**LAB REPORT**

1. Adafruit IO Basics: Feeds Feeds: chứa metadata về dữ liệu bạn đẩy tới Adafbean IO và cũng chứa các giá trị dữ liệu cảm biến được đẩy tới server từ thiết bị cảm biên, hãy tạo, chỉnh sửa và xóa các feeds:

− Nhiệt độ: Nhận dữ liệu từ cảm biến nhiệt độ

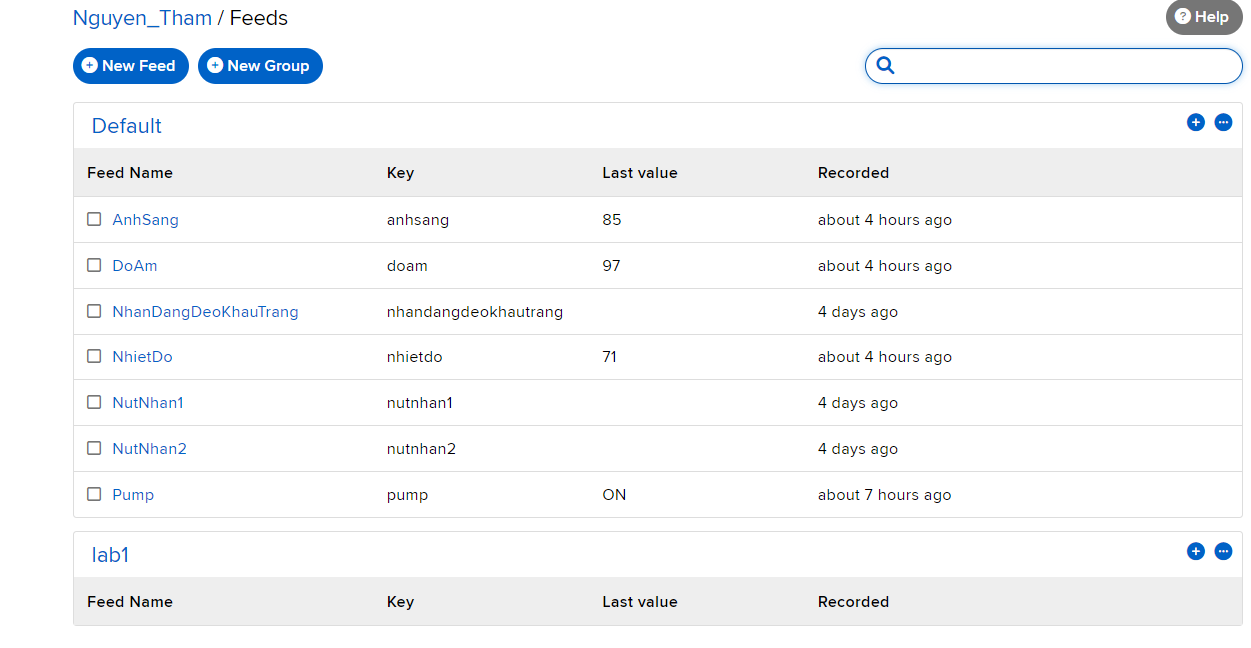
− Độ ẩm: Nhận dữ liệu từ cảm biến độ ẩm

− Ánh sáng: Nhận dữ liệu từ cảm ánh sáng

− Nút nhấn 1: Dùng để điều khiên thiết bị đèn

− Nút nhấn 2: Dùng để điều khiên thiết bị máy bơm

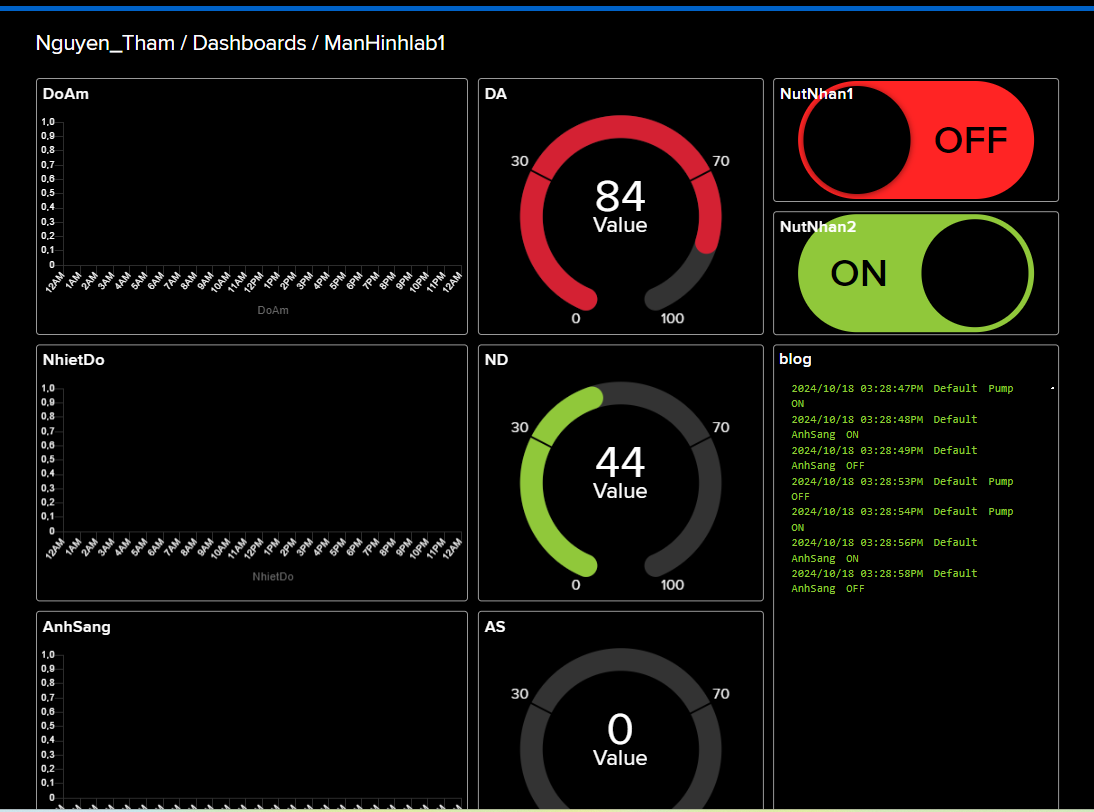
− Nhận dạng đeo khẩu trang: Nhận dữ liệu là thông tin một người người đeo khẩu trang, không đeo khẩu trang hay không có người bằng thuật toán học máy qua một camera quan sát



2. Adafruit IO Basics: Dashboards Dashboards cho phép bạn trực quan hóa dữ liệu và kiểm soát các dự án được kết nối Adafruit IO từ bất kỳ trình duyệt web- Thêm các khối vào bảng điều khiển mới để hiển thị trực quan thông tin.

Nút nhấn 1: để điều khiển thiết bị đèn

Nút nhấn 2: để điều khiển thiết máy bơm Các khối khác mà nó phù hợp cho các thiết bị cảm biến.- Chọn các khối phù hợp với các feeds: một số khối có thể được sử dụng làm đầu ra và một số khối có thể được sử dụng



