HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING


# Project 3

## Topic : Cigarette-Smoking-Detection

### Lecturer : Huỳnh Xuân Phụng

1. **Nguyễn Đình Thiên Phước 17110062**
2. **Mai Nguyễn Anh Thư 17110080**

Ho Chi Minh City, 1/2021

# Content

# Score

| Criteria | Content | Presentation | Overall |
|---|---|---|---|
| Score | | | |

Observations of Instructor

.....
.....
.....
.....

Evaluation:

Nguyễn Đình Thiên Phước :.....

Mai Nguyễn Anh Thư :.....

Lecturer

Huynh Xuan Phung

# 1. OVERVIEW

## 1.1 Feature and necessity of identity problems

### 1.1.1 Feature

- Identification is the process of classifying the objects represented by a certain model and assigning them to a class (assigning the object a name) based on the rules and standard patterns. The process of identifying based on known learning patterns is called supervised learning; in the opposite case, it is called non- supervised learning.

- Identification is an important problem in machine learning subject.

### 1.1.2 Necessary

- Along with the ongoing development of socio-economic and technical branches. Requires precise management and processing of information that is beyond human power. So we need machines or dynamics to reduce the load or replace heavy, precise, boring human work. Helping machines to identify (collect, classify information) as humans will help machines operate effectively like humans with much higher accuracy.

- Some applications of identity problems.

Fingerprint recognition

Voice recognition

License plate recognition

## 1.2 Model and feature of the identification process

### 1.2.1 Model

The choice of an identification process is closely related to the type of description that people use to specify the object. In identity, people are divided into two class families:

- They describe according to parameters.
- They describe according to the structure.

The description chosen determines the model of the object. Thus, they will have two types of models: the parametric model and the structural model.

### 1.2.2 Supervised learning

The technique of classifying through prior knowledge is called mastered learning. The basic feature of this technique is that one has a library of standard templates. The identifiable sample is compared with the standard sample to see which category it belongs to. For example, in a remote sensing image, one wants to distinguish a rice field, a forest, or a wasteland that has a description of that object. The main problem is to design a system to be able to match the

objects in the image with the standard template and decide to assign them to a class. The comparison is done through decision-making procedures based on a tool called a classifier or decision-making function.

### 1.2.3 Non supervised learning

This technique has to manually define the different classes and define the parameters that are specific to each class. Learning without a teacher is more difficult. On the one hand, because the number of layers is unknown, on the other hand, the class characteristics are also unknown. This technique is to do all possible grouping and choose the best. Starting from the data set, many different processing procedures to classify and upgrade gradually to achieve a classification plan. In general, an identity system can summarize the following diagram.

# 2. THEORY

## 2.1. The reason for choosing the topic

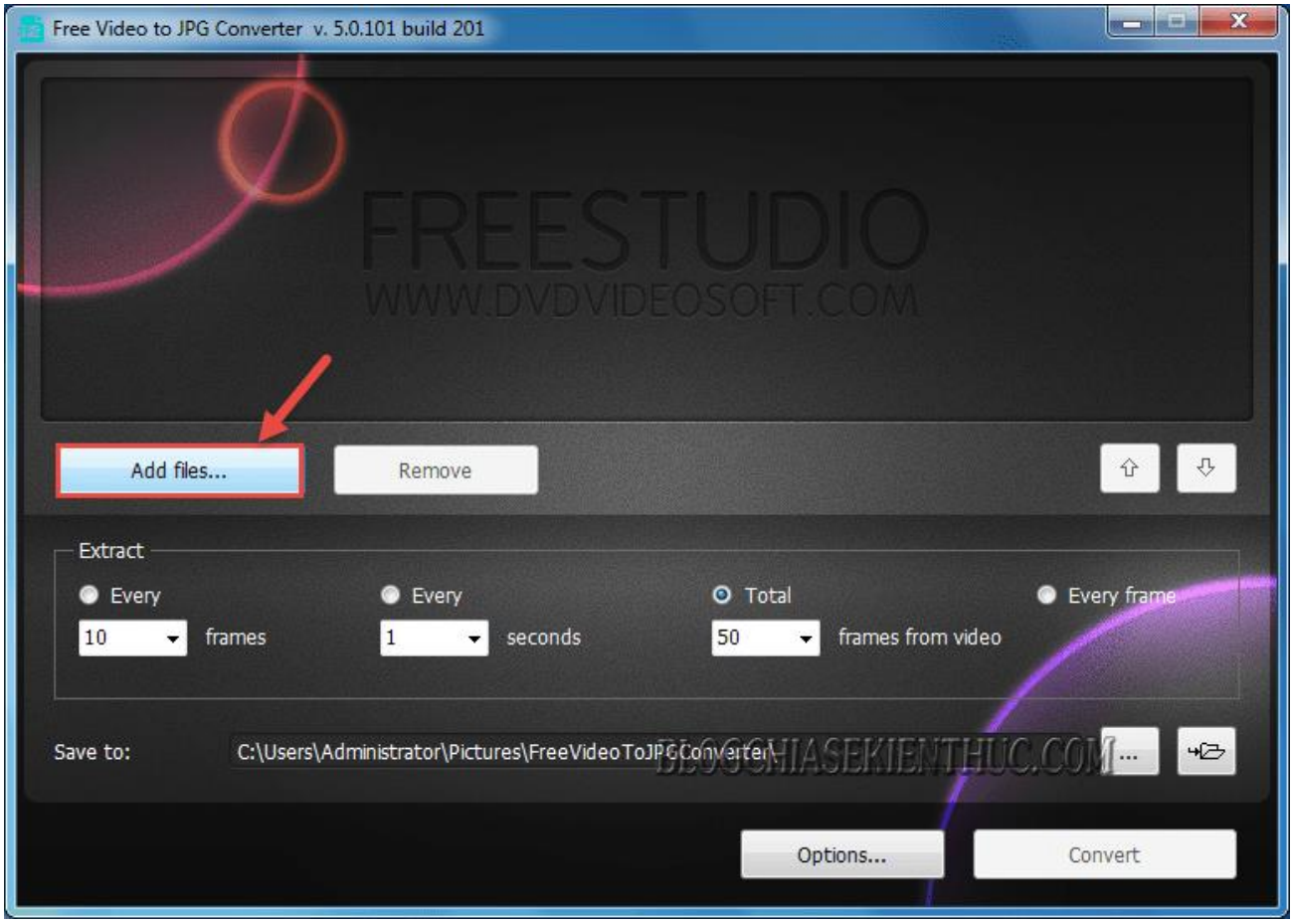
Today technology is evolving and AI is going up day by day. Smoker identification is a practical application to identify the act of smoking and consider smoking in the right place or not.

