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

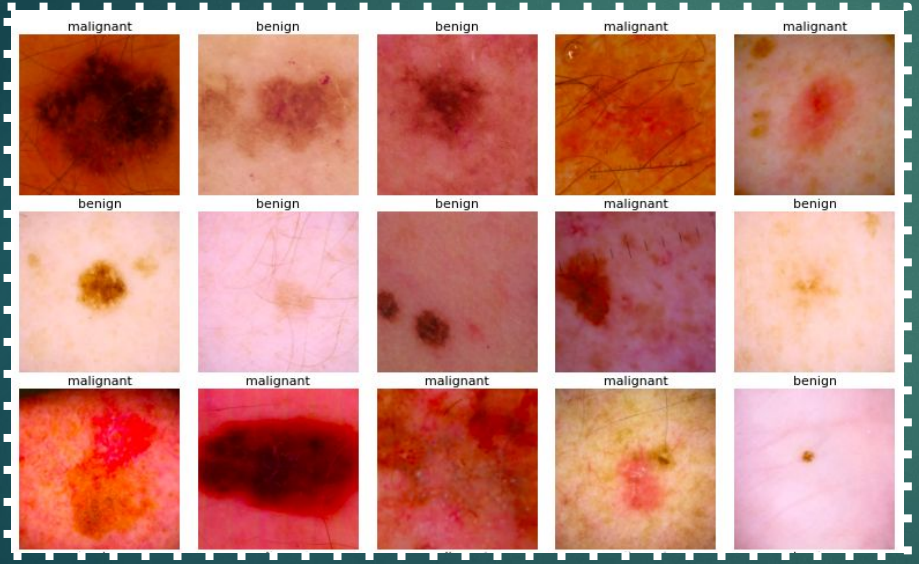
Skanner App

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

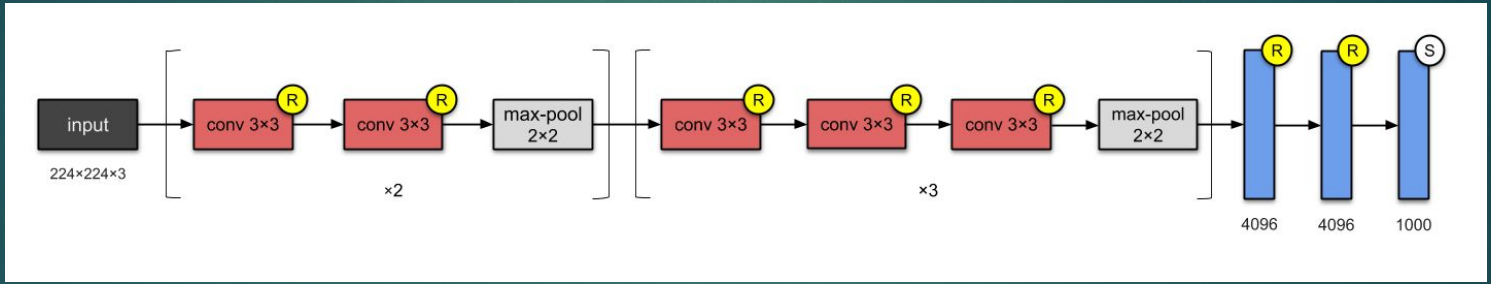

Image Zooming

Horizontal flip


Image Brightness

Model Architecture