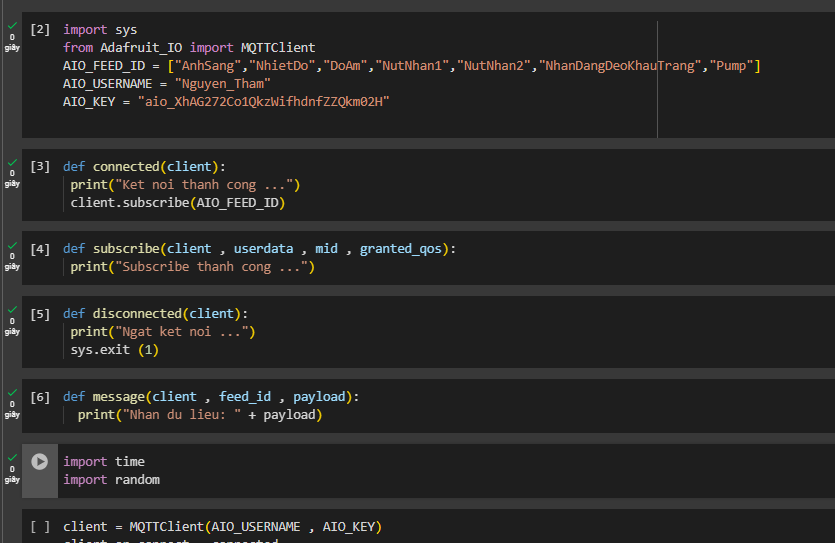
**3. Thực nghiệm**

**- Dữ liệu:**

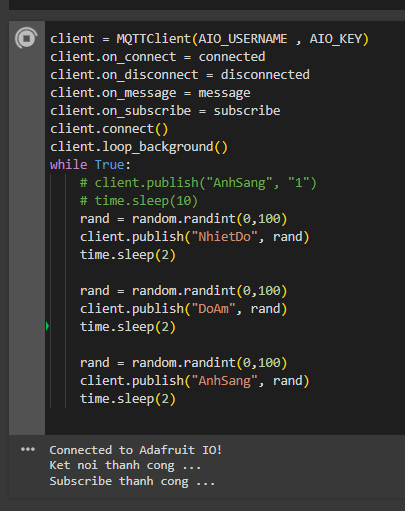
**Thu thập hoặc tạo ra dữ liệu mẫu về người đeo khẩu