

# Computer Vision

Catalin Stoean

[catalin.stoean@inf.ucv.ro](mailto:catalin.stoean@inf.ucv.ro)

<http://inf.ucv.ro/~cstoean>

# Informatii generale

- Pagina web a cursului
  - <http://inf.ucv.ro/~cstoean>
- Nota
  - Se obtine in urma realizarii unor proiecte.
  - Teme de proiect vor fi enuntate pe parcursul cursului si cuprind:
    - in principal, procesari de imagini
    - dar si extrageri de informatii din imagini (masuratori, calcule etc) folosind sau nu invatare automata
    - fiecare tema are asociat un numar de puncte si are un termen limita.
  - Studentii care nu au realizat (suficiente) proiecte vor da un examen scris.

# Bibliografie 1/2

- Robert Laganière, OpenCV 2 Computer Vision Application Programming Cookbook, Packt Publishing, Birmingham, UK, 2011.
- Gary R. Bradski, Vadim Pisarevsky, Jean-Yves Bouguet, Open Source Computer Vision Library, Springer, 1st ed. 2006.
- Tutoriale OpenCV:
  - <http://docs.opencv.org/3.0-beta/doc/tutorials/tutorials.html>
  - [http://docs.opencv.org/trunk/d9/df8/tutorial\\_root.html](http://docs.opencv.org/trunk/d9/df8/tutorial_root.html)
  - <http://opencvexamples.blogspot.com/p/learning-opencv-functions-step-by-step.html>
  - <https://www.learnopencv.com/>

# Bibliografie 2/2

- D.A. Forsyth, Jean Ponce, Computer Vision - A Modern Approach (2nd Edition), Jean Ponce, 2011.
- Gary Bradski and Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008.

<http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/OReilly%20Learning%20OpenCV.pdf> .

- Peter Corke, Robotics, Vision & Control, Springer 2011.

# Continutul cursului

- Computer Vision cu OpenCV
- Încărcarea, afisarea si salvarea imaginilor
  - Crearea unei aplicatii GUI folosind QT pentru procesare de imagini
- Accesarea valorilor pentru pixeli din cadrul unei imagini
- Definirea de regiuni de interes in imagini
- Procesare de imagini cu clase
- Histograma unei imagini
- Detectarea continutului unei imagini folosind histograma
- Transformarea imaginilor cu operatii morfologice
- Extragerea de linii, contururi si componente
- Detectarea de puncte de interes
- Detectarea de obiecte dupa culoare
- Procesarea de secvente video



# Ce este Computer Vision?

- Transformarea datelor de la o camera foto sau video intr-o reprezentare noua sau chiar in decizii.
  - Camera poate fi montata pe o masina
  - Un laser poate indica faptul ca te apropii mai aproape de 1 metru de un obiect
- Aceste observatii se fac in general simplu, intuitiv de catre om.

# Ce este Computer Vision?

- Este banal pentru un om sa identifice o masina intr-o poza
  - S-a focusat in imagine doar pe acea regiune
  - A vazut deja suficiente masini anterior ca sa aiba o reprezentare clara asupra lor



**Dar o masina vede:**

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	74	65
20	41	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

# De ce Computer Vision?

- Imagini (si filme) sunt pretutindeni
- Aplicatii utile care sa extraga informatii din imagini:
  - Identificarea automata a numarului de la masina
  - Identificarea feței
  - Identificarea unor regiuni de interes intr-o imagine
  - Procesarea filmelor
- Exista deja multe soft-uri care fac astfel de procesari.
  - Dar cand avem nevoie sa procesam zeci, sute sau chiar mii de imagini/filme, este esential sa ne cream propria aplicatie care sa realizeze aceste lucruri.



# OpenCV

- Reprezinta o librerie gratuita (Open) pentru dezvoltare si cercetarea in Computer Vision
  - <http://sourceforge.net/projects/opencvlibrary/>
- Contine peste 2500 de algoritmi
- Functioneaza sub Windows, Linux, Android, Mac OS.
- Sunt dezvoltate interfete pentru limbaje precum: C++, C, Java, Python, Matlab.
- Oferă infrastructura pentru Computer Vision pentru a construi rapid aplicatii sofisticate
- Cursul ne va ajuta sa alegem ce algoritm sa utilizam pentru scopul avut si in ce moment

# OpenCV

- Are peste 14 mil de descarcari
- Printre utilizatori se numara si companii mari precum Google, Yahoo, Microsoft, Intel, IBM, Sony etc.
- Printre algoritmi continuti sunt unii dedicati pentru:
  - Recunoasterea feței
  - Identificarea de obiecte
  - Urmărirea obiectelor in miscare
  - Gasirea de imagini similare intr-o baza de date cu imagini
  - Eliminarea ochilor rosii din poze
  - Urmărirea ochilor in miscare
- Functioneaza sub diferite medii de dezvoltare integrate (IDE) pentru C++.

# Descarcarea OpenCV

- **Presupunem instalat deja Visual Studio 2012 (sau 2013, 2015, 2017...)**
  - Gratuit prin contul personal de la DreamSpark  
<http://e5.onthehub.com/d.ashx?s=bc81baqimt>
- Cea mai recenta versiune de OpenCV se descarca de la <http://sourceforge.net/projects/opencvlibrary/>
- Arhiva se extrage intr-un folder, de exemplu in D:\OpenCV2



OpenCV | Free Science & ...

sourceforge.net/projects/opencvlibrary/

sourceforge Search Browse Enterprise Blog Help Jobs

SOLUTION CENTERS Go Parallel Smarter IT Resources Newsletters

VISUAL ANALYTICS See your data for all it's worth. LIVE DEMO: TRY IT NOW sas THE POWER TO KNOW

Home / Browse / Science & Engineering / Human Machine Interfaces / OpenCV

OpenCV Open Source Computer Vision Library Brought to you by: akamaev, ashishkov, etalanin, ganbradski, and 4 others

Summary Files Reviews Support Wiki Donate

★ 4.8 Stars (128)  
↓ 37,042 Downloads (This Week)  
📅 Last Update: 2014-02-05

Download opencv-2.4.8.exe

🐦 Tweet 🍷 +1 177 📘 Like 357 🌐 Browse All Files

Description

The Open Source Computer Vision Library has >2500 algorithms, extensive documentation and sample code for real-time computer vision. It works on Windows, Linux, Mac OS X, Android and iOS.

Homepage: [opencv.org](http://opencv.org)  
Q&A forum: [answers.opencv.org](http://answers.opencv.org)  
Documentation: [docs.opencv.org](http://docs.opencv.org)

Please pay special attention to our tutorials! <http://docs.opencv.org/doc/tutorials/tutorials.html>

# Instalarea OpenCV

- Dupa descarcare, biblioteca se instaleaza
- Pentru aceasta, descarcati CMake (open-source)

CMake

www.cmake.org

About Resources Developer Resources Download

**CMake, ParaView, & VTK Courses in Carrboro, NC**  
Oct 20: Scalable Vis with VTK, ParaView, & Python  
Oct 21: Project Lifecycle Management with CMake  
[Details & Registration](#)

**Kitware** offers robust, cross-platform software development solutions. Find out how we can help your team efficiently manage the build, test, and package process for your software project.

[Contact Us](#) or [Learn More](#)

**CMake News & Blogs**

- 09.24.2014 Kitware Announces Development of Real-Time Image Guidance to Impr...
- 09.24.2014 Fall New Hires & Promotions at Kitware
- 09.04.2014 Kitware Announces Ice Bucket Challenge Network Website
- 10.02.2014 New Open Source Policies from the U.S. Government
- 09.15.2014 SciPy 2014 In Review: Posters and Presentations
- 09.11.2014 CMake 3.0.2 available for download

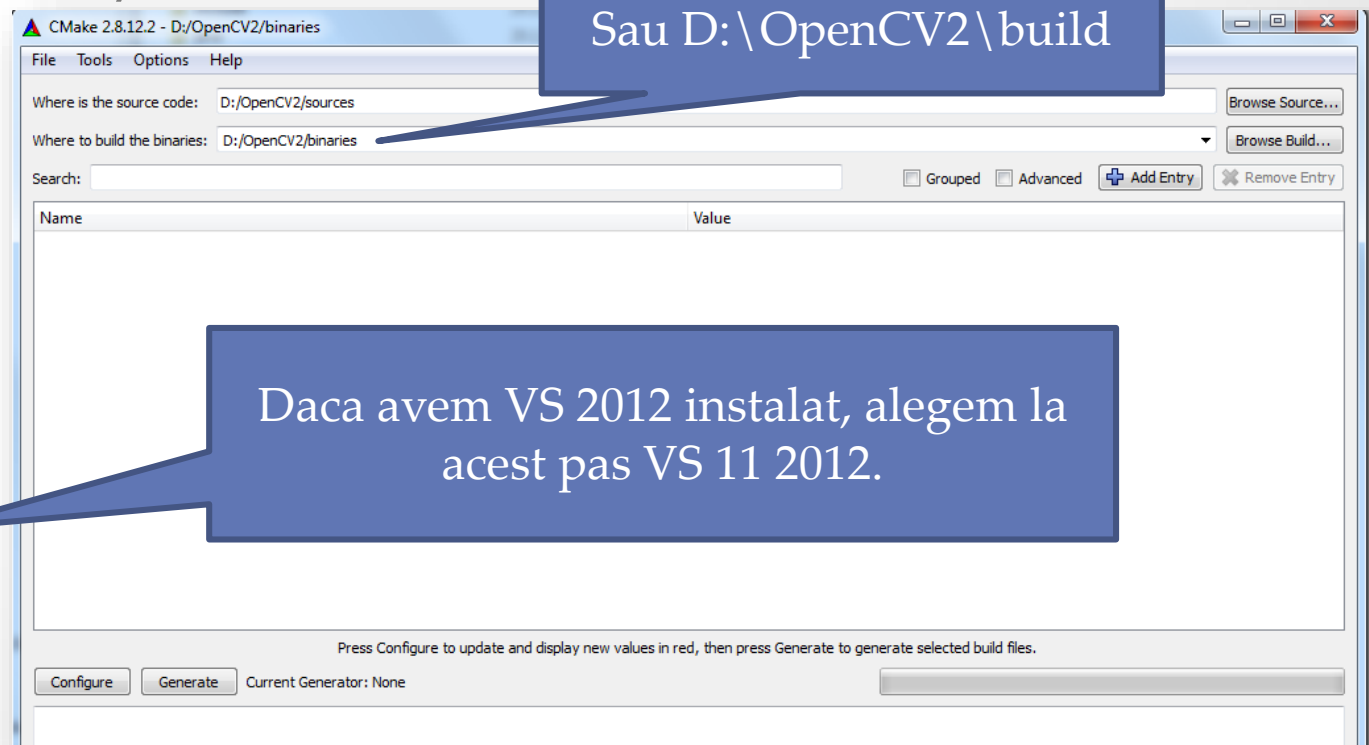
Welcome to CMake, the cross-platform, open-source build system. CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.

**Download the Latest Version of CMake**  
[Download CMake Version 3.0.2](#)

# Instalarea OpenCV

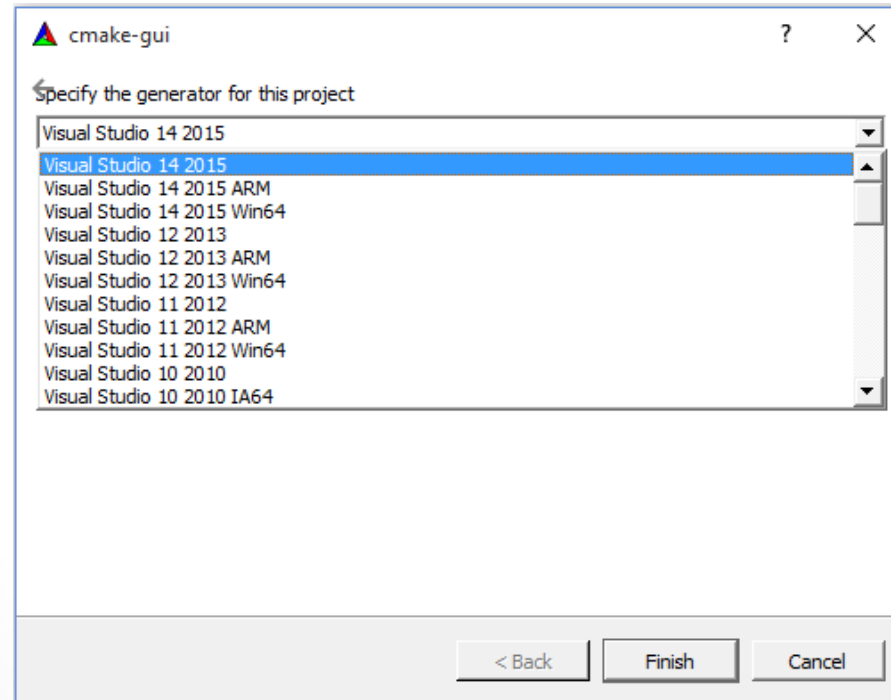
- Dupa descarcare, biblioteca se instaleaza
- Pentru aceasta, descarcati si instalati CMake (open-source)

- Se pun:
  - Calea catre sursa
  - Calea catre biblioteca compilata
- Configure
  - Se alege Visual Studio 10
- Generate



# Alegerea generatorului CMake

- Generatorul se selectează în funcție de versiunea de Visual Studio pe care o avem și de ce platformă folosim la Configuration Platform (Win 32 sau X64).



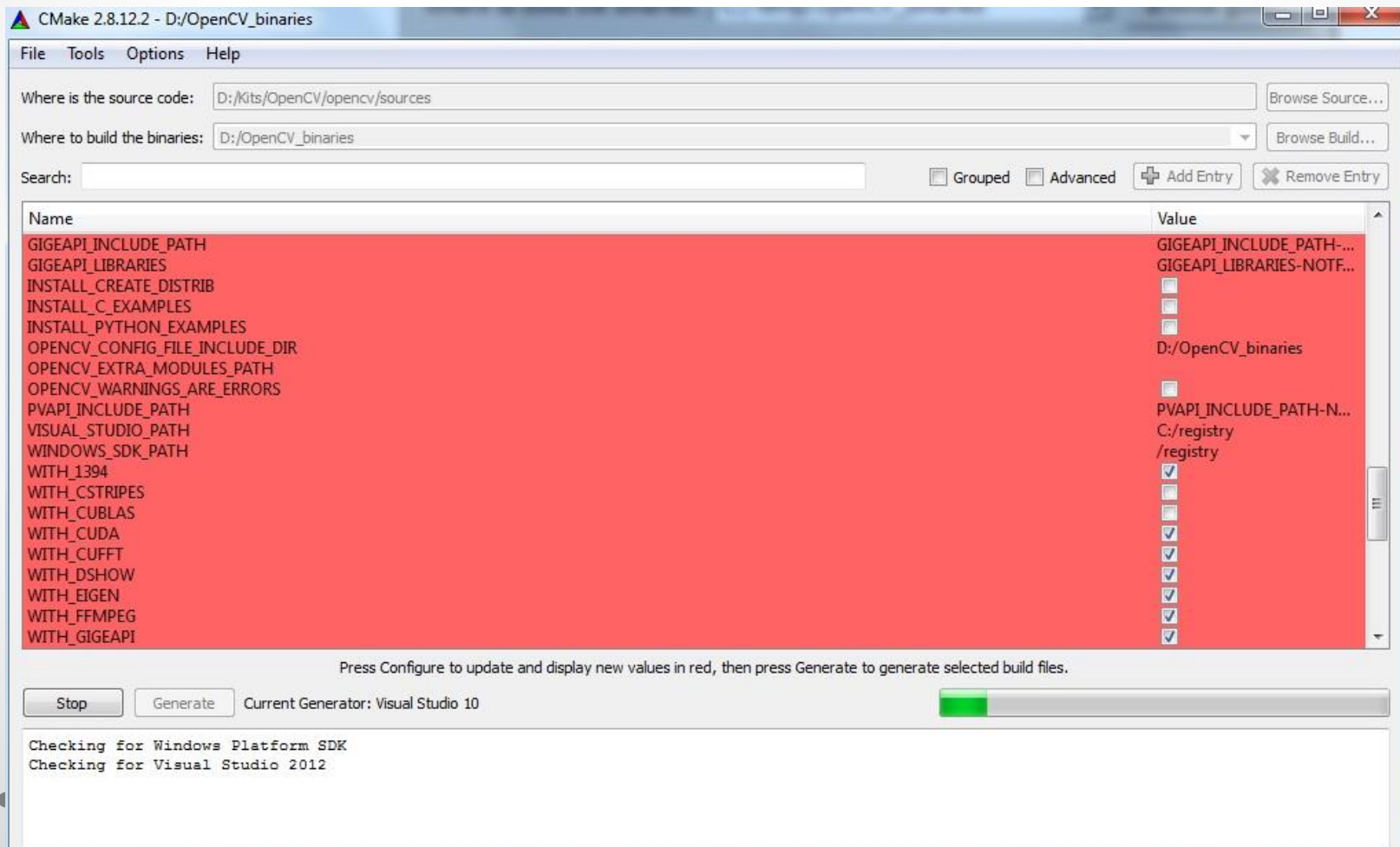
# Instalarea OpenCV

- Se obtine o fereastră ca mai jos.
- Apasam din nou **Configure**.



# Instalarea OpenCV

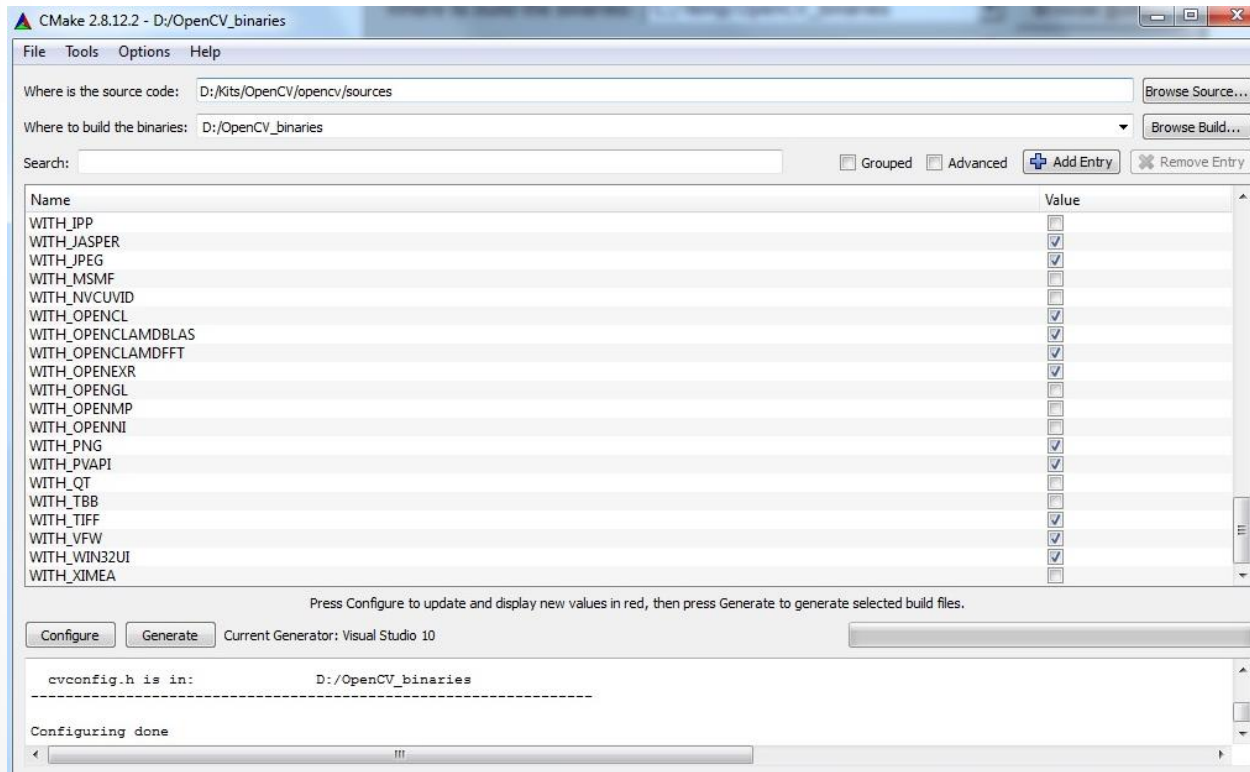
- Dupa ce a fost apasat din nou **Configure**





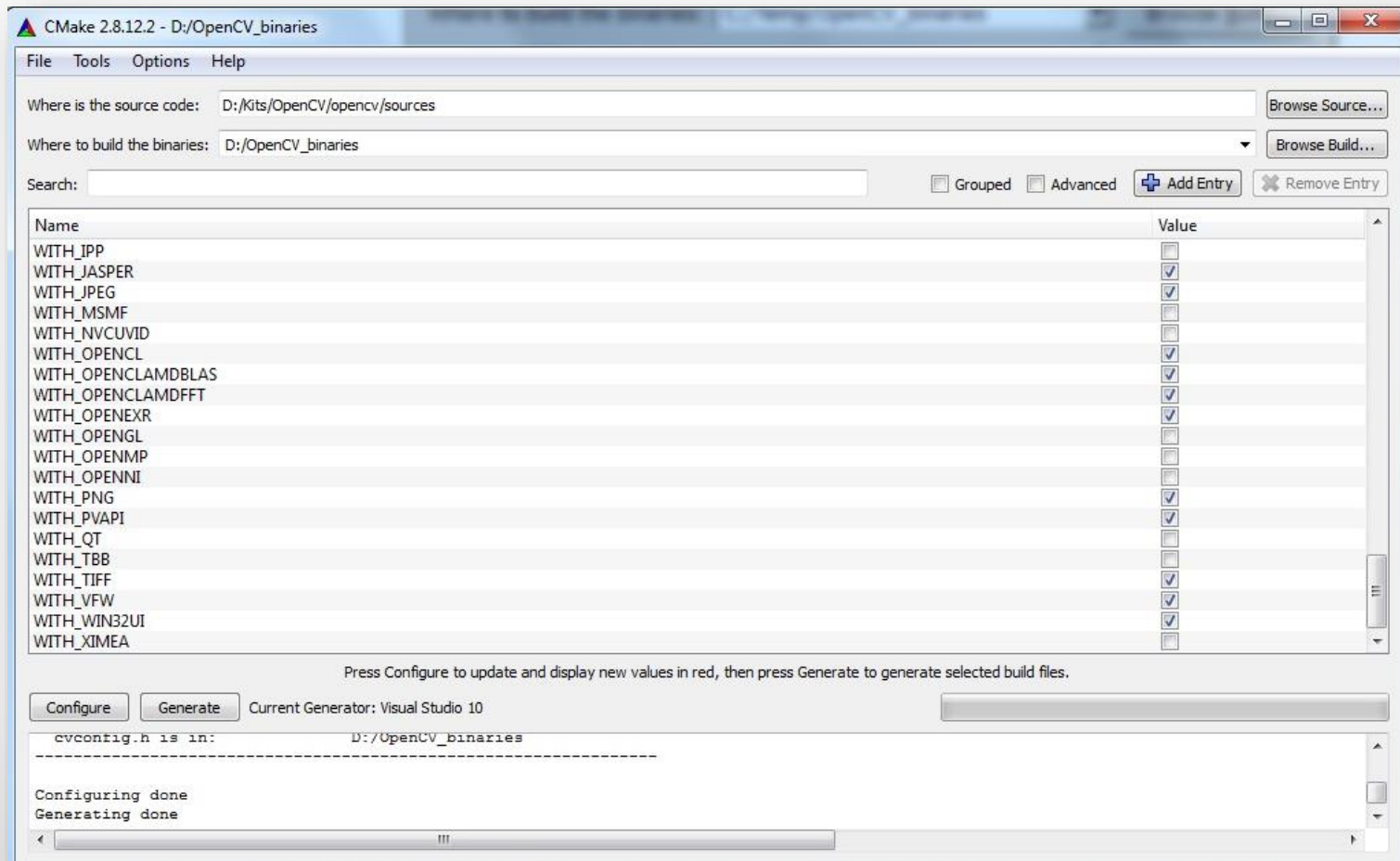
# Instalarea OpenCV

- Configurarea este gata. Acum apasam din nou **Generate**



# Instalarea OpenCV

- Gata si generarea



# Instalarea OpenCV

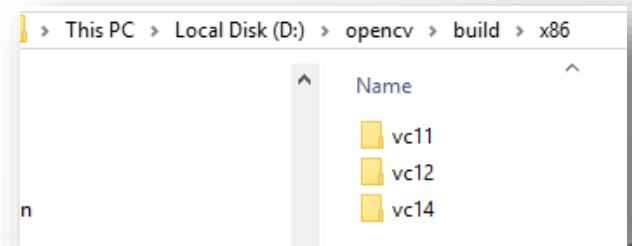
- Verificam folderul in care am trimis libraria

- Dam dublu-click pe OpenCV.sln (solutia creata) pentru a o deschide in Microsoft Studio 2010 (sau ce versiune avem)
- Dam Build Solution atat cu *Debug*, cat si cu *Release*
  - Dureaza in general peste 5 minute...

3rdparty	21.02.2014 13:59	File folder	
apps	21.02.2014 14:07	File folder	
CMakeFiles	21.02.2014 14:07	File folder	
data	21.02.2014 14:07	File folder	
doc	21.02.2014 14:07	File folder	
include	21.02.2014 14:07	File folder	
junk	21.02.2014 13:59	File folder	
modules	21.02.2014 14:07	File folder	
opencv2	21.02.2014 14:06	File folder	
unix-install	21.02.2014 14:06	File folder	
win-install	21.02.2014 14:06	File folder	
ALL_BUILD.vcxproj	21.02.2014 14:07	VC++ Project	27 KB
ALL_BUILD.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
cmake_install.cmake	21.02.2014 14:07	CMAKE File	7 KB
cmake_uninstall.cmake	21.02.2014 14:00	CMAKE File	2 KB
CMakeCache.txt	21.02.2014 14:07	TXT File	129 KB
cvconfig.h	21.02.2014 14:00	C Header File	4 KB
INSTALL.vcxproj	21.02.2014 14:07	VC++ Project	6 KB
INSTALL.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
opencv_modules.vcxproj	21.02.2014 14:07	VC++ Project	20 KB
opencv_modules.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
opencv_perf_tests.vcxproj	21.02.2014 14:07	VC++ Project	19 KB
opencv_perf_tests.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
opencv_tests.vcxproj	21.02.2014 14:07	VC++ Project	20 KB
opencv_tests.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
OpenCVConfig.cmake	21.02.2014 14:00	CMAKE File	14 KB
OpenCVConfig-version.cmake	21.02.2014 14:00	CMAKE File	1 KB
OpenCVModules.cmake	21.02.2014 14:00	CMAKE File	22 KB
uninstall.vcxproj	21.02.2014 14:07	VC++ Project	17 KB
uninstall.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
version_string.tmp	21.02.2014 14:06	TMP File	5 KB
ZERO_CHECK.vcxproj	21.02.2014 14:07	VC++ Project	27 KB
ZERO_CHECK.vcxproj.filters	21.02.2014 14:07	VC++ Project Filte...	1 KB
OpenCV.sln	21.02.2014 14:07	Microsoft Visual S...	95 KB

# Daca avem Visual Studio 2015 si OpenCV 3.0

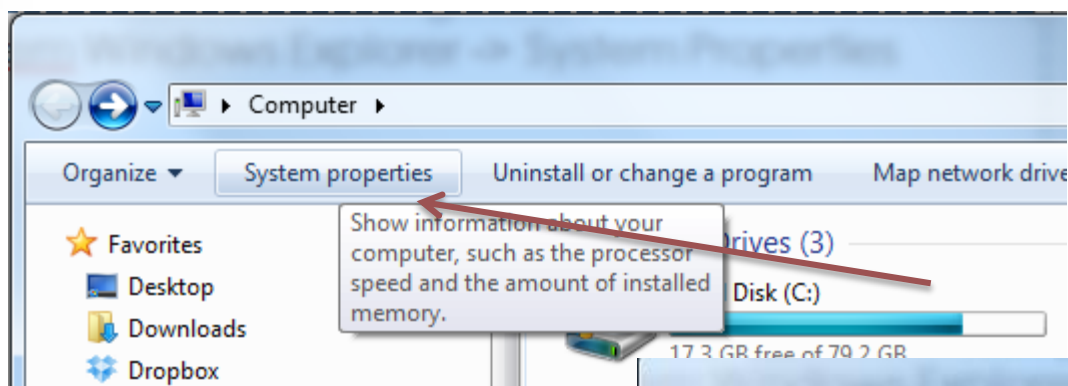
- Este necesar sa generam folderul vc14 prin o compilare aditionala.
- Tot in cadrul OpenCV.sln, alegem din cadrul Folderului CMakeTargets fisierul INSTALL
  - Cu click dreapta pe el alegem Build (o data cu *debug* si o data cu *release*)
- Cautam apoi in Windows Explorer, in folderul *install*, la versiunea x64 sau x86, dupa caz, folderul vc14.
  - Acesta se copiaza in folderul build, apoi x64 sau x86, dupa caz.
  - Acolo va sta langa folderele deja compilate existente, de exemplu vc11 si vc12.
- In continuare, caile vor fi date in functie de ce Visual Studio avem:
  - VS15 -> vc14
  - VS13 -> vc12
  - VS12 -> vc11 etc



