

Project 5: A Simple CNN Model

Please implement a simple convolutional neural network (CNN) model using C/C++. You can follow <https://github.com/ShiqiYu/SimpleCNNbyCPP> where a pretrained CNN model is provided. The model contains 3 convolutional layers and 1 fully connected layer. The model can predict if the input image is a person (upper body only) or not. More details about the model can be found at SimpleCNNbyCPP web site.

You are welcome to implement more CNN layers, and to make the implemented CNN to be more general (such as the convolutional layer can be for any size of kernels, not just 3x3).

Do not use any third-party library except OpenCV to read image data from image files. You should implement all CNN layers using C/C++.

Hints

1. Only the forward part is required to implement, and the backward part (the training part) is not mandatory.
2. You can implement an unoptimized version firstly to verify the correctness of the implementation. Then you can optimize it for a better speed. Make your source code simple, beautiful and efficient.
3. The convolutional operation can be implemented by matrix multiplication. Do not waste your experience and the source code in Project 2, Project 3 and Project 4.
4. The parameters trained have been put into a CPP file `/weights/face_binary_cls.cpp`. You can just include it into your project.
5. You can use OpenCV to read images. The image data stored in `cv::Mat` is unsigned char type. You should convert the data to `float` and normalize it to range `[0.0, 1.0]` firstly before operate it in a convolutional layer. Be careful with the order of pixel colors (channels) in `cv::Mat`. It is BRG, not RGB. The input of the CNN model should be a 3x128x128 (channel, height, width) data blob (matrix). You can adapt your class in Project 4 for this project.
6. The output of the CNN model is a vector with two float numbers `[c0, c1]`. $c0 + c1 = 1.0$ and `c1` is the possibility that the input is a person (upper body only).
7. Test your program on X86 and ARM platforms, and describe the differences. Be careful that `char` is different on X86 and ARM. It is signed char on one, and unsigned char on another.

Rules:

1. Please submit your project report before its deadline. After the deadline, **23:59 on Dec. 19th**, (even 1 second), **0 score!**.
2. Please submit the files as: report.pdf, source1.c, source2.c, header.h. I think you do not need too many files for the project. Please do **NOT** put the files into a compressed one.
3. Your score will also depend on the quality of your source code and your report. Your report should be easy to understand and describe your work well, especially the highlights of your work.
4. Do **NOT plagiarize** your classmates' projects or past students' projects. The instructor may use plagiarism tools to check all your source code including past year students'.
5. Please pay more attention to your code style. After all this is not ACM-ICPC contest. You have enough time to write code with both correct result and good code style. You will get deduction if your code style is terrible. You can read Google C++ Style Guide (<http://google.github.io/styleguide/cppguide.html>) or some other guide for code style.