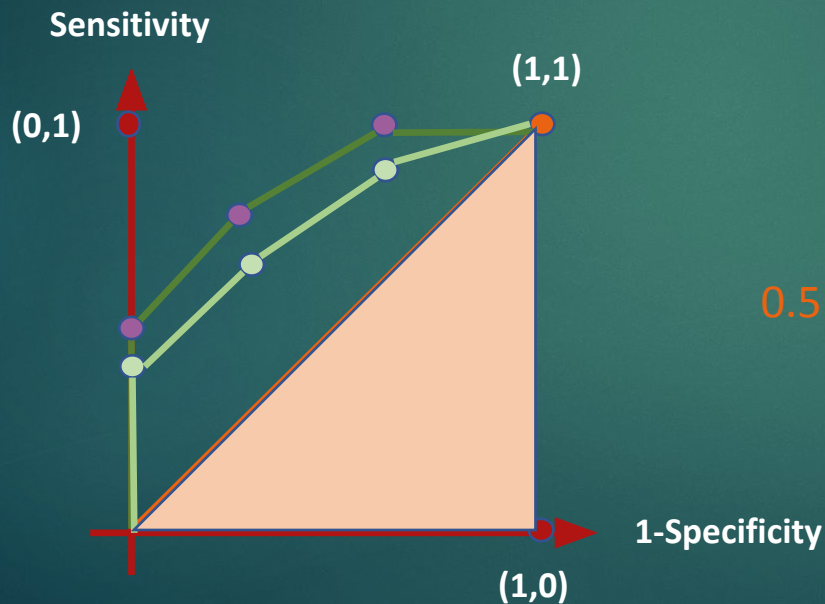
## Confusion Matrix

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
		<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$

# Model Evaluation

## ROC & AUC metrics

**AUC or Area Under the Curve**, is a value between 0 and 1 that computes the area under the ROC curve. It is usually higher than 0.5 for non-random models. The higher the value the better is the model overall.

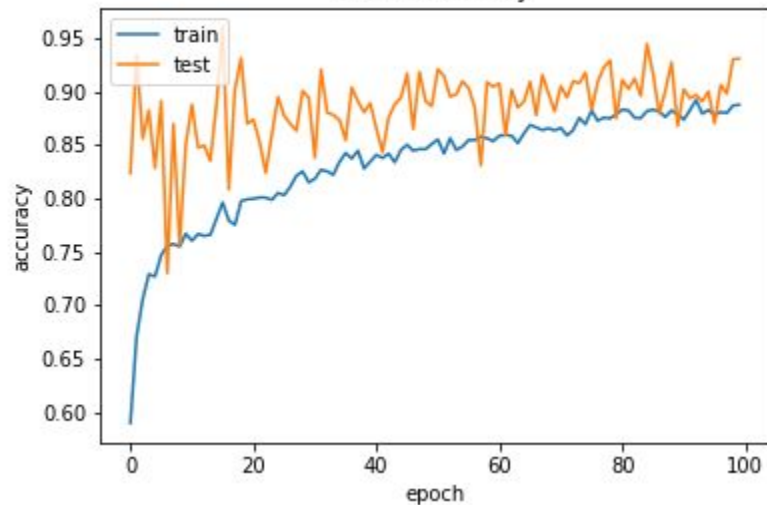


0.5 represents the displayed area.

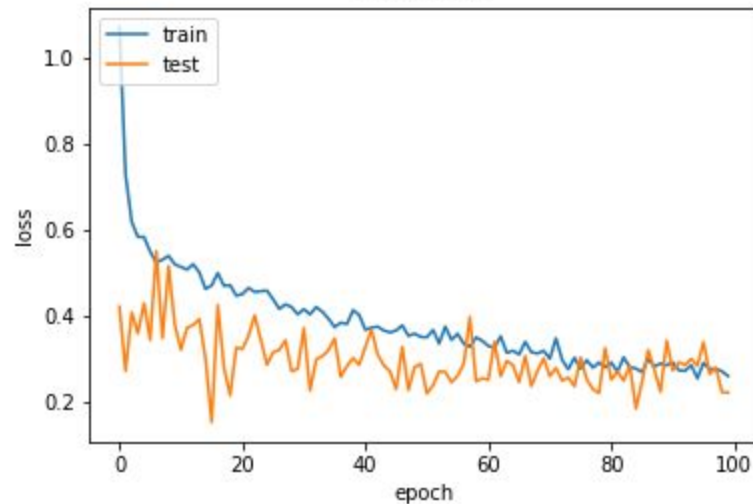
# Model Evaluation

## Accuracy & Loss

model accuracy

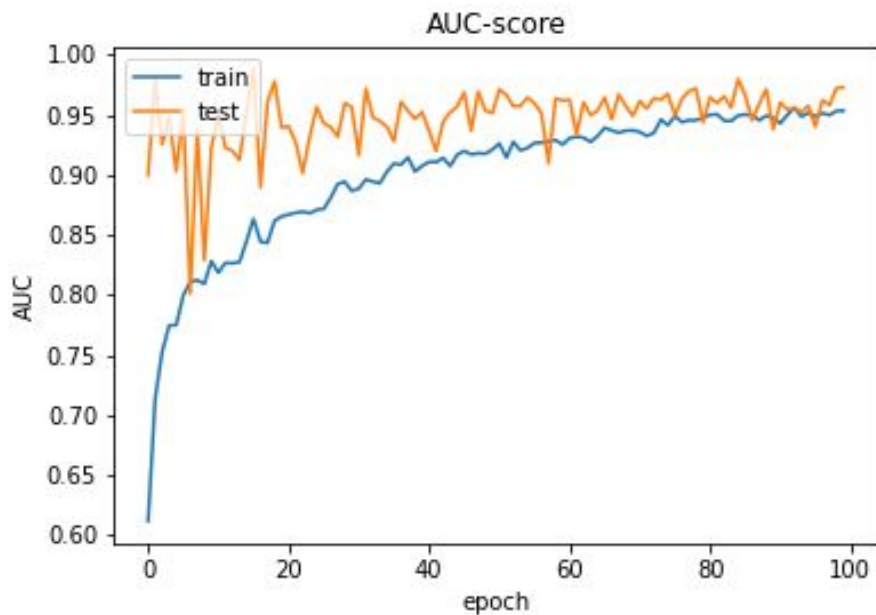


model loss



# Model Evaluation

## AUC metric



# Summary

## Our Solution :

- ❖ Technology used
- ❖ Test the code locally
- ❖ EC2
- ❖ FileZilla
- ❖ Android Studio
- ❖ Demonstration
- ❖ Limits of the solution



## Possible Improvements (1) :

- ❖ Technology used
- ❖ Lambda
- ❖ Lambda Settings
- ❖ Lambda demo
- ❖ Demonstration

## Possible Improvements (2) :

- ❖ Technology used
- ❖ Cloud Formation
- ❖ Lambda
- ❖ Demonstration







# Our Solution

# Technology used

## Architecture



Python



Colab



Amazon  
EC2



FileZilla



Android  
Studio

## Libraries



TensorFlow



Flask



Scikit-image



# EC2

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

skanner2

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances



Amazon  
EC2



```
C:\Users\OlivierRANDAVEL\Downloads>ssh -i "skanner2.pem" ubuntu@ec2-100-25-33-156.compute-1.amazonaws.com
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1023-aws x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

```
System information as of Sun Jul  5 11:06:36 UTC 2020
```

```
System load:  0.0      Processes:      91
Usage of /:   48.7% of 7.69GB   Users logged in:  0
Memory usage: 18%      IP address for eth0: 172.31.59.186
Swap usage:   0%
```

