

Estimados alumnos,

bienvenidos al curso de Visión Artificial del Grado en Ingeniería Informática.

Este es un curso de introducción a la visión por computador que tiene una parte teórica y una parte práctica. El material de estudio (parte teórica) está cubierto por los apuntes proporcionados por el equipo docente, salvo en algunos temas puntuales en los que se dará material adicional. También ponemos a vuestra disposición algunas otras referencias por si tenéis alguna duda (todo este material es gratuito). Además, en internet hay una gran cantidad de material de muy buena calidad. En el fichero “correspondenciaTemario.pdf” tenéis la correspondencia entre el temario y el material de estudio.

Para la parte práctica, usaremos el entorno de programación PYTHON pues tiene una comunidad muy activa de desarrolladores y es un lenguaje para prototipado fácil de aprender, orientado a objetos y fácilmente extensible con librerías de otros lenguajes como C ó C++.

Os recomendamos utilizar anaconda para configurar el entorno.

1. Anaconda (<https://www.anaconda.com/download>) es un gestor de paquetes que contiene todos los paquetes necesarios y es independiente de la plataforma. Dispone de instaladores para Windows, Linux y Mac. Es posible crear configuraciones independientes para distintos proyectos. Tiene un gestor de paquetes y de aplicaciones. En el fichero “crear_entorno_CV.pdf” tenéis las instrucciones para la instalación.

Otras opciones, para los más aventureros:

1. Linux (debian ó ubuntu): la última distribución, en modo TESTING, tiene todos los paquetes necesarios para hacer pruebas con python3 y openCV.
2. Máquina virtual de google. <https://colab.research.google.com/> Google Colaboratory proporciona una máquina virtual python 3 en la nube que permite instalar paquetes con el comando pip como si estuvieran en local. Hay tutoriales en google. Es más lento y hay que aprender a utilizar el entorno de la máquina virtual para subir y bajar material.

Una vez seleccionado el entorno, el entorno que vamos a utilizar está compuesto por los siguientes programas. Dado que cada paquete tiene su propio manual de instalación y de ayuda, no nos extenderemos aquí con los detalles de la instalación, solo comentaremos algunos detalles importantes:

—
Python3: Python 3.7

Código con extensión “.py”. Ojo, a veces python hace referencia a python 2.7 y python 3 se referencia como python3. Dado que python 3 llega ya por la versión 3.7, parece recomendable trabajar con python3 y no con python 2 (el nuevo código se hace para python 3).

—
ipython es el entorno interactivo de python (una terminal python).

—
Spyder: entorno integrado de programación y depuración. Permite depuración paso a paso y evaluar el estado de las variables o realizar pruebas tras un paso de la ejecución del programa.

Jupyter Notebook: entorno que facilita combinar código con explicaciones. Ideal para documentar y compartir código (y para realizar memorias). Los ficheros tienen extensión “.ipynb”.

Paquetes de python:

- En el código de un programa python, al inicio, se hace mención a los paquetes necesarios (import xxx). Unos paquetes son genéricos de python (os, sys, ...), otros dependen de Numpy (una extensión para cálculo científico) y otros son específicos de procesamiento de imagen (pillow, scikit-image) y de visión por computador (opencv cv2).
- se instalan desde anaconda con conda o con el comando pip desde la línea de comandos del terminal. Conda y pip son dos gestores de paquetes, por lo que lo recomendable es utilizar el gestor de anaconda, pero si algún paquete no existe en anaconda, es posible cargarlo desde pip (ver ejemplo de opencv-contrib-python en “crear_entorno_CV.pdf”).

Libros recomendados:

* Además de los libros recomendados para la asignatura, los siguientes libros digitales, disponibles en safaribooks, contienen abundantes ejemplos de código básico y avanzado de visión artificial:

- Saurabh Kapur (2017). Computer Vision with Python 3. Packt Publishing <https://proquest-safaribooksonline-com.ezproxy.uned.es/book/programming/python/9781788299763>
- <https://github.com/PacktPublishing/Computer-Vision-with-Python-3>
- Alexey Spizhevoy and Aleksandr Rybnikov (2018). OpenCV 3 Computer Vision with Python Cookbook. Packt Publishing <https://proquest-safaribooksonline-com.ezproxy.uned.es/book/programming/python/9781788474443> <https://github.com/PacktPublishing/OpenCV-3-Computer-Vision-with-Python-Cookbook>
- Abhinav Dadhich (2018) Practical Computer Vision (**con OpenCV**) <https://proquest-safaribooksonline-com.ezproxy.uned.es/book/programming/machine-learning/9781788297684>
- A Byte of Python. <https://python.swaroopch.com>
- Solem J.E (2012). Programming Computer Vision with Python. O'Reilly. <http://programmingcomputervision.com/>

En resumen, se recomienda seguir los siguientes pasos:

1. Instalar Anaconda, con ello tendréis disponible 1) una aplicación desde la que configurar el entorno de python, 2) un terminal donde ejecutar los comandos pip para instalar otros paquetes no disponibles en anaconda y 3) enlaces a las aplicaciones spyder y jupyter notebook configurados para ser ejecutados sobre ese entorno. Si trabajáis en linux, mirad si la distribución tiene ya configurada openCV con python 3 (paquete python3-opencv ó opencv-contrib-python).

Dentro del entorno Anaconda, en cada entorno python se parte de una configuración mínima de paquetes. ES NECESARIO instalar los paquetes que se desee que estén accesibles. Los paquete necesarios para utilizar opencv dentro de python son: opencv, libopencv y py-opencv.

Para instalar una versión concreta de la librería openCV:

```
pip install opencv-contrib-python==3.4.2.17
```

para ver qué versiones hay de una librería podéis usar:

```
pip install opencv-contrib-python==
```

2. Utilizad Spyder para programar y depurar el código y después pasadlo a Jupyter notebook para documentarlo combinando el código con la salida en puntos concretos del programa (se configura desde el submenú “Environments” de Anaconda Navigator).

3. Comenzad a ejecutar el código de ejemplo que viene con los libros recomendados para familiarizaros con el lenguaje de programación y con las librerías de visión. Con Spyder se puede ejecutar el programa paso a paso y evaluar el estado de las variables. Los jupyter notebooks (.ipynb) contienen código mezclado con explicaciones, fórmulas, resultados, etc. Para analizar los resultados, es posible pasar los campos de código a un programa python (.py) y analizarlo en Spyder. O mejor, convertir el fichero .ipynb a .py: **jupyter nbconvert --to script xxx.ipynb**

Por ahora nada más. Probad a ver qué problemas tenéis. Después podéis comenzar a implementar en python los ejemplos ya resueltos de PECs de años anteriores.

Saludos.