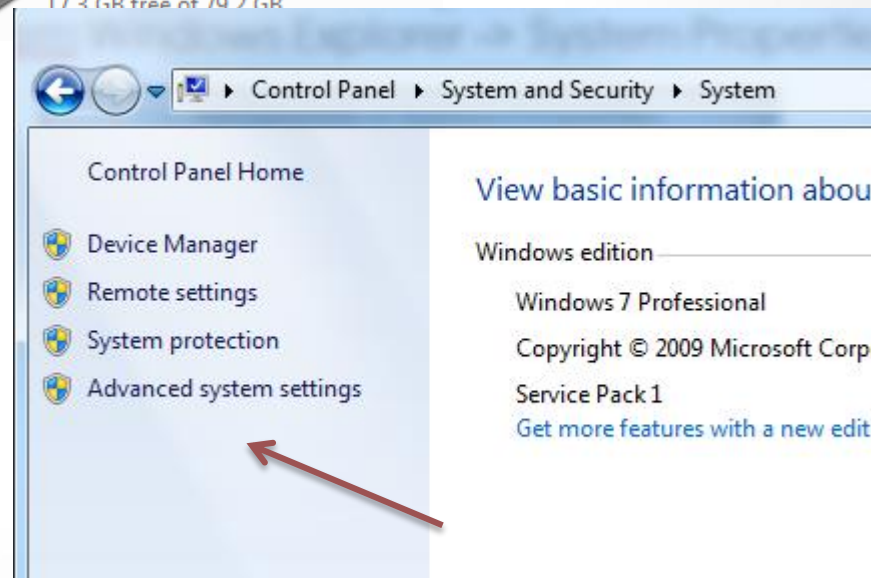
# Instalarea OpenCV

## Setare variabile de mediu

- Alegem Windows Explorer -> System Properties



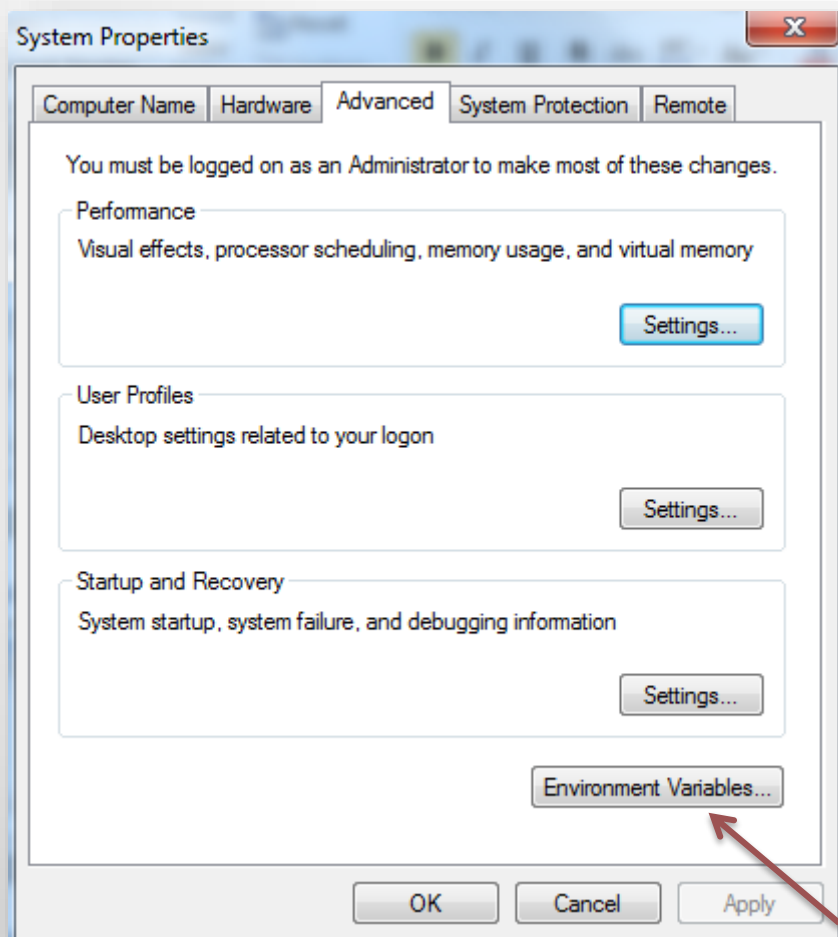
- Apoi **Advanced system settings**



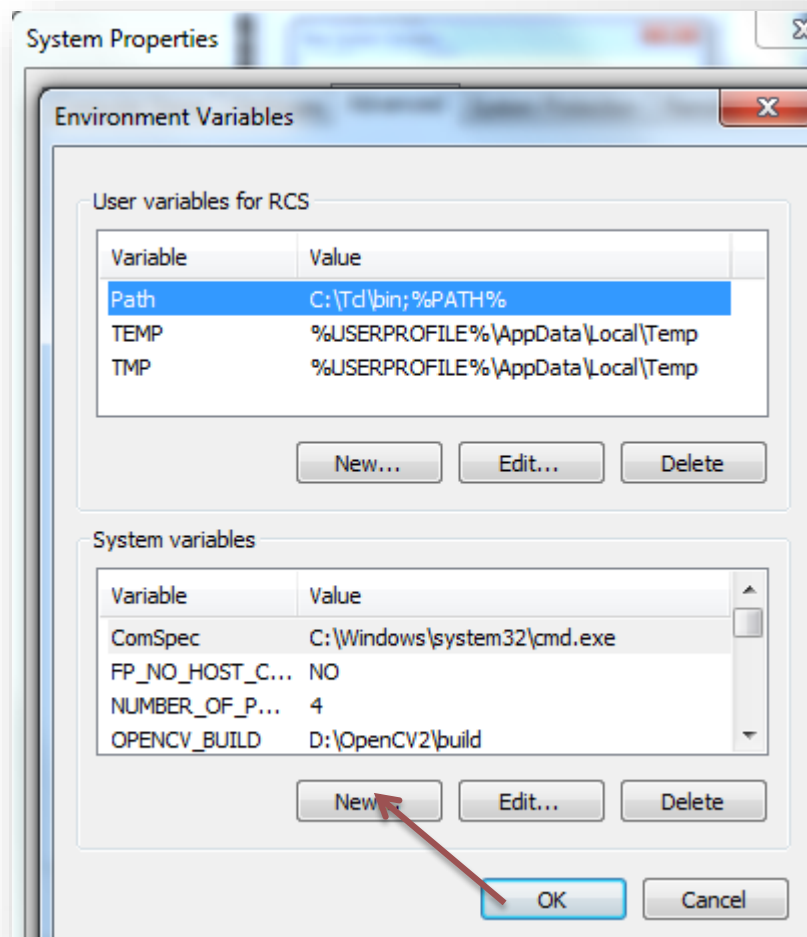
# Instalarea OpenCV

## Setare variabile de mediu

- Environment Variables



- Cream o variabila de sistem

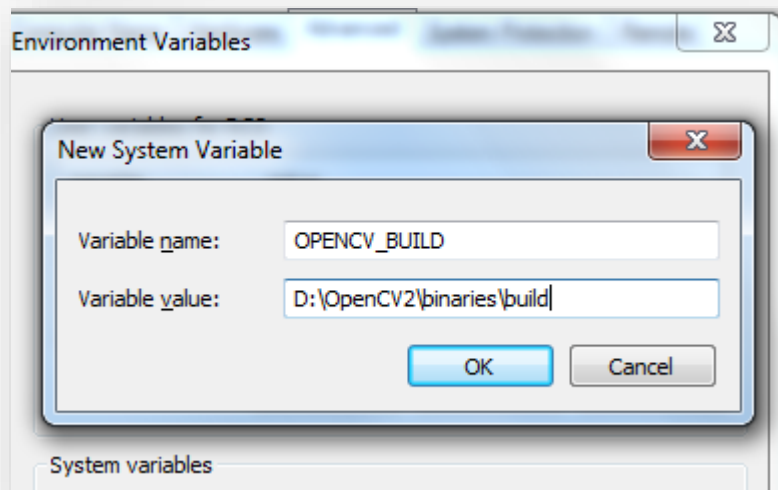


# Instalarea OpenCV

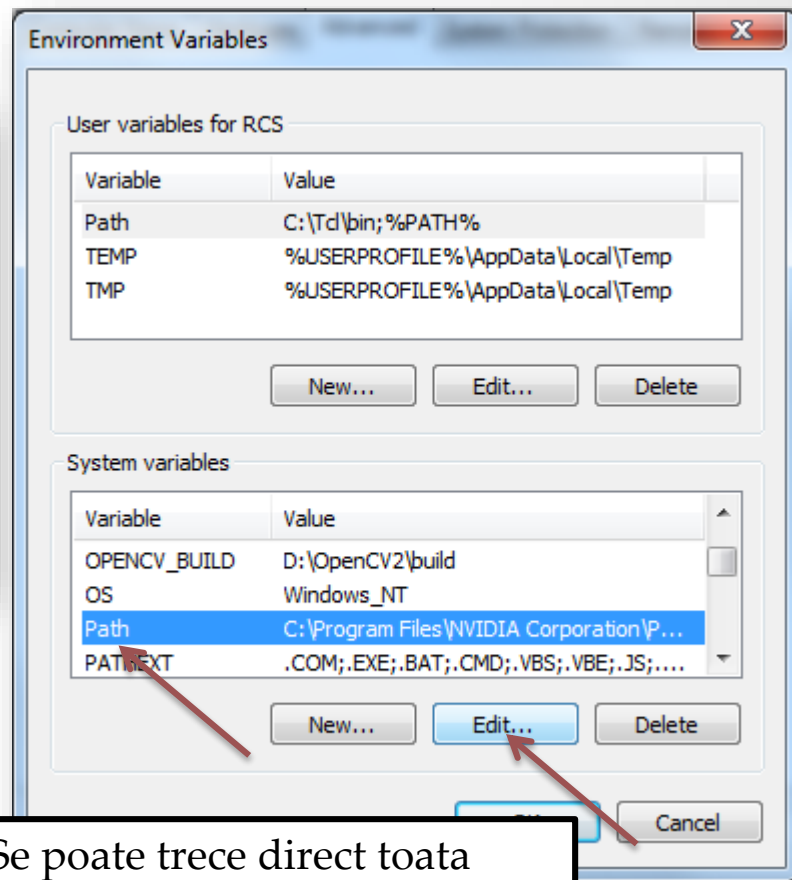
## Setare variabile de mediu

- Environment Variables

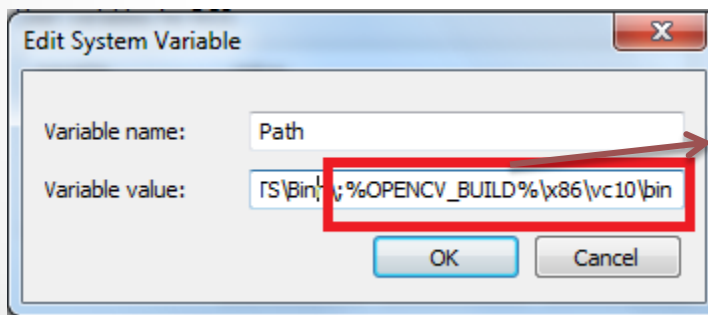
1



2



3

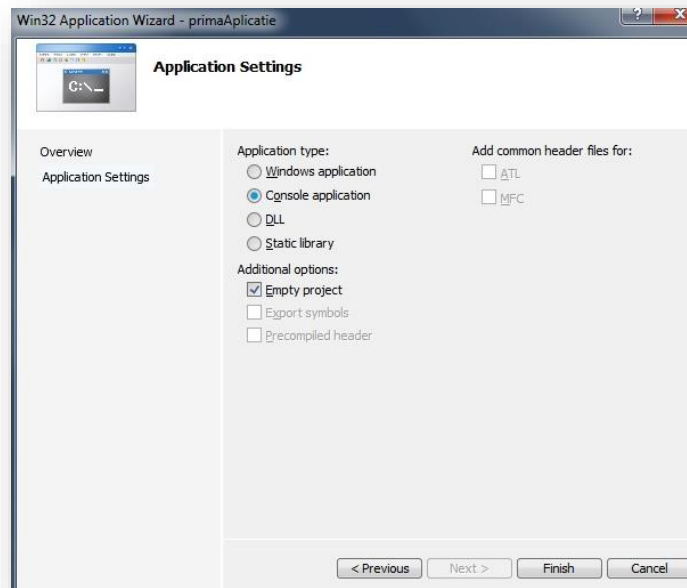


Se poate trece direct toata calea in loc sa se foloseasca variabila de mediu

Daca IDE este pe 64 de biti, se alege folderul x64 in loc de x86.