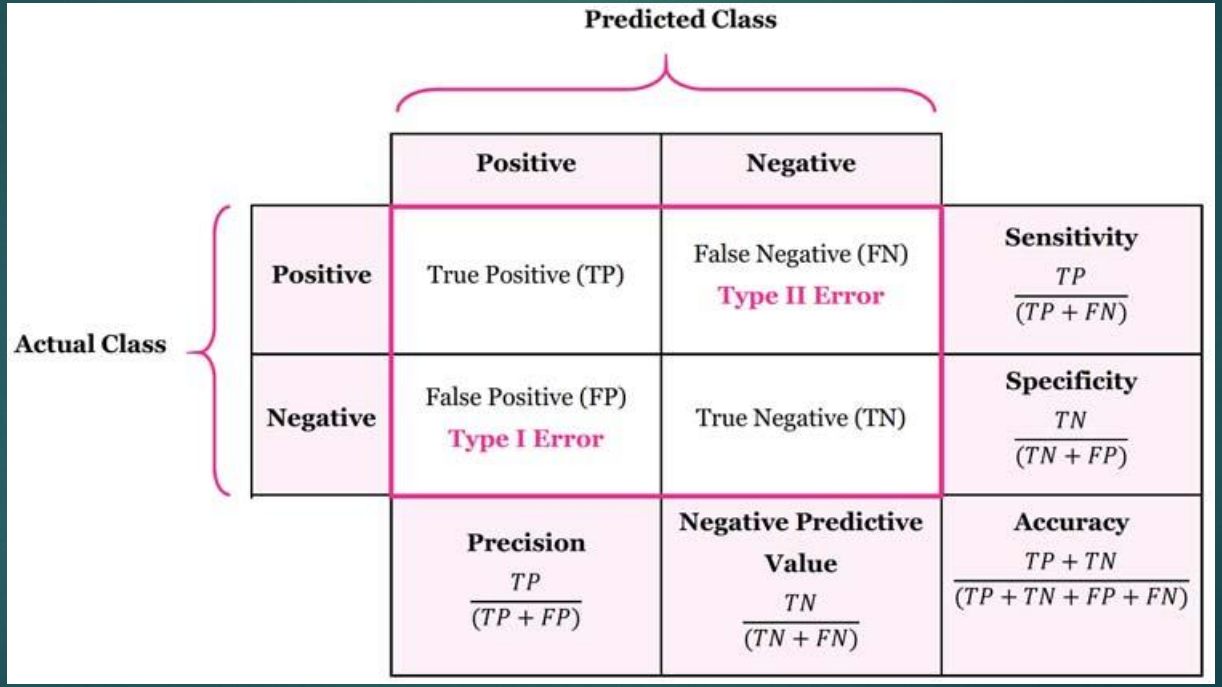
Transfer Learning



VGG16 architecture

Model Evaluation

Confusion Matrix

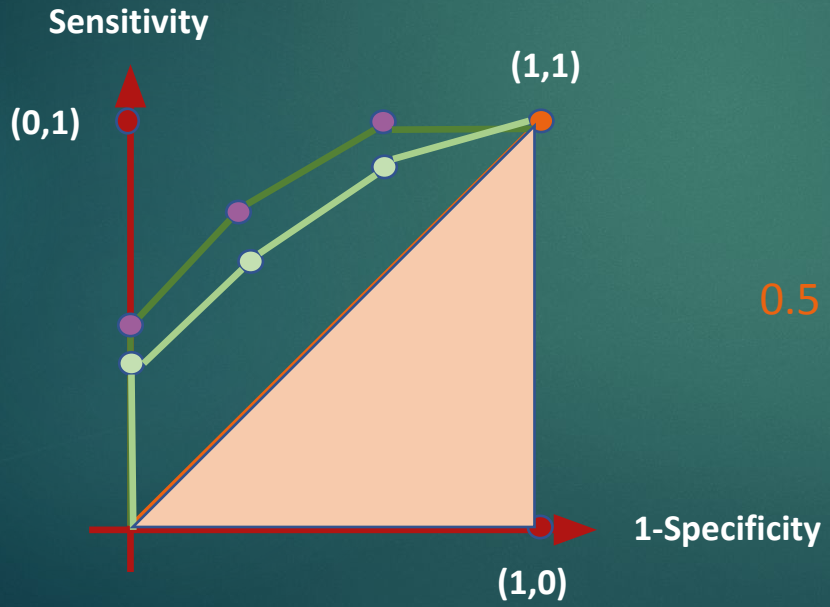


		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$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

