

# 画像・映像情報処理 演習

## OpenCVによる画像処理

演習ページ

[https://fujis.github.io/ivip\\_enshu/](https://fujis.github.io/ivip_enshu/)

にアクセスし、演習課題を行うこと

演習レポート締め切り：12/22(月) 17:00

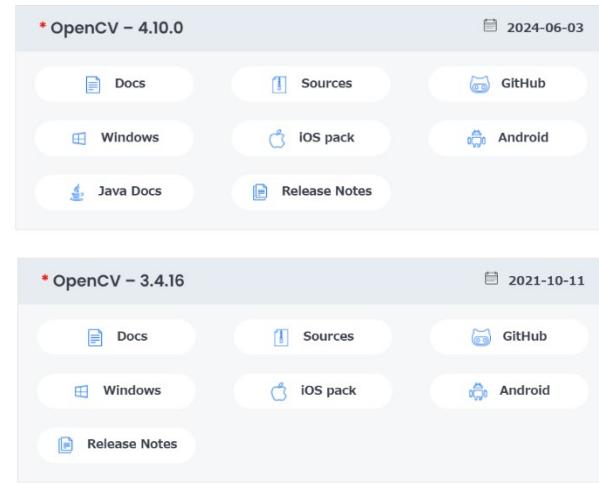
演習レポート提出場所：manaba



# 画像・映像情報処理 演習

## 演習で用いるOpenCVについての注意

- OpenCVには**Ver3系統(3.4.16)**と新しいバージョンである**Ver4系統(4.12.0)**がある。
- Ver3系統は2021年10月から更新なし
- Ver3とVer4のコード上の差異は主に**定数の名前**:  
例) `CV_THRESH_BINARY` ⇒ `THRESH_BINARY`  
(Ver3では両方とも, Ver4では後者のみが使える)
- 演習ページのコードは基本的にVer4系統で動作確認しているので自分のPCを用いる場合は注意

