

# VS2019+OpenCV2 環境セットアップ

---

# 環境のセットアップ (※演習室PCには、「1.相当」および「2.」が設定済み)

2

## 1. Visual Studio Community 2019(以下VS)のインストール

- 必要なもの: Windows PC、インターネットへの接続環境、Microsoftアカウント
- ※システム要件: Windows 7 SP1以上、RAM 2GB以上、HDDに20GB以上の空き容量
- <https://www.visualstudio.com/ja/vs/older-downloads/> から2019を[ダウンロード]  
(要Microsoftアカウント(学校のOffice365のアカウントでOK))。

## 2. OpenCV2.4のインストールと環境変数の設定

- <https://opencv.org/releases/> から、  
[OpenCV - 2.4.13] OpenCV2.4.13.???.exe (Windows)をダウンロードする。
- ダウンロードした「opencv-2.4.13.???.exe」を実行し、インストール。  
ここでは、展開先を「C:\openCV2」と指定したものとする。  
(※別の場所を指定した場合は、以下のパスを適宜読み替えること。)
- 環境変数の設定(いわゆる「パスを通す」作業)
  - 「エクスプローラから[PC]の右クリック→プロパティ」(または[Win]+[Pause])で開いた画面で、左の[システムの詳細設定]をクリック → 「システムのプロパティ」が開く。
  - [詳細設定]タブの[環境変数]ボタンをクリック → 「環境変数」が開く。
  - 下の「システム環境変数」内にある、変数「Path」をダブルクリックし、「変数値」の一番後ろに「;C:\openCV2\opencv\build\x86\vc12\bin」を追記する。(※区切り文字が「;」)
  - [OK]を何度かクリックして設定を反映。

※パスの **vc12** の部分は、vc14 等、実際に存在するパスに読み替えて下さい。

この説明には、未検証の部分があります。自分のPCにインストールする場合は、OpenCV2.4系列が使えるように各自で調べて設定してみてください。

# VS2019上での新規プロジェクトの作成方法

3

1. メニューから「ファイル > 新規作成 > プロジェクト」をクリックし、  
「VisualC++ > Win32コンソールアプリケーション」で任意のプロジェクト名を入れて[次へ]
2. 「空のプロジェクト」にチェックを入れて[完了]
3. プロジェクトの設定(メニューの「プロジェクト>プロパティ」)から、以下を設定
  - ① include: 「VC++ディレクトリ」の「インクルードディレクトリ」の先頭に、  
「C:¥openCV2¥opencv¥build¥include;」を追記する。
  - ② LIB: 「VC++ディレクトリ」の「ライブラリディレクトリ」の先頭に、  
「C:¥openCV2¥opencv¥build¥x86¥vc12¥lib;」を追記する。
  - ③ リンカーへの設定: 「リンカー > 入力」の「追加の依存ファイル」の先頭に、  
「opencv\_core2413d.lib; opencv\_highgui2413d.lib; opencv\_imgproc2413d.lib;」を追記。
  - ④ 以上の①～③の設定は、構成(Debug/Release)毎に行う必要があるので、切り替えて両方の構成について設定しておくこと。  
※リンカーへの設定部分は、Debugの場合は「d」付き、Releaseの場合は「d」無しで指定する。
4. ソリューションエクスプローラの「ソースファイル」を右クリックして、  
「追加 > 新しい項目」で「C++ファイル」を選び、任意の名前をつけたcppファイルを新規作成
5. 新規作成したcppファイル内の先頭で、  
通常は右記のように2つのファイルを #include し、  
main() 関数内にコーディングをしていく形となる。

※パスの vc12 の部分は、vc14 等、  
実際に存在するパスに読み替えて下さい。

```
#include <stdio.h>
#include <opencv/highgui.h>
void main() {
    ...
}
```

※この状態で、メニューの「プロジェクト > テンプレートのエクスポート」からプロジェクトテンプレートを作成しておけば、  
3.～5. の作業を毎回行う手間が省けるのでオススメ。(「新しいプロジェクトの作成」でテンプレートを選択)

