

Uživatelská příručka

V kapitole jsou rozepsány jednotlivé kroky, které bychom měli udělat, abychom správně spustili program.

Tento návod poskytuje postup pro instalaci knihovny OpenCV jazyka C++ v prostředí Visual Studio. Postup budeme ilustrovat na počítači se systémem Windows, ale je také použitelný s mírnými úpravami pro macOS .

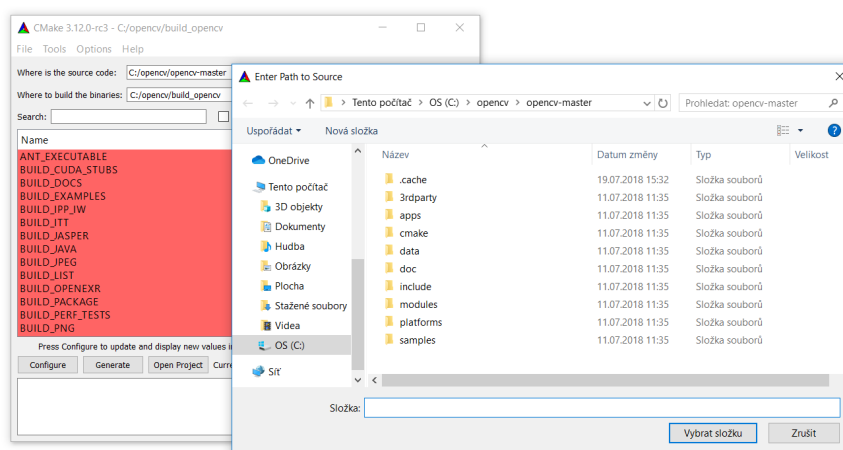
Vývojové prostředí

Jako vývojové prostředí použijeme Visual Studio. Na oficiální stránce <https://visualstudio.microsoft.com/cs/vs/> můžeme stáhnout *installer*, pomocí kterého stáhneme a nainstalujeme celé Visual Studio prostředí (v práci je použita verze roku 2015).

Instalace OpenCV knihovny

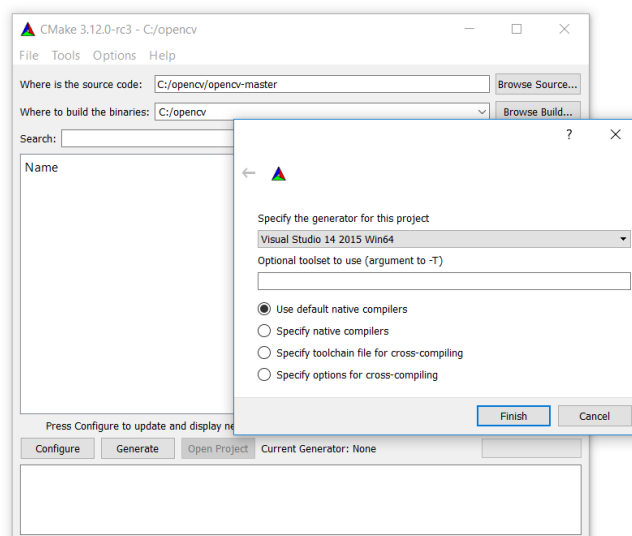
Stáhneme si OpenCV z oficiální stránky <https://opencv.org/releases/>. V práci je použita knihovna verze 3.4.2 stažená ze stránky <https://github.com/opencv/opencv>. Nejlépe je umístit knihovnu v kořenovém adresáři, např. C:\opencv 3.4.2.

Navíc potřebujeme instalovat modul *contrib* (stačí zatím jen extrahovat vnitřní obsah) obsahující potřebné soubory pro detekci a rozpoznání obličeje. Modul umístíme stejně v kořenovém adresáři. Před nainstalováním modulu je potřeba také instalovat *cmake*, který můžeme stáhnout z oficiální stránky <https://cmake.org/download/>. Spojíme (nainstalujeme) extra modul contrib s knihovnou OpenCV. K tomu použijeme již stažený a nainstalovaný program *cmake*. Spustíme ho a vybereme si v položce „Where is the source code“ adresář s knihovnou OpenCV. V položce „Where to build the binaries“ si vybereme adresář kam chceme uložit vytvořenou knihovnu (viz obrázek 1) např. „opencvtest“.



