OpenCV(Python)
Deeplearning4j(Java)

Data Mining (Dr. Mohsen Qholami) _ Kasra Samadi Darestani _ Semester 402-2

TABLE OF CONTENTS

OpenCV

Deeplearning4j

OpenCV Introduction

1. was started at Intel in 1999 and the first release came out in 2000
2. now supports a multitude of algorithms related to Computer Vision and Machine Learning and is expanding day by day
3. supports a wide variety of programming languages such as C++, Python, Java, etc.
4. is available on different platforms including Windows, Linux, OS X, Android, and iOS.
5. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development
6. OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language

OpenCV-Python

1. is a library of Python bindings designed to solve computer vision problems
2. allows you to perform various tasks related to image and video processing, such as loading and saving images, applying various filters and transformations, detecting and recognizing objects, image segmentation, and much more.
3. It provides a wide range of functions and algorithms to handle images and videos efficiently
4. All the OpenCV array structures are converted to from Numpy arrays

OpenCV-Python

- Image and Video I/O

allows you to read and write images and videos in various file formats. You can use functions like cv2.imread() to load an image, cv2.imwrite() to save an image, cv2.VideoCapture() to capture video from a camera or a file, and cv2.VideoWriter() to save video output.

- Image Manipulation

provides functions for resizing, cropping, rotating, and flipping images. You can change the image properties, such as brightness, contrast, and saturation, using functions like cv2.resize(), cv2.flip(), cv2.rotate(), and others

OpenCV-Python

- Image Filtering

You can apply various filters and enhancements to images using functions like cv2.filter2D(), cv2.GaussianBlur(), cv2.medianBlur(), and cv2.equalizeHist(). These functions allow you to perform operations like smoothing, sharpening, noise reduction, and histogram equalization

- Image Segmentation

provides functions for image segmentation, which involves dividing an image into meaningful regions or objects. Functions like cv2.findContours(), cv2.drawContours(), and cv2.watershed() can be used for contour detection, region filling, and watershed segmentation

OpenCV-Python

- Object Detection

includes pre-trained models and functions for object detection and recognition. The most commonly used method is the Haar cascades classifier, which can be used to detect objects like faces, eyes, and smiles. OpenCV-Python also supports deep learning-based object detection frameworks like TensorFlow and PyTorch.

- Feature Detection

offers various algorithms for detecting and describing image features, such as corners, edges, and keypoints. Common functions include cv2.goodFeaturesToTrack(), cv2.Canny(), cv2.cornerHarris(), cv2.ORB_create()

Canny Edge Detection in OpenCV

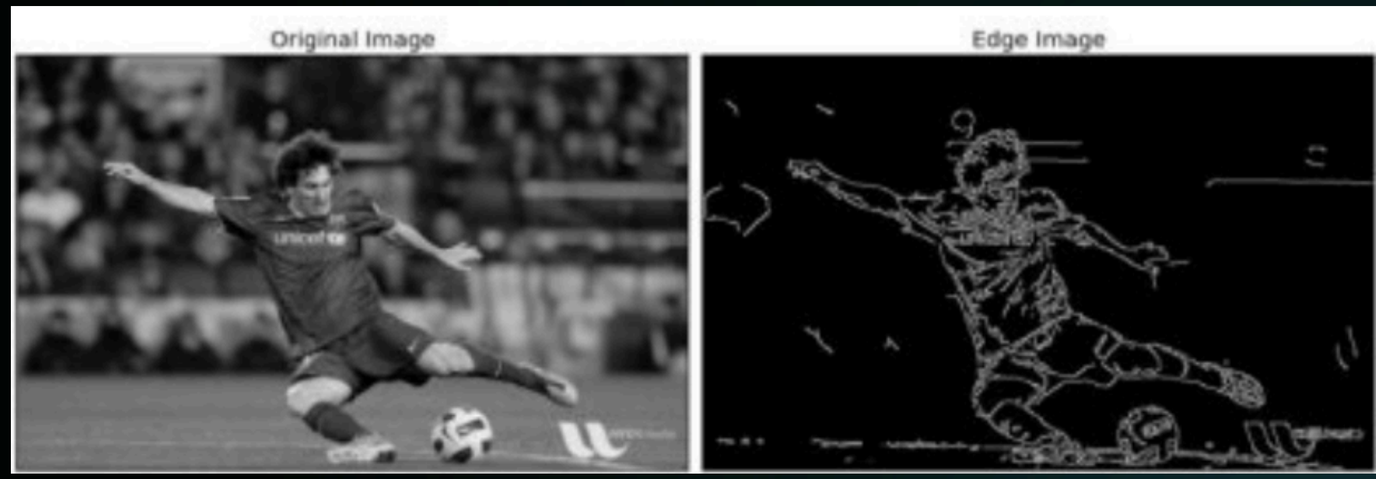
```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('messi5.jpg',
cv.IMREAD_GRAYSCALE)
assert img is not None

edges = cv.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([ ]), plt.yticks([ ])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([ ]), plt.yticks([ ])

plt.show()
```



Deeplearning4j Introduction

1. Eclipse Deeplearning4j is a suite of tools for running deep learning on the JVM
2. is open-source released under Apache License
3. It is powered by its own open-source numerical computing library, and works with both CPUs and GPUs
4. includes implementations of the restricted Boltzmann machine(RBM), deep belief net, deep autoencoder, recursive neural tensor network, word2vec, doc2vec. These algorithms all include distributed parallel versions that integrate with Apache Hadoop and Spark

Deeplearning4j Introduction

1. includes a vector space modeling and topic modeling toolkit, implemented in Java and integrating with parallel GPUs for performance. It is designed to handle large text sets
2. includes implementations of term frequency–inverse document frequency (tf–idf), deep learning, Mikolov's word2vec algorithm, doc2vec, reimplemented and optimized in Java
3. Tensorflow, Keras and Deeplearning4j work together. Deeplearning4j can import models from Tensorflow and other Python frameworks if they have been created with Keras

Installation requirements

Java (developer version) 11 or later

Apache Maven 3.3.x (automated build and dependency manager)

IntelliJ IDEA or Eclipse

Neural network configuration in Deeplearning4j

```
MultiLayerConfiguration config = new NeuralNetConfiguration.Builder()
    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
    .iterations(1000)
    .learningRate(0.1)
    .list()
    .layer(0, new DenseLayer.Builder().nIn(2).nOut(4).activation(Activation.RELU).build())
    .layer(1, new OutputLayer.Builder(LossFunctions.LossFunction.MSE)
        .nIn(4).nOut(1).activation(Activation.SIGMOID).build())
    .pretrain(false).backprop(true)
    .build();
```

Thanks for your attention!