# とりあえずコンパイルが通るかチェック...

4

```
// Example 05-09: OpenCV check
#include <stdio.h>
#include <opencv/highgui.h>

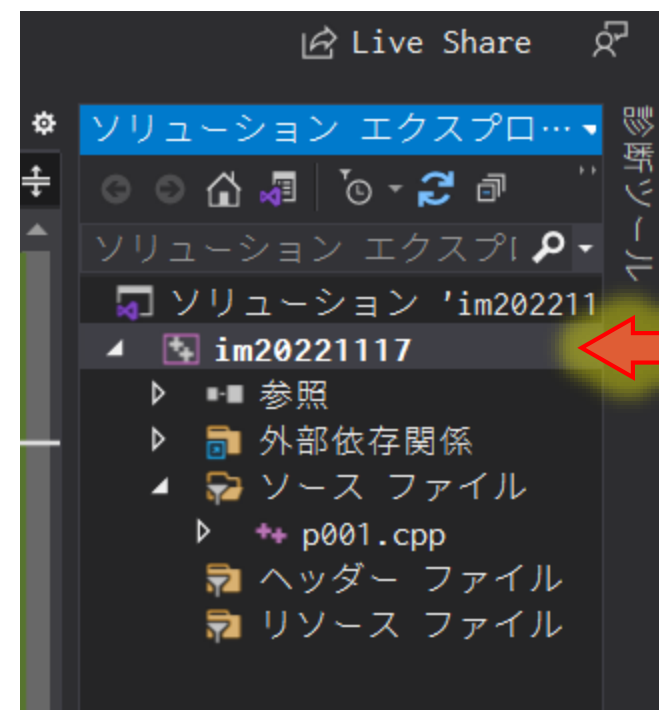
int main(void)
{
    IplImage* img;
    cvNamedWindow("Test Window");
    img = cvLoadImage("C:/.../mono/LENNA.bmp", CV_LOAD_IMAGE_UNCHANGED);

    if (img == NULL) {
        printf("Load Error.¥n");
        cvWaitKey(0);
    }
    else {
        cvShowImage("Test Window", img);
        printf("Channels = %d¥ndepth = %d¥nwidth = %d, height = %d¥n",
            , img->nChannels, img->depth, img->width, img->height);
        cvWaitKey(0);

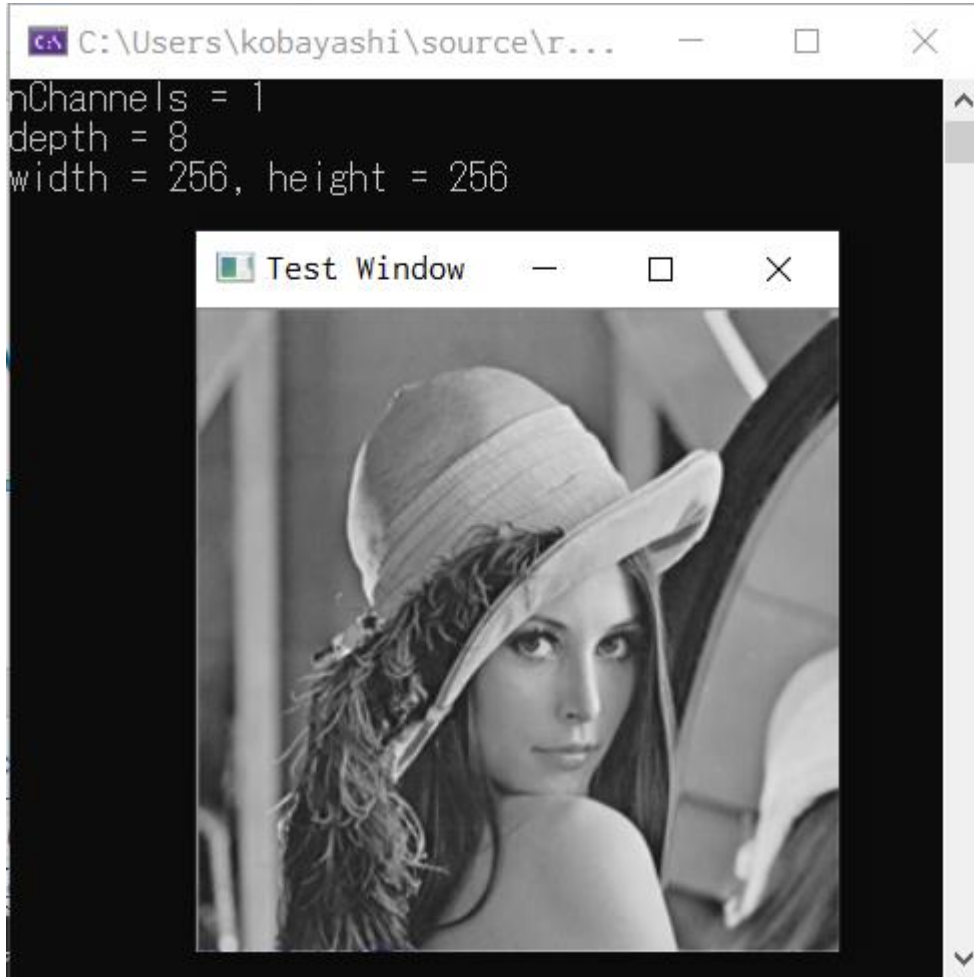
        cvDestroyAllWindows();
        cvReleaseImage(&img);
    }
    return 0;
}
```

