

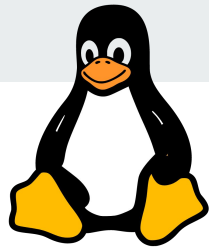


La biblioteca de visión artificial OpenCV

Informática I - Curso R1042 - 2019

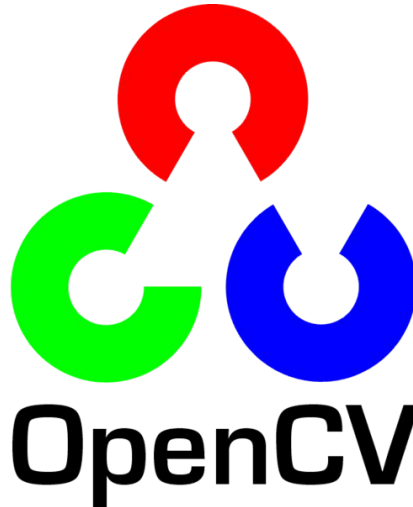
Docente: Ing. Nahuel González
Ayudantes: Ing. Héctor Spataro
 Ing. Javier Balloffet
 Sr. Matías Schwalb

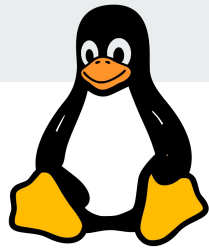
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES



Historia

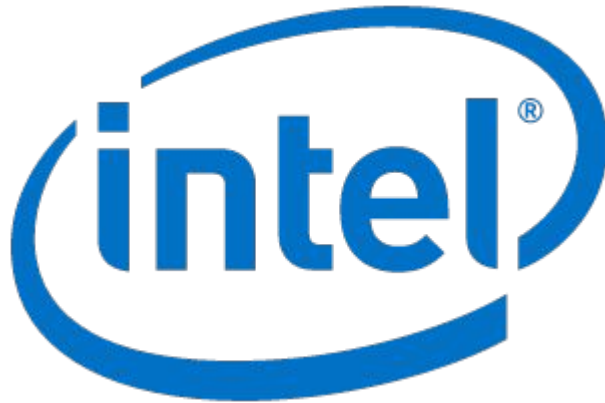
- OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial y machine learning.

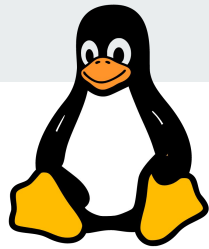




Historia

- Creada originalmente en Intel por Gary Bradski en el año 1999.

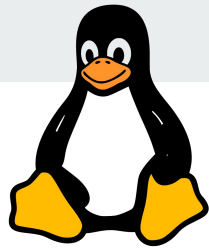




Historia

- En el año 2008, varios miembros del proyecto comenzaron a trabajar en robótica en la empresa Willow Garage, quien comenzó a liderar activamente el proyecto.



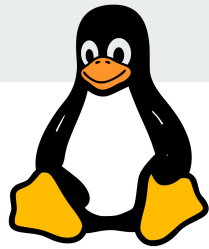


Historia

- En Agosto de 2012, se crea la fundación OpenCV.org quien lidera el proyecto en la actualidad.

OPENCV.ORG

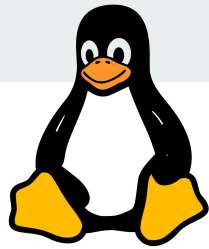




Características

- Multiplataforma: GNU/Linux, Mac OS X, Windows y Android.
- Desarrollada íntegramente en C/C++.
- Interfaces: C (v3.4.7), C++, Python, Java and MATLAB (v4.1.1).