## 2.2. Build a program to detect

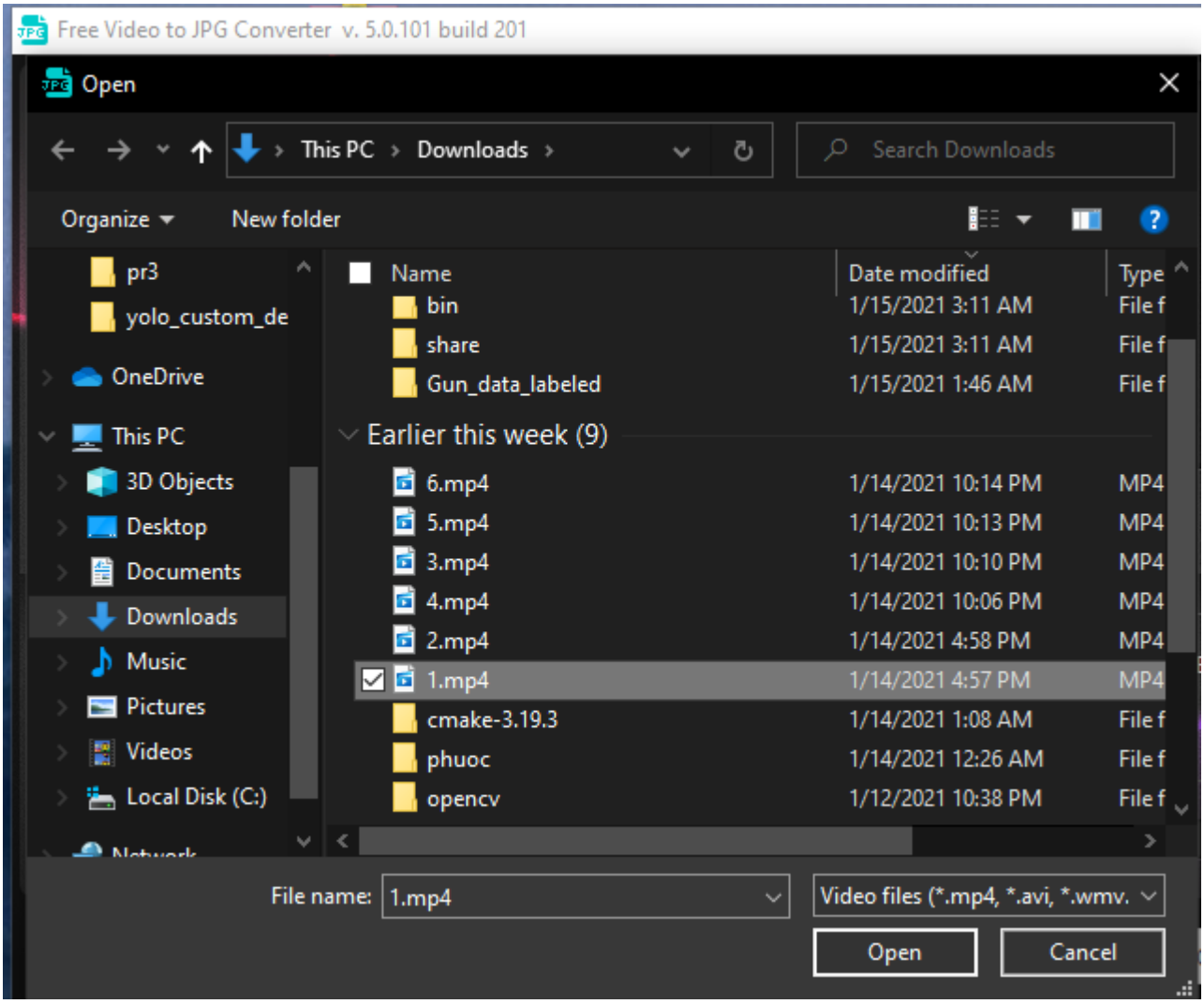
### 2.2.1 Prepare the Image dataset

Here our team has recorded 6 videos of smokers and using the software Free Video to JPG Converter to output a picture frame of about 1000 photos where there is the custom object you want to detect by steps :

**Step 1**: After you open the program => at the application interface, clickAdd files....




And find the video file you need to split photos on your computer. Click on Video, then click Opento add Video to the application.

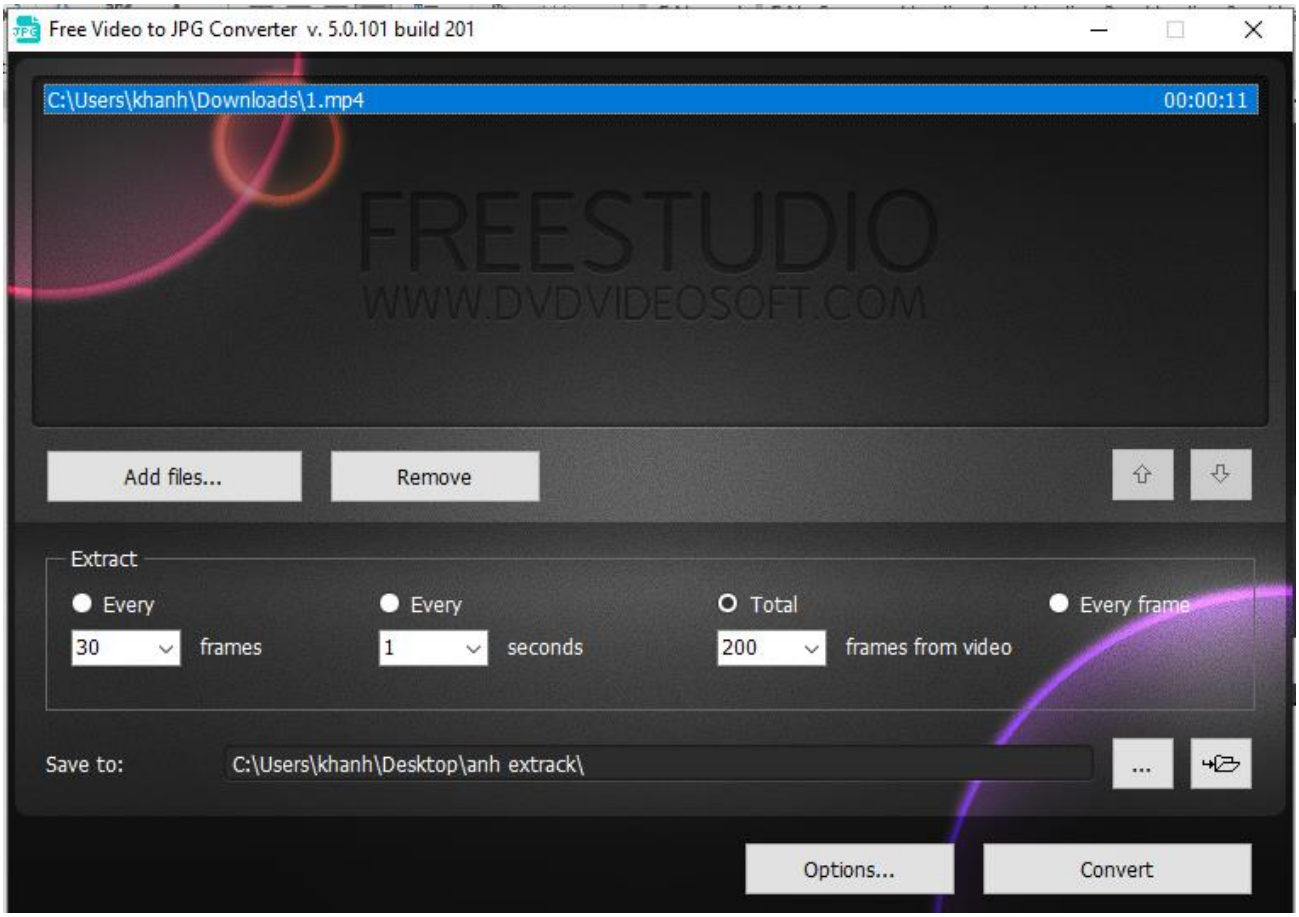



+ **Step 2**: After the upload is complete, click the icon Browse...to change the file export path. Or you can also keep the default file saving directory according to the path:

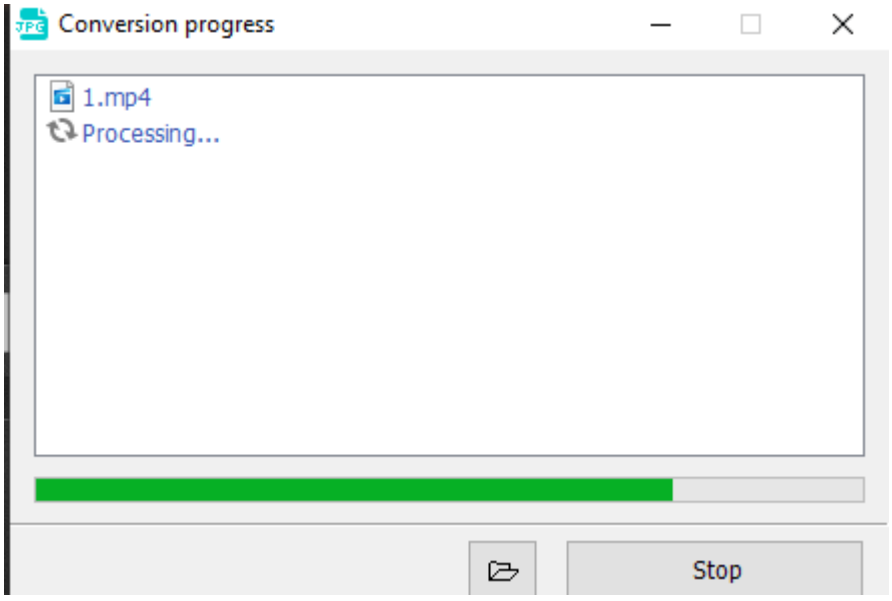
In addition, you can add settings in the section Extractas shown:

- Export 10 seconds / 1 image.
- Export 1 second / 1 image.
- Export 50 images across the entire video.
- Export images in 24 frames per second.

=> Then click Convertto convert

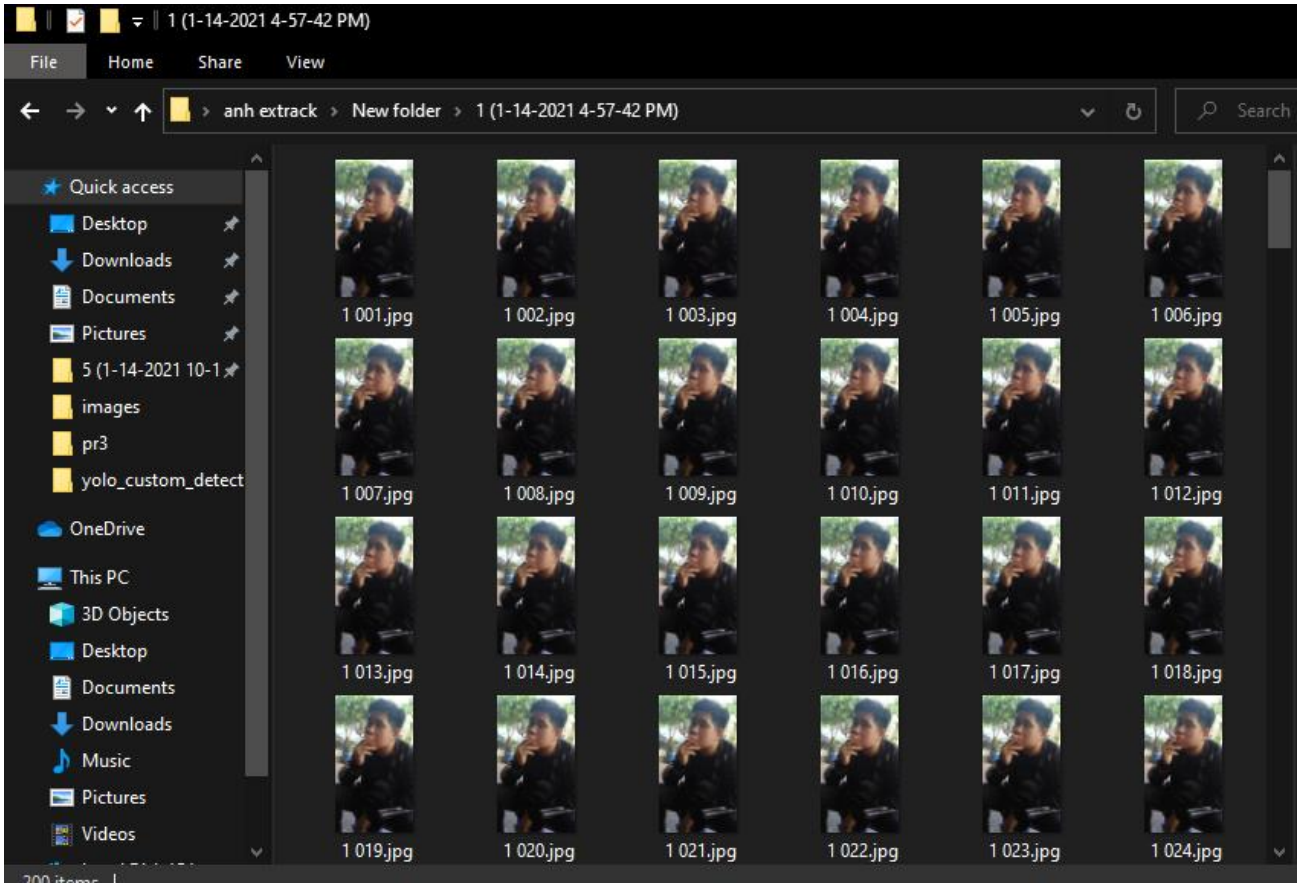



+ **Step 3**: Wait… the process of extracting images from the Video




After the process is Convertcompleted, a text will appear Process is completed. You click Closeto exit the application.

**Step 4**: Then you go to the folder where the file is saved in Step 2 to get the pictures that the program has extracted.

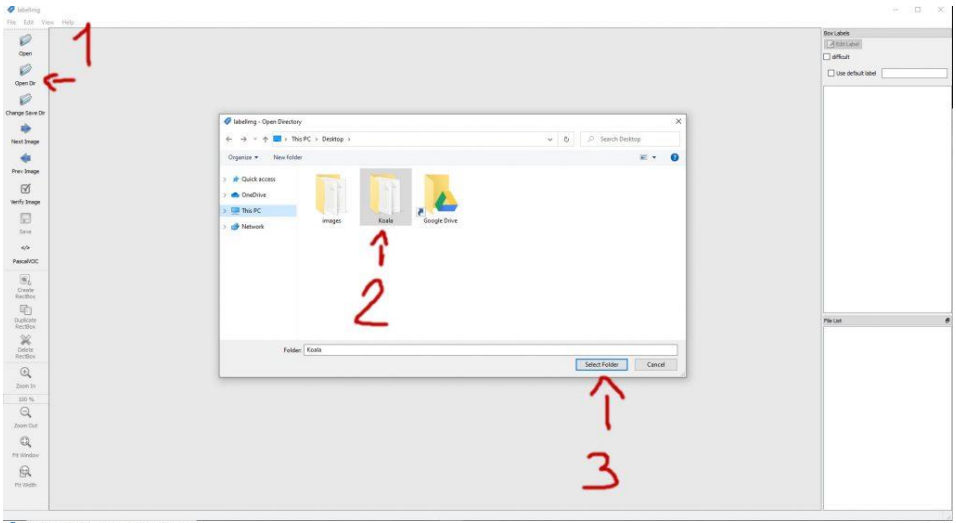


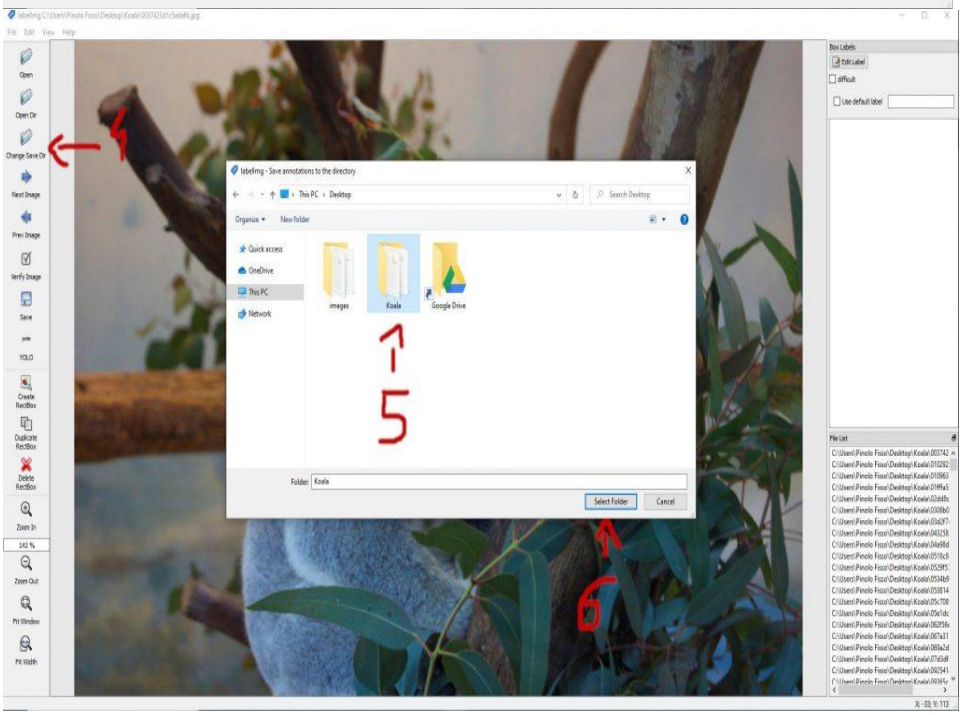

Having the iamges is not enough, but we also need to specify where the custom object is located on the specific image.

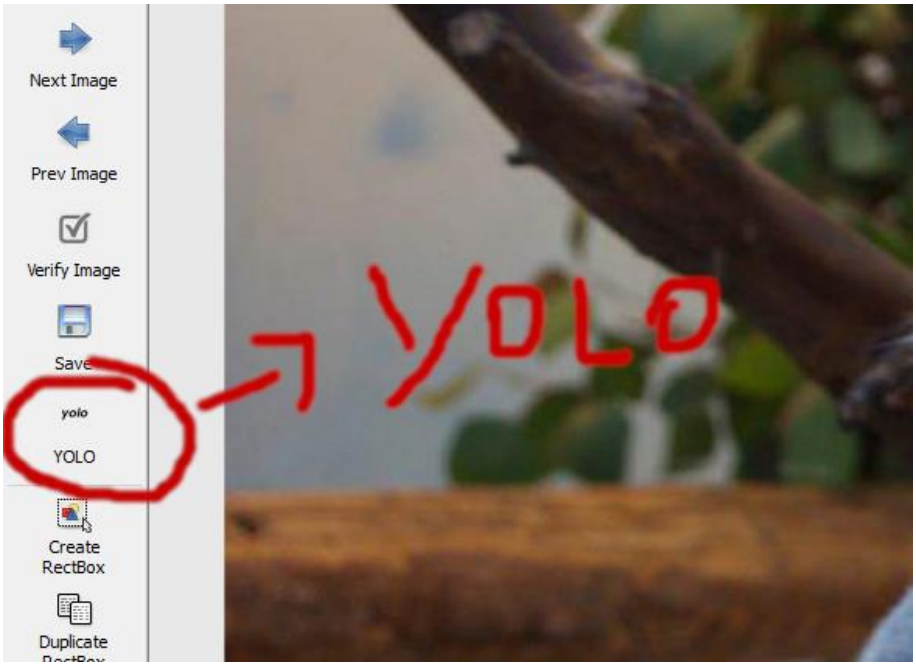
For this operation we will need an external software: **LabelImg.**
You can download it here for Windows and Linux: https://tzutalin.github.io/labelImg/
If you have the MAC, you can install it by following the instructions here: https://github.com/tzutalin/labelImg#macOS

**Let's set labelimg for our dataset:**

1. Once we run LabelImg let's click on "Open Dir".
2. We choose the folder where the images are located
3. Then we click on "Select folder"
4. We then click on "Change save dir".
5. We select the folder where the images are located (same folder we selected on step 2).
6. Then we click on "Select folder".
7. Finally make sure that we're using the settings for YOLO.
   If pascalVOC is written, then let's click and we will see YOLO.


1
2
3


4
5
6

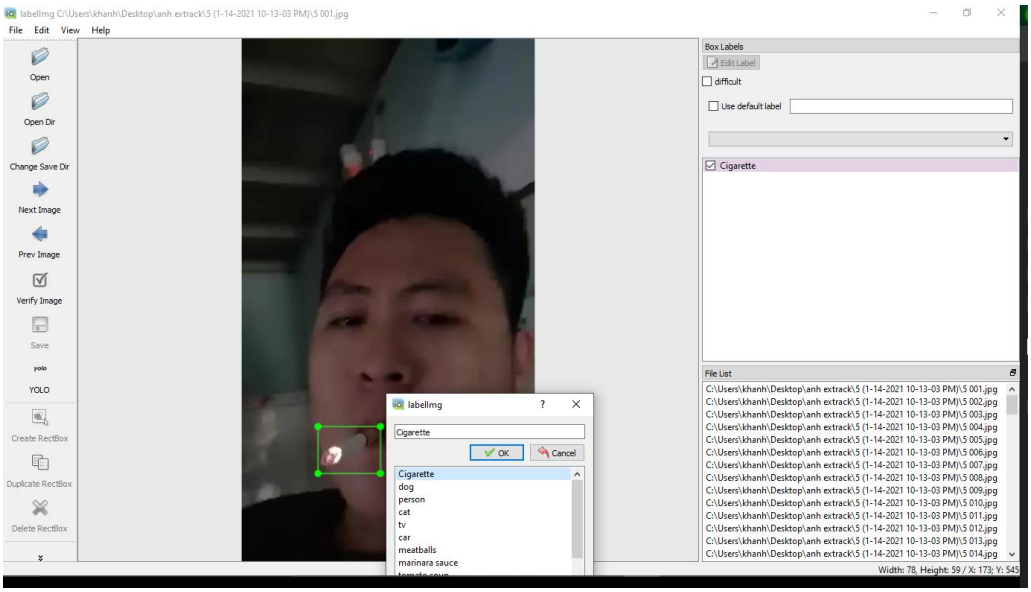



**Now we're ready to label the images.**

1. Let's click on "Create RectBox"
2. Let's select the area where our object is located (in my case I'm going to select the Koala)
3. We add the label with the name of our object. In my case typed Koala and press Ok.
4. We click on "Save"

We're going to do this operation for all the images we have on the dataset.