画像ファイルの絶対パス  
or  
実行時のカレントディレクトリ  
からの相対パスで指定

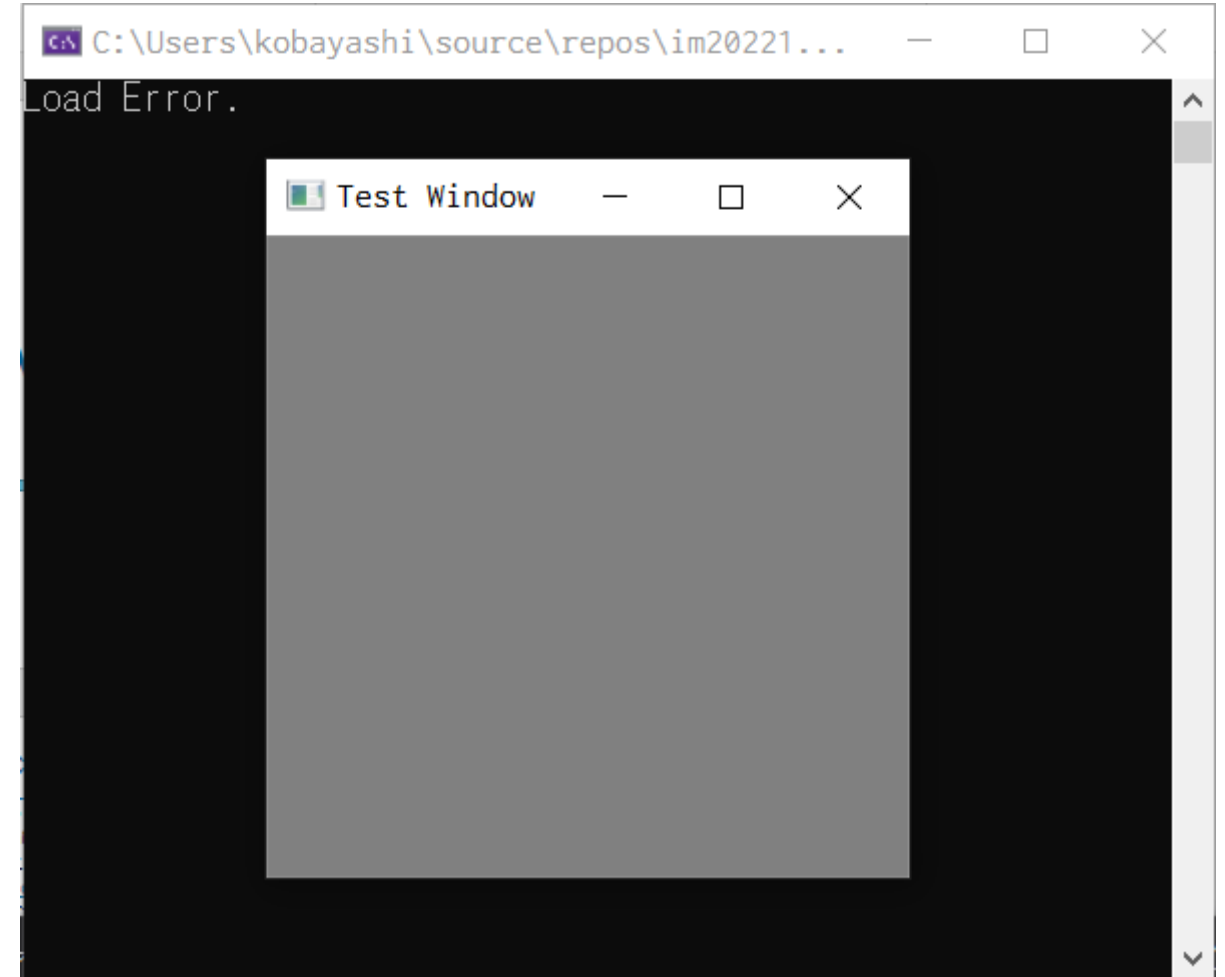
※実行時のカレントディレクトリは、  
ソリューションエクスプローラで  
プロジェクトを右クリック  
➤ エクスプローラーでフォルダを開く  
として確認できる。  
(例えばここに画像ファイルを置いておく)



## ●成功時



## ●失敗時 (画像ファイルのパスが不適切)



そもそもコンパイルに失敗する場合はエラーメッセージをよく確認しましょう...