trang, không đeo khẩu trang, không**

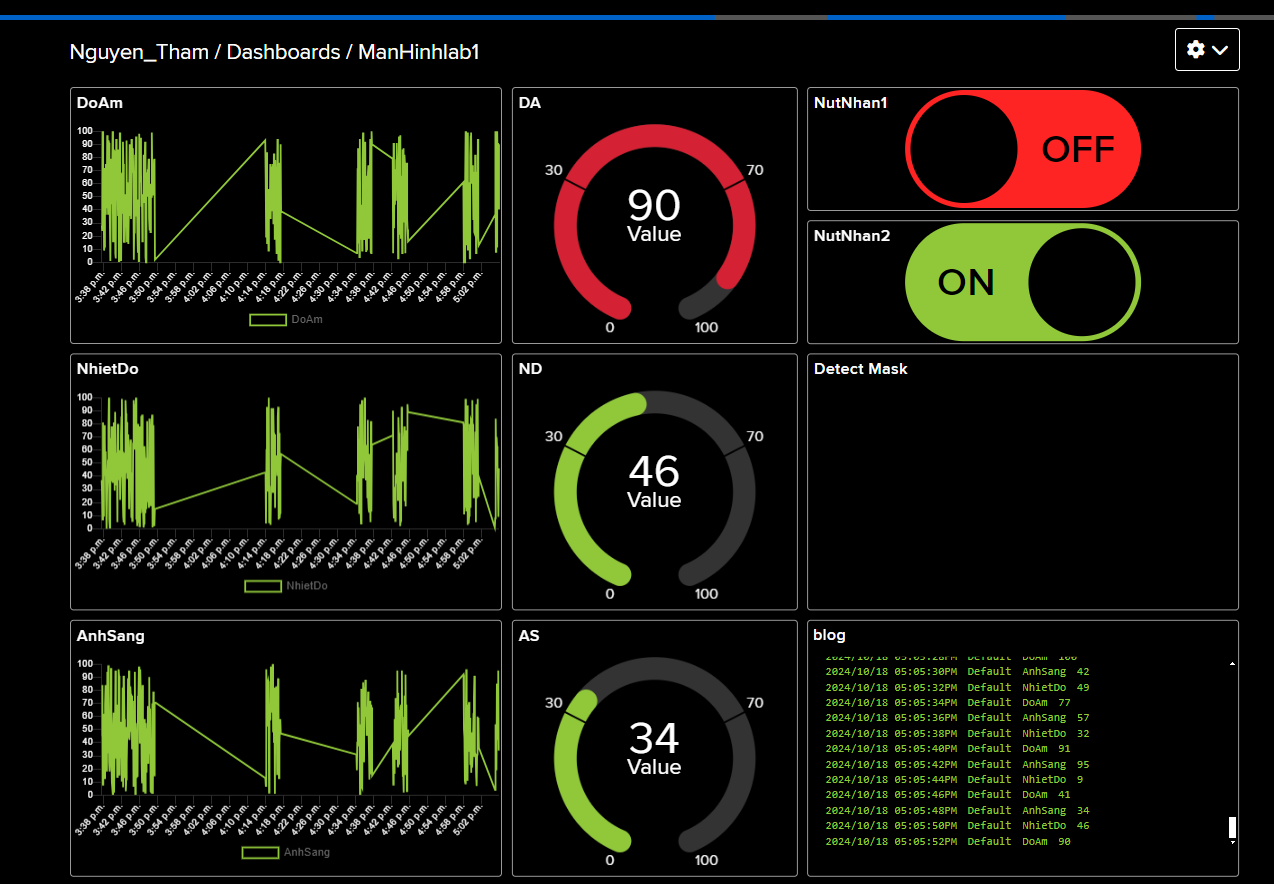
**có người. Sinh ngẫu nhiên dữ liệu trên các khoảng giá trị để mô phỏng cho dữ liệu của các thiết bị**