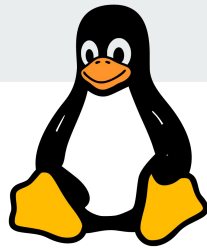




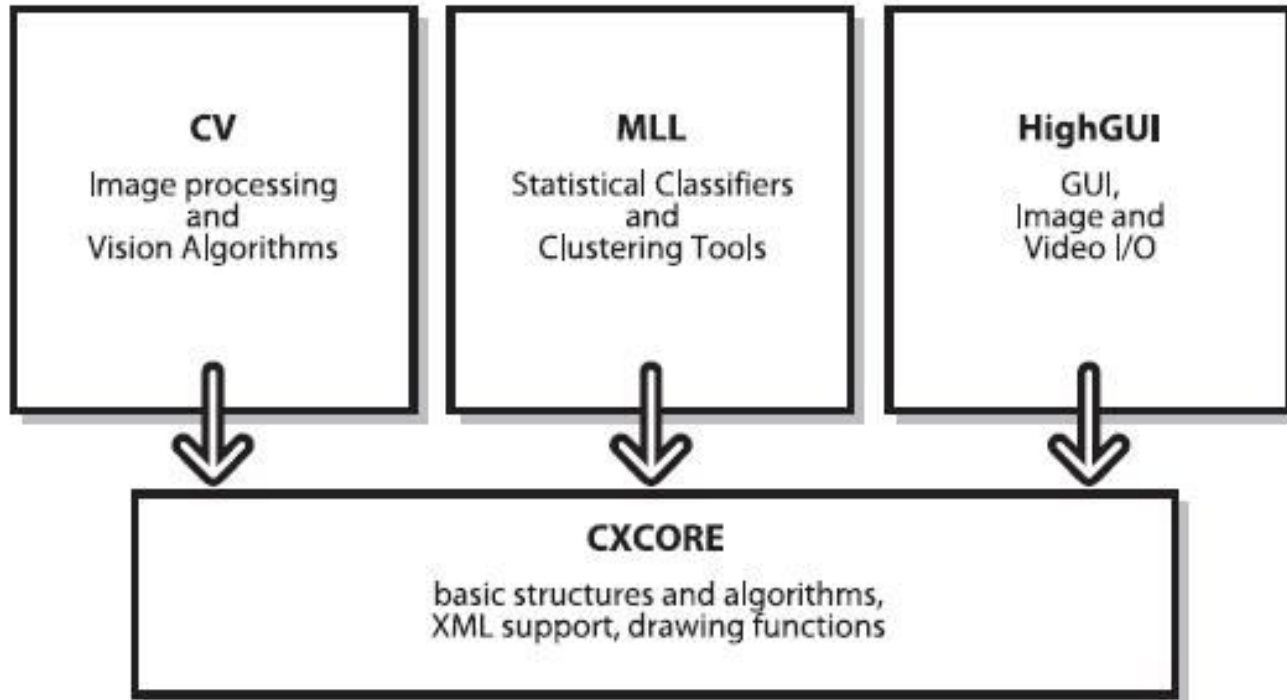
Características

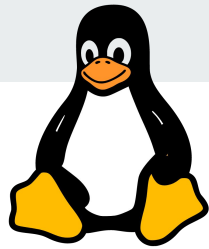
- Tiene estructuras básicas de datos para manipulación y procesamiento de matrices, imágenes y video.
- Posee más de 2500 algoritmos, por ejemplo para detectar y reconocer caras, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, etc.
- Utiliza instrucciones avanzadas del procesador (MMX, SSE).
- Se está trabajando en aumentar la interfaz con instrucciones de GPU.



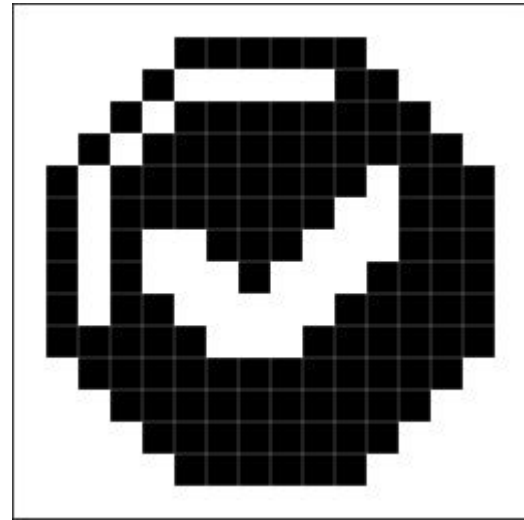


Arquitectura

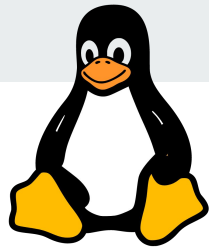




¿Qué es una imagen?



Unidad de información: **Píxel**

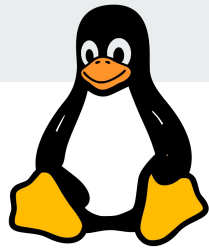


¿Qué es una imagen?

Imagen Blanco y Negro

1 bit por píxel

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	1	1	1	1
1	1	1	1	0	1	1	1	1	1	0	0	1	1	1
1	1	1	0	1	0	0	0	0	0	0	0	0	1	1
1	1	0	1	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	1	1	0	0	0
1	0	1	0	1	1	0	0	0	1	1	1	0	0	0
1	0	1	0	1	1	1	0	1	1	1	0	0	0	0
1	0	1	0	0	1	1	1	1	1	0	0	0	0	0
1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1
1	1	1	1	1	0	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