# Proiect OpenCV folosind Visual Studio 10

- Cream un proiect de tip **Console Application**, **Empty project**, fara precompiled header.

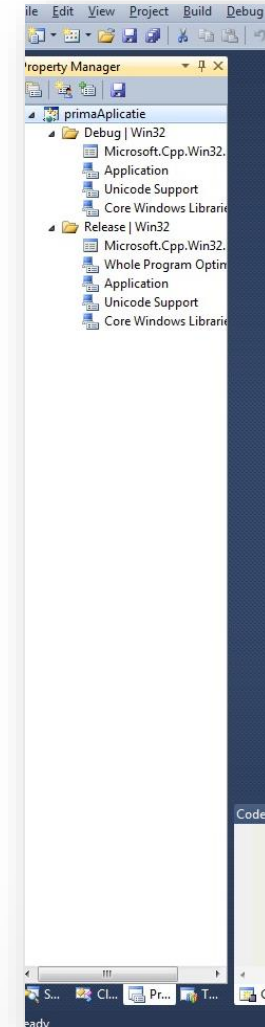


- In continuare, trebuie sa specificam unde se gasesc librariile OpenCV.



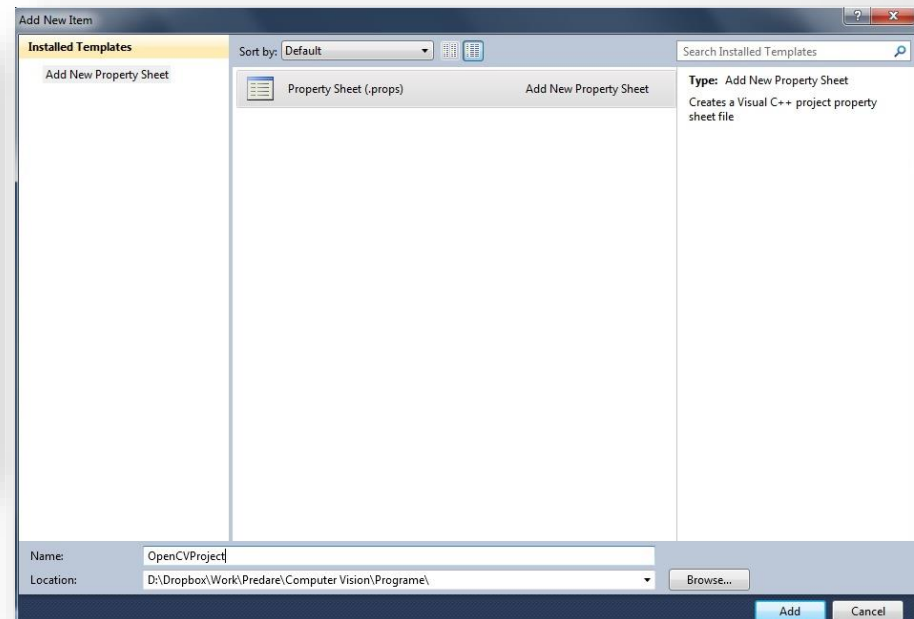
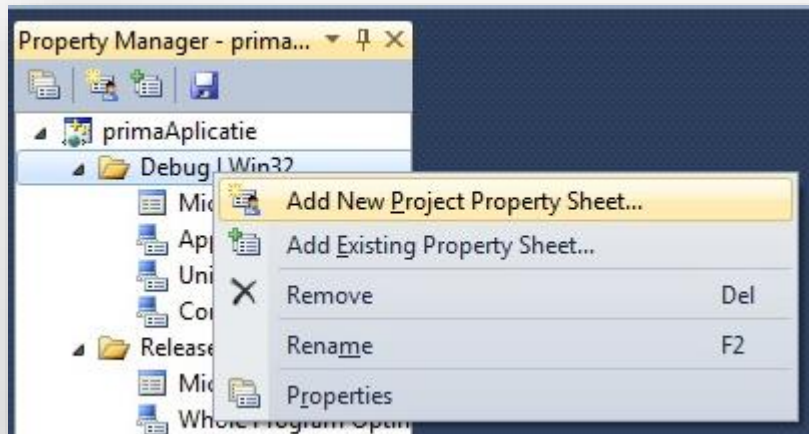
# Proiect OpenCV folosind Visual Studio 10

- Cea mai buna optiune pentru a specifica unde se gasesc librariile OpenCV este sa cream un **Property Sheet** pe care sa il putem utiliza si in alte proiecte.
- Pentru aceasta, mergem la Property Manager.
  - Avem de adaugat cate unul pentru **Debug** si **Release**.



# Property Sheet

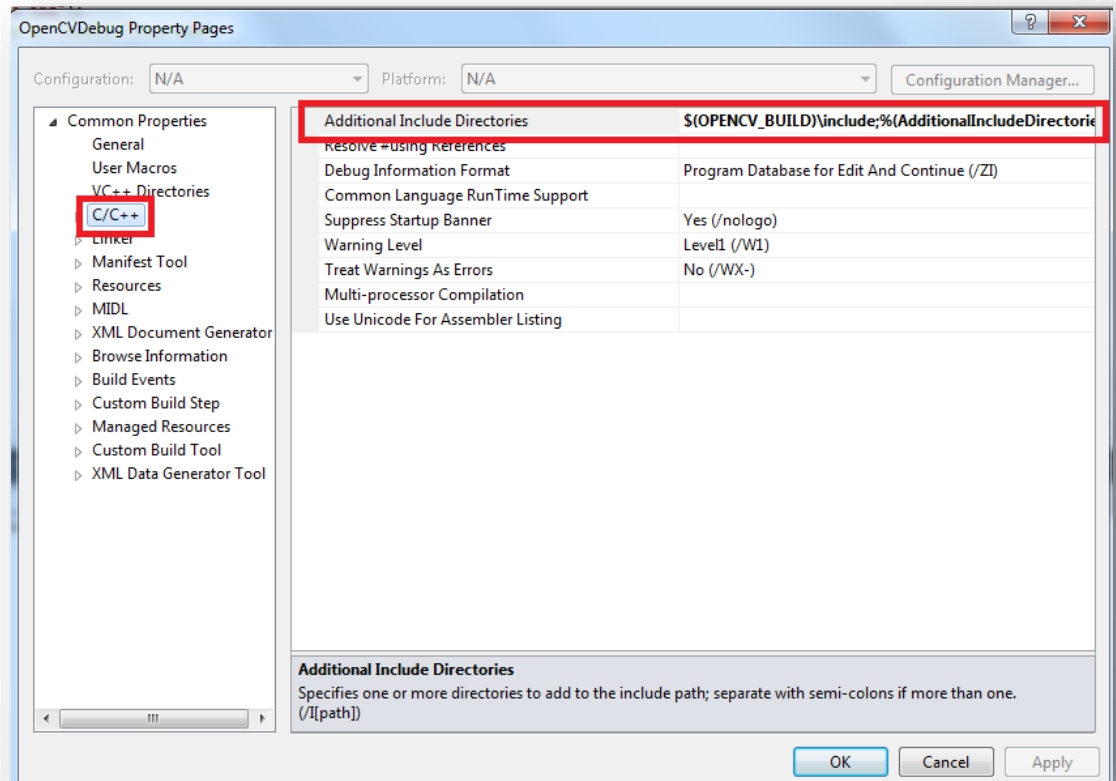
- Click-dreapta pe Debug si selectam **Add New Property Sheet...**
- In fereastra care se deschide ii punem un nume, de exemplu **OpenCVDebug**, apoi **Add**.



