
Manual de Instalación y configuración de la librería OpenCV4 en Windows

---

http://umh1782.edu.umh.es

Requisitos: *Windows 7, Windows 10 - (64bits)*

Más información en http://opencv.org

NOTA: necesitarás más de 100G de espacio libre en el disco para poder compilar la librería completa con los módulos adiciones debido a los ficheros temporales de compilación.

1) Instalar Microsoft Visual Studio 2017 y Cmake:

- En el blog de la asignatura (Apartado ***OpenCV***) disponemos de un enlace para descargar el instalador tanto de Visual Studio 2017 como de CMake

  http://umh1782.edu.umh.es/opencv/#Software

- Al arrancar el compilador por primera vez, el programa nos pedirá una licencia que podemos conseguir de forma gratuita registrándonos en la web de Microsoft, basta seguir el enlace que nos indica.

- Si deseamos que se compilen las librerías de Python deberemos tener instalado también el intérprete correspondiente. Puedes encontrar instrucciones para su instalación en:

  http://umh1782.edu.umh.es/phyton

3) Instalar la librería OpenCV:

- Podemos hacer la instalación de una versión precompilada desde el portal de la asignatura, pero en este caso solo dispondremos de la librería base, no pudiendo usar las librearías adicionales (*contrib*)

  http://umh1782.edu.umh.es/opencv/#Software

- En nuestro caso vamos a compilar la librería incluyendo los módulos adiciones (contrib). Para ello descargaremos el código fuente de la librería OpenCV y del paquete contrib.
  - opencv-4.5.5.zip
  - opencv_contrib-4.5.5.zip

– Descomprimiremos ambos ficheros en una carpeta (**Figura 1**):

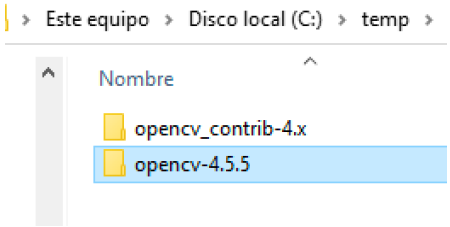


Figura 1. Carpetas código fuente

4) Compilar la librería OpenCV:

4.1) Generar proyecto en Visual Studio 2017:

Ejecutaremos el programa ***CMake-Gui***:

- Seleccionaremos la ubicación de la carpeta del código de la librería (***opencv-4.5.5***) (**Figura 2**)
- Seleccionaremos la carpeta donde estará ubicada la librería compilada (***opencv-contrib)*** ***(***Figura 2***)***.

- Pulsaremos el botón "Configure" par a configurar el proyecto. Nos abrirá una ventana de configuración (**Figura 3**), donde seleccionaremos el compilador a utilizar, en nuestro caso **Visual Studio 2017 (VC15)**, así como la plataforma opcional **x64**.

