

ProgressSoft Assignment

Problem 3 - Wrapper Class Library

Approach:

For this problem, I created a **.net 6** class and **Python** class (I will talk about it later) that loads a trained model and uses it to make predictions on images to classify if the person wearing a mask or not.

The C# implementation:

I used the **ML.NET** framework to create a prediction engine that loads the **ONNX** model file, processes the input image, and returns the predicted result. The input image is first resized to the required size and then converted to a pixel array. Finally, the processed image is passed to the prediction engine, which returns the predicted result.

The solution has two main files:

- **FaceMaskClassifier.cs**: this file has the code for the class and what it does.
- **Program.cs**: this file is the main file that has the code for making objects from the class giving it the image path as input for it and printing the prediction.

That was what I planned HAHA, but unfortunately, there is a bug with the code and I can't fix it, the class is built in a good way, and the main program also have to bugs, but when I run the code, an error pops up that related to loading the model, and it says that there is an error for initializing python when loading the model. I tried many things to solve this issue but unfortunately, I couldn't. There is a lack of bugs solved in stack overflow for the **C#** programming language. That is why I solved this problem using Python.

The Python Implementation:

I created a wrapper class that loads the model and uses it to make predictions on images. The input image is first load the model then the image using **Keras preprocessing** library and then resized to the required size (the size that the model trained on), then the image is normalized and passed to the loaded model, which returns the predicted class.

This solution has two main files:

- ***Face_mask_classifier_py.py***: which has the wrapper class that does everything mentioned above.
- ***Main.py***: which is the main file called the class that makes predictions and prints the result.

With the ***main.py*** file, I called the class with “***from face_mask_classifier_py import FaceMaskClassifier***” which is the name of the class. This file has a code for the **argparse** library just to run the script in an easy way. When initializing the class instance, we pass the image path and the model path as the parameters to the class, and after that call the predict function that returns the prediction, then print the results.

Test Results:

The program can be tested by providing different images as input and verifying that the prediction is accurate.

How to Run the Program:

- For **.net 6 (C#)**: As it should be, run the following command in cmd: ***dotnet run <model path>*** but the code will not run and it didn't handle the argument passes through the CMD because that the program will not run because of the bug I mentioned before.
- For **Python**: run the following command in cmd after running ***pip install -r requirements.txt*** :
python main.py --img_path path/to/img --model_path path/to/model

Note: I only tested it on Windows

Python Version:

The python script was developed using Python **3.9.13**

Future Enhancement:

Possible future enhancements for this problem could be to solve the problem with the .net 6 program, also include adding the ability to predict multiple images at once.