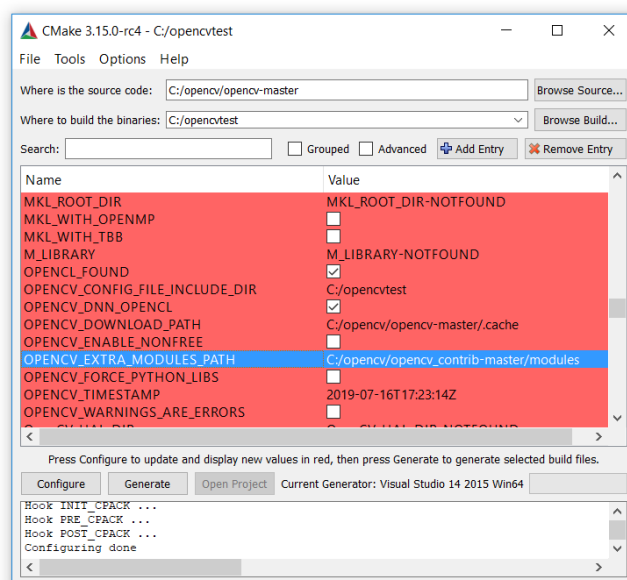
Obrázek 1: Instalace pomocí *cmake*

Pokud již máme vybrané adresáře, provedeme konfiguraci zmáčknutím tlačítka *Configure*. V nově vytvořeném okénku si vybereme verze Visual Studio pro který chceme instalovat knihovny (obrázek 2).



Obrázek 2: Konfigurace

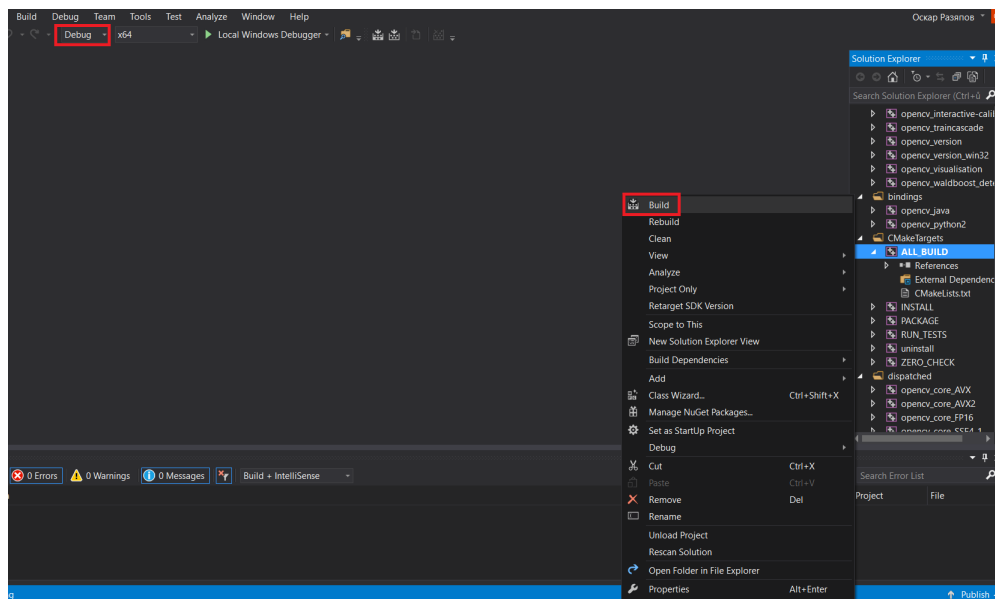
Výsledkem konfigurace je seznam knihoven. Ze seznamu si můžeme vybrat jednotlivé knihovny, které chceme instalovat. Zajímá nás položka „OPENCV_EXTRA_MODULES_PATH“, kde musíme ukázat cestu k jednotlivým modulům OpenCV contrib (viz obrázek 3). Je to cesta, kam jsme extrahovali modul *contrib*. Znovu si zmáčkneme tlačítko *configure* a následně tlačítko *Generate*.