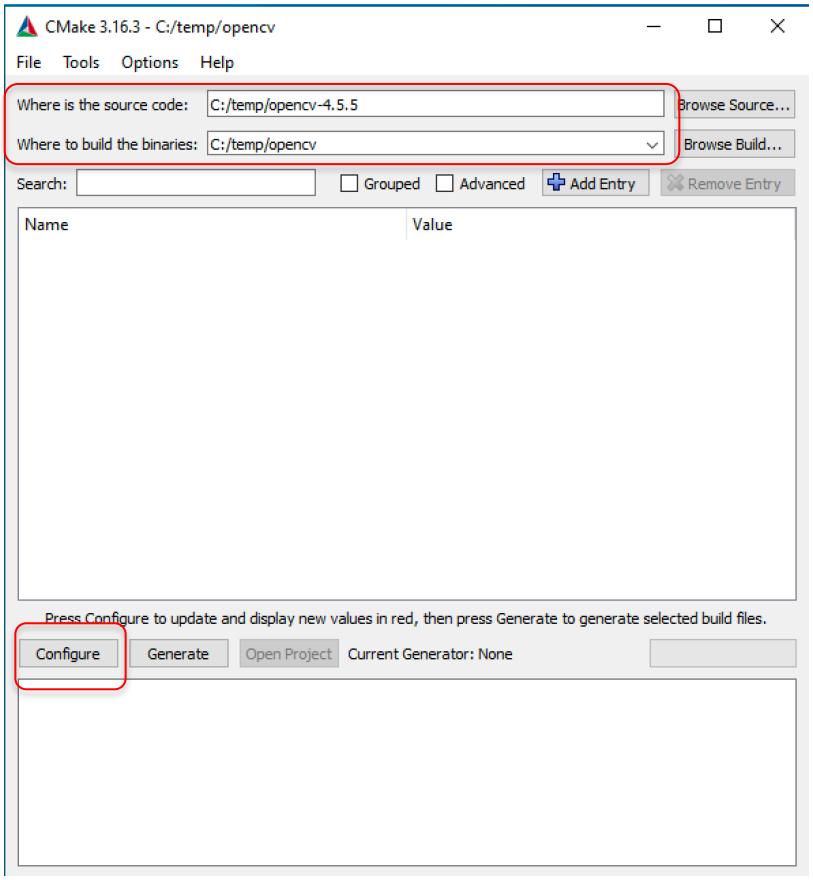


Figura 2 Cmake-gui

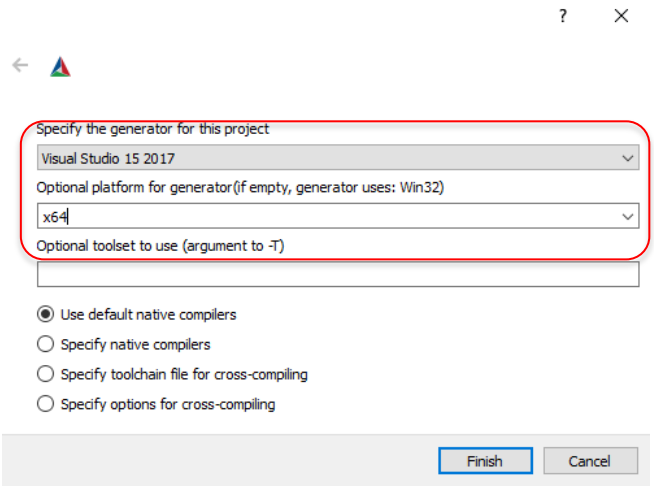


Figura 3. Configuración Compilador

- Al pulsar el botón "***Finish***" CMake iniciará la búsqueda en el código fuente de OpenCV del fichero "***CMakeLists.txt***" y chequeará la configuración del compilador y otras herramientas necesarias. En la ventana (**Figura 4**) nos mostrará en la parte superior las opciones configurables en la compilación de OpenCV (se marcan en rojo las opciones añadidas desde la configuración previa, no se trata de errores). En la parte inferior nos mostrará avisos de la ejecución con posibles problemas (**texto en rojo**). Deberemos resolver cualquier aviso que nos dé. Cada vez que hagamos una modificación podemos testearla pulsando de nuevo el botón "***Configure***".

  Como podemos ver hay muchas opciones de librerías externas que pueden ser configuradas de forma opcional si las tenemos instaladas. En esta compilación solo usaremos las librerías estándar.

- **Si fuera necesario** (en caso de no estar bien configurado el interprete Python), editaremos el fichero en la carpeta del código opencv-4.5.5: "***cmake/OpenCVDetectPython.cmake***", para configurar correctamente la versión de Python a utilizar (en nuestro caso Python 3),

> Editar al final (último bloque) del fichero **cmake/OpenCVDetectPython.cmake** y cambiar el orden de búsqueda de la versión de Python poniendo primero la versión 3:
>
> ```
> if(PYTHON_DEFAULT_EXECUTABLE)
>  set(PYTHON_DEFAULT_AVAILABLE "TRUE")
> elseif(PYTHON3INTERP_FOUND) # Use Python 3 as fallback Python interpreter (if there is no Python 2)
>  set(PYTHON_DEFAULT_AVAILABLE "TRUE")
>  set(PYTHON_DEFAULT_EXECUTABLE "${PYTHON3_EXECUTABLE}")
> elseif(PYTHON2INTERP_FOUND) # Use Python 2 as default Python interpreter
>  set(PYTHON_DEFAULT_AVAILABLE "TRUE")
>  set(PYTHON_DEFAULT_EXECUTABLE "${PYTHON2_EXECUTABLE}")
> endif()
> ```

- Ejecutaremos de nuevo el comando (botón) "***Configure***", veremos que el aviso en rojo de la configuración de Python ha desaparecido.

