

# CV

## Project 1: Binary and Non-binary Threshold Operations

### C++

**Adrian Noa**

**Due 9/2/2022**

Algorithm steps:

1. Check for correct number of arguments: input file name, and 2 output file names
2. Set input stream to the first argument and output streams to the output files
3. Read in first 4 integers from input file header and use for row, col, min and max values
4. Request user input for threshold value. Incorrect input sets threshold to 6
5. Write to output files the corresponding header
6. Nested loop row \* col times and write values
  - a. 0 or 1(greater than threshold) if binary : output 1
  - b. 0 or pixel value(if greater than threshold) for non binary : output 2
7. Close streams

```
/*
Created by Adrian Noa

Objective: To take an image asset and smoothen each of its pixels using binary
and nonbinary threshold operations

Usage:

g++ noa_adrian_main.cpp -o main.exe && ./main.exe data.txt out1.txt out2.txt

OR

g++ noa_adrian_main.cpp -o main.exe

./main.exe data.txt out1.txt out2.txt

*/
```

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main(int argc, const char* argv[]) {

    if (argc != 4){
        printf("Not enough arguments\n");
        return 1;
    }
    ifstream myFile(argv[1]);
    ofstream outFile1(argv[2]); // binary threshold operation
    ofstream outFile2(argv[3]); // non binary
    if (!myFile.is_open()) {
        cout << "Unable to open file" << endl;
        exit(2);
    }
    int numRows, numCols, minVal, maxVal, thrValue;

    myFile >> numRows >> numCols >> minVal >> maxVal;

    printf("What is the threshold value? ");
    cin >> thrValue;
    thrValue = ( cin.fail() || thrValue > maxVal || thrValue < 0 ) ? 6 : thrValue;
    // thrValue=6;
    cout << "Using threshold value " << thrValue
        << " to perform smoothening operations" << endl;

    outFile1 << numRows << " " << numCols << " " << 0 << " " << 1;
    outFile2 << numRows << " " << numCols << " " << minVal << " " << maxVal;
    int pixelVal;
    for (int i = 0; i<numRows; i++){
        outFile1 << endl;
        outFile2 << endl;
        for (int j=0; j<numCols; j++){
            myFile >> pixelVal;
            if(pixelVal >= thrValue){
                outFile1 << "1 ";
                outFile2 << pixelVal << " ";
            } else {
                outFile1 << "0 ";
                outFile2 << "0 ";
            }
        }
    }

    cout << "Image processing complete\n";
    myFile.close();
    outFile1.close();
    outFile2.close();
    return 0;
}

```

### Input screenshot (data.txt)

[illegible]

### BINARY OUTPUT screenshot (out1.txt)

[illegible]

### NON Binary OUTPUT screenshot (out2.txt)

[illegible]