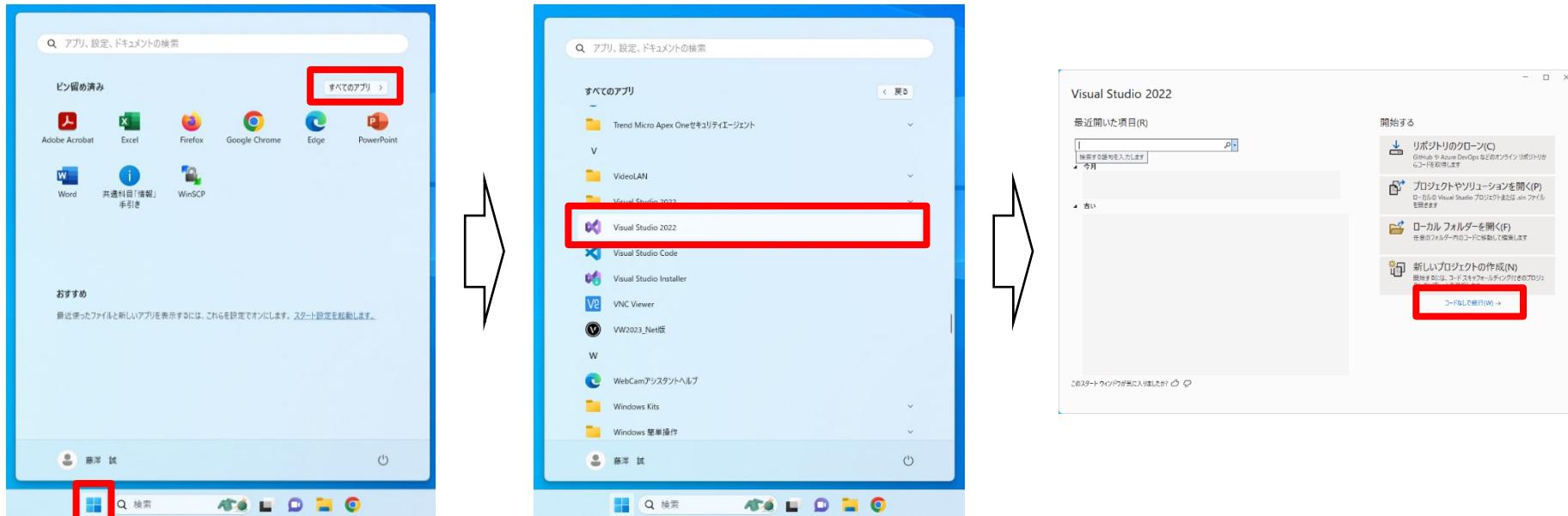


# 計算機室のWindows環境での演習1

全学計算機のWindows環境における演習方法を説明します。

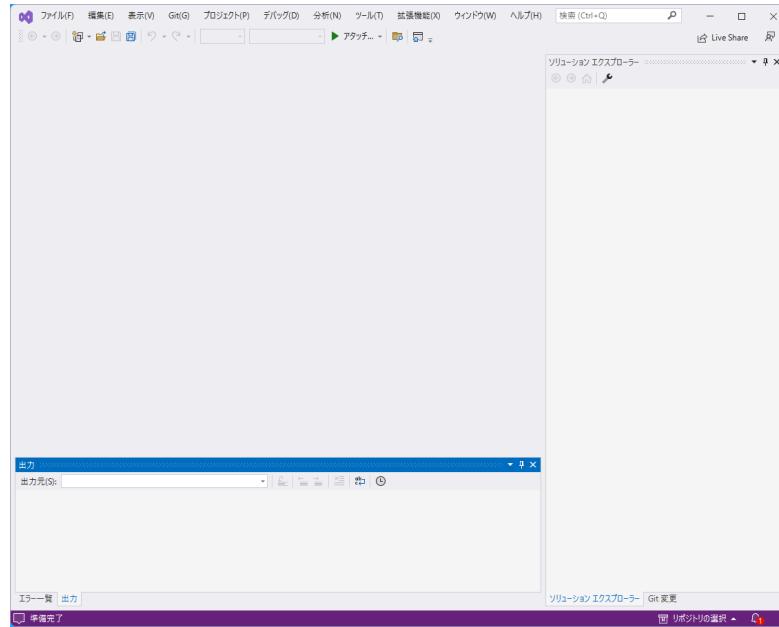
全学計算機には統合開発環境である**Visual Studio 2022**(以下VS2022)がインストールされています。以下はVS2022での演習の進め方の説明です。

1. 真ん中下のウインドウアイコンをクリックしてスタートメニューを出す。
2. 「すべてのアプリ」から「Visual Studio 2022」をクリックする(下の方にあります)。  
もしくは、検索ウインドウにVisual Studioなどと入力するのでもOK。
3. Visual Studioの初期ウインドウが出たら「コードなしで続行」をクリック  
(初期設定ウインドウが出たら適当に設定しておく)