```
20 packages can be updated.
13 updates are security updates.
```

```
Last login: Sun Jul  5 07:12:59 2020 from 185.228.230.179
ubuntu@ip-172-31-59-186:~$
```

Create a VM :

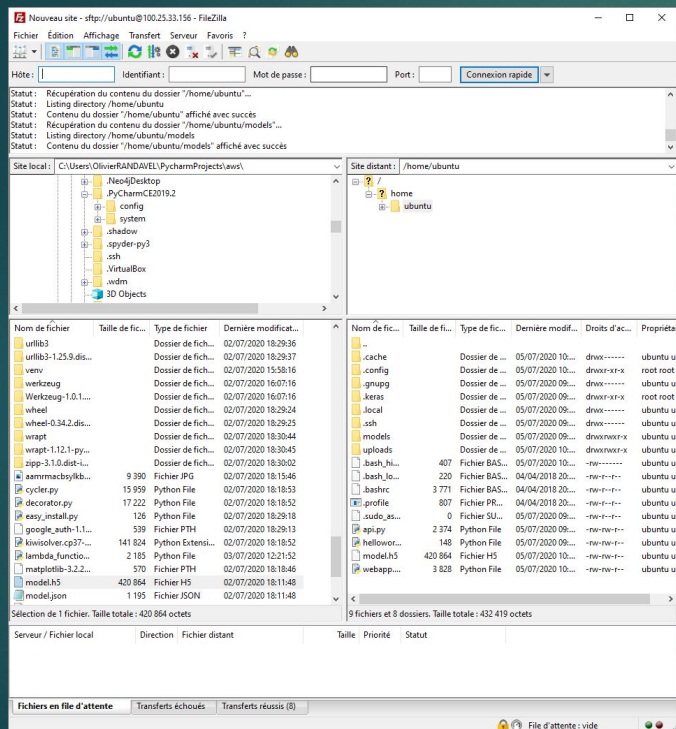
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type  
Python3, Pip3

Download the key pair model to access to the EC2

Install all libraries



# FileZilla



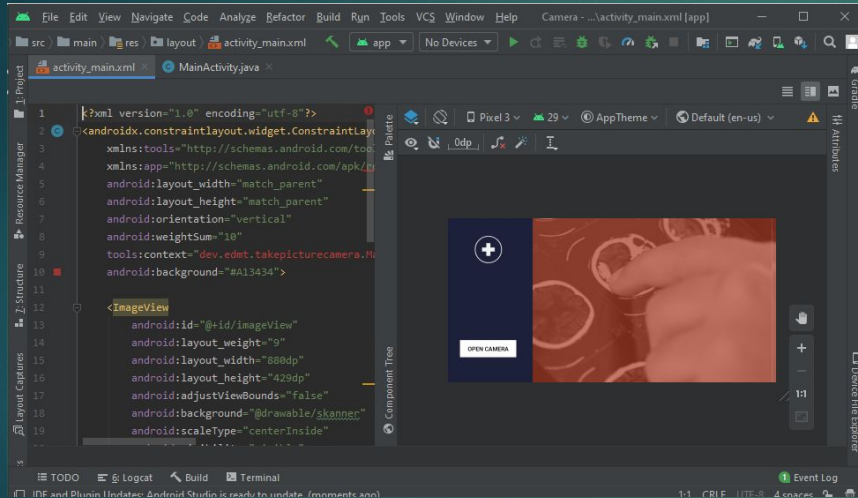
FileZilla

Send file from Windows to EC2 (VM)

Use ssh key with the pem file downloaded previously



# Android Studio

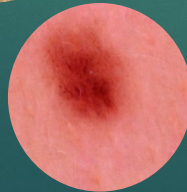
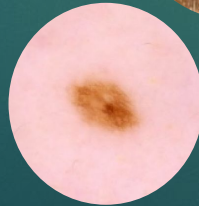
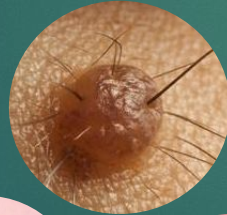
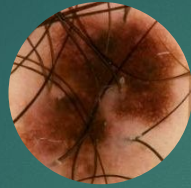


Implement the app using javascript

Adding a button to take picture

Make a call API to the EC2

# Demonstration



Visit the website and follow the instruction

You will upload a picture and get the prediction  
regarding the lesion

# Limits of the solution



The VM need to be on in order to use the app,  
this imply huge cost.

Besides the EC2 is not scalable in fact multiple  
calls at the same time will consume more RAM.  
The VM could run out of memory





# Possible Improvements (1)

# Technology used

## Architecture



AWS Lambda



Python



API Gateway



Cloud Watch



Colab



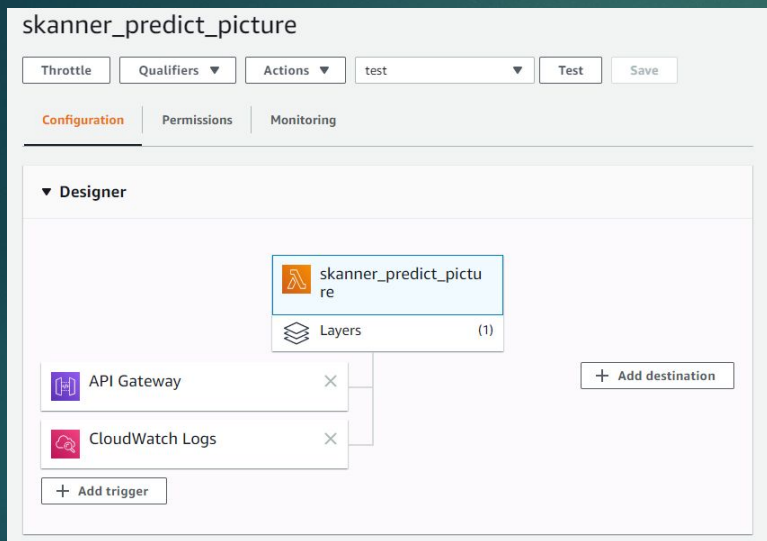
Android Studio

## Libraries

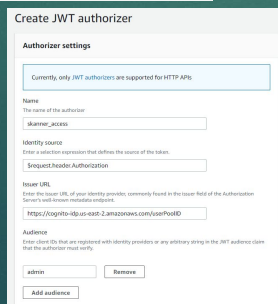
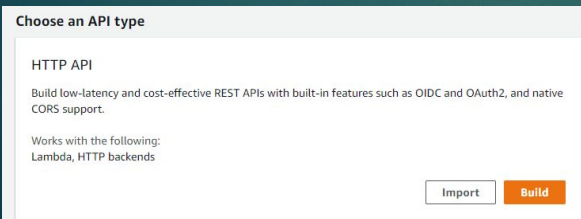




# Lambda



AWS Lambda

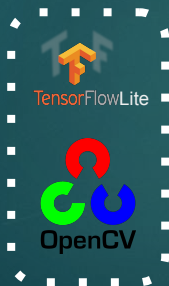


- Use a lambda function and launch it using an API