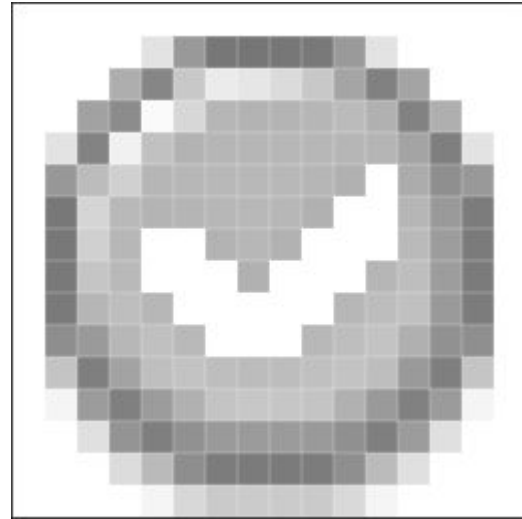


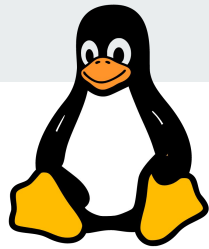
¿Qué es una imagen?

Imagen en **Escala de Grises**

8 bits (1 byte) por píxel

Negro -> 0 Blanco -> 255



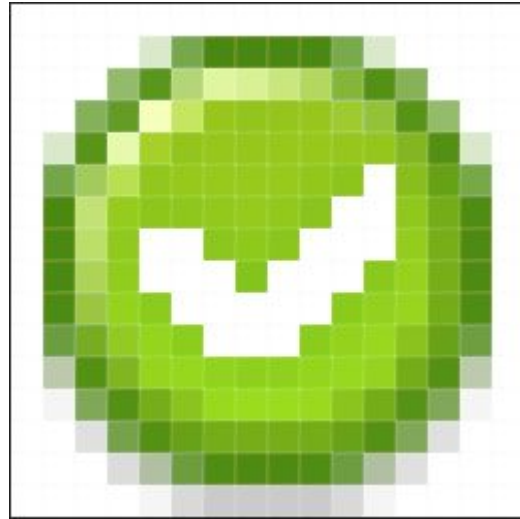


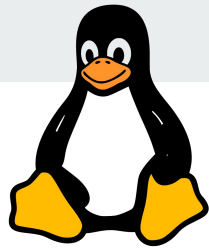
¿Qué es una imagen?

Imagen **Color** (RGB)

24 bits (3 bytes) por píxel

Red, Green y Blue





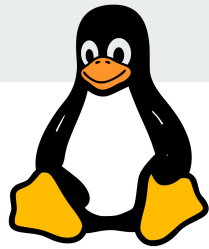
¿Qué es una imagen?

Imagen **Color** (RGBA)

32 bits (4 bytes) por píxel

Red, **Green**, Blue y **Alpha**



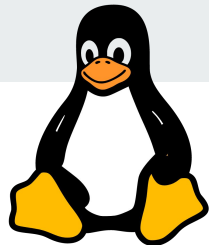


¿Qué es una imagen?

#FF0000	RGB (255, 0, 0)
#FF7F00	RGB (255, 127, 0)
#FFFF00	RGB (255, 255, 0)
#00FF00	RGB (0, 255, 0)
#0000FF	RGB (0, 0, 255)
#4B0082	RGB (75, 0, 130)
#8F00FF	RGB (143, 0, 255)

SuddenWhims.com





Funciones básicas

Cargar una imagen

```
IplImage* image = cvLoadImage("lenna.png");
```

Crear y ubicar una ventana

```
cvNamedWindow("Ejemplo Color", CV_WINDOW_AUTOSIZE);  
cvMoveWindow("Ejemplo Color", 100, 100);
```

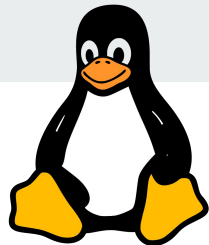
Mostrar la imagen en la ventana creada

```
cvShowImage("Ejemplo Color", image);
```

Liberar recursos

```
cvDestroyAllWindows();  
cvReleaseImage(&image);
```





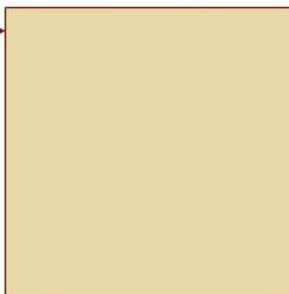
La estructura IplImage

Image structure

IplImage *

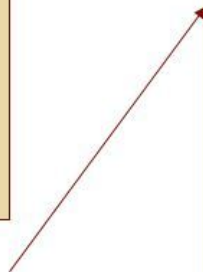
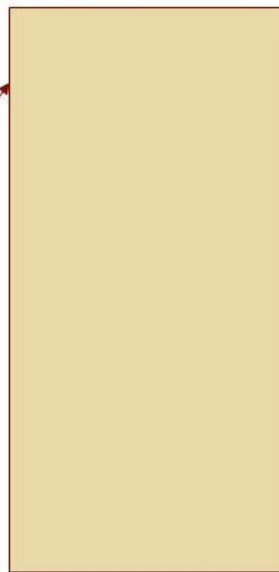