Model Evaluation

ROC & AUC metrics

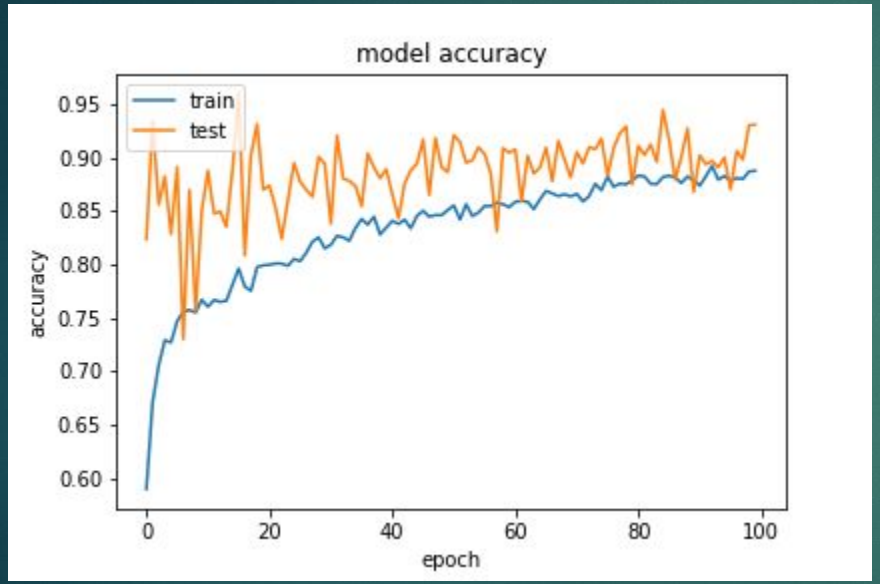
AUC or Area Under the Curve, is a value between 0 and 1 that computes the area under the ROC curve. It is usually higher than 0.5 for non-random models. The higher the value the better is the model overall.

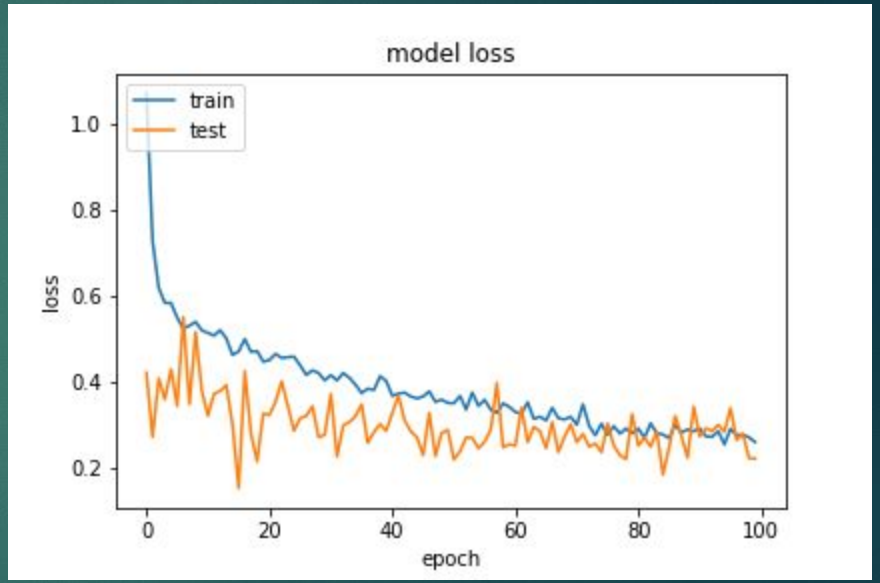


0.5 represents the displayed area.