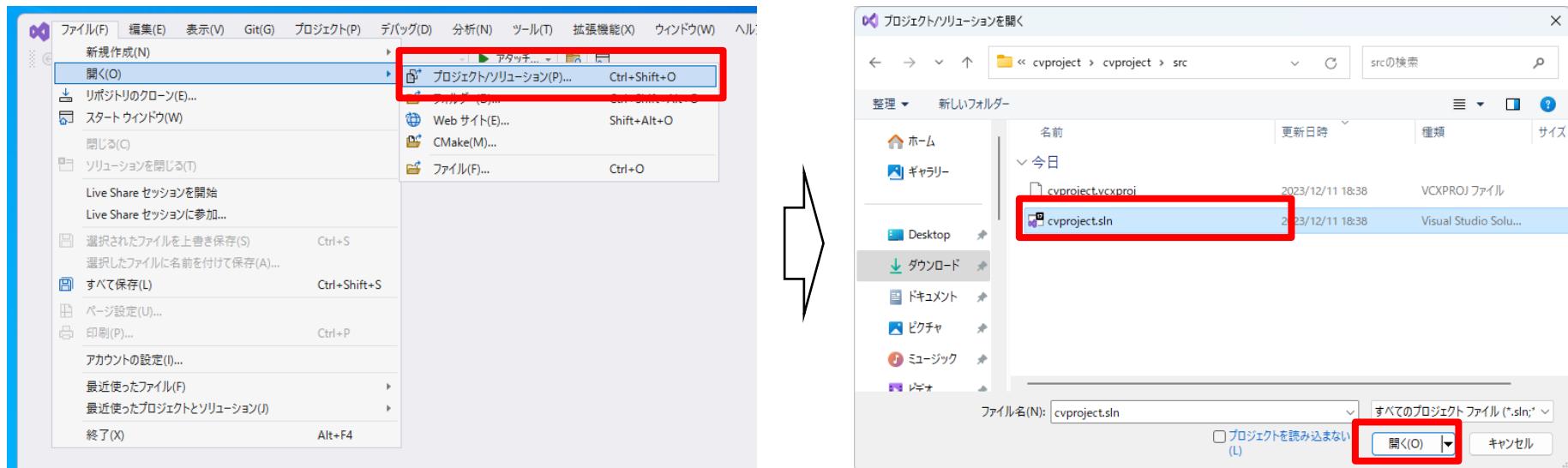
# 計算機室のWindows環境での演習2

4. 下のようなVisual Studioのウィンドウが出たらOK.
5. サンプルプロジェクトファイル(cvproject.zip)を以下のページから  
リモートデスクトップにダウンロードする.  
サンプル配布ページ：<https://fujis.github.io/ivip/sample/index.html>
6. ダウンロードしたzipファイルを右クリックして、「すべて展開」で解凍する(ダブル  
クリックだと解凍されず圧縮したまま中身を見るモードとなってしまうので注意).



# 計算機室のWindows環境での演習3

7. VS2022を開き、「ファイル」→「開く」→「プロジェクト/ソリューション」をクリックする。
8. 「プロジェクトを開く」ウィンドウが出るので、手順7で解凍したフォルダ内の cvproject/src/cvproject.sln ファイルを選択して、「開く」をクリックする。(セキュリティ警告が出たらそのままOKをクリック)。



# 計算機室のWindows環境での演習4

10. VS2022ウィンドウの右にある「ソリューションエクスプローラー」から, cvtestプロジェクトをダブルクリックして開き, cvtest.cppファイルを開く.
11. コードを確認したら、「ビルド」メニュー→「cvtestのビルド」を選択, 下の出力領域にビルドの様子が出るので, 「ビルド: 成功 1、失敗 0…」と出ればOK.