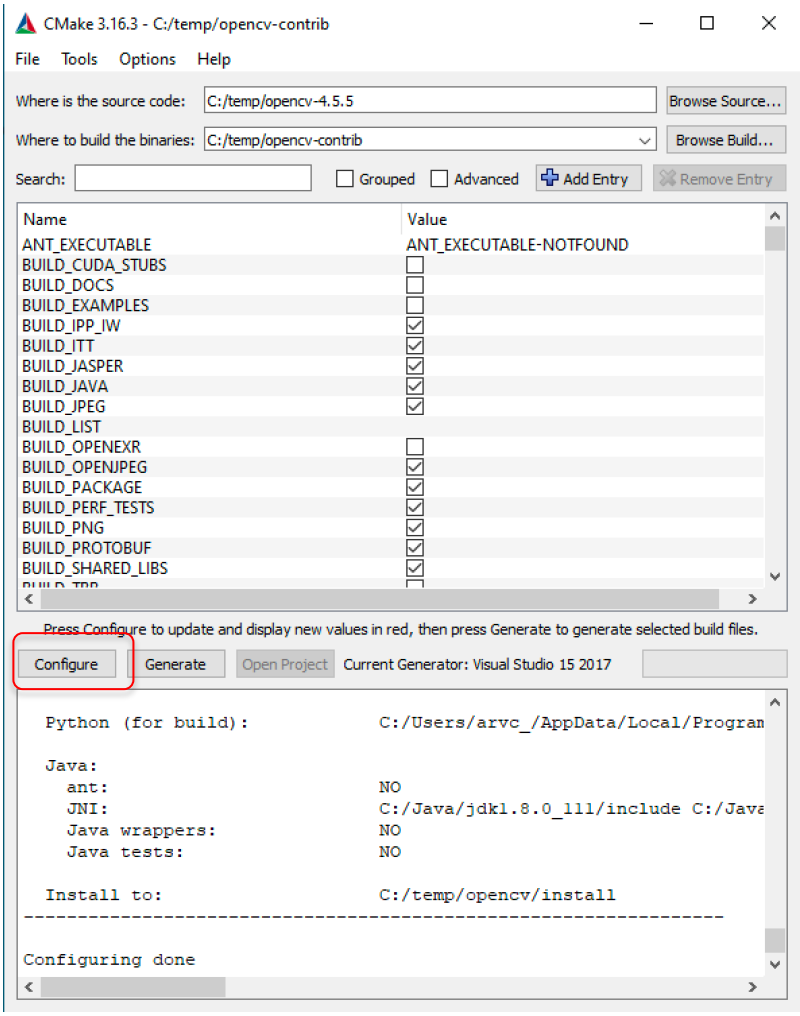


Figura 4. Resultado ejecución Cmake

- Al editar el fichero en la carpeta del código: "***cmake/OpenCVGenSetupVars.cmake***", (línea 54) podemos ver que el error que nos muestra la Figura 5 indica que está usando una ruta absoluta para el interprete de python y no lo soporta, Para evitar este error deseleccionaremos la opción **OPENCV_GENERATE_SETUPVARS**. Ejecutaremos de nuevo el comando (botón) "***Configure***", veremos que el aviso en rojo de la configuración ha desaparecido
    - **OPENCV_GENERATE_SETUPVARS** (Desactivado)

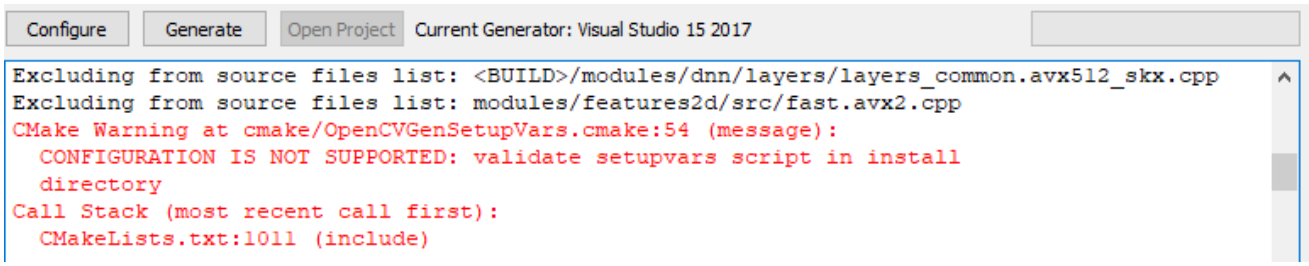


Figura 5. Error path absoluto python

```
if(IS_ABSOLUTE "${__python_path}")
  set(OPENCV_PYTHON_DIR_RELATIVE_CMAKECONFIG "${__python_path}")
  message(WARNING "CONFIGURATION IS NOT SUPPORTED: validate setupvars script in install directory")
 else()
```