Obrázek 3: Instalace extra modulů

Pokud konfigurace a generování proběhly úspěšně, otevřeme si složku „*opencvtest*“ a spustíme soubor *OpenCV.sln* pomocí Visual Studio.

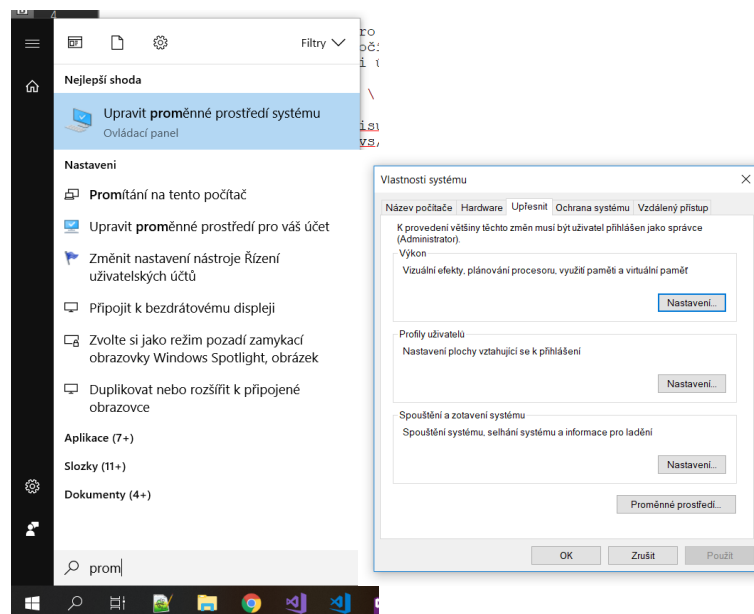
Ověříme si, jestli jsme v režimu *Debug* (viz obrázek 4 levý horní roh). V „*Solution Explorer*“ pro položku *ALL_BUILD* sestrojíme projekt zmáčknutím tlačítka *build* (viz obrázek 4), což bude trvat kolem 20 minut. Změníme režim *Debug* na *Release* a sestrojíme projekt znovu. Analogicky musíme sestavit projekt ve dvou režimech pro položku



Obrázek 4: Build

INSTALL.

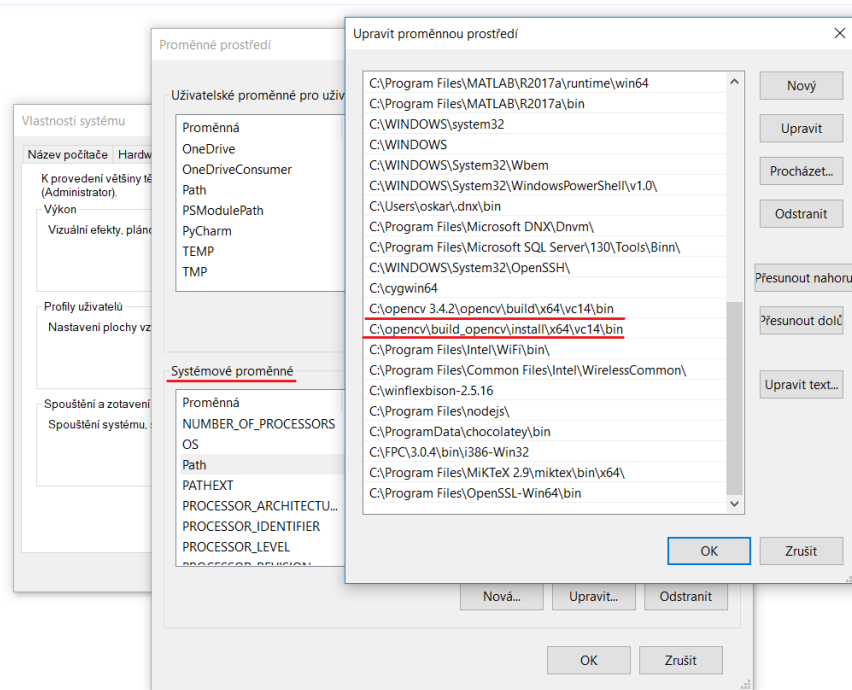
Pokud vše proběhlo úspěšně, napíšeme v *proměnném prostředí* cesty k jednotlivým souborům viz obrázek 5.