Model Evaluation

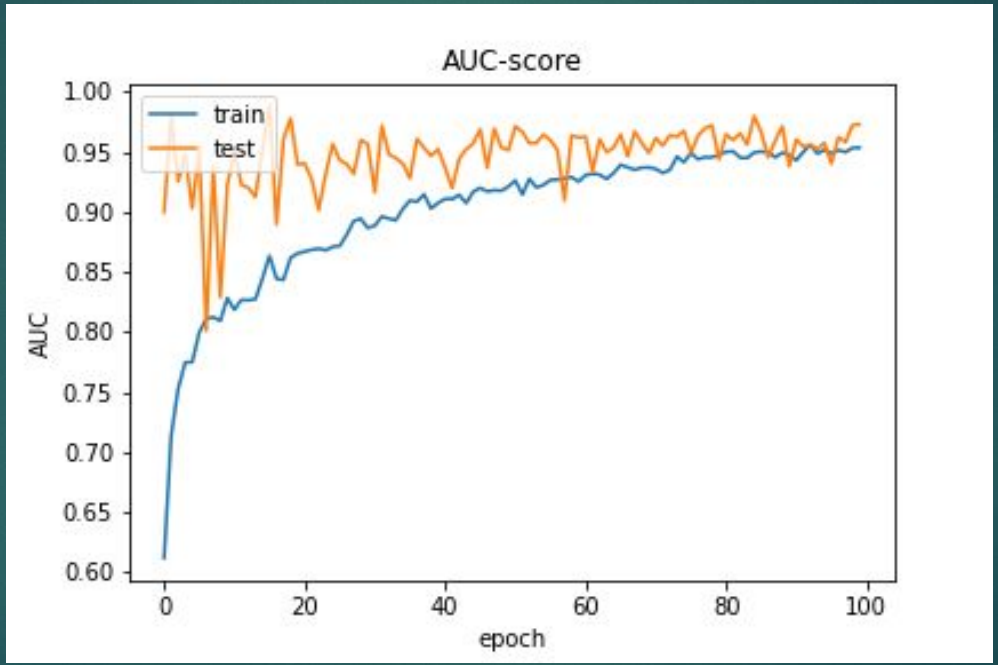
Accuracy & Loss





Model 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

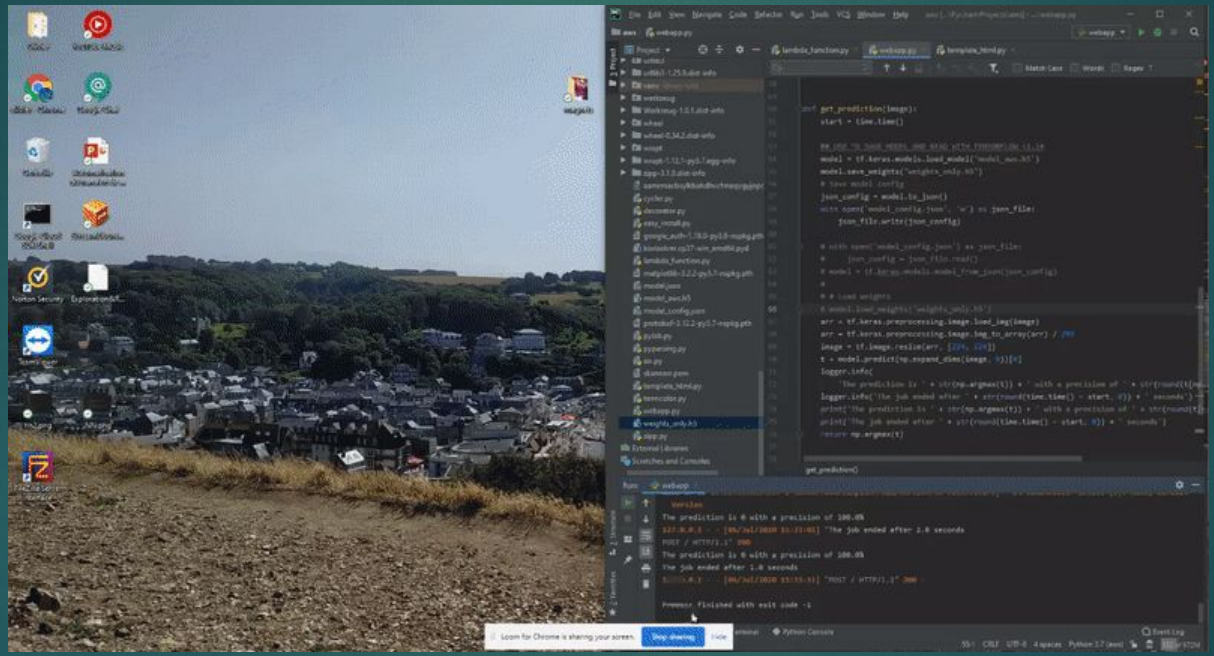
Libraries

TensorFlow

