**GUÍA DE INSTALACIÓN Y CONFIGURACIÓN DEL ENTORNO DE DESARROLLO**

1. Descargar e instalar la distribución de **Linux Ubuntu 15.04** desde la página oficial de ubuntu, accesible a través de la siguiente línea de comandos:

# wget <http://releases.ubuntu.com/15.04/ubuntu-15.04-desktop-amd64.iso>

2. Si disponemos de una versión de Debiun o Ubuntu que no esté actualizada, es conveniente actualizarlo previamente a través de:

# sudo apt-get update

# sudo apt-get upgrade

# sudo apt-get dist-upgrade

# sudo apt-get autoremove

2. Para instalar la última versión de OpenCV de forma manual en el sistema es preciso eliminar algunas librerías del repositorio previa a su actualización posterior. Se procederá ejecutando, como primera orden, la siguiente línea de comandos desde el terminal de ubuntu 15.04:

# sudo apt-get autoremove libopencv-dev python-opencv

3. El sistema linux donde vamos a proceder con la instalación de OpenCV es instalar cualquier instalación anterior de OpenCV, FFmpeg y otras dependencias previamente instalado.

# sudo apt-get -qq remove ffmpeg x264 libx264-dev

4. Instalar las dependencias requeridas por OpenCV. Las clasificaremos en los siguientes apartados:

4.1 Herramientas de desarrollo

# sudo apt-get install build-essential cmake automake libopencv-dev checkinstall pkg-config aptitude

4.2 GUI

# sudo apt-get install libvtk6-dev

Opcionalmente podremos instalar la biblioteca multiplataforma QT5, empleada para desarrollar aplicaciones y GUI's.

# sudo apt-get install qt5-default

4.3 Componentes de entrada y salida para Video y Media. Algunas de estas librerías son opcionales en función de lo que vayamos a desarrollar, como es el caso del empleo de librerías de streaming para vídeo o soporte de Firewire de videocámaras.

# sudo apt-get install zlib1g-dev libjpeg-dev libwebp-dev libpng-dev libtiff5-dev libjasper-dev libopenexr-dev libgdal-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev v4l-utils libv4l-dev libmp3lame-dev libopus-dev

Gstreamer es un framework multimedia libre y multiplataforma que permite crear aplicaciones audiovisuales. Para instalarlo en Ubuntu 15.04 se lanzará la siguiente orden:

# sudo apt-get install gstreamer1.0

Algunas librerías han dejado de formar parte del repositorio oficial de Ubuntu para algunas versiones, como sucede para la versión 14.04 LTS. Podemos instalarlas desde otros repositorios no oficiales o desde la página de FFmpeg (<https://www.ffmpeg.org/download.html>), que consiste en una solución multiplataforma para grabar, convertir y transmitir audio y video.

Si la opción elegida es realizar la instalación del paquete FFmpeg usando repositorios, han de ejecutarse las siguientes líneas de comando:

# sudo add-apt-repository ppa:mc3man/trusty-media

# sudo apt-get update

# sudo apt-get install ffmpeg gstreamer0.10-ffmpeg

O bien, descargarlo desde la página e instalarlo en el sistema ejecutando las siguientes instrucciones desde el terminal:

# mkdir ~/ffmpeg

# cd ~/ffmpeg

# wget <http://ffmpeg.org/releases/ffmpeg-2.6.2.tar.bz2>

# sudo tar jxvf ffmpeg-2.6.2.tar.bz2

# cd ffmpeg-2.6.2

# PATH="$HOME/bin:$PATH" PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig"

# ./configure

# make

# make install

Si trabajamos sobre Ubuntu 15.04, la multiplataforma FFmpeg regresa al repositorio oficial de este, lo que nos permite su instalación mediante la siguiente línea de comandos:

# sudo apt-get install ffmpeg

4.4 Instalación opcional de la multiplataforma Xine para reproducir y decodificar archivos multimedia.

# sudo apt-get install libxine2-dev

4.4 Biblioteca de procesamiento de código paralelo y lineal (**opcional**):

# sudo apt-get install libtbb-dev libeigen3-dev

# sudo apt-get install libblas-dev liblapack-dev // Subrutinas básicas de álgebra lineal.

4.5 Instalar entornos de desarrollo de **python** y librerías de uso más común necesarias para determinados procesos de OpenCV. **NumPy** es el paquete fundamental para la computación científica con Python que puede ser usado como un contenedor multi-dimensional eficiente de datos genéricos.