Obrázek 5: Proměnné prostředí

A to tak, že si v *systémových proměnných* vybereme proměnnou *path* viz následující obrázek 6. V proměnné napíšeme cesty ke složkám *bin* modulu contrib a samotné knihovně OpenCV.

Vytvořili jsme všechny potřebné soubory, tedy můžeme se pokusit napsat náš první program s použitím knihovny OpenCV. Vytvoříme si nový C++ projekt a vložíme do c++ souboru nějaký kus kódu (viz obrázek 7).



Obrázek 6: Proměnné prostředí

```

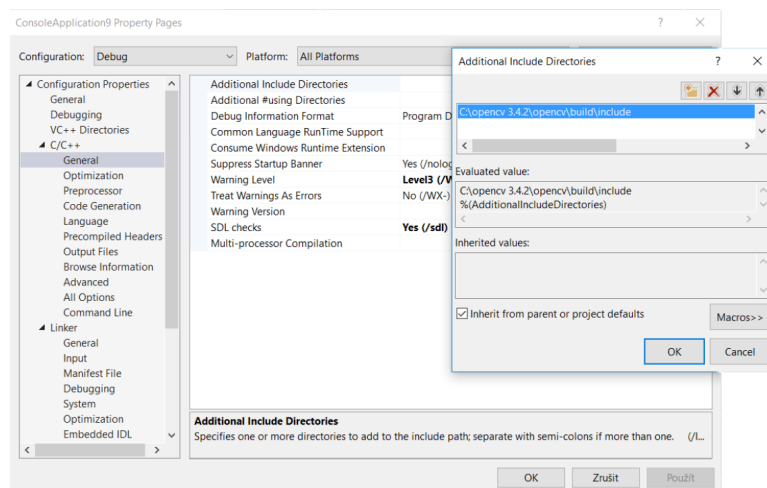
1  #include<opencv2/opencv.hpp>
2  #include<iostream>
3
4  using namespace std;
5  using namespace cv;
6  int main()
7  {
8      Mat img = imread("lena.jpg");
9      namedWindow("image", WINDOW_NORMAL);
10     imshow("image", img);
11     waitKey(0);
12     return 0;
13 }

```

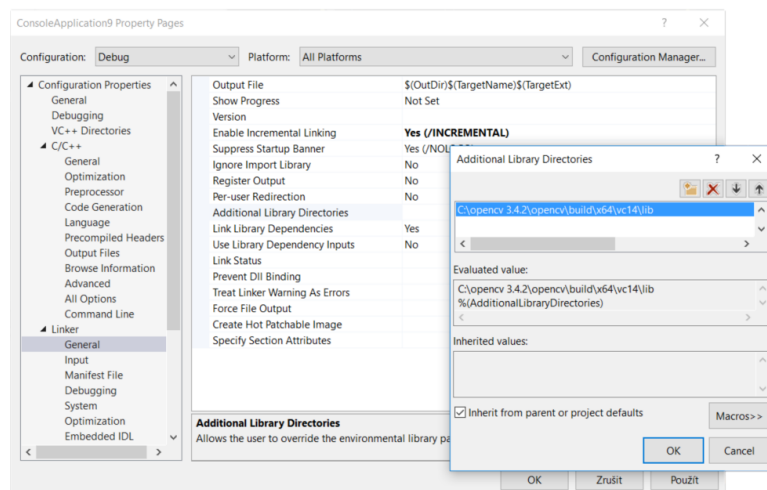
Obrázek 7: Testový kód

Ukážeme cestu ke složce *include* v položce „Additional Include Directories“ ve vlastnosti *General* pro *C/C++* (viz obrázek 8) a ke složce *lib* pro *Linker* (viz obrázek 9) v nastavení projektu.