The screenshot shows the Microsoft Visual Studio 2022 interface. The title bar says "cvproject". The Solution Explorer on the right shows a single project named "cvtest" with one item "cvtest.cpp". The code editor on the left contains the following C++ code:

```
// ライブライアリの読み込み(VisualStudio用)
#include <iostream>
#include <opencv2/opencv.hpp>
using namespace cv; // 名前空間の設定

int main(void)
{
    Mat img, gray;
    img = imread("sample512.png", IMREAD_COLOR); // 画像ファイルの読み込み
    if(img.empty()){
        printf("Unable to load the image");
        return 1;
    }

    cvtColor(img, gray, COLOR_BGR2GRAY); // グレースケールに変換
    namedWindow("Display Image", WINDOW_AUTOSIZE); // ウィンドウ生成
    imshow("Display Image", gray); // 画像表示

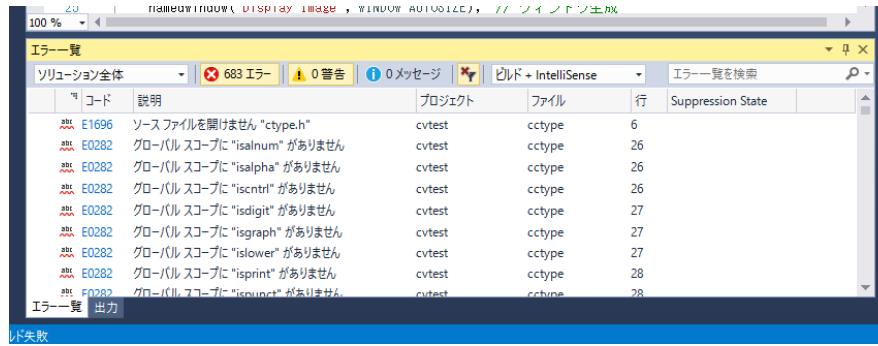
    // キー待ち状態に移行
    while(1)
    {
        int c = waitKey(10); // 引数はdelay
        switch((char)c)
        {
            case 'q': // qキーでプログラム終了
            {
                return 0;
            }
        }
    }
}
```

The Output window at the bottom shows the build log:

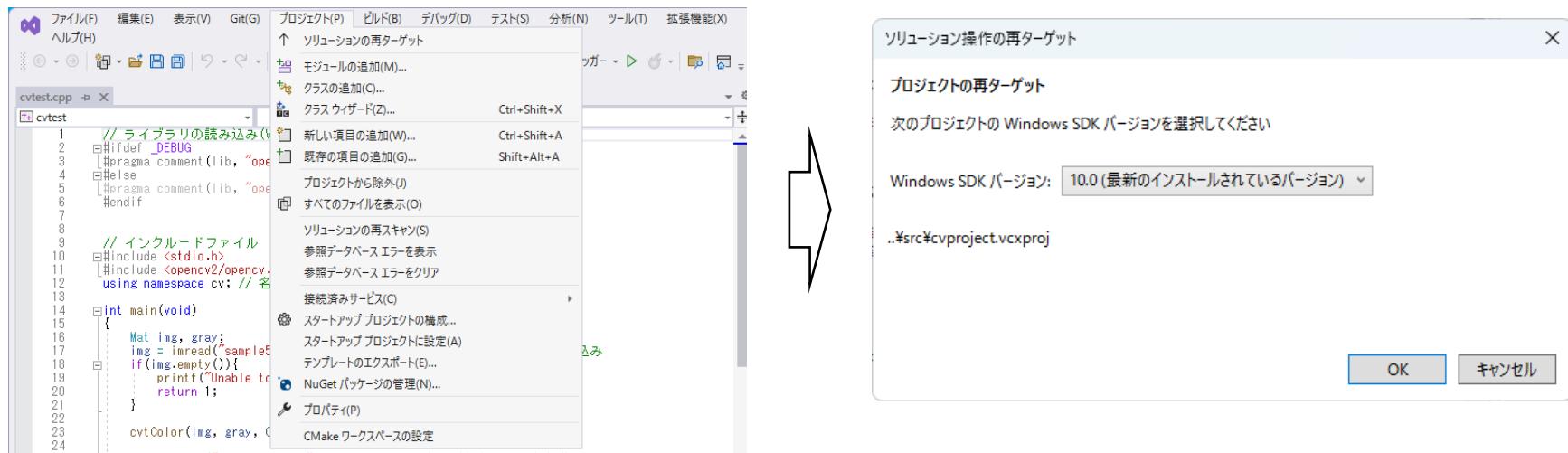
```
出力
出力元(S) ビルド
ビルドを開始しました...
-----[ビルド開始: プロジェクト: cvtest, 構成: Debug x86]-----
cvtest.vcxproj
cvtest.vcxproj -> W:\cho23\home\1\home01\Phone\fuji\ssws\makoto.fuji\Downloads\cvproject\cvproject\src\x86\Debug\cvte
===== ビルド: 成功 1、失敗 0、最新の状態 0、スキップ 0 =====
===== ビルド (は 8:40 PM)に開始され、07.587 秒かかりました =====
```

# 計算機室のWindows環境での演習5

[補足1] ビルドしたときに下図のようなエラーが出て、ビルドが失敗する場合は、



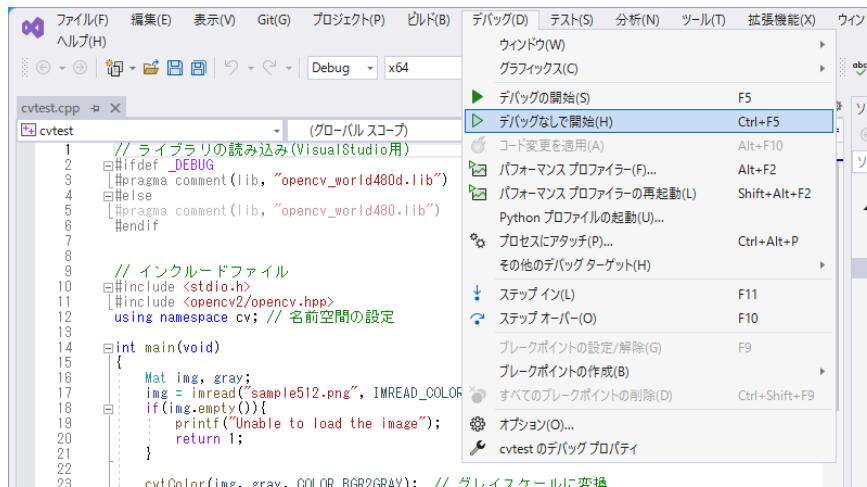
「プロジェクト」メニューから「ソリューションの再ターゲット」をクリックし、  
プロジェクトの再ターゲットウィンドウが出たら、そのままOKをクリック。



# 計算機室のWindows環境での演習6

12. ビルドが正常に終了したら、「デバッグ」メニュー→「デバッグなしで開始」をクリックする(「デバッガを使う場合は「デバッグの開始」でもOK).
13. 下図のようにグレースケール化された画像が表示されればOK.

演習ページ(<https://fujis.github.io/ivip/>)にアクセスして、  
演習問題を進めていこう!



```
cvtest.cpp  x (グローバルスコープ)
cvtest
1 // ライブライアリの読み込み(VisualStudio用)
2 #ifdef _DEBUG
3 #pragma comment(lib, "opencv_world480d.lib")
4 #else
5 #pragma comment(lib, "opencv_world480.lib")
6 #endif
7
8 // インクルードファイル
9 #include <stdio.h>
10 #include <opencv2/opencv.hpp>
11 using namespace cv; // 名前空間の設定
12
13 int main(void)
14 {
15     Mat img, gray;
16     img = imread("sample512.png", IMREAD_COLOR);
17     if(img.empty()){
18         printf("Unable to load the image");
19         return 1;
20     }
21
22     cvtColor(img, gray, CV_RGB2GRAY); // ダミィケート用に変換
23 }
```

Debug x64

デバッグ(D) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W)

デバッグの開始(S) F5

デバッグなしで開始(H) Ctrl+F5

コード変更を適用(A) Alt+F10

パフォーマンス プロファイル(F)... Alt+F2

パフォーマンス プロファイルの再起動(L) Shift+Alt+F2

Python プロファイルの起動(U)...

プロセスアタッチ(P)...