# Property Sheet

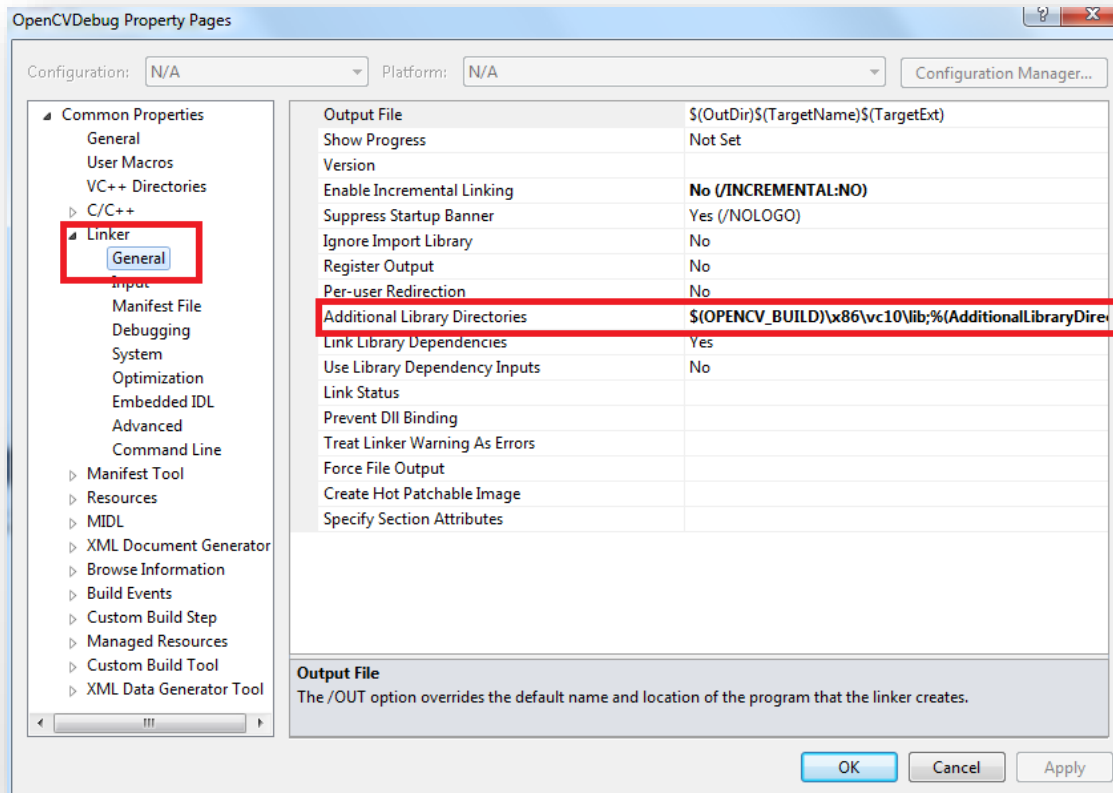
- Dublu-click pe noul property sheet creat, **OpenCVDebug**.
- La **C/C++**, alegem **Additional Include Directories**, apasam **Edit** si in fereastra deschisa adaugam **\$(OPENCV\_BUILD)\include**
  - Astfel furnizam calea catre bibliotecile din OpenCV pe care le vom utiliza

In loc de **\$(OPENCV\_BUILD)** se poate pune direct calea catre folderul **build**.



# Property Sheet

- Mergem apoi la **Linker** -> **General** -> **Additional Library Directories**, apasam **Edit...** si introducem **\$(OPENCV\_BUILD)\x86\vc10\lib**.
  - Din nou, daca avem IDE pe 64 de biti alegem folderul **x64**.

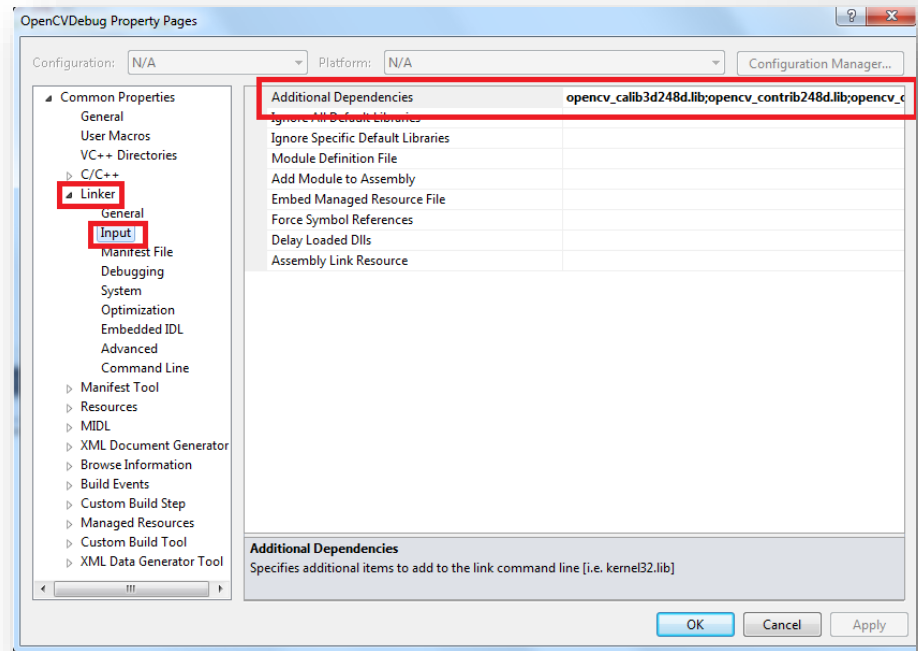


# Property Sheet

- Tot la **Linker**, **Input** si **Additional Dependencies**, **Edit**.
- Daca avem OpenCV 3.0 si Visual Studio 2012 sau 2013 adaugam in

fereastra care se deschide:

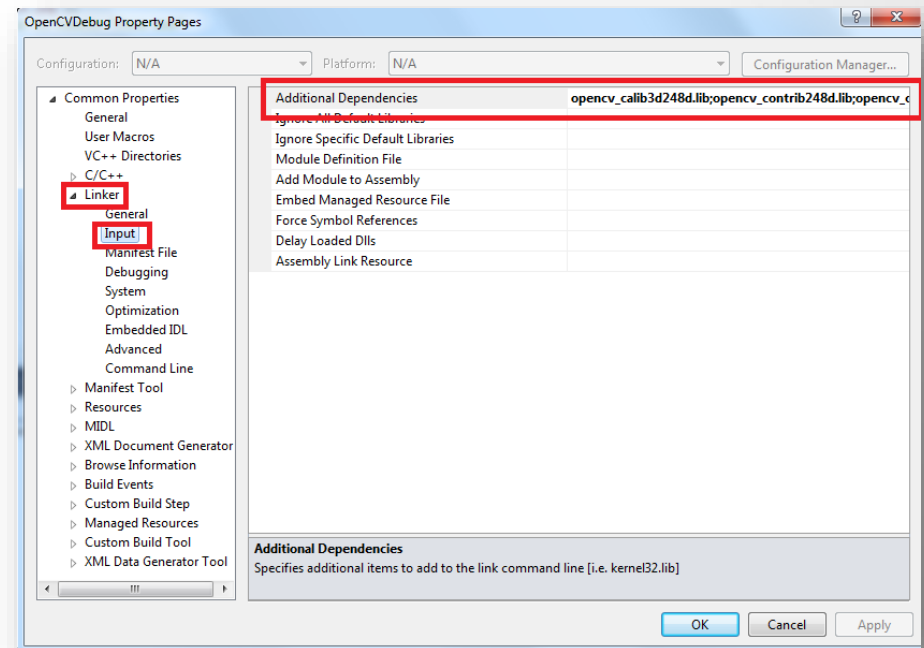
- opencv\_ts300d.lib
- opencv\_world300d.lib



- 300 vine de la versiunea OpenCV. Daca aveti o versiune diferita, schimbati valorile.
- Librariile se mai pot schimba de la o librerie la alta
  - se pot introduce unele noi si/sau scoate altele existente anterior.