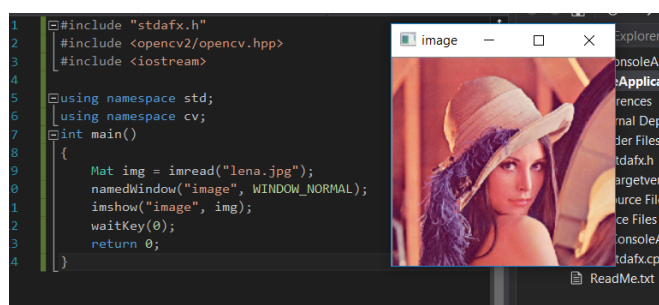
V položce *Additional Dependencies* ve vlastnosti *Input* ukážeme na soubor *opencv_world342d.lib*, který má cestu *C:\opencv 3.4.2\opencv\build\x64\vc14\lib*. Nakonec můžeme spustit testovací soubor (obrázek 10). Pokud všechno funguje, můžeme spustit i náš program.



Obrázek 8: C/C++



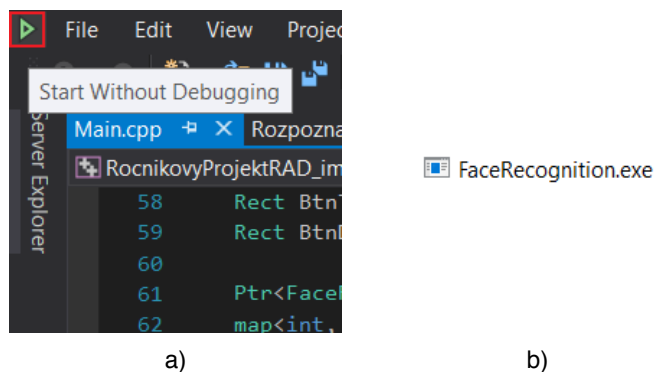
Obrázek 9: Linker



Obrázek 10: Lena, testovací kód

Příručka uživatele

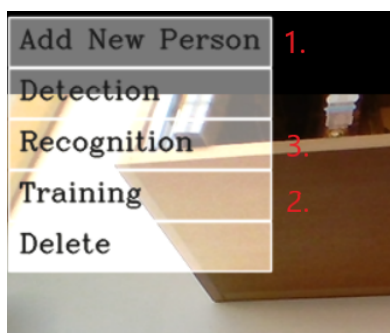
V tomto odstavci se podíváme jak spustit aplikaci, jaké příslušenství je potřeba použít a v jakém pořadí spouštět jednotlivé režimy programu. Spustit aplikaci můžeme pomocí prostředí Visual Studio pomocí tlačítka „Start without debugging“, anebo pomocí *exe* souboru vygenerovaného prostředím Visual Studio (viz obrázek 11).



Obrázek 11: Spuštění aplikace

Pro načtení snímku musíme použít kameru počítače.

Jakmile jsme spustili aplikaci, můžeme začít s ní pracovat. Výchozím režimem je detekce. V aplikaci existuje možnost načtení již připraveného modelu, ale předpokládáme, že spouštíme poprvé, a model zatím neexistuje.



Obrázek 12: Režimy



Obrázek 13: Přidání nového uživatele

Nejdříve přidáme nového uživatele zmáčknutím tlačítka „*Add New Person*“ (viz obrázek). V nově vytvořeném okně vyplníme jeho jméno a uložíme (pro rozpoznávací metodu FisherFace je potřeba přidat minimálně dva uživatele). Druhým krokem musíme natrénovat model zmáčknutím tlačítka „*Training*“. Jakmile model se natrénuje, můžeme začít samotný proces rozpoznávání zmáčknutím tlačítka „*Recognition*“. Pro vrácení do režimu detekce musíme zmáčknout tlačítko „*Detection*“. Pokud chceme smazat model (např. pokud je potřeba vytvořit nový), musíme zmáčknout tlačítko „*Delete*“. Pro přesnější rozpoznání je potřeba udělat hodně trénovacích snímků, čím je jich více, tím přesnější je rozpoznání.

Pokud model již existuje, můžeme rovnou spouštět rozpoznání.