

A short horizontal bar with a teal segment on the left and an orange segment on the right.

Introdução ao Opencv

Prof.: Ivan Oliveira Tarifa
SENAI CIMATEC CV Research Group



Sumário da Aula

1. O que é o Opencv
2. Como instalar o Opencv
 - a. Instalando dependências
 - b. Setando as variáveis de cmake
 - c. Instalando opencv e contrib
3. Opções de instalação do Opencv
4. Manipulando entradas
 - a. Arquivos
 - b. Câmera
5. Remapping
6. Filtrando imagens

O que é o Opencv

O que é Open Computer Vision



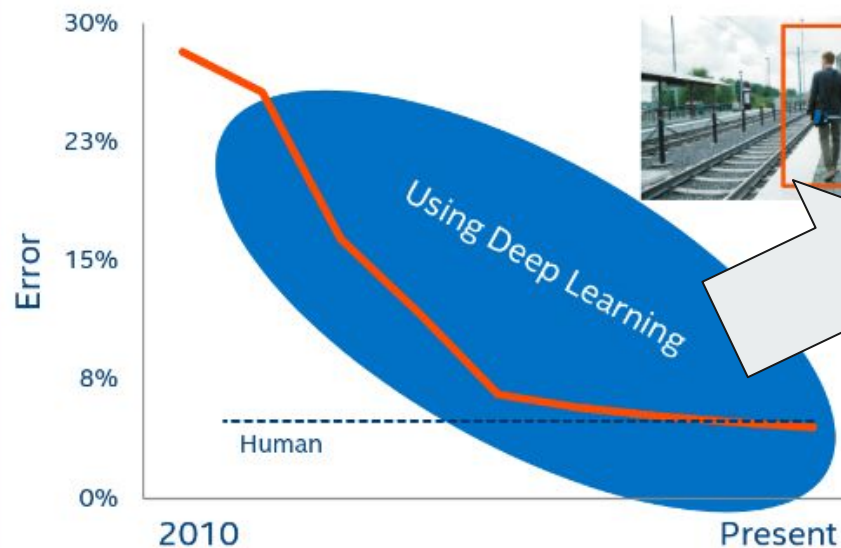
1. O que é o Opencv

- OpenCV
 - ◆ Open Source Computer Vision Library
 - ◆ Publicado no começo dos anos 2000, iniciado na intel
 - ◆ **3-clause BSD License**
- OpenCV 4
 - ◆ c++11
 - ◆ Melhorias na performance
 - ◆ Iniciando o suporte ao android

Deep Learning Usage Is Increasing

Deep learning revenue is estimated to grow from \$655M in 2016 to **\$35B** by 2025¹.

Image Recognition



Traditional
Computer Vision
Object Detection



Deep Learning
Computer Vision
Person
Recognition



Market Opportunities + Advanced Technologies Have Accelerated Deep Learning Adoption

OpenCV 4 Changelog (Parcial)

Major Changes

OpenCV 4.0 is the evolution of OpenCV 3.0. We follow the same design principles and the same library layout as in OpenCV 3.0. So most of the code and most of the scripts should work fine, with a few notable exceptions:

- Most of the C API has been excluded. In particular, instead of `cv::cvtColor(src, dst, CV_RGB2GRAY)` one should use `cv::cvtColor(src, dst, cv::COLOR_RGB2GRAY)`; instead of `cv::VideoCapture cap(0); cap.set(CV_CAP_PROP_WIDTH, 640);` one should use `cv::VideoCapture cap(0); cap.set(cv::CAP_PROP_WIDTH, 640);` etc. Also, the classical C data structures, such as `CvMat`, `IplImage`, `CvMemStorage` etc., as well as the corresponding functions, such as `cvCreateMat()`, `cvThreshold()` etc., are mostly excluded from API and will be completely excluded in further OpenCV 4.x updates. Please, replace them with the corresponding C++ structures and functions, like `cv::Mat`, `std::vector`, `cv::threshold()` etc.

NEW NETWORKS REGISTERED BY THE MODEL ZOO

- Extended OpenCV DNN samples to support various models from [OpenVINO Open Model Zoo](#)
- New networks from TensorFlow Object Detection API: Faster-RCNNs, SSDs and Mask-RCNN with dilated convolutions, FPN SSD



• Performance improvements:

- More optimization using AVX2 instruction set.
- Automatic runtime dispatching is available for large set of functions from `core` and `imgproc` modules.