**cảm biến**



**Tạo biến random để sinh ngẫu nhiêu dữ liệu truyền lên cloud**

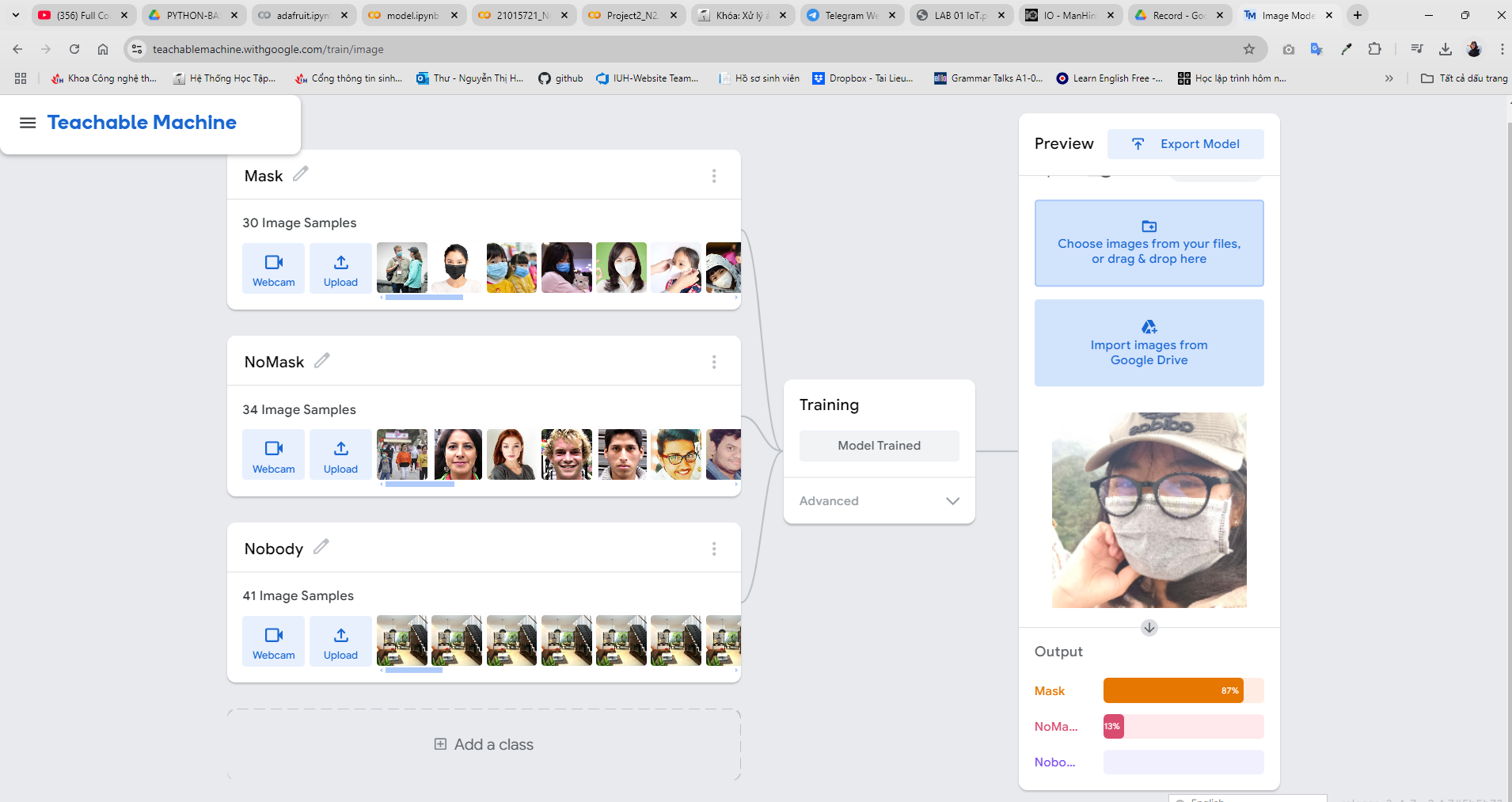




4. Đánh giá mô hình nhận dạng.

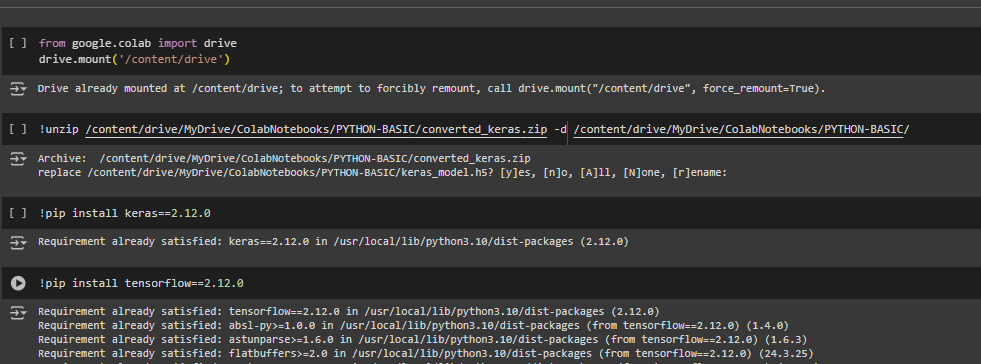
Mô tả (các) tập dữ liệu bạn đang sử dụng (cung cấp tài liệu tham khảo). Nếu nó vẫn chưa rõ ràng, hãy đảm bảo rằng nhiệm vụ liên quan được mô tả rõ ràng. Thí dụ kích thước tập dữ liệu, bao nhiêu dữ liệu đeo khẩu trang, bao nhiêu dữ liệu không đeo khẩu trang và bao nhiêu là không có người..

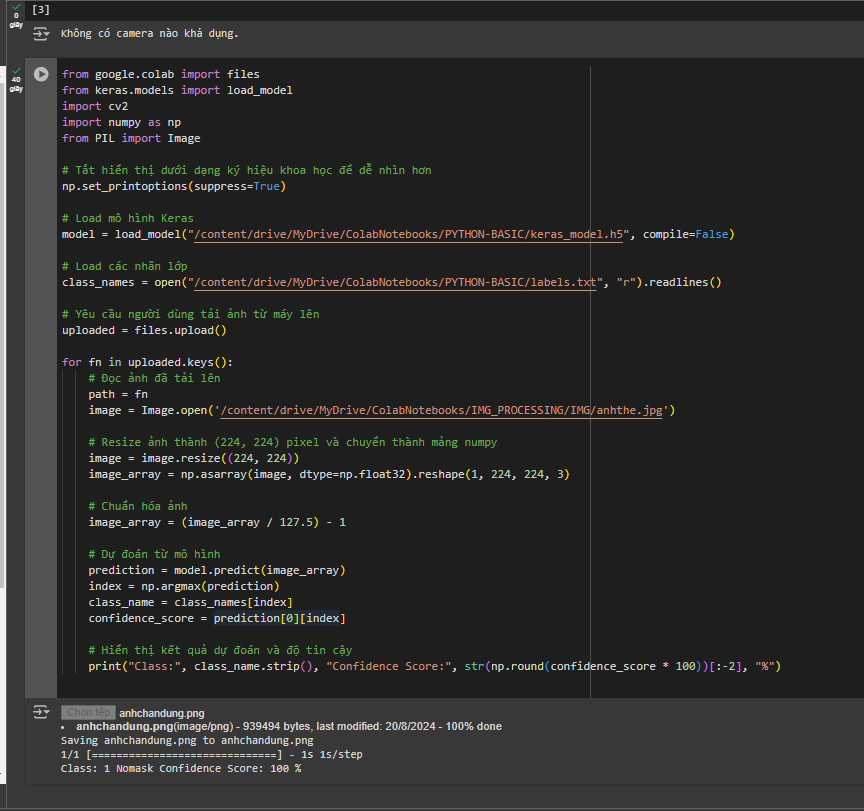
**Train model**

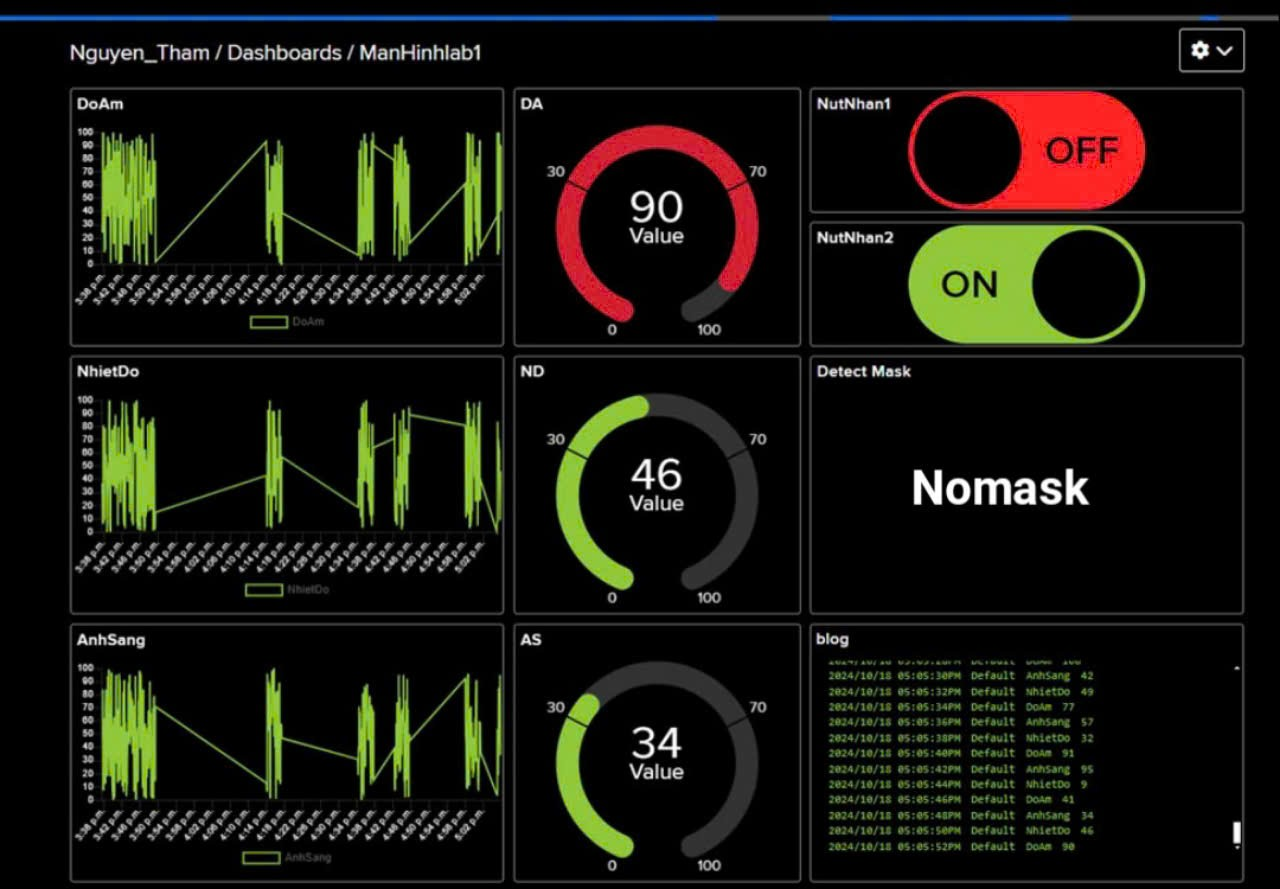


**Export model**

**Tải lên gg colab**

 **Dô google coleb không hỗ trợ camera cho máy tính nên tải file lên**



****

from google.colab import files

from keras.models import load\_model

import cv2

import numpy as np

from PIL import Image

# Tắt hiển thị dưới dạng ký hiệu khoa học để dễ nhìn hơn

np.set\_printoptions(suppress=True)

# Load mô hình Keras

model = load\_model("/content/drive/MyDrive/ColabNotebooks/PYTHON-BASIC/keras\_model.h5", compile=False)

# Load các nhãn lớp

class\_names = open("/content/drive/MyDrive/ColabNotebooks/PYTHON-BASIC/labels.txt", "r").readlines()

# Yêu cầu người dùng tải ảnh từ máy lên

uploaded = files.upload()

for fn in uploaded.keys():

    # Đọc ảnh đã tải lên

    path = fn

    image = Image.open('/content/drive/MyDrive/ColabNotebooks/IMG\_PROCESSING/IMG/anhthe.jpg')

    # Resize ảnh thành (224, 224) pixel và chuyển thành mảng numpy

    image = image.resize((224, 224))

    image\_array = np.asarray(image, dtype=np.float32).reshape(1, 224, 224, 3)

    # Chuẩn hóa ảnh

    image\_array = (image\_array / 127.5) - 1

    # Dự đoán từ mô hình

    prediction = model.predict(image\_array)

    index = np.argmax(prediction)

    class\_name = class\_names[index]

    confidence\_score = prediction[0][index]

    # Hiển thị kết quả dự đoán và độ tin cậy

    print("Class:", class\_name.strip(), "Confidence Score:", str(np.round(confidence\_score \* 100))[:-2], "%")

link chạy model

<https://colab.research.google.com/drive/1D9AU8e90Q3oxVB--nh6-Ilo1rqh988Fz?usp=sharing>

link code thêm data vào dashbroad

<https://colab.research.google.com/drive/1NYF48b_eHuctO1wQYsFEiEI6tM3sqfIr?usp=sharing>