- Configuración de otras opciones de OpenCV: en la ventana superior dispones varias casillas con opciones que podemos activar o desactivar (están por orden alfabético). En nuestro caso configuraremos las siguientes opciones:
    - **BUILD_EXAMPLES** (Activo)
    - **BUILD_SHARED_LIBS** (Activado) : creará una librería dinámica compartida **DLL**. Si lo dejamos desactivado crea un una librería estática que no precisa de DLLs.

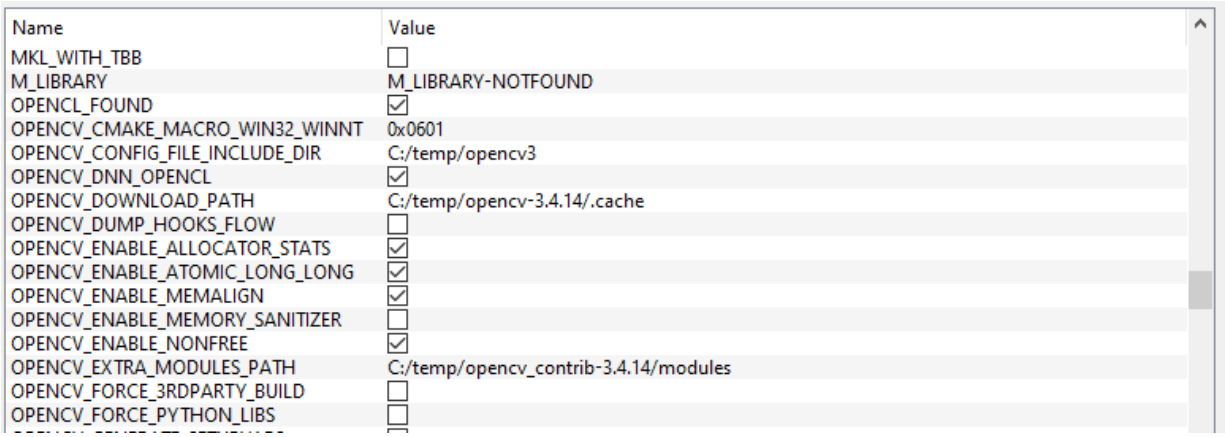
    - **BUILD_opencv_world** (Activo) : crea un único fichero con toda la librería
    - **BUILD_opencv_python3** (Activo) : crea librería python

    - **OPENCV_ENABLE_NONFREE** (Activo): incluye los algoritmos SURF
    - **PROTOBUF_UPDATE_FILES** (desactivado)

    - **OPENCV_EXTRA_MODULES_PATH**. Añadiremos la carpeta de los módulos adicionales: ***c:/temp/opencv_contrib-4.x/modules***. (Figura 6) (**Nota**: al pulsar ***Configure***, añadirá nuevas opciones a esta ventana correspondientes a cada uno de los módulos añadidos Figura 7). Recuerde volver a pulsar ***Configure***.

    - Si tenemos una **GPU NVidia** con soporte **CUDA,** podemos activar la compilación del módulo CUDA de OpenCV (precisa instalar previamente la librería CUDA): https://developer.nvidia.com/cuda-downloads https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html

        - **BUILD_CUDA_STUBS** (Activo)
        - **OPENCV_DNN_CUDA** (Activo)
        - **WITH_CUDA** (Activo)
        - **WITH_CUDNN** (Activo)
        - **WITH_CUBLAS** (Activo)



Name	Value
MKL_WITH_TBB	☐
M_LIBRARY	M_LIBRARY-NOTFOUND
OPENCL_FOUND	☑
OPENCV_CMAKE_MACRO_WIN32_WINNT	0x0601
OPENCV_CONFIG_FILE_INCLUDE_DIR	C:/temp/opencv3
OPENCV_DNN_OPENCL	☑
OPENCV_DOWNLOAD_PATH	C:/temp/opencv-3.4.14/.cache
OPENCV_DUMP_HOOKS_FLOW	☐
OPENCV_ENABLE_ALLOCATOR_STATS	☑
OPENCV_ENABLE_ATOMIC_LONG_LONG	☑
OPENCV_ENABLE_MEMALIGN	☑
OPENCV_ENABLE_MEMORY_SANITIZER	☐
OPENCV_ENABLE_NONFREE	☑
OPENCV_EXTRA_MODULES_PATH	C:/temp/opencv_contrib-3.4.14/modules
OPENCV_FORCE_3RDPARTY_BUILD	☐
OPENCV_FORCE_PYTHON_LIBS	☐