その他のデバッグターゲット(H)

ステップイン(L) F11

ステップオーバー(O) F10

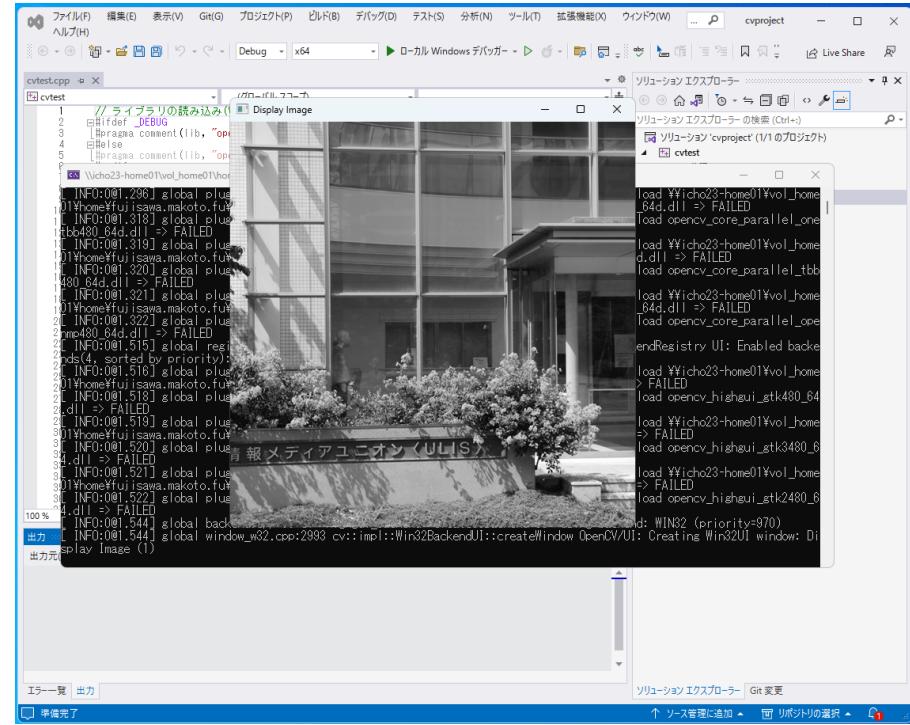
ブレークポイントの設定/解除(G) F9

ブレークポイントの作成(B)

すべてのブレークポイントの削除(D) Ctrl+Shift+F9

オプション(O)...

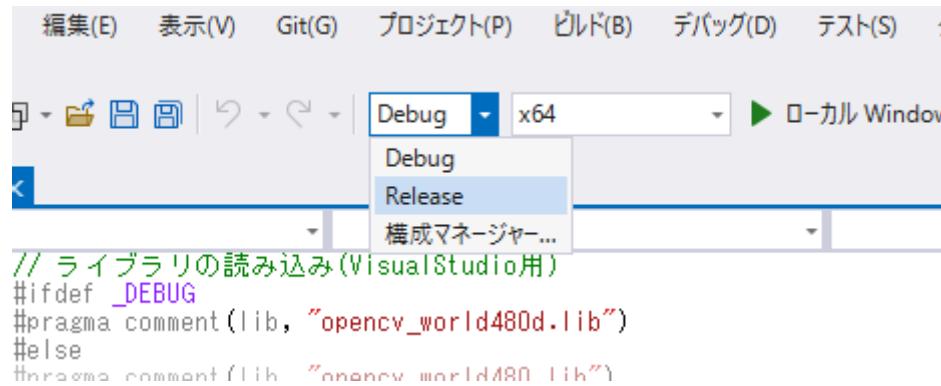
cvtest のデバッグ プロパティ



# 計算機室のWindows環境での演習7

[補足2] 演習課題2で処理が遅い(数秒ウィンドウが固まるなど)

ツールバーで、「Debug」とあるところの右の▼をクリックして、「Release」を選択し、ビルドし直そう。



- 「Debug」モード: デバッグ用にプログラムを最適化なしでそのままコンパイル
- 「Release」モード: コンパイル時にプログラムを最適化する。最適化で高速になるが、デバッグで1行1行実行することはできない(最後の製品リリースのためのモード)