- Use logs to get information regarding execution

# Lambda : settings



Amazon  
**EC2**



The code needs to be compatible with  
Amazon-Ubuntu

The code including all packages need to be lower  
than 250 Mo



AWS Lambda

# Lambda : Demo

skanner\_predict\_picture

Throttle Qualifiers ▼ Actions ▼ test ▼ Test Save

✓ Execution result: succeeded ([logs](#)) ✕

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

"rc"



AWS Lambda

CloudWatch > 2020/07/05/[\$LATEST]18201302bbc54769aebcb880e7ece8d6 Switch to the orig

Log events ↺ Actions ▼ Create Metric Filter

Filter events Clear 1m 30m 1h 12h custom ⚙

Timestamp	Message
There are older events to load. <a href="#">Load more.</a>	
2020-07-05T13:45:32.199+02:00	START RequestId: b74b9881-d5f6-4a5b-a324-1922c32d11
2020-07-05T13:45:32.199+02:00	rc
2020-07-05T13:45:32.200+02:00	END RequestId: b74b9881-d5f6-4a5b-a324-1922c32d11cb
2020-07-05T13:45:32.200+02:00	REPORT RequestId: b74b9881-d5f6-4a5b-a324-1922c32d1

No newer events at this moment. Auto retry paused. 5



Cloud Watch

# Demonstration



The function receives the call and return "rc"



# Possible Improvements (2)



# Technology used

## Architecture



AWS Lambda



Python



API Gateway



S3



CloudFormation



Android Studio

## Libraries



# Cloud Formation

deploy-ml-api

Stack info | **Events** | Resources | Outputs | Parameters | Template | Change sets

Events (81)

Q Search events

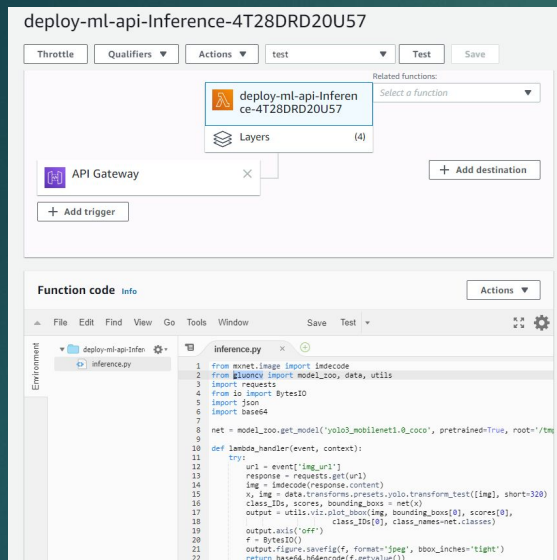
Timestamp	Logical ID	Status
2020-07-04 20:42:05 UTC+0200	deploy-ml-api	CREATE_COMPLETE