Flask

Scikit-picture

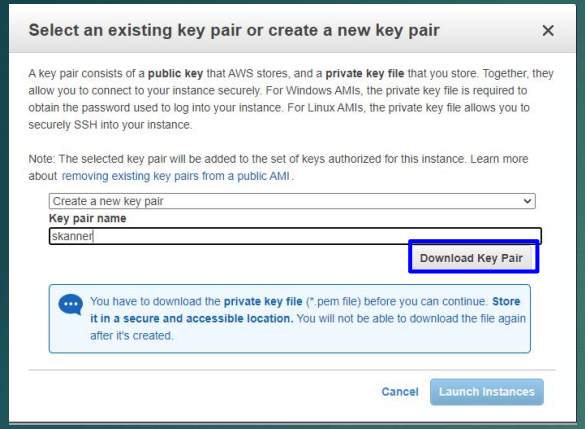
Test the code locally

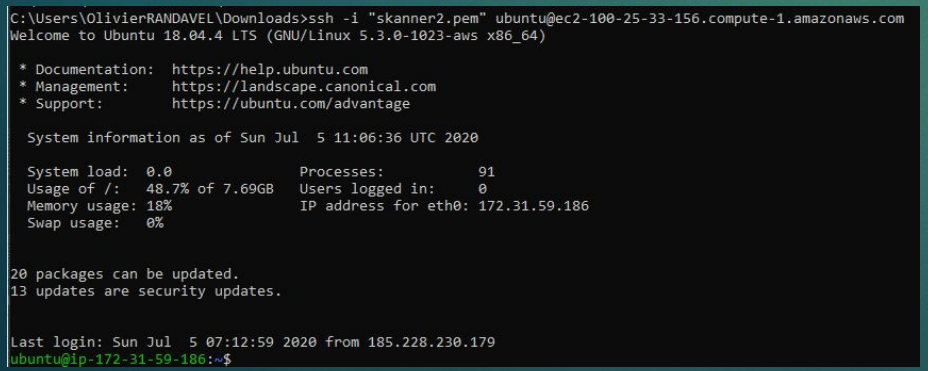


Python

PyCharm

EC2





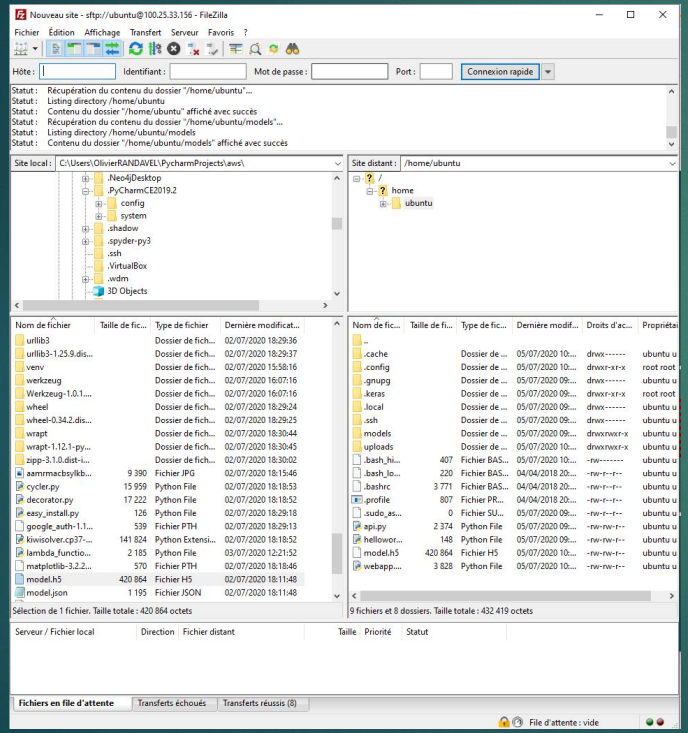
Create a VM :

