

# 國立臺北科技大學自動化科技研究所

## 嵌入式工業機器視覺

### Integration of Industrial Machine Vision in Embedded Systems

#### *Assignment #1*

Starts from Wednesday, 10/11

Due Wednesday, 10/25

#### **Objective**

The main objective of this assignment is to create the APIs of image processing that actively communicates C#. You are asked to use P/Invoke technique in C#. This assignment has three parts.

You have to add “save bmp” function to **NImage** class (provided in **Lab 4**) in the first part. The second and third parts are to design “filtering” and “adaptive thresholding” functions with **NImgProcess** class (**Lab 5**), respectively. This assignment must be submitted using the “NTUT Tech. i School” submission system (see the website for details). You may submit it multiple times if you wish. Only the last submission will be counted. For example, it might be a good idea to go ahead and submit it after you have done the instructions and gotten that part working.

#### **Materials**

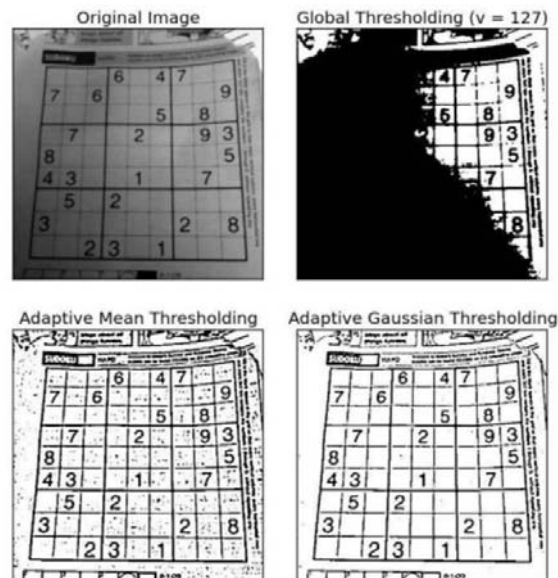
Before you start this assignment, you have to download Visual Studio 2017 (or 2022) and the source codes of “filtering” from NTUT Tech. i School. It is critical that you get the source codes downloaded and functioning as early as possible so that if you have problems you will have a chance to talk me and get help.

#### **Required Tasks**

1. Follow the instructions of **Lab 4** and **Lab 5** to design and run the “save bmp” function with **NImage** class. You have to implement this function in C# [P/Invoke]. Do not proceed to the next steps unless your function as expected and executes without errors. Using P/Invoke for saving bmp file that can follow the instructions of **Lab 4** (20%).
2. Refer to the downloaded source codes; please add “filtering” functions to **NImgProcess** class. Your class are wrapped by using C# [P/Invoke]. Please ensure they are executable. (40%)
  - (1) Image filtering functions are implemented to **NImgProcess** class including mean filter (3x3), Sobel and Laplacian. (30%)
  - (2) Using P/Invoke in C# for image filtering that can follow the instructions of **Lab 5**. (10%)

3. Develop the algorithms of adaptive thresholding for `NImgProcess` class. And the wrappers of “adaptive thresholding” functions are applied in C# [**P/Invoke**]. Please ensure they are executable. **(40%)**

- (1) Adaptive thresholding functions are implemented to **NImgProcess** class including `AdaptiveMean` and `AdaptiveGaussian`. **(30%)** Please refer to [https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html).



- (2) Using P/Invoke in C# for adaptive thresholding that can follow the instructions of **Lab 5**. **(10%)**

4. Avoiding the problems listed in the Evaluation section below is part of the required tasks of every assignment. This list grows as the semester progresses, so be sure to check it again with each assignment.

## **Evaluation**

In all of the assignments this semester, designing quality API that executes without errors and then testing the resulting application and iterating until it functions properly is the goal.

Here are the most common reasons that assignments are marked down:

- Project does not build.
- Project does not execute without errors.
- One or more items in the **Required Tasks** section were not satisfied.
- A fundamental concept was not understood.
- Code is sloppy and hard to read (e.g. indentation is not consistent, etc.).

- Assignment was turned in late.

**Submission (Deadline: 2023/10/25 23:59)**

1. Source code of your design for the tasks (including .h / .cpp / .cs / .py / .ui / .dll / .exe files)
2. Documents for describing your design and how it works