

C++ Project: 实时隐私保护工具

赵英迪 12311714

一、摘要

本项目开发了一款高效的实时隐私保护工具，利用C++和OpenCV技术，实现了人脸检测及多种隐私保护模式。包括模糊处理(blur)、像素化处理(Pixel)和图像遮盖(mask)，此外，用户还可以动态调整处理参数，并上传自定义遮罩图片，以实现个性化的隐私保护效果。

二、环境配置

1. OpenCV

出于对OpenCV进一步了解与使用的兴趣，我在GitHub页面下载了OpenCV和OpenCV_contrib的源码，并在Windows系统下通过CMake工具进行源代码的编译。因为网络代理等原因，有些文件无法下载，我通过手动检查CMakeDownloadLog对相关文件进行单独的下载和替换，完成了对OpenCV及其contrib模块的配置。

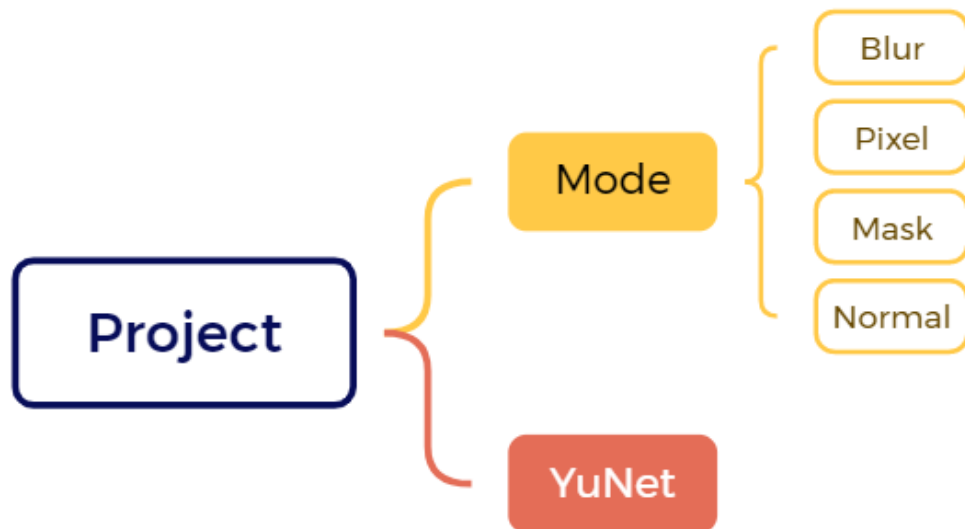
2. Visual Studio

Visual Studio为微软研发的IDE，对cpp开发和相关依赖环境的配置有较好的支持，故使用其作为项目开发的工具平台。首先将OpenCV加入系统环境变量，然后再具体项目中完成对OpenCV库的附加包含目录、附加库目录与附加依赖项的配置。注意到OpenCV源码被编译为opencv_world4100d.dll文件，故在此处应为动态库连接。

3. YuNet模型

随着深度学习技术的发展，深度学习方法在人脸识别方面的性能远超传统方法，YuNet具有轻量化，高效，准确、易用等特点。本项目引入了face_detection_yunet_2023mar.onnx文件，以及GitHub上于仕琪老师提供的demo代码，在YuNet实现人脸检测的基础上实现了隐私保护模式的实现。

三、代码实现



我将项目设计为以上结构，其中Mode类主要负责具体模糊模式的实现，YuNet主要负责模型的调用及对具体视频图像的处理。下面将对各个部分代码进行详细介绍。

1. Mode

该类为BlurMode, PixelMode, MaskMode, NormalMode四个类的基类，主要提供了一个内部枚举类 Type 以表示子类的模式，方便将来更好的代码拓展，以及纯虚函数 handle 供子类实现对人脸具体的模糊处理。

2. BlurMode

该类为Mode的派生类，具体负责对于传入的Mat进行高斯模糊处理。通过调用CV库中的 `GaussianBlur` 函数实现，将传入Mat和传出Mat均设置为img_，通过调整高斯核的大小来设置模糊程度，后两个参数代表高斯核在X, Y方向的标准差，设置为0表示模糊程度仅与核大小有关。

```
1 void BlurMode::handle() {
2     if (img_.empty()) {
3         cerr << "Error: 无法读取图像" << '\n';
4         return;
5     }
6     GaussianBlur(img_, img_, Size(val_, val_), 0, 0);
7 }
```