Figura 6. Configuración módulos adicionales (contrib)

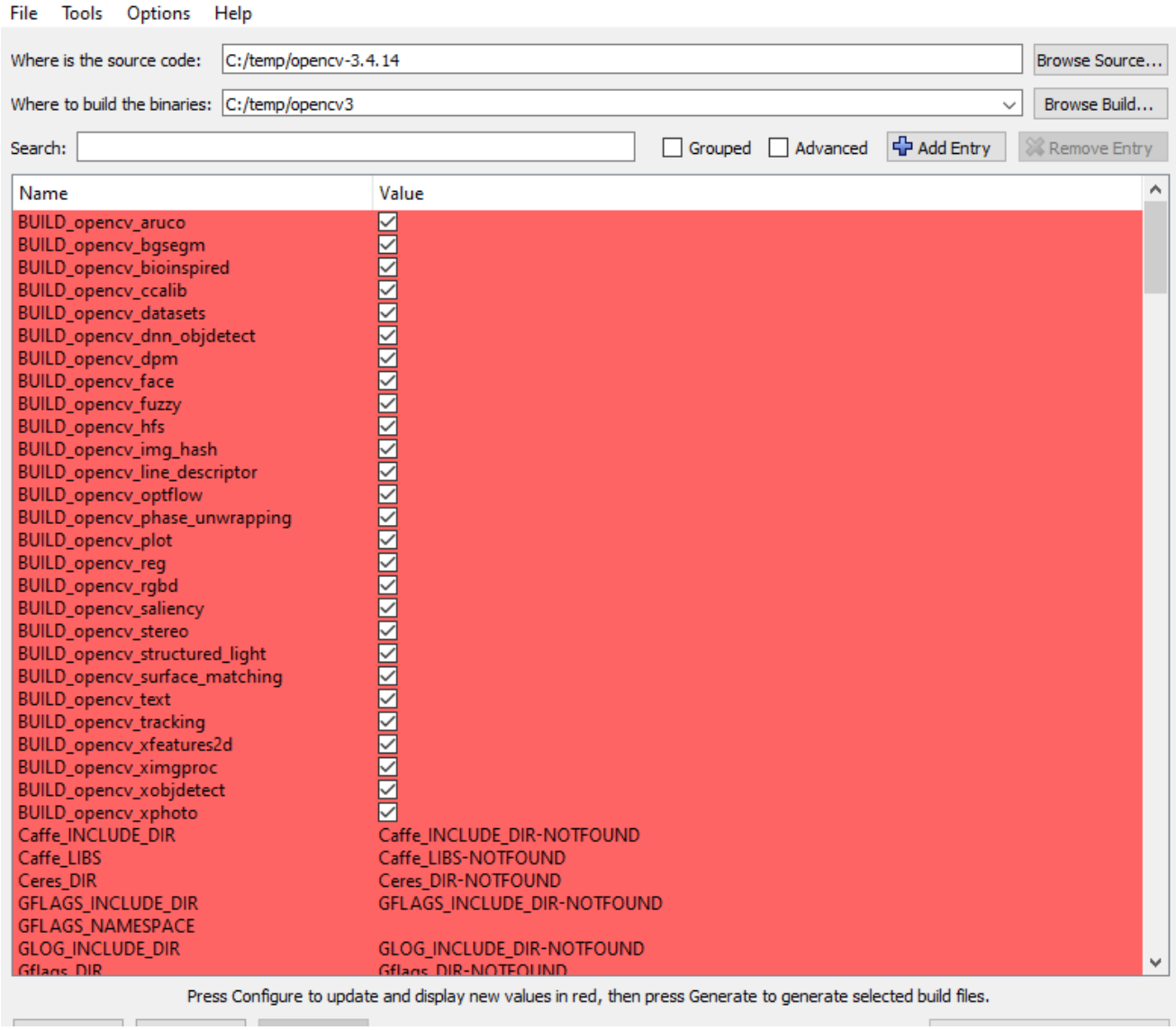


Figura 7. Nuevas opciones del paquete Contrib

- Deberemos ir resolviendo cualquier aviso en rojo que nos aparezca, generalmente es que no tenemos instalado o no es capaz de encontrar algún paquete o librería. En internet podemos buscar la forma de resolverlo. Al finalizar volveremos a pulsar el botón ***Configure*** para comprobar que la configuración es correcta.

