CSCI446/946 Big Data Analytics

Lab Week 9
Image Analytics

School of Computing and Information Technology
University of Wollongong Australia

Introduction

- Deep Neural Networks can be very powerful in image analytics.
  - But they require a lot of training samples.
- If we do not have enough training samples use image augmentation:
  - Increase the number of samples by adding or using distorted versions of the training samples.
  - For example: Create and use randomly scaled, sheared, rotated, … versions of images.

Introduction

- There are many ways by which image augmentation can be performed.
    - https://link.springer.com/article/10.1186/s40537-019-0197-0
- Toolboxes and libraries such as EBImage, magick (in R), torchvision, Augmenter (in python) can make the task easier.
- Examples:
  - EBImage
    - https://www.bioconductor.org/packages/devel/bioc/vignettes/EBImage/inst/doc/EBImage-introduction.html
  - magick:
    - https://cran.r-project.org/web/packages/magick/vignettes/intro.html
  - Augmenter
    - https://towardsdatascience.com/elastic-deformation-on-images-b00c21327372
    - https://augmentor.readthedocs.io/en/master/userguide/mainfeatures.html#elastic-distortions

Task (For Advanced Students)

Use your tool of choice to create 100 randomly distorted versions of the images LetterA.gif and LetterO.gif by:

- Random scale by 80%-120%, **and**
- Random rotation by +-20 degrees, **and**
- Random shear by +-20 degrees, **and**
- Random Gaussian elastic distortion using spread 5, and 18x amplification .

For Beginners (1)

- Image distortions can be performed in a multitude of different ways.
- The following demonstrates a simple way by which image distortions can be achieved.
- Follow those steps:

1. Download the images LetterA.gif and LetterO.gif
2. Start python

```
import Augmentor                #pip3 install Augmentation
p = Augmentor.Pipeline('C:/Users/joe/path_to_image_directory')
```

For Beginners (2)

The images are in B/W. Some of the operations in the "Augmenter" toolbox requires images in grayscale, moreover, "Augmenter" uses black as the background color whereas the background of the images is while. So lets convert the images accordingly:

#Convert to grayscale then invert colors

```
p.greyscale(probability=1)
p.invert(probability=1)
```

For Beginners (3)

#Randomly scale by +-5%
p.zoom(probability=1, max_factor=1.05, min_factor=0.95)

#Randomly rotate by +-5degrees
p.rotate(probability=1, max_left_rotation=5, max_right_rotation=5)

#Randomly shear by +-8degrees
p.shear(probability=1, max_shear_left=8, max_shear_right=8)

#Random Gaussian elastic distortion by spread 3 and amplification 5
p.gaussian_distortion(probability=1, grid_width=3, grid_height=3, magnitude=5, corner='bell', method='in')

#Convert colors back (white background)
p.invert(probability=1)

For Beginners (4)

#Create 10 distorted samples

p.sample(10)

- The output will be saved in the folder “output”.
- Verify the output (view all 10 of the distorted samples) then execute the task for “Advanced students”.

Prep for next week

Install mxnet in R

In R3.4

```
cran <- getOption("repos")
cran["dmlc"] <- "https://s3-us-west-2.amazonaws.com/apache-mxnet/R/CRAN/"
options(repos = cