INSTRUCTION FOR INSTALLATION AND IMPLEMENTATION OF THE PROGRAM (WATERMARKING DETECTION)

Group information (group 7)

StudenID	Full name	Email	Phone number
21127289	Đoàn Việt Hưng	dvhung21@fitus.edu.vn	0394 529 869
21127412	Hồ Bạch Như Quỳnh	lnpuyen21@fitus.edu.vn	0762 902 938
21127476	Lê Nguyễn Phương Uyên	hbnquynh21@fitus.edu.vn	0945 420 790

1. Installation:

a) Programming Language:

In this program, we are using Python - a universal programming language for machine learning and deep learning.

There are several reasons why Python is a popular choice for training deep learning models:

1. Easy to learn and use: Python is an easy-to-learn and easy-to-use programming language, especially for beginners and those without a strong programming background.
2. Large community: Python has a large and active community, meaning you can easily search for and use libraries, tools, and support from the community for training deep learning models.
3. Rich libraries: Python has many powerful deep learning libraries such as TensorFlow, PyTorch, and Keras, making it easy to build and train complex models.
4. Good integration: Python integrates well with other tools and libraries in the process of developing and training deep learning models.
5. High performance: While Python is not the fastest language, using libraries like NumPy and TensorFlow, which are written in C++ and have high performance, still allows for efficient model training.

Download Python at this site: https://www.python.org/downloads/

b) Code editor:

In order to run the program, users should have a suitable IDEs (or Web-based platform) for an easier, faster and more convenient because of necessary built-in tools.

We recommend using Visual Studio Code because of its integration of multiple programming languages and web-based interactive computing platform (Python and Jupiter Notebook are our two main things in this program).

Download Visual Studio Code at this site: https://code.visualstudio.com/Download

c) Vital Libraries:

There are lots of available libraries in Python used for deep learning such as Torch, Tensorflow, Keras, Fastai,...

In this program, we are going to use Pytorch and Torchvision as two main libraries and there are some supported libraries such as PIL (ImageFile), timm and os.

For installing these libraries above, we can do it through our computer command promt by using **'pip install <library_name1> <library_name2>'**

EG: if you want to install Torch and Keras, our command line should be: 'pip install torch keras'

The prerequisite is you have to download Python first (read section 'a').

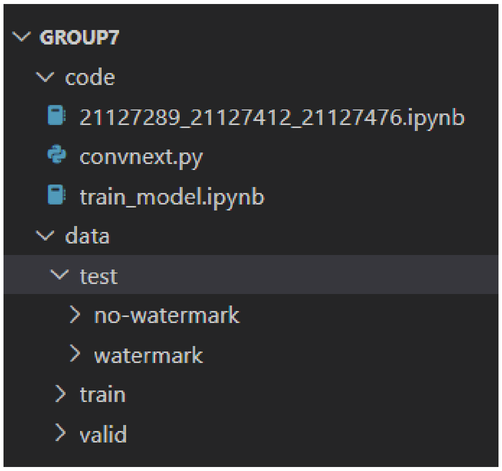
d) Dataset:

You can create your own dataset. Make sure your images, some have watermark and some do not, and you have splitted the dataset into 3 folders for 3 different purposes: test, train and validation, Each folder consists of 2 subfolders "watermark" and "no-Watermark".

OR you can directly download available datasets from the Internet.

https://drive.google.com/drive/folders/1RcHyDj9JBS6Tl0r8L2IeeCrrsUcUFVLR

After downloading, put the folder containing dataset right outside the folder containing source code, the directory tree should look like this:



```
GROUP7
  code
    21127289_21127412_21127476.ipynb
    convnext.py
    train_model.ipynb
  data
    test
      no-watermark
      watermark
    train
    valid
```

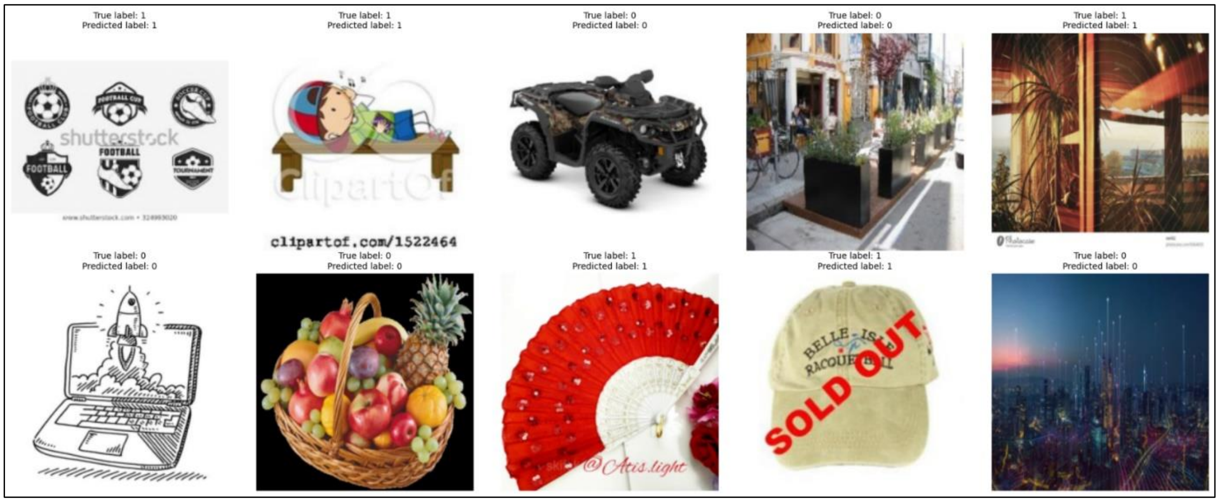
2. Implementation

****Note**: you can download the model had been trained and then use it to run file "21127289_21127412_21127476.ipynb" immediately.**