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

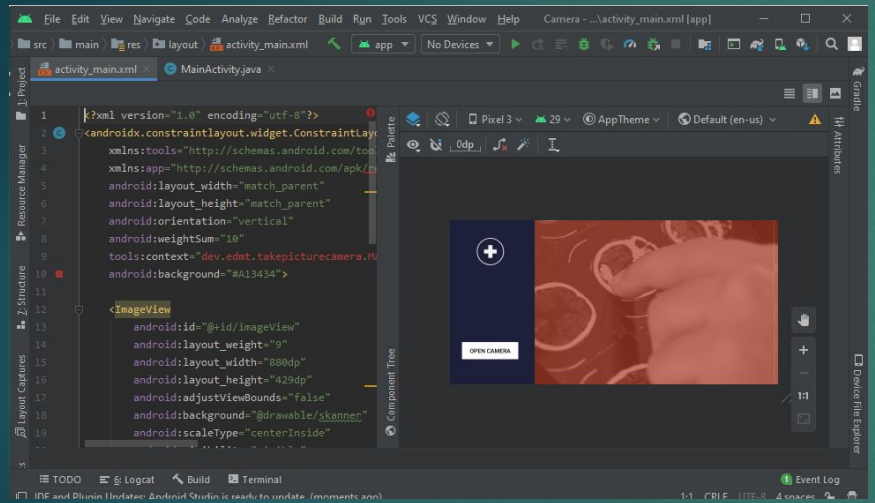


FileZilla

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

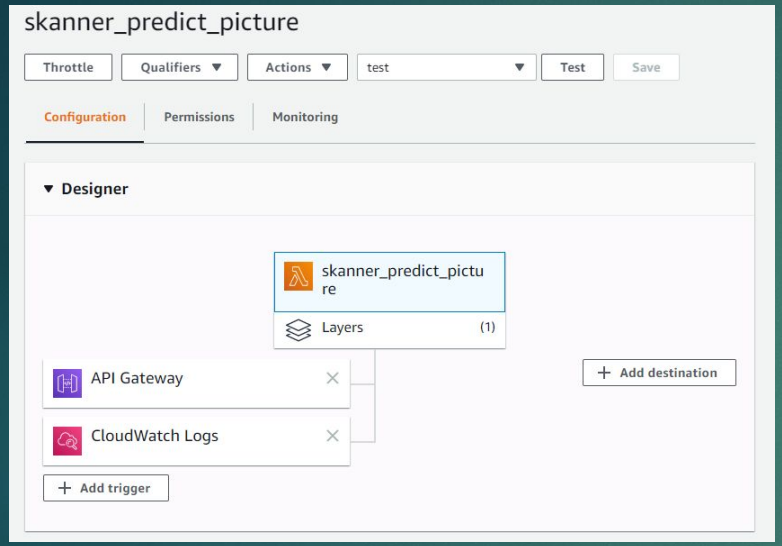
Android Studio

Libraries

TensorFlow Lite

OpenCV

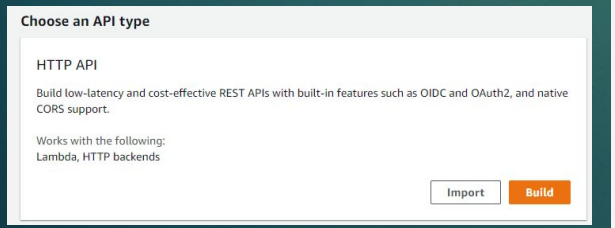
Lambda

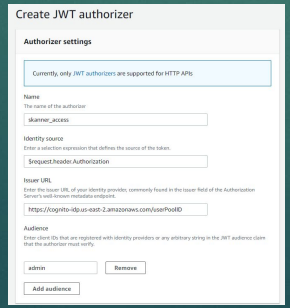


AWS Lambda

Use a lambda function and launch it using an API

Use logs to get information regarding execution





Lambda : settings

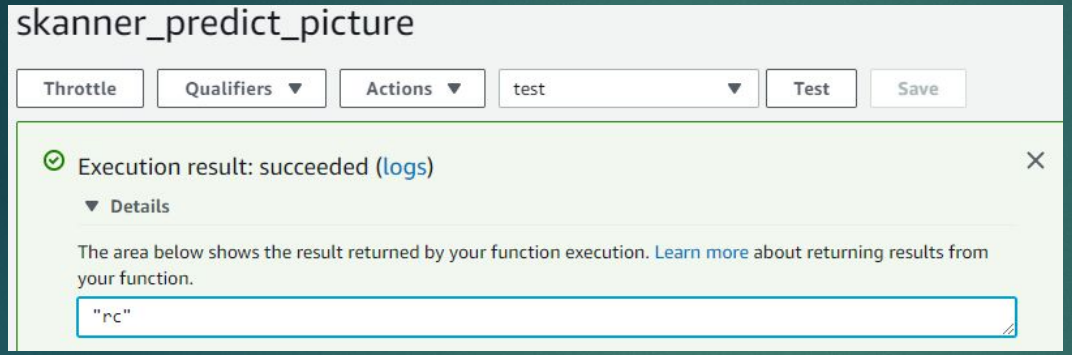
Amazon
EC2