- A continuación(cuando hayamos resulto todos los errores) pulsaremos una última vez el botón "***Configure***" y a continuación el botón "***Generate***" para que genere los ficheros del proyecto en Visual Studio. Esto puede tardar algún tiempo

Nota: si quisiéramos empezar la configuración de CMake de cero, iremos al menú ***File*** y seleccionaremos la opción "***Delete Cache***".

4.2) Compilar el proyecto en Visual Studio 2017 (Linea de comandos):

Ejecutaremos los siguientes comandos en una consola con la carpeta de instalación seleccionada: c:/temp/opecv-contrib

```
cmake --build . --config Debug --target INSTALL

cmake --build . --config Release --target INSTALL
```

Nota: la compilación puede tardar bastante tiempo y ocupar más de 10 GB de espacio en el disco.

En la carpeta (***c:/temp/opencv-contrib/install***) (Figura 8) disponemos de los ficheros de la librería con toda su estructura de directorios tal como tal como aparece en la versiones precompiladas. Podemos copiarlas en la carpeta de instalación estándar de OpenCV (***c:/opencv-contrib***). Para ello ejecutaremos los siguientes comandos:

```
cd c:\temp\opencv-contrib
xcopy /e /y .\install\  c:\opencv-contrib\build\
xcopy /e /y .\python_loader\  c:\opencv-contrib\build\python\
xcopy /e /y .\lib\python3\Release\ c:\opencv-contrib\build\python\cv2\python-
3.10\
xcopy /e /y c:\temp\opencv-4.5.5\samples\ c:\opencv-contrib\samples\


copy /y .\bin\Debug\*.dll c:\opencv-contrib\build\x64\vc15\bin\
copy /y .\bin\Debug\*.pdb c:\opencv-contrib\build\x64\vc15\bin\
copy /y .\lib\Debug\*.lib c:\opencv-contrib\build\x64\vc15\lib\
copy /y .\lib\Debug\*.pdb c:\opencv-contrib\build\x64\vc15\lib\
```

Nombre	Fecha de modificación	Tipo	Tamaño
bin	31/05/2021 11:42	Carpeta de archivos	
etc	30/05/2021 22:09	Carpeta de archivos	
include	30/05/2021 22:06	Carpeta de archivos	
x64	30/05/2021 22:06	Carpeta de archivos	
LICENSE	01/04/2021 14:37	Archivo	3 KB
OpenCVConfig.cmake	30/05/2021 11:10	Archivo CMAKE	7 KB
OpenCVConfig-version.cmake	30/05/2021 11:10	Archivo CMAKE	1 KB

Figura 8. Directorio "install"

- Deberemos editar la variable de entorno ***'Path'*** *y en la ventana de edición añadiremos al final la siguiente ruta.*
  - ***c:\opencv-contrib\build\x64\vc15\bin***
- Deberemos también editar los directorios de trabajo en las hojas de propiedades del proyecto (Ver documento de creación de un proyecto en Visual Studio), o descargar las hojas de propiedades ya configuradas desde el blog.

ANEXO.- Compilar el proyecto en Visual Studio 2017:

A continuación buscaremos la carpeta con el proyecto generado (***c:/temp/opencv-contrib***) y abriremos el fichero (***OpenCV.sln***). Al hacer doble click, se ejecutará el compilador y cargará el proyecto (puede tardar mientras analiza todos los ficheros del mismo, en el pie de la ventana podemos ver el progreso).

Nota: NO debemos cerrar la aplicación **CMake-Gui** por si debemos modificar alguna opción si tenemos problemas con la compilación.

En la ventana izquierda del compilador (Figura 9) marcaremos la pestaña "Explorador de Soluciones". Podemos ver la configuración que ha realizado CMake para generar la compilación de cada parte de la librería y los ejemplos.