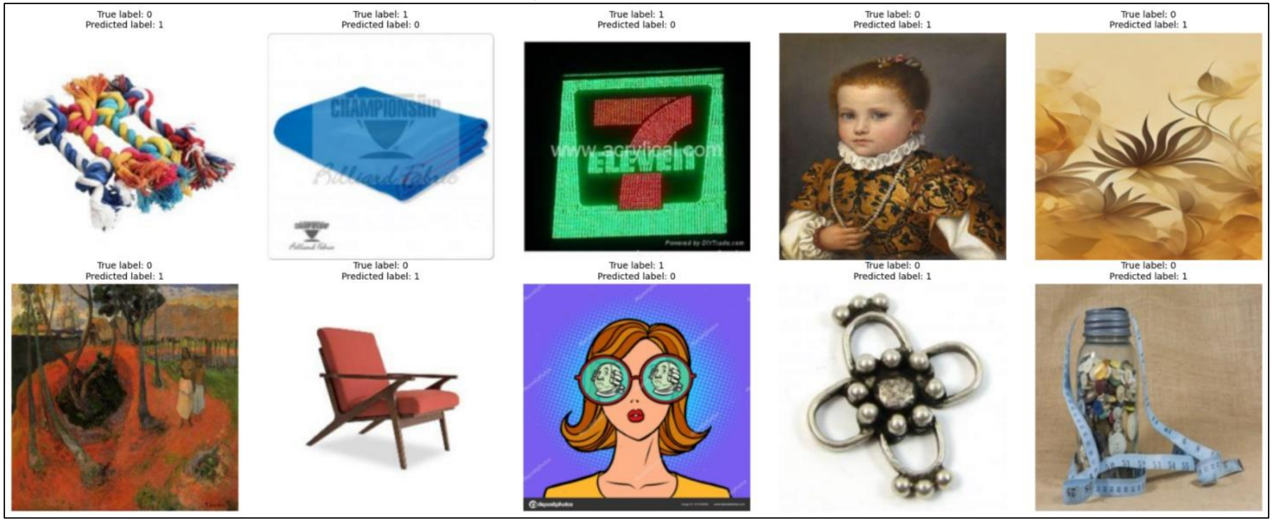
To run the program:

1. **Run file "train_model.ipynb":** This file is used to train a CNN model to classify images into two categories: those containing watermarks and those without. It's likely part of a larger project aimed at automating the detection or removal of watermarks from images.
2. **Run file "21127289_21127412_21127476.ipynb":** This file is used to evaluate the performance of the trained watermark detection model on a separate test dataset. It provides insights into the model's accuracy, showcases examples of correct and incorrect predictions, and visualizes the model's performance through a confusion matrix. This evaluation process is essential for assessing the model's effectiveness and identifying areas for improvement.
3. **Finally, you will receive your expected output:**

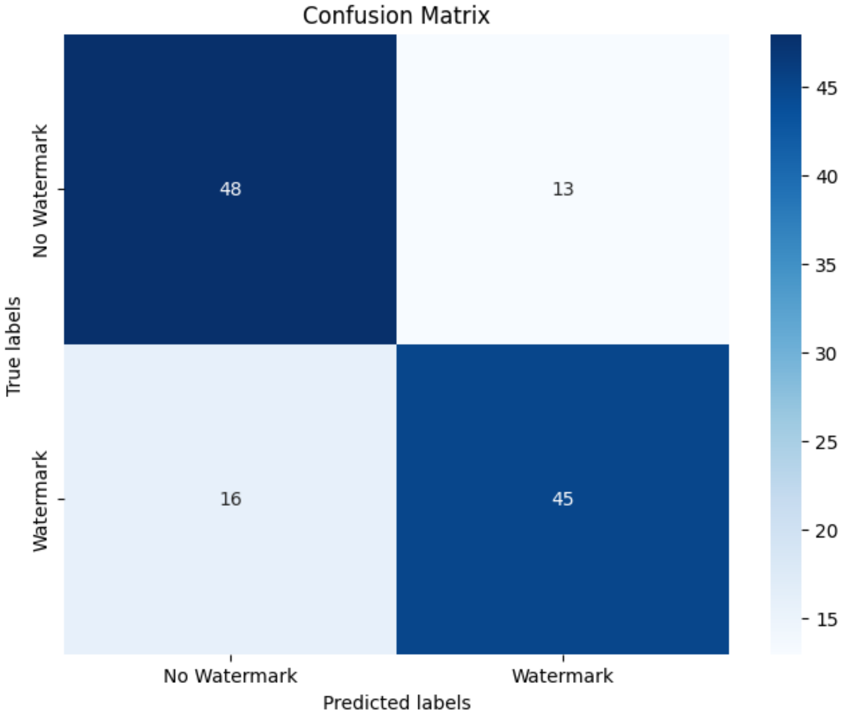
   some correct prediction images:



some incorrect prediction images:



Confusion matrix


Confusion Matrix
True labels
No Watermark
Watermark
Predicted labels
No Watermark
Watermark