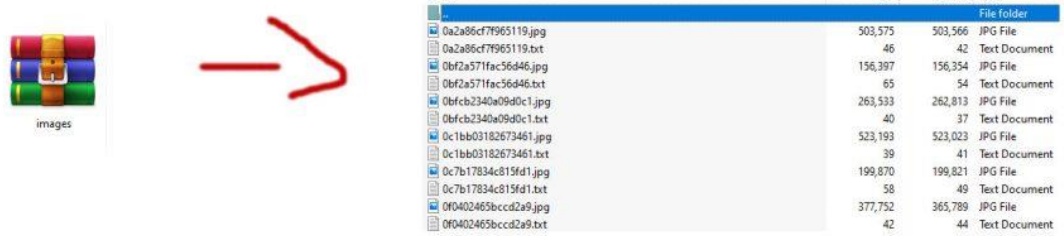
.



At the end of this operation, we should see on the folder of the images, for each image a .txt file with the same name.

Inside the file will look like this: 15 0.392578 0.504395 0.566406 0.791992

**Let's now put all the images and txt file into a .zip archive called images.zip**



### 2.2.2. Train the Image dataset online

To train the image dataset we're going to use the free server offered by google colab.

Google colab is a free service offered by google where you can run python scripts and use machine learning libraries taking advantage of their powerful hardware.
It's for free with the only disadvantage the you can use it for 12 hours in a row, after that you'll be disconnected and your files will be deleted.
you can restart it again but doing everything from scratch.

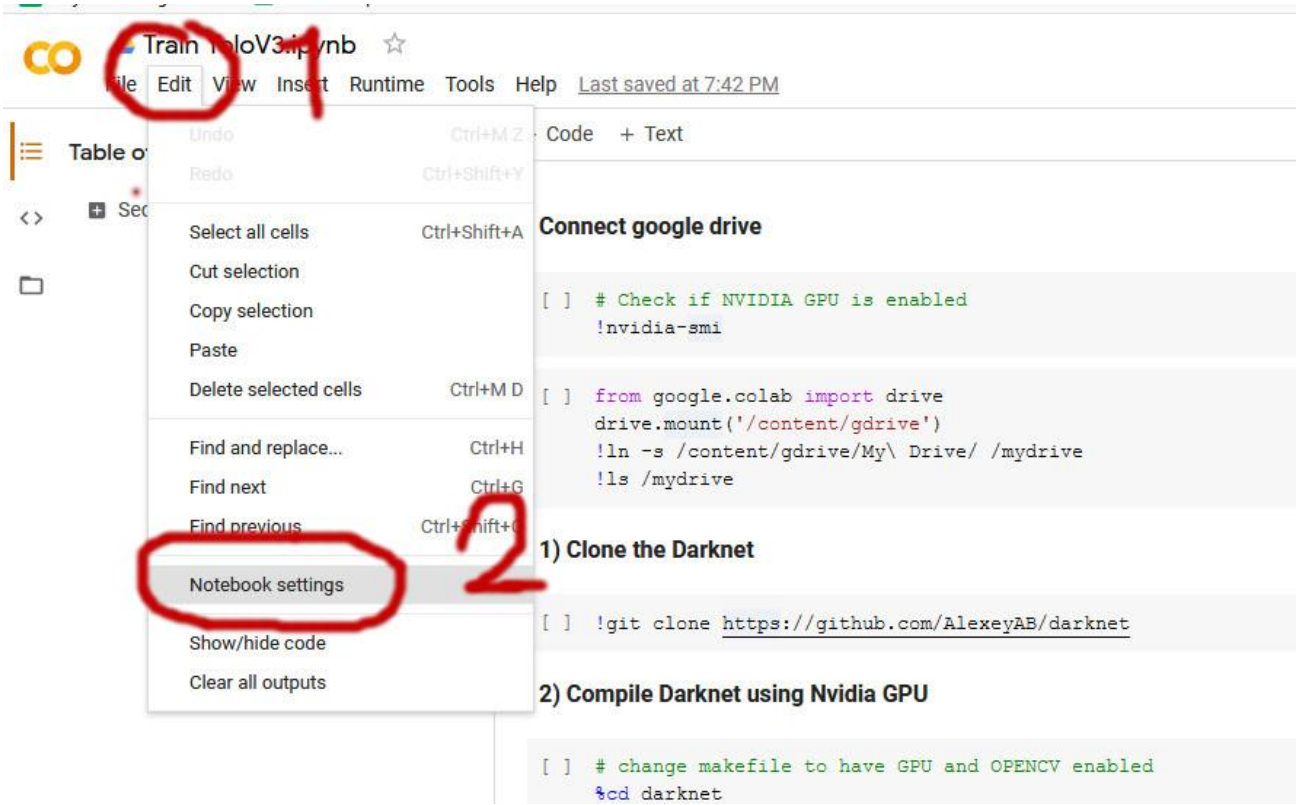
We can solve this problem by connecting google colab with google drive, so we won't lose the files in case of disconnection.

**Set up google drive:**

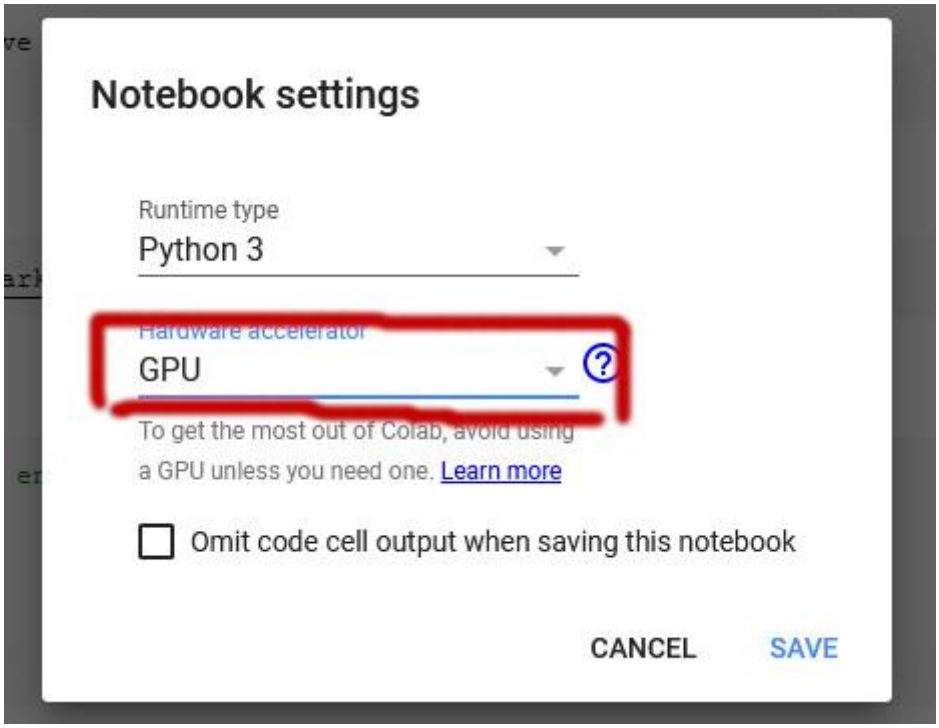
1. Go on google drive and log in. If you' don't have an account, create one and log in.
2. Create a new folder called "yolov3".
3. Then upload the file "images.zip" you created before inside the yolov3 folder.

**Set up google colab:**

1. Go on google colab and log in with the same account you used to log in on google drive.
2. Upload this file "Train_YoloV3.ipynb"
   *n.b. You can get this file by clicking on "Click here to download the Source code" at the beginning of the post.*
3. Then we need to enable the GPU. So click on "Edit".



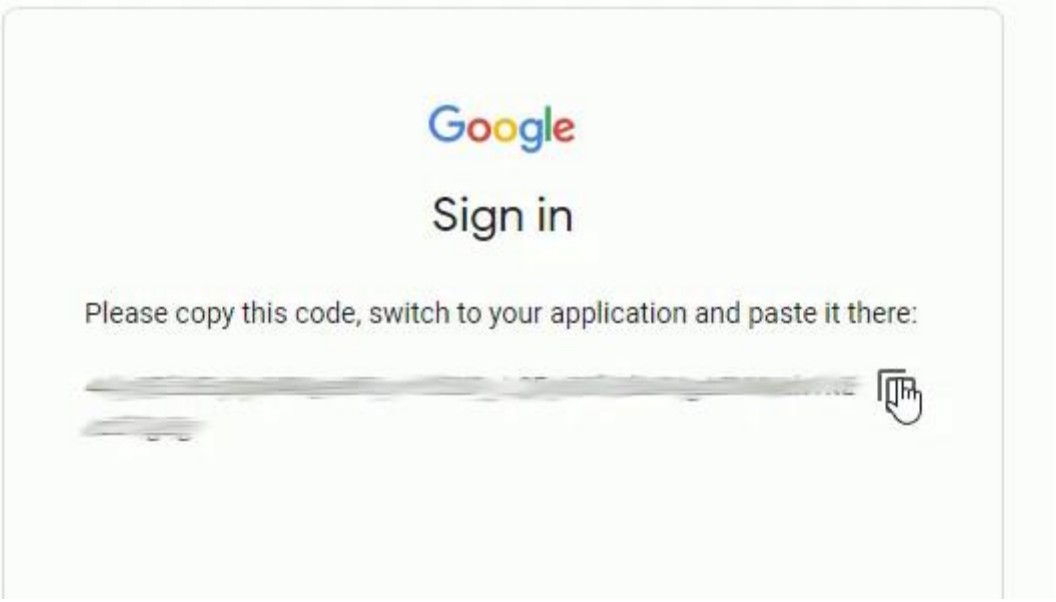
4. Then we select “GPU” and click save.



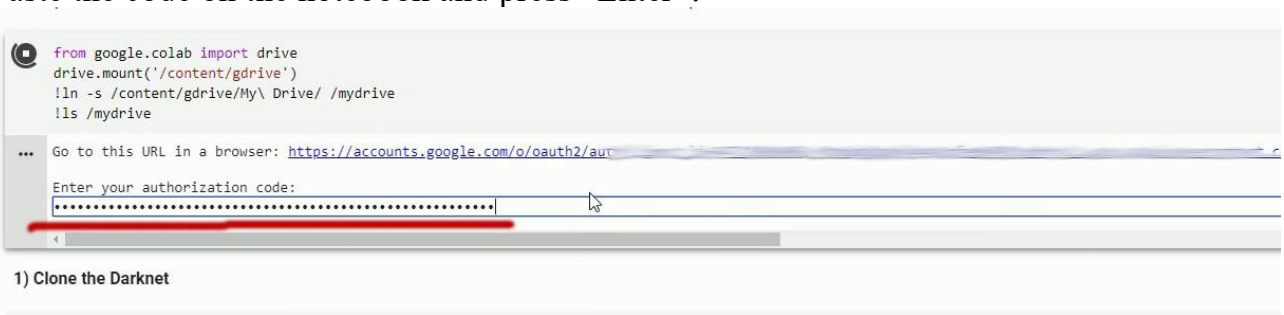

5. Now we’re ready to connect Colab with our drive.
Run the cell where it’s written “from google.colam import drive” and then clicl on the link that appears.


6. This links is to grant access to your google drive. Once you enter it asks you to allow the Google drive file Stream. Click on “Allow”. And then copy the code that

appears.




7. Paste the code on the notebook and press “Enter”.




Your colal is ready for the training.

Now you can **start the training** by simply clicking on “Runtime”, then “Run all”.

The command above is going to run all the cells.

The code will automatically install the darknet (framework used to run and train YOLO), it will make the configuration and it will run the training.

If you see an output similar to the one below, then well done, your model is training.


## 2.2.3. Test the model we created

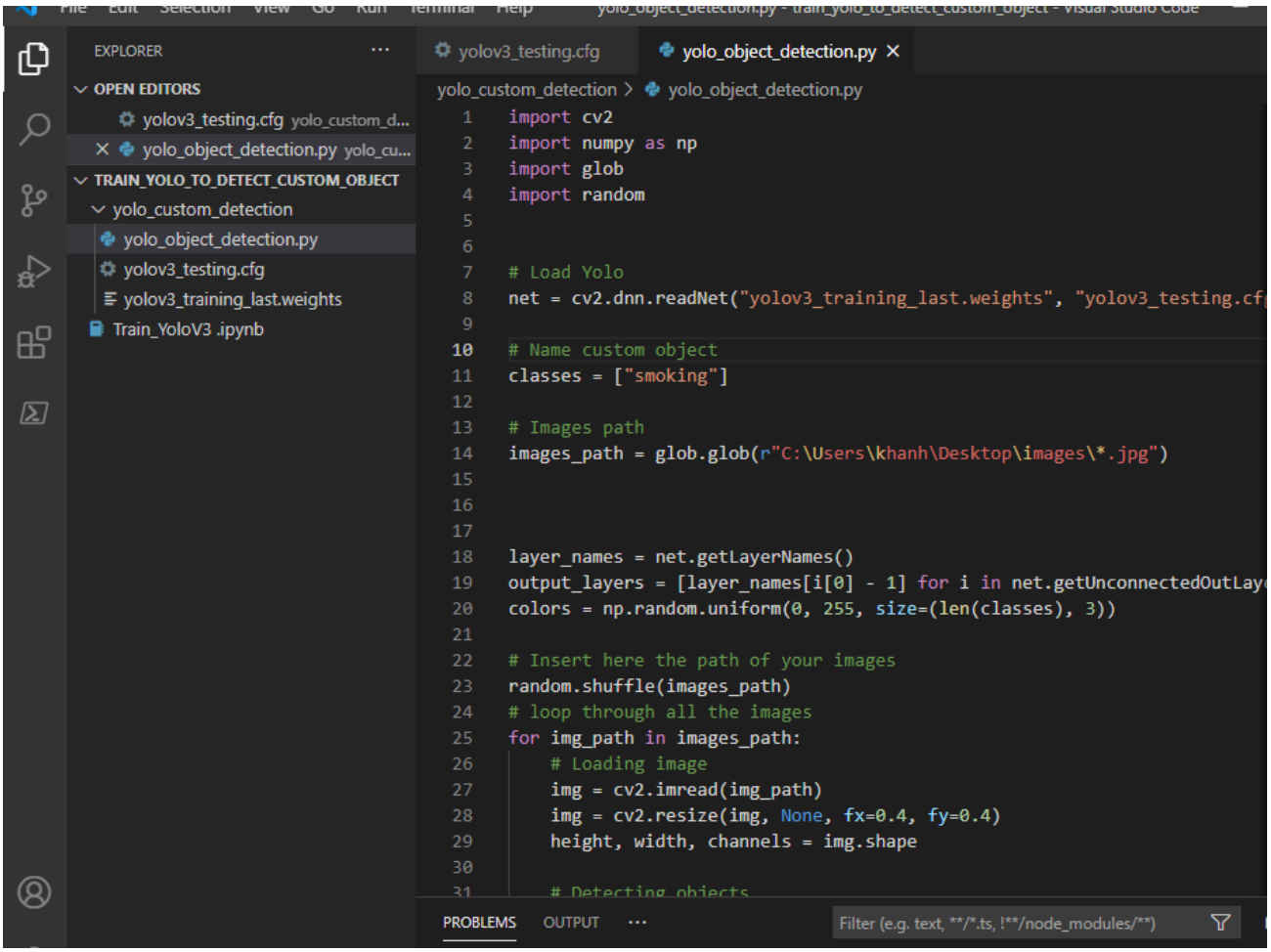
Each 100 iterations, our custom object detector is going to be updated and saved on our Google drive, inside the folder “yolov3”.

The **file that we need is “yolov3_training_last.weights”.**

You might find that other files are also saved on your drive, “yolov3_training__1000.weights”, “yolov3_training_2000.weights” and so on because the darknet makes a backup of the model each 1000 iterations.

I created a python project to test your model with Opencv.

The project is on the folder yolo_custom_detection, which contains 2 files (yolo_object_detection.py and yolov3_testing.cfg).




## 2.3 Pros and cons

### a. Advantages

The quick setting algorithm can recognize cigarettes

### b. Disadvantages

- Poor recognition of slightly blurred images.

- Not running on the app can only read images

### c. Development

- Can read the video and identify it in advance

- It runs out a basic app that the user can easily use

## 2.4 Difficulties encountered

- Identification was not highly accurate. There are some images that are still unrecognized
- Haven't figured out how to read the video file

## *Preference*

https://github.com/mehulpurohit97/Cigarette-Smoking-Detection-using-Deep-Learning/
https://github.com/AlexeyAB/darknet#how-to-compile-on-windows-using-cmake
https://github.com/TharunAts/People-Smoking-in-restricted-places-Object-Detection
https://pysource.com/2020/04/02/train-yolo-to-detect-a-custom-object-online-with-free-gpu/
https://www.youtube.com/watch?v=_FNfRtXEbr4&t=1314s