# Property Sheet

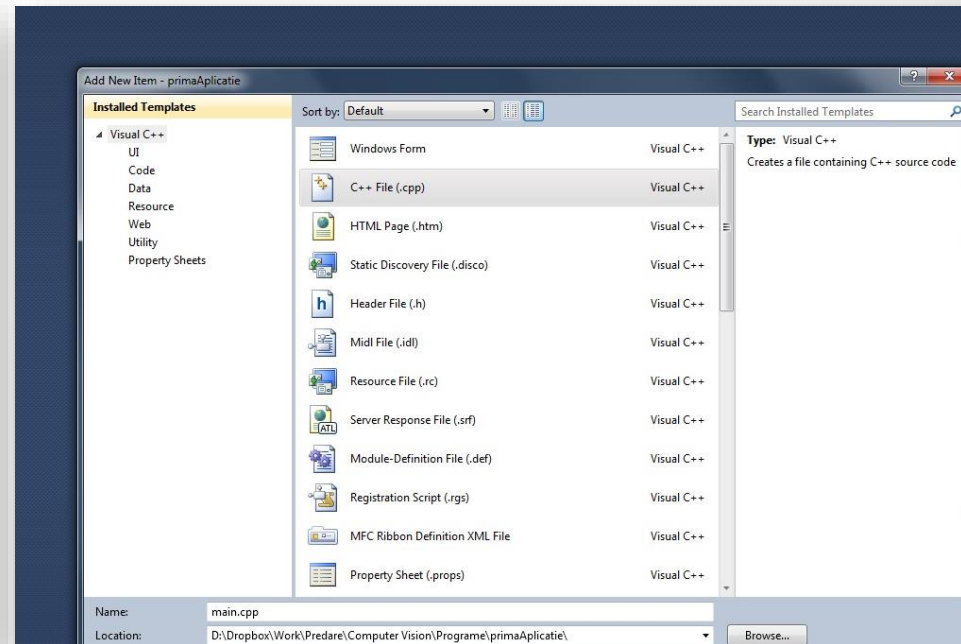
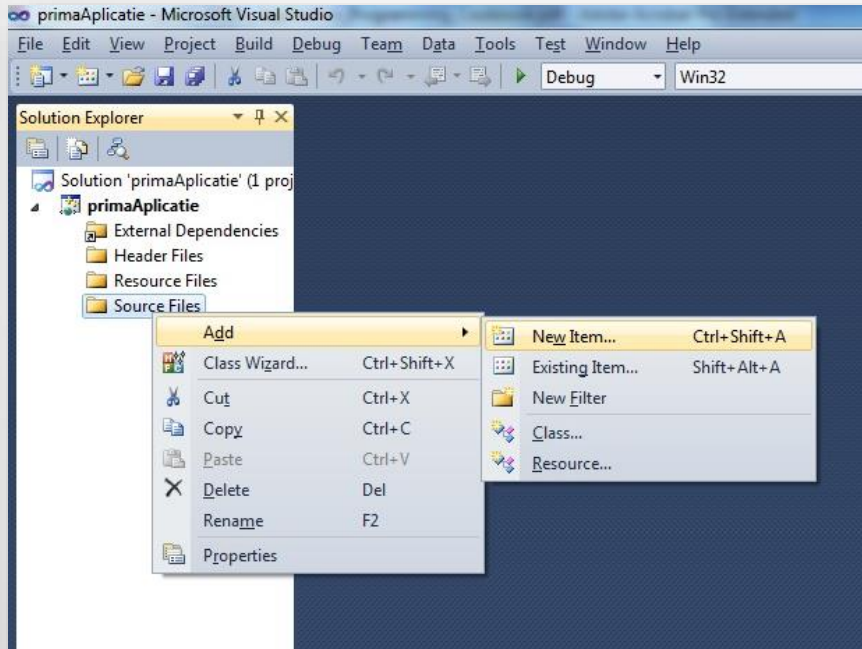
- Tot la **Linker**, **Input** si **Additional Dependencies**, **Edit**.
- Daca avem OpenCV 3.0 si Visual Studio 2015 (cu librarii generate dupa indicatiile de la slide-ul 20) librariile sunt:
- opencv\_calib3d300d.lib
- opencv\_core300d.lib
- opencv\_features2d300d.lib
- opencv\_flann300d.lib
- opencv\_highgui300d.lib
- opencv\_imgproc300d.lib
- opencv\_ml300d.lib
- opencv\_objdetect300d.lib
- opencv\_photo300d.lib
- opencv\_stitching300d.lib
- opencv\_ts300d.lib
- opencv\_video300d.lib
- opencv\_videostab300d.lib
- opencv\_imgcodecs300d.lib



# Property Sheet

- Caracterul *d* de la finalul librariilor de pe slide-ul anterior vine de la **debug**.
- Cum am facut pentru Debug va trebui sa facem si pentru varianta de Release.
  - **Add New Property Sheet** cu numele **OpenCVRelease**
  - **C/C++ -> Additional Include Directories**
  - **Linker -> General -> Additional Library Directories**
  - **Linker -> Input -> Additional Dependencies**
    - opencv\_ts300.lib
    - opencv\_world300.lib
- Sau, pentru Visual Studio 2015 cu generarea de la slide-ul 20:
  - opencv\_calib3d300.lib
  - opencv\_core300.lib
  - ...
  - **Toate fara *d*-ul de la final**

# Proiect OpenCV folosind Visual Studio 10





# Build si Run

- Daca nu merge si nu identificati o eroare de sintaxa, trebuie reluati pasii anteriori. 😊

```
main.cpp X
(Global Scope)
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace std;
using namespace cv;

int main()
{
    //citim o imagine
    Mat poza = imread("D:/poza.jpg");

    if(! poza.data ) // Verific daca citirea este valida
    {
        cout << "Nu am gasit poza" << endl ;
        waitKey(5000);
        return -1;
    }

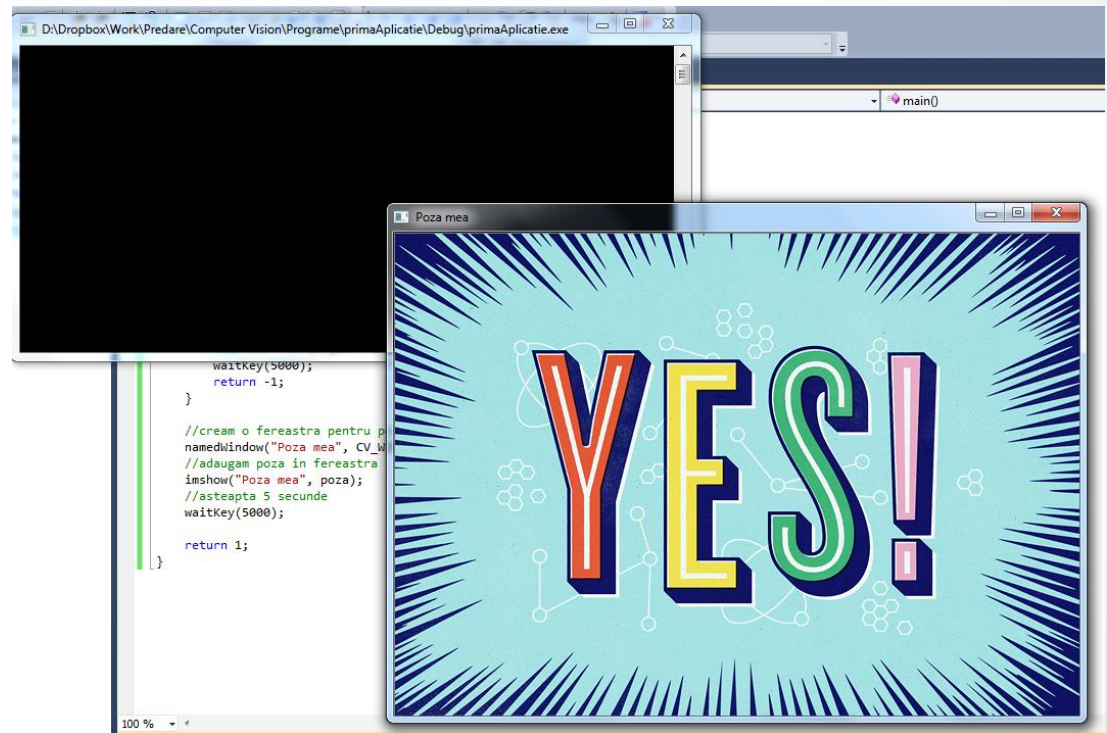
    //cream o fereastră pentru poza cu numele "Poza mea"
    namedWindow("Poza mea", CV_WINDOW_AUTOSIZE);
    //adaugam poza in fereastră
    imshow("Poza mea", poza);
    //asteapta 5 secunde
    waitKey(5000);

    return 1;
}
```

Pentru OpenCV 3.0 adaugati  
`#include <opencv2/imgcodecs.hpp>`  
imread a fost mutate in imgcodecs.hpp

# Ce face programul

- Afiseaza o poza

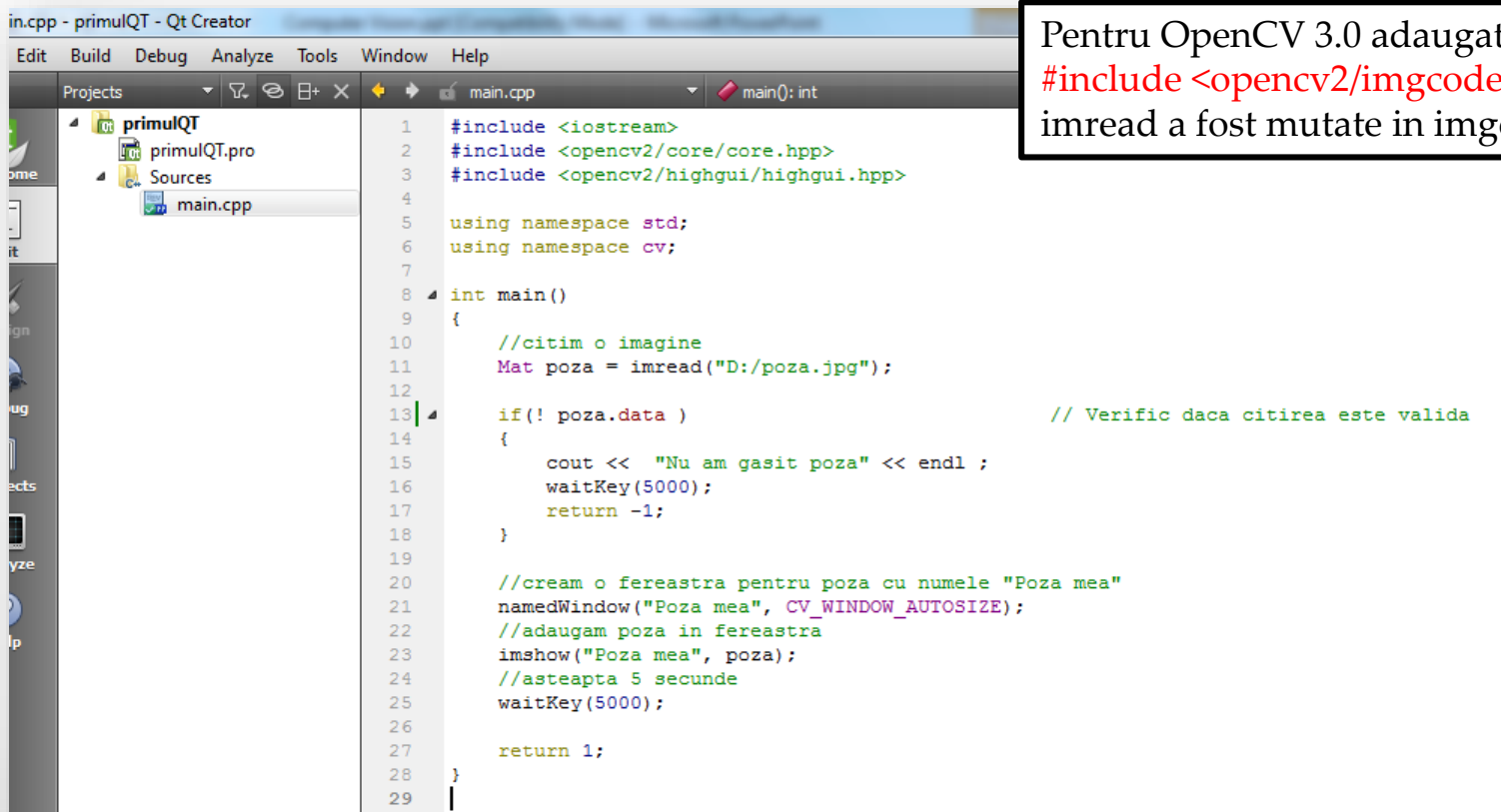


# Proiect OpenCV folosind QT

- QT se descarca de la  
<http://qt-project.org/downloads>
- Dezvoltat de compania norvegiana Trolltech si cumparat de Nokia in 2008.
- Open source
- Cross-platform

# Proiect OpenCV folosind QT

- Cream un proiect de tip **Console Application**.

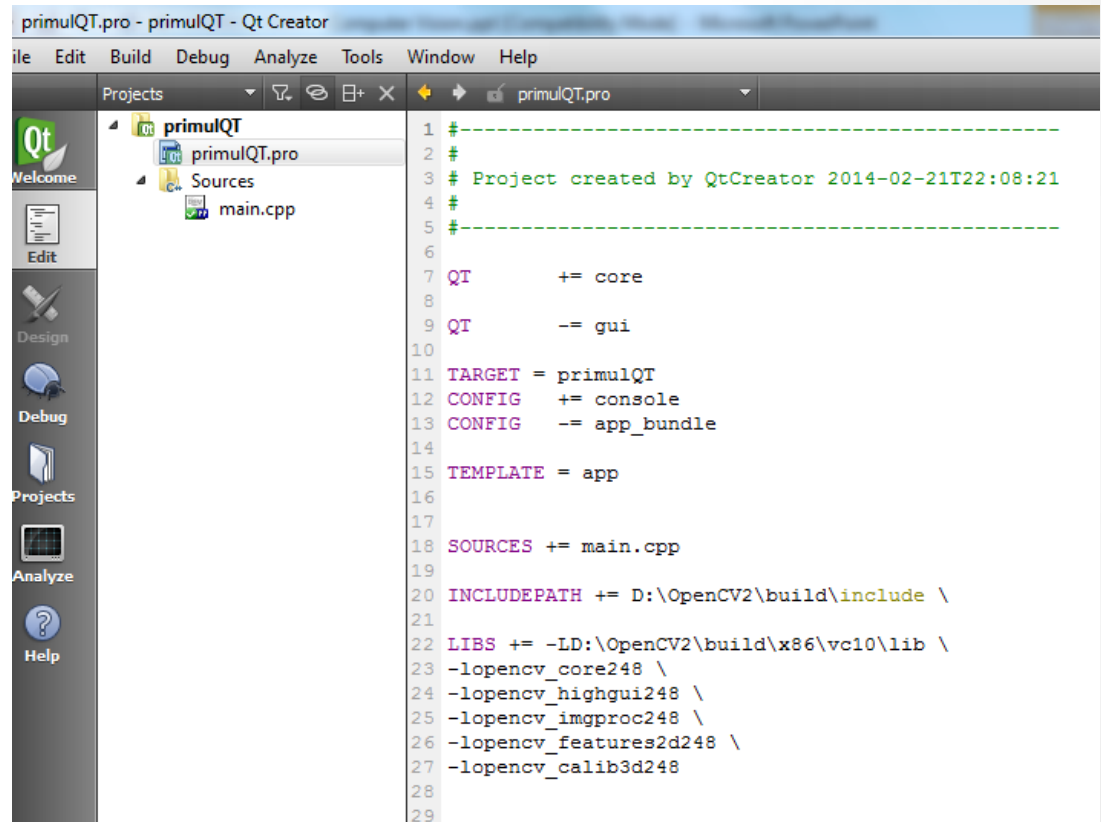


```
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/highgui/highgui.hpp>
4
5 using namespace std;
6 using namespace cv;
7
8 int main()
9 {
10     //citim o imagine
11     Mat poza = imread("D:/poza.jpg");
12
13     if(! poza.data ) // Verific daca citirea este valida
14     {
15         cout << "Nu am gasit poza" << endl ;
16         waitKey(5000);
17         return -1;
18     }
19
20     //cream o fereastră pentru poza cu numele "Poza mea"
21     namedWindow("Poza mea", CV_WINDOW_AUTOSIZE);
22     //adaugam poza in fereastră
23     imshow("Poza mea", poza);
24     //asteapta 5 secunde
25     waitKey(5000);
26
27     return 1;
28 }
29
```

Pentru OpenCV 3.0 adaugati  
`#include <opencv2/imgcodecs.hpp>`  
imread a fost mutata in imgcodecs.hpp

# Proiect OpenCV folosind QT

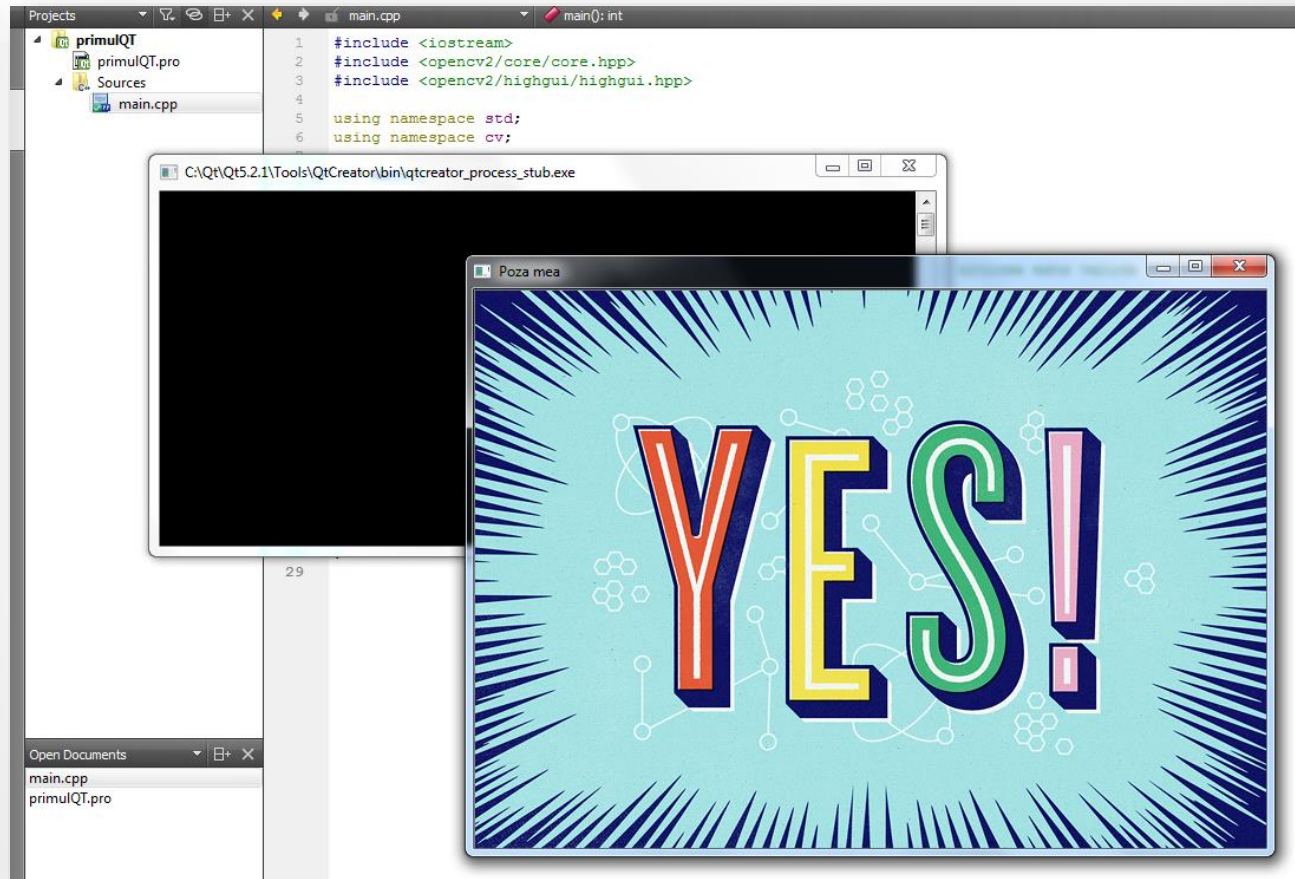
- In fisierul .pro se specifica calea catre folderul **include** si catre librarii.
- Din meniul Build, dam Run qmake.
- Nu este nevoie de setari aditionale.



The screenshot shows the Qt Creator interface with the 'primulQT.pro' file open. The left sidebar shows the project structure with 'primulQT' containing 'primulQT.pro' and 'Sources' containing 'main.cpp'. The main editor displays the following .pro file content:

```
1 #-----
2 #
3 # Project created by QtCreator 2014-02-21T22:08:21
4 #
5 #-----
6
7 QT     += core
8
9 QT     -= gui
10
11 TARGET = primulQT
12 CONFIG += console
13 CONFIG -= app_bundle
14
15 TEMPLATE = app
16
17
18 SOURCES += main.cpp
19
20 INCLUDEPATH += D:\OpenCV2\build\include \
21
22 LIBS += -LD:\OpenCV2\build\x86\vc10\lib \
23 -lopencv_core248 \
24 -lopencv_highgui248 \
25 -lopencv_imgproc248 \
26 -lopencv_features2d248 \
27 -lopencv_calib3d248
28
29
```

# Proiect OpenCV folosind QT



# Exercitii

- Folosind documentatia OpenCV, modificati programul anterior pentru a:
  - Stabili pentru fereastra in care s-a afisat imaginea dimensiunea 800x600
  - Citi o poza in format Grayscale.
  - Salva cu un alt nume poza citita in format Grayscale
  - Scrie la consola cati pixeli are o poza in lungime si cati are in latime
  - Afisa imaginea citita intr-o fereastra de dimensiuni la jumatate din marimile sale originale.