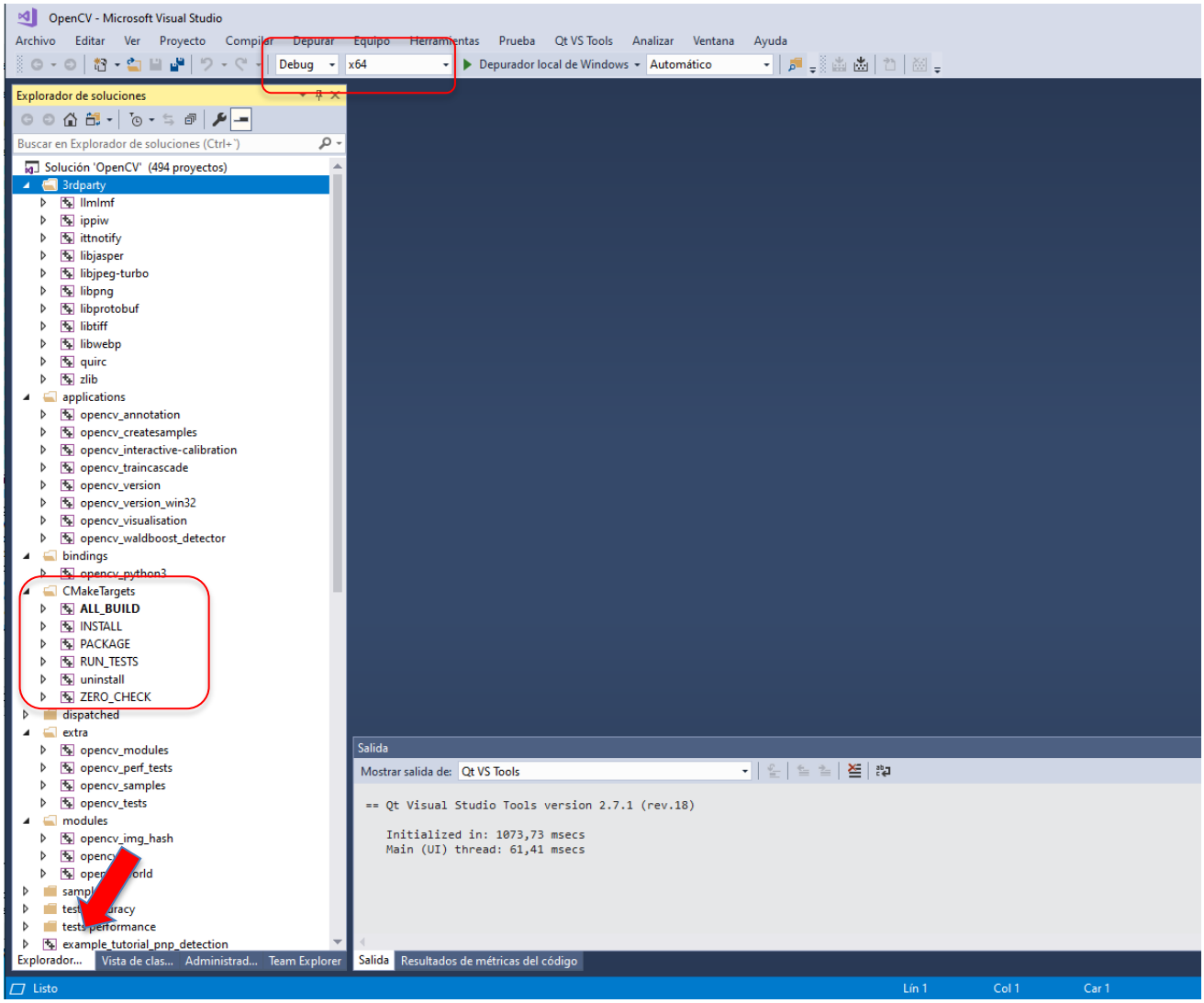


Figura 9. Proyecto librería OpenCV en Visual studio 2017

Buscaremos la carpeta ***CMakeTargets*** que nos permitirá hacer una compilación completa:

a) Compilar el elemento **ALL_BUILD (Debug-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **ALL_BUILD** click derecho – opción compilar (Figura 10) ó seleccionar "***Compilar ALL_BUILD"*** desde menú ***Compilar***. En la ventana central (Salida) se muestra el avance de la compilación que puede tardar bastante tiempo.

b) Compilar el elemento **ALL_BUILD (Release-x64)**: en la pestaña superior seleccionares el modo ***Release*** y la plataforma ***x64***. Seleccionaremos **ALL_BUILD** click derecho – opción compilar (Figura 10) ó seleccionar "***Compilar ALL_BUILD***" desde menú ***Compilar***.

