Vision and Cognitive Systems Project

Berselli Werther, Casarotti Luca, Valentini Laura

Video demo (https://drive.google.com/file/d/1qs-peosM_GF03QwlCEE_H6P9YLmIKaoN/view)

Paintings and people detection weights (https://drive.google.com/file/d/1sbhh2JMP20V8aeaabpkZ2PJkTaRgwKPs/view)

Tested on:

```
Ubuntu 18.04 LTS, Ubuntu 20.04 LTS
CUDA v10.2
CUDNN v7.6.5
OpenCV 4.3
NVIDIA GT 840M, GTX 970, GTX 1650, GTX 1080
```

External dependencies

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install python3-pip
sudo apt-get install git
sudo apt-get install python3-tk
sudo apt-get install python3-pil.imagetk
sudo apt install build-essential
pip3 install matplotlib
pip3 install scikit-image
pip3 install dhash
pip3 install pandas
pip3 install keyboard
```

Make sure that your default gcc and g++ versions are <= 8

CUDA

1. Update your Nvidia GPU driver with the latest Nvidia proprietary one
2. Install CUDA Toolkit following the CUDA Installation Guide (https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html)

cuDNN

Install cuDNN following the cuDNN Installation Guide (https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html)

CUDA Enabled OpenCV with Contrib

1. Install OpenCV dependencies

```
$ sudo apt install build-essential cmake git pkg-config libgtk-3-dev \
    libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \
    libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \
    gfortran openexr libatlas-base-dev python3-dev python3-numpy \
    libtbb2 libtbb-dev libdc1394-22-dev
```

2. Clone opencv and opencv_contrib

```
$ git clone https://github.com/opencv/opencv
$ git clone https://github.com/opencv/opencv_contrib
```

3. Make a directory i.e. build inside opencv directory, build and install the library

```
$ mkdir opencv/build && cd opencv/build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
 -D CMAKE_C_COMPILER=/usr/bin/gcc-8 \
 -D CMAKE_INSTALL_PREFIX=/usr/local \
 -D WITH_CUDA=ON \
 -D ENABLE_FAST_MATH=1 \
 -D CUDA_FAST_MATH=1 \
 -D WITH_CUBLAS=1 \
 -D INSTALL_PYTHON_EXAMPLES=OFF \
 -D INSTALL_C_EXAMPLES=OFF \
 -D OPENCV_GENERATE_PKGCONFIG=ON \
 -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
 -D PYTHON_EXECUTABLE=~/.virtualenvs/cv/bin/python \
 -D WITH_GTK=ON \
 -D ENABLE_PRECOMPILED_HEADERS=OFF \
 -D BUILD_opencv_cudacodec=OFF \
 -D WITH_NVCUVID=OFF \
 -D OPENCV_ENABLE_NONFREE=ON \
 -D BUILD_EXAMPLES=ON ..

$ nproc
# use the number that nproc returns which is the number of cores of your
processor. Let's say it returns 4.
$ make -j4
$ sudo make install
```

Darknet

1. Clone darknet

```
$ git clone https://github.com/AlexeyAB/darknet
```

2. Build it

```
$ ./darknet/build.sh
```

Running the project:

1. Copy libdark.so (or libdarknet.so) from darknet directory to VCS-Project/detection directory and name it libdarknet.so

2. Put the weight linked above in your `/home/username` directory
3. Run the application

```
$ python3 gui.py
```