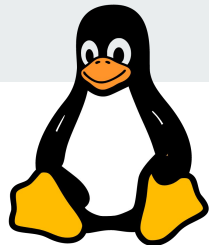
IplImage



- Width
- Height
- Bits per pixel
- Channel sequence
- ...
- pointer to image data

Data

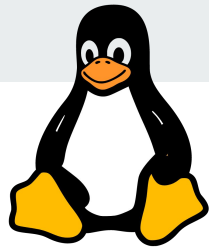




La estructura IplImage

```
typedef struct _IplImage {
    int    nSize;
    int    ID;
    int    nChannels;
    int    alphaChannel;
    int    depth;
    char    colorModel[4];
    char    channelSeq[4];
    int    dataOrder;
    int    origin;
    int    align;
    int    width;
    int    height;
    struct _IplROI *roi;
    struct _IplImage *maskROI;
    void    *imageId;
    struct _IplTileInfo *tileInfo;
    int    imageSize;
    char    *imageData;
    int    widthStep;
    int    BorderMode[4];
    int    BorderConst[4];
    char    *imageDataOrigin;
} IplImage;
```

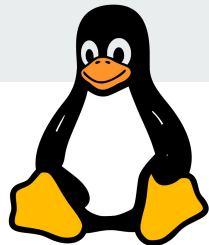




La estructura `IplImage`

- **`char* imageData`**: Imagen en memoria, donde cada elemento es un píxel (1 a 4 bytes dependiendo de si es color o no).
- **`int imageSize`**: Tamaño en bytes de la imagen.
- **`int nChannels`**: 1 canal para escala de grises, 4 canales para RGBA (colores + transparencia).
- **`int width`**: Ancho de la imagen en píxels.
- **`int height`**: Alto de la imagen en píxels.
- **`int widthStep`**: Tamaño en bytes de una fila de la imagen alineada en bytes.

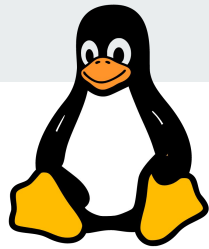




Otras funciones útiles

- **IplImage*** cvCreateImage(**CvSize** size, **int** depth, **int** channels)
- **CvSize** cvSize(**int** width, **int** height)
- **int** cvWaitKey(**int** delay)
- **CvScalar** cvGet2D(**IplImage*** img, **int** idx0, **int** idx1)
- **void** cvSet2D(**IplImage*** img, **int** idx0, **int** idx1, **CvScalar** value)
- **double** cvGetCaptureProperty(**CvCapture*** capture, **int** property_id)
- **IplImage*** cvQueryFrame(**CvCapture*** capture)
- **CvCapture*** cvCreateCameraCapture(**int** index)
- **void** cvCvtColor(**IplImage*** img1, **IplImage*** img2, **int** code)

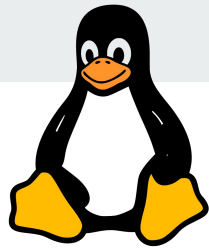




Más funciones útiles

- `CvPoint cvPoint(int x, int y)`
- `CvScalar cvScalar(double red, double green, double blue, double alpha)`
- `void cvCircle(IplImage* img, CvPoint center, int radius, CvScalar color, int thickness, int line_type, int shift)`
- `void cvLine(IplImage* img, CvPoint point1, CvPoint point2, CvScalar color, int thickness, int line_type, int shift)`
- `void cvRectangle(IplImage* img, CvPoint point1, CvPoint point2, CvScalar color, int thickness, int line_type, int shift)`

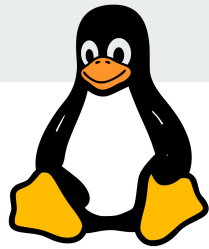




Instalación

<http://www.codebind.com/linux-tutorials/install-opencv-ubuntu-18-04-lts-c-cpp-linux/>



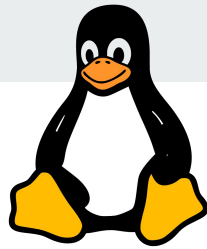


Compilación



```
g++ main.c -o program -Wall `pkg-config --cflags --libs opencv`
```

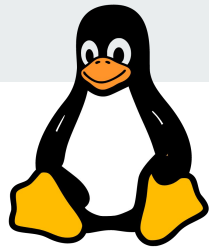




¿Qué aprendimos?

¿Dudas?





Referencias

Diapositivas basadas en:

- Material de referencia del Ing. Alejandro Furfaro
- <https://opencv.org/>
- <https://docs.opencv.org/>
- <http://www.willowgarage.com/>
- <http://www.willowgarage.com/pages/software/opencv>
- <https://en.wikipedia.org/wiki/OpenCV>
- <https://github.com/opencv/opencv>