2020-07-04 20:42:02 UTC+0200	s3Operations	CREATE_COMPLETE
2020-07-04 20:42:02 UTC+0200	IntegrationAPI	CREATE_COMPLETE
2020-07-04 20:42:02 UTC+0200	s3Operations	CREATE_IN_PROGRESS
2020-07-04 20:41:58 UTC+0200	s3Operations	CREATE_IN_PROGRESS
2020-07-04 20:41:57 UTC+0200	InferenceAPIProductionStage	CREATE_COMPLETE
2020-07-04 20:41:56 UTC+0200	InferenceAPIProductionStage	CREATE_IN_PROGRESS
2020-07-04 20:41:55 UTC+0200	s3Lambda	CREATE_COMPLETE
2020-07-04 20:41:55 UTC+0200	s3Lambda	CREATE_IN_PROGRESS
2020-07-04 20:41:55 UTC+0200	InferenceAPIProductionStage	CREATE_IN_PROGRESS
2020-07-04 20:41:53 UTC+0200	AppDistribution	CREATE_COMPLETE
2020-07-04 20:41:53 UTC+0200	InferenceAPIDeployment799996c84	CREATE_COMPLETE



CloudFormation

We specify the template of the html page, the code of the lambda function, and all libraries in the form of layers stored in the S3

# Lambda



AWS Lambda



Use a lambda function and launch it using an API

Use logs to get information regarding execution

Layers Info				Add a layer	Edit
Merge order	Name	Layer version	Version ARN		
1	MXNet	1	arn:aws:lambda:us-east-1:222789047898:layer:MXNet:1		
2	SciPy	1	arn:aws:lambda:us-east-1:222789047898:layer:SciPy:1		
3	Pillow	1	arn:aws:lambda:us-east-1:222789047898:layer:Pillow:1		
4	GluonCV	1	arn:aws:lambda:us-east-1:222789047898:layer:GluonCV:1		

# Demonstration



The model recognize objects among 100 items  
(person, plane, bag..)

It helps to segment item and give the prediction  
accuracy

You can use picture example to test the model



Merci de votre attention