Instalando o Opencv

Instalando opencv e opencv contrib



Instalação do Opencv

- Dependências
- OpenCV
 - ◆ Determinar as flags de compilação
- OpenCV contrib
 - ◆ Baixar o contrib e os seus módulos



Dependências

→ Python

◆ python-dev python-numpy ...

→ Qt

◆ libqt4-dev libqt4-opengl-dev ...

→ Imagens

◆ libpng-dev libpng++-dev libjpeg-dev libjpeg-progs ...

→ Codecs

◆ libatlas-base-dev libv4l-dev libx264-dev libxvidcore-dev ...



Escolher a versão e clonar o repositório

- Versão do OpenCV e do OpenCV contrib
 - ◆ Para referência vamos utilizar a 4.0, mas pode escolher a de sua preferência



Flags de compilação

- OpenCV utiliza o sistema de build através do cmake (Sendo ele apenas um gerador o sistema de build é através do make)
- Como será estruturado o ambiente:
 - ◆ Teremos o source, build e a instalação em pastas separadas (Não misturar código com build)
 - ◆ Instalação não acontecerá no sistema (Dar mais liberdade para utilizar diferentes versões)
 - ◆ Contrib será instalado
 - ◆ Gráficos será através do Qt
 - ◆ Exemplos serão instalado (para que possamos testar as funcionalidades do OpenCV com facilidade caso precisemos)



→ Flags

- ◆ cmake -D CMAKE_BUILD_TYPE=RELEASE \ -> Tipo de build (pode ser debug)
- ◆ -D CMAKE_INSTALL_PREFIX=/opt/opencv \ (Onde será instalado o OpenCV)
- ◆ -D INSTALL_C_EXAMPLES=ON \ (Se vamos instalar o exemplo em C)
- ◆ -D INSTALL_PYTHON_EXAMPLES=ON \ (Instação de exemplos em python)
- ◆ -D OPENCV_EXTRA_MODULES_PATH=/source/opencv_contrib/modules \ (Pasta com o OpenCV contrib)
- ◆ -D BUILD_opencv_cvv=OFF \ (Esta dependência será desligada devido a uma issue do contrib [Issue](#))
- ◆ -D WITH_TBB=ON \ Threading Building Blocks (intel) [Ex](#)
- ◆ -D WITH_V4L=ON \ (Backend Video I/O para linux)
- ◆ -D WITH_QT=ON \ (Biblioteca gráfica)
- ◆ -D WITH_OPENGL=ON \ (Biblioteca de renderização)
- ◆ -D WITH_GDAL=ON \ (Geographic image [Ex](#))
- ◆ -D WITH_XINE=ON \ (Reprodutor de video)
- ◆ -D BUILD_EXAMPLES=ON (Compilar exemplos)
- ◆ /source/opencv (Local de execução do CMakeLists.txt)



Agora é o make

Primeiros Passos

Entrada e saída





Entrada e saída de dados

```
cv::Mat color= cv::imread("../samples/corrosion.png");
```

```
    cv::Mat gray= cv::imread("../samples/corrosion.png", cv::IMREAD_GRAYSCALE);
```

```
auto video capture = cv::VideoCapture(video file);
```

```
auto video capture = cv::VideoCapture(0); -> Capturar do socket de video /dev/video0
```



Interpretador de linha de comando

https://docs.opencv.org/3.4/d0/d2e/classcv_1_1CommandLineParser.html#a190215863579a6d244dcfd261ee0224



Codando

Cores



Remapeando as cores

- Filtros simples porém interessante
 - ◆ RC (red, cyan): Technicolor (< 1940)
 - ◆ RGV (red, green, value).
 - ◆ CMV (> 1980)
 - ◆ HSV (Pode ser preferido ao processar imagens)



Curve Remapping

- Aplicação de uma função de interpolação dentro do espaço da imagem
 - ◆ Linear
 - ◆ Quadrático
 - ◆ Cúbico



Kernels e convolução

Pixel fonte -> Pixel Saída

-> RC

-> RGV

...

Região fonte -> Pixel Saída

-> Filtro Bilinear

-> Blur

-> Laplacian (Filtro de contorno)



Codando

Referências

A horizontal bar with a teal segment on the left and an orange segment on the right.

Olhe a wiki e a documentação

- <https://github.com/opencv/opencv/wiki/>
- <https://docs.opencv.org/>



Realização



Federação das Indústrias do Estado da Bahia

AtoS