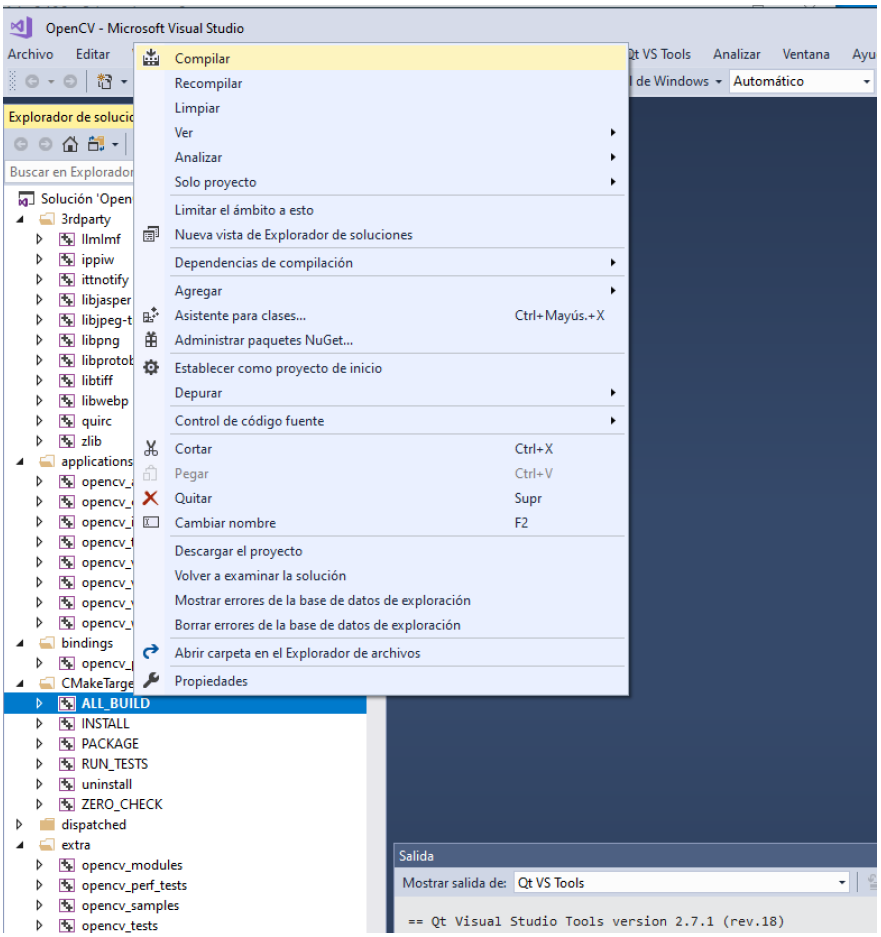


Figura 10. Compilación ALL_BUILD

c) Compilar el elemento **INSTALL (Debug-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **INSTALL** click derecho – opción compilar (Figura 10) ó seleccionar "***Compilar* INSTALL"** desde menú ***Compilar***.

d) Compilar el elemento **INSTALL (Release-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **INSTALL** click derecho – opción compilar (Figura 10) ó seleccionar "***Compilar* INSTALL"** desde menú ***Compilar***.

En la carpeta (***c:/temp/opencv-contrib/install***) (Figura10) disponemos de los ficheros de la librería con toda su estructura de directorios tal como tal como aparece en la versiones precompiladas. Podemos copiarlas en la carpeta de instalación estándar de OpenCV (***c:/opencv-contrib***)

Nombre	Fecha de modificación	Tipo	Tamaño
bin	31/05/2021 11:42	Carpeta de archivos	
etc	30/05/2021 22:09	Carpeta de archivos	
include	30/05/2021 22:06	Carpeta de archivos	
x64	30/05/2021 22:06	Carpeta de archivos	
LICENSE	01/04/2021 14:37	Archivo	3 KB
OpenCVConfig.cmake	30/05/2021 11:10	Archivo CMAKE	7 KB
OpenCVConfig-version.cmake	30/05/2021 11:10	Archivo CMAKE	1 KB

Figura 11. Directorio "install"