# sudo apt-get install python-dev python-tk python-numpy python3-dev python3-tk python3-numpy python-support

4.6 También es opcional instalar la documentación asociada a la plataforma:

# sudo apt-get install doxygen sphinx-common texlive-latex-extra

4.7 El proyecto va a ser desarrollado en **JAVA**, por lo que instalaremos el jdk y el IDE NetBeans 8.0.2 como entorno de desarrollo.

4.7.1 El primer paso consiste en descargar el jdk y el jre desde la página de ORACLE, instroduciendo desde el terminal de ubuntu las siguientes líneas de órdenes:

# cd Descargas

# wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u45-b14/jdk-8u45-linux-x64.tar.gz>

# tar xvzf [jdk-8u45-linux-x64.tar.gz](http://download.oracle.com/otn-pub/java/jdk/8u45-b14/jdk-8u45-linux-x64.tar.gz)

# rm [jdk-8u45-linux-x64.tar.gz](http://download.oracle.com/otn-pub/java/jdk/8u45-b14/jdk-8u45-linux-x64.tar.gz)

# cd jdk1.8.0\_45/

# sudo mv jdk1.8.0\_45/ /usr/local/java

# sudo gedit /etc/profile

Añade al final de este fichero las siguientes líneas:

JAVA\_HOME=/usr/local/java/jdk1.8.0\_45

JRE\_HOME=$JAVA\_HOME/jre

PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin:$JRE\_HOME/bin

export JAVA\_HOME

export JRE\_HOME

export PATH

Informar a sistema Ubuntu la ubicación de JAVA ORACLE JDK/JRE:

# sudo update-alternatives --install "/usr/bin/java" "java" "/usr/local/java/jdk1.8.0\_45/bin/java"

# sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/local/java/jdk1.8.0\_45/bin/javac" 1

# sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/local/java/jdk1.8.0\_45/bin/javaws" 1

Informar al sistema Ubuntu que JAVA 8 ORACLE JDK/JRE debe ser el predeterminado:

# sudo update-alternatives --set java /usr/local/java/jdk1.8.0\_45/bin/java

# sudo update-alternatives --set javac /usr/local/java/jdk1.8.0\_45/bin/javac

# sudo update-alternatives --set javaws /usr/local/java/jdk1.8.0\_45/bin/javaws

4.7.2 Otra forma de proceder con la instalación de JAVA 8 ORACLE JDK/JRE a través del repositorio consiste en ejecutar, a través del terminal de Ubuntu 15.04, la siguiente línea de comandos:

# sudo apt-get install openjdk-8-jdk ant

Opcionalmente podremos instalar scritps de ayuda para empaquetar programas en Java:

# sudo apt-get install javahelper

4.7.3 Descargar el IDE NetBeans 8.0.2 ejecutando, desde el terminal, las siguientes líneas de comando para proceder con su instalación:

# cd ~/Descargas

# sudo wget <http://download.netbeans.org/netbeans/8.0.2/final/bundles/netbeans-8.0.2-linux.sh>

# sudo chmod +x [netbeans-8.0.2-linux.sh](http://download.netbeans.org/netbeans/8.0.2/final/bundles/netbeans-8.0.2-linux.sh)

# ./[netbeans-8.0.2-linux.sh](http://download.netbeans.org/netbeans/8.0.2/final/bundles/netbeans-8.0.2-linux.sh)

4.8 Descargar e instalar la última versión estable de OpenCV. La instalación expuesta en este informe se referirá a la versión OpenCV 3.0. Los pasos a ejecutar desde un terminal son los expuestos a continuación:

# sudo -i

# mkdir /usr/local/opencv

# cd /usr/local/opencv/

# wget <https://github.com/Itseez/opencv/archive/3.0.0.zip>

# unzip 3.0.0.zip

# cd opencv-3.0.0/

# mkdir build

# cd build

# sudo cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D WITH\_TBB=ON -D WITH\_CUBLAS=ON -D BUILD\_EXAMPLES=ON -D INSTALL\_C\_EXAMPLES=ON -D INSTALL\_PYTHON\_EXAMPLES=ON -D WITH\_PNG=ON -D WITH\_FFMPEG=ON -D WITH\_JPEG=ON -D WITH\_TBB=ON -D BUILD\_DOCS=ON -D WITH\_V4L=ON -D WITH\_VTK=ON -D WITH\_QT=ON -D WITH\_OPENGL=ON -D OPENCV\_EXTRA\_MODULES\_PATH=../../opencv\_contrib-master/modules ..

Debemos comprobar que el comando cmake ejecutado en la última línea no produce ningún error. La salida de este comando nos indica qué dependencias se incluirán en la instalación de OpenCV, así como qué otras no se encuentran en el sistema y, por lo tanto, podrán incurrir en errores durante la creación de proyectos futuros usando determinadas librerías.