3. PixelMode

该类为Mode的派生类，具体负责对传入的Mat进行两次resize，达到像素化的效果。val_值表示将目标区域压缩为 $\{h/val_, w/val_\}$ 后在放大填充回原来区域。 `INTER_LINEAR` 与 `INTER_NEAREST` 与插值算法有关，保证了较好的像素化效果。

```

1 void PixelMode::handle() {
2     if (img_.empty()) {
3         cerr << "Error: 无法读取图像" << '\n';
4         return;
5     }
6     if (val_ < 1) {
7         cerr << "Error: 像素画参数错误" << '\n';
8         return;
9     }
10
11     Mat imgSmall;
12     int h = img_.rows;
13     int w = img_.cols;
14     int hNew = h / val_;
15     int wNew = w / val_;
16
17     resize(img_, imgSmall, Size(wNew, hNew), INTER_LINEAR);
18     resize(imgSmall, img_, Size(w, h), INTER_NEAREST);
19 }

```

4. MaskMode

该类为Mode的派生类，具体负责用传入图片遮罩Mat。通过对传入图片进行resize，使其大小与人脸识别框相匹配，再通过Mat类的深拷贝函数 copyTo 达成遮罩效果。通过windows.h库中的 `GetOpenFileName` 函数，可以在程序运行时打开对话框，手动选择图片路径来更改遮罩图片。

```

1 void MaskMode::handle() {
2     if (img_.empty() || mask_.empty()) {
3         cerr << "Error: 无法读取图像" << '\n';
4         return;
5     }
6     Mat resizedMask;
7     int h = img_.rows;
8     int w = img_.cols;
9     resize(mask_, resizedMask, Size(w, h));
10
11     resizedMask.copyTo(img_);
12 }

```

5. Normal Mode

该类为Mode的派生类，并未对Mat做任何处理，保留了YuNet demo中对人脸识别区域的显示。

6. YuNet

该类参考了YuNet demo中的代码，对YuNet模型进行了封装，并实现了对人脸识别区域的模糊化处理。具体实现了 `createMode` 和 `visualize` 两个函数，其中 `createMode` 负责针对每一个识别框模糊任务的创建，`visualize` 通过调用所有模式的 `handle` 函数执行模糊任务，并处理其它对捕捉视频画面的显示修改内容。

```
1 static void visualize(const Mat& faces, vector<shared_ptr<Mode>>& modes,
2     Mat& output_image, Mode::Type type, float fps = -1.f, int lev=0);
3
4 static void createMode(const Mat& faces, vector < shared_ptr<Mode>>& modes,
5     Mat& output_image, Mode::Type type, Mat mask, int val=0);
```

7. 命令行实现

该项目对命令行相关操作也进行了支持，借助了OpenCV的 `CommandLineParser`，实现了对一下命令行命令的支持：

关键字	默认参数	解释
help	NIL	解释命令行参数功能
model	face_detection_yunet_2023mar.onnx	选择model路径
mode	blur	选择进入时打开的模糊模式
blur_size	1	设置高斯模糊程度，级别为1-6
pixel_size	1	设置像素化程度，级别为1-6
mask_image	default.jpg	设置默认遮罩图片路径
device	0	设置默认打开摄像头

四、问题反思

- 帧率问题，经过测试，未进行模糊处理时视频约有25帧左右，模糊处理后在10帧左右徘徊。分析原因是对矩阵像素的修改操作拖慢了程序效率，是否存在更高效的模糊处理方式可以提高帧率，有待于进一步探索。
- OpenCV在打开罗技C930c摄像头时在默认设置下需要较长时间，在设置环境变量为：

```
_putenv("OPENCV_VIDEOIO_MSMF_ENABLE_HW_TRANSFORMS=0");
auto cap = VideoCapture(device_id, CAP_MSMF);
```

并加入参数 `CAP_MSMF` 后打开效率有了显著提高，问题得到了解决。查阅文档发现：

CAP_MSMF

官方描述：Microsoft Media Foundation (via videoInput).

Microsoft Media Foundation是微软公司的新一代多媒体开发平台，用以取代原来的Directshow，为了满足现在多媒体播放的高清晰，高品质，颜色管理，以及充分利用硬件加速等功能而建立的新一代开发平台。

其中体现了OpenCV这个项目的复杂性与对多平台适配性。

五、项目总结

本项目制作了一个基于OpenCV的隐私保护系统，实现了高斯模糊，像素化，图片遮罩三种模糊模式，对OpenCV库有了基本的了解与使用，加深了C++类与对象特性的使用和掌握。

六、引用

[OpenCV: Flags for video I/O](#)

[Visual Studio + OpenCV + OpenCV_contrib 源代码编译流程 - 知乎](#)

[C++计算机视觉库OpenCV在Visual Studio 2022的配置方法_visual studio 2022 opencv-CSDN博客](#)