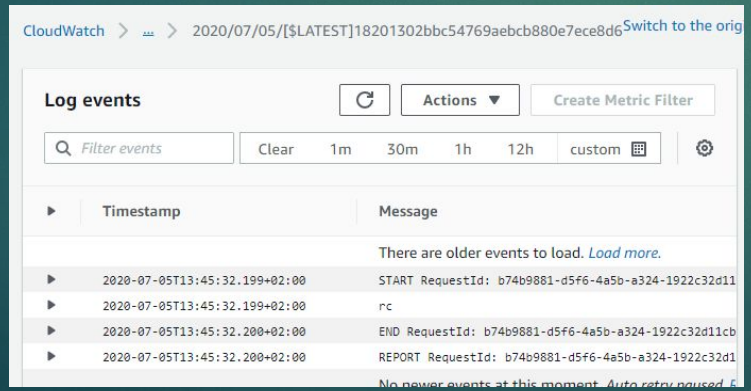
The code needs to be compatible with Amazon-Ubuntu

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

AWS Lambda

Lambda : Demo





AWS Lambda

Cloud Watch

Demonstration

The function receives the call and return “rc”

Possible Improvements (2)

Technology used

Architecture

AWS Lambda

Python

API Gateway

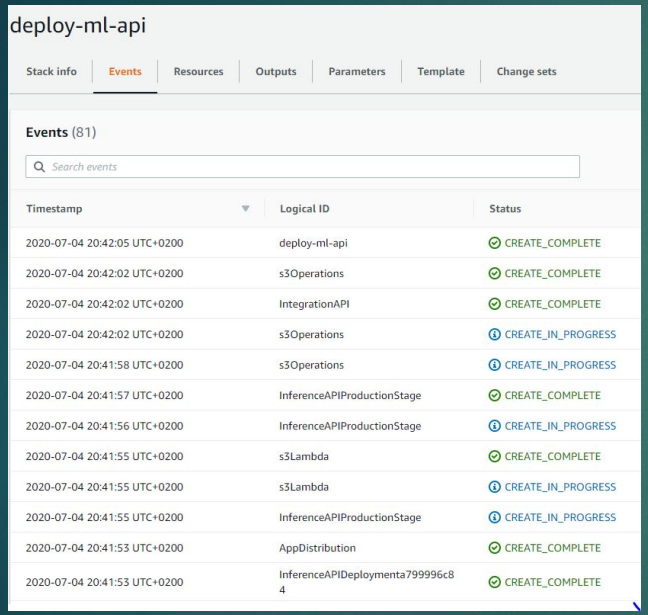
S3

CloudFormation

Android Studio

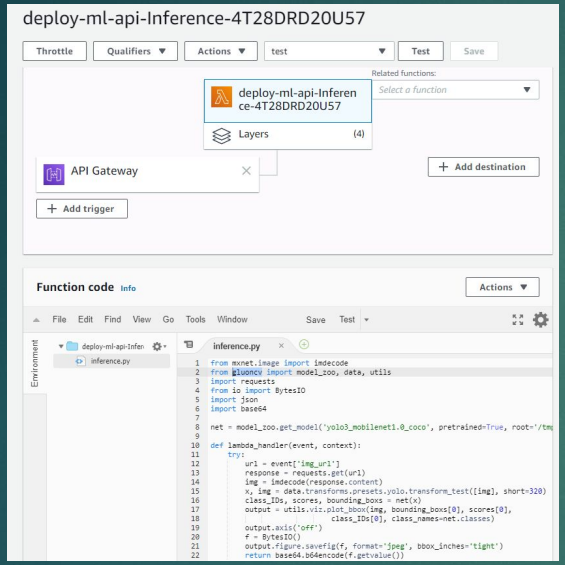
Libraries

Cloud Formation



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