OpenCV cuenta con una estricta relación de dependencias sin las cuales no funcionará adecuadamente. Pero también se estima conveniente la existencia de ciertos módulos opcionales en el sistema para potenciar las funcionalidades de OpenCV. Entre estas dependencias se recomienda detectar favorablemente Java, Python, TBB, OpenGL, V4L, OpenGL y Qt para obtener un entorno suficientemente funcional y adaptado.

Uno de los posibles problemas encontrados en la salida de cmake es no encontrar las dependencias de Java: JNI, ant, Java wrappers y Java tests. Si hemos instalado el jdk siguiendo los pasos expuestos en esta guía es muy probable que las mencionadas dependencias se encuentren instaladas en tu sistema. Sólo debejos ajustas la variable de entorno $JAVA\_HOME para localizarlas. Para ello ejecuta las siguientes instrucciones:

# sudo su

# gedit ~/.profile

Añade al final de este fichero las siguientes líneas:

JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

JRE\_HOME=$JAVA\_HOME/jre

PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin:$JRE\_HOME/bin

export JAVA\_HOME

export JRE\_HOME

export PATH

Si tras reiniciar el sistema no se encuentra la ruta del framework JNI, así como Java wrappers y Java Tests : *Could not find JNI (missing: JAVA\_INCLUDE\_PATH JAVA\_INCLUDE\_PATH2 JAVA\_AWT\_INCLUDE\_PATH)*, podremos exportar manualmente la variable $JAVA\_HOME mediante la siguiente línea de comandos:

# export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Reinicia el sistema operativo y vuelve a ejecutar la instrucción 'cmake' con los mismos parámetros. La salida perteneciente a Java deberá identificar correctamente la ruta de las dependencias descritras.

# make -j $(nproc)

# make install

Para que OpenCV funcione correctamente es necesario configurar la ruta de búsqueda de librerías del sistema. Se procederá creando el fichero **opencv.conf** y agregando la línea '/ usr / local / lib'. Todo este proceso puede automatizarse lanzando el comando siguiente:

# sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'

# sudo ldconfig

Por último se debe reiniciar el sistema para que los cambios tengan efecto.

Ajustar la variable de entorno PKG\_CONFIG\_PATH ya que las bibliotecas de OpenCV no están en una localización estándar. Se soluciona agregando las siguientes líneas al fichero **/etc/bash.bashrc:**

PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig

export PKG\_CONFIG\_PATH

5. Compilar y ejecutar los ejemplos

# cd opencv/opencv-3.0.0/samples/

# sudo cmake .

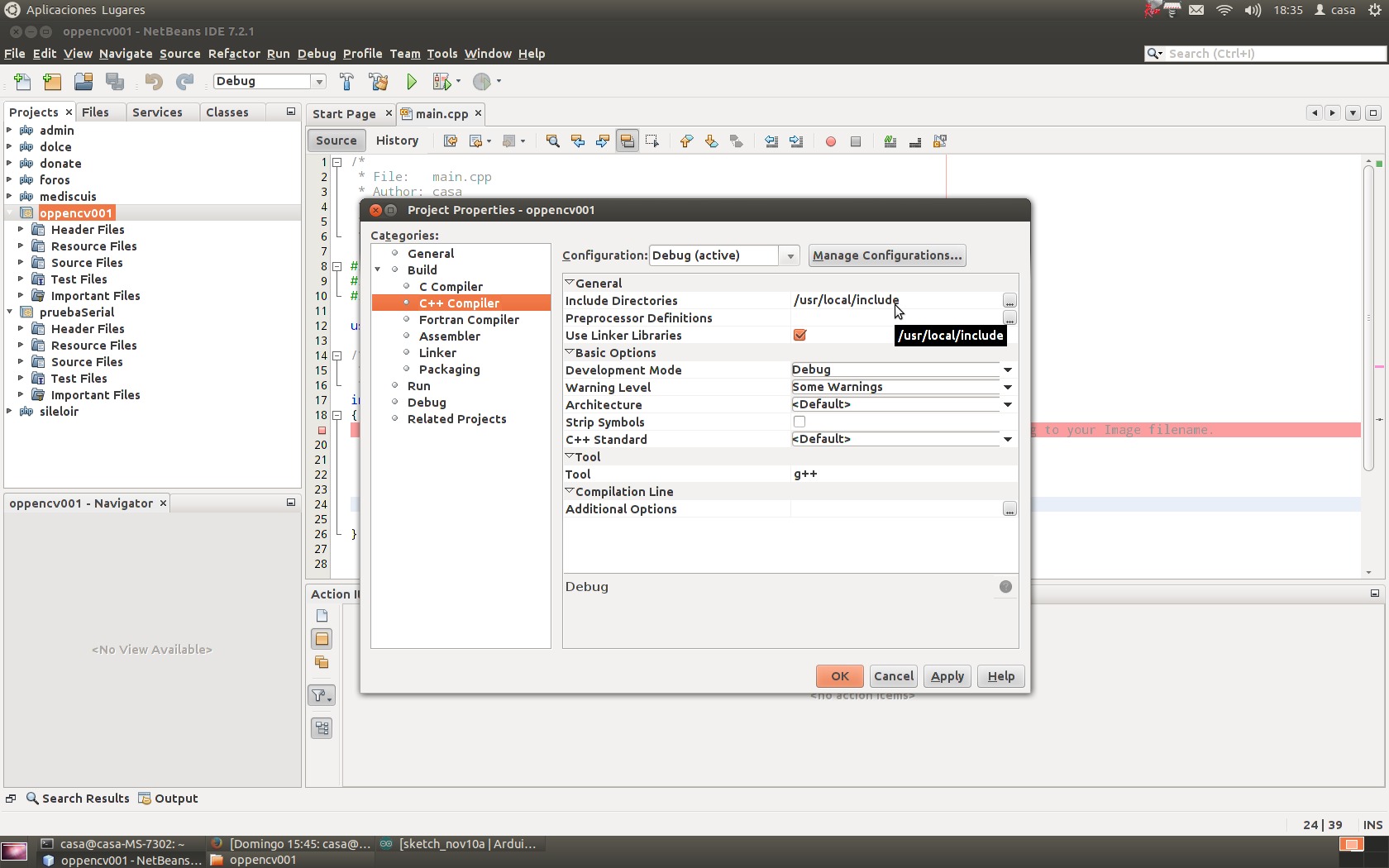
# sudo make

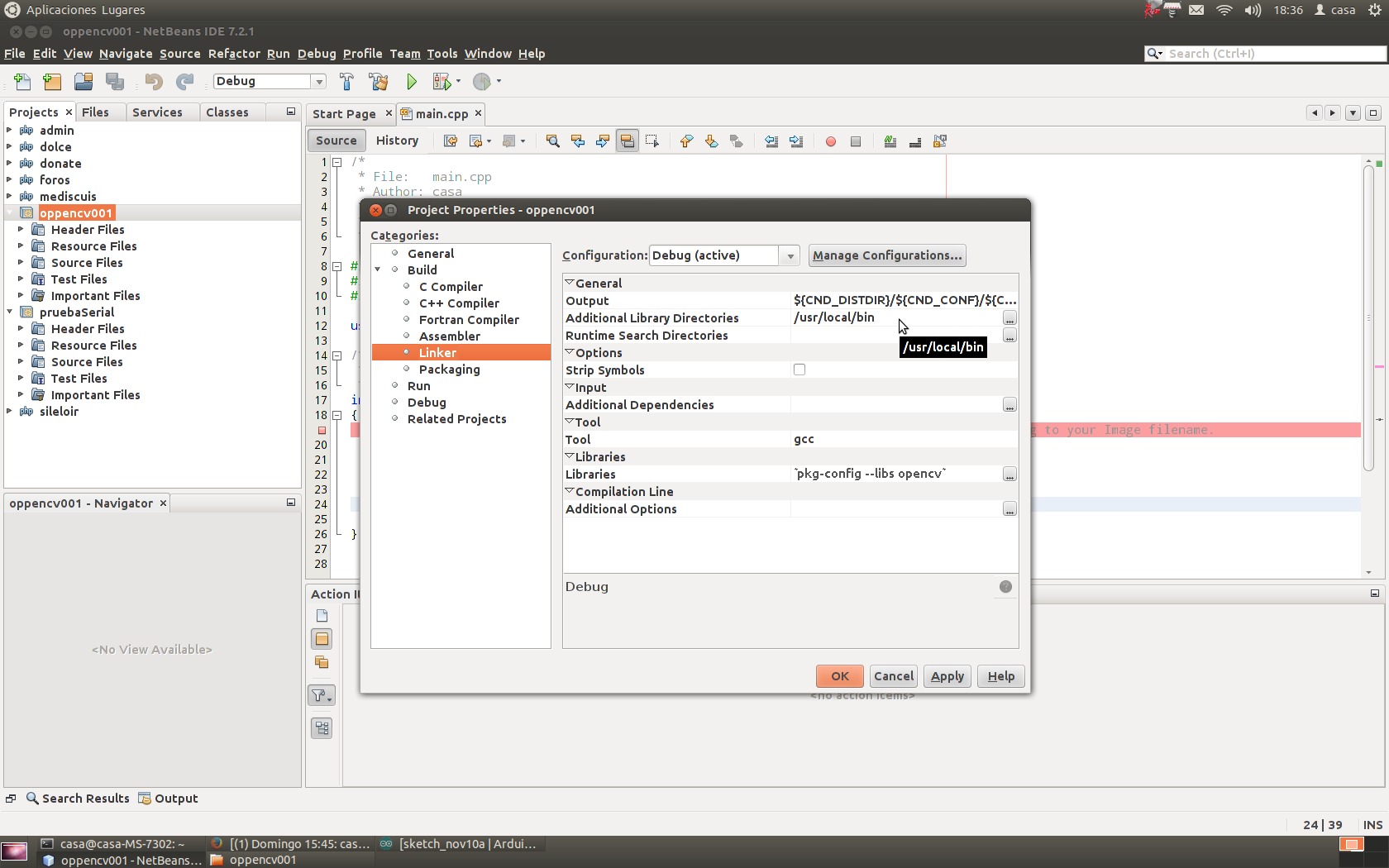
# cd cpp/

# ./cpp-example-facedetect ../data/lena.jpg

6. Configurar NetBeans 8.0.2 para desarrollar un proyecto en C++ usando OpenCV 3.0.

Luego de descargar y/o configurar/instalar el netbean. Procederemos a crear un proyecto de C++.   
Una vez creado el proyecto, le daremos click derecho en subventana de navegacion y luego iremos a propiedades.   
  
Entramos a Build y luego a Linker.   
Ahí:   
pondremos:   
/usr/local/bin   
En Aditional librery directories   
  
Y en Libreries:   
`pkg-config --libs opencv`   
(O lo pueden hacer manualmente. En libreries le dan click al boton que tiene "..." y luego se abre una ventana, ahi hacen click en Add PkgConfig Librery y busca OpenCV o opencv   
Aceptar y listo)   
  
Y dentro de Build pero en la subpestaña que dice C++ Compiler:   
  
En Include DIrectories: /usr/local/include   
  
y Cambiamos en Tool, gcc por g++.   
  
Aceptamos y listo.   
  
Ya tenemos configurado nuestro primer proyecto en OpenCV





7. Instalación del OCR Tesseract:

7.1 Requiere Leptonica :

# cd /usr/local

# wget <http://www.leptonica.com/source/leptonica-1.72.tar.gz>

# sudo tar xvzf [leptonica-1.72.tar.gz](http://www.leptonica.com/source/leptonica-1.72.tar.gz)

# cd leptonica-1.72

# sudo ./configure

# sudo make

# sudo make install

7.2 Descargar e installar Tesseract OCR:

# cd /usr/local

# wget <http://archive.ubuntu.com/ubuntu/pool/universe/t/tesseract/tesseract_3.04.00.orig.tar.gz>

# sudo tar -xvzf [tesseract\_3.04.00.orig.tar.gz](http://archive.ubuntu.com/ubuntu/pool/universe/t/tesseract/tesseract_3.04.00.orig.tar.gz)

# cd tesseract-3.04.00/

# ./autogen.sh

# ./configure

# sudo make

# sudo make install

# sudo ldconfig

If you want the training tools (3.04), you will also need to run the following commands:

# sudo make training

# sudo make training-install

7.3 Descargar el fichero de datos del idioma

# cd /usr/local/tesseract-3.04.00/

# wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tesseract-spa/tesseract-spa\_3.04.00.orig.tar.gz

# tar -xvzf tesseract-spa\_3.04.00.orig.tar.gz

# sudo cp -R tesserct-ocr-spa/tessdata/\* tessdata/

# sudo cp -R tessdata/\* /usr/local/share/tessdata

Si no podemos llevar a cabo este procedimiento por no estar el directorio de destino vacío, se procederá copiando los ficheros contenidos en **tesseract-3.04.00/tessdata** (que acabamos de descomprimir) en el directorio **/usr/local/share/tessdata**.

Compilar con el siguiente código:

**g++ -I/usr/local/include `pkg-config --cflags --libs opencv tesseract`**

Descargar el repositorio de módulos extras de OpenCV:

# cd /usr/local/opencv/[opencv-3.0.0](https://github.com/Itseez/opencv_contrib/archive/master.zip)

# wget <https://github.com/Itseez/opencv_contrib/archive/master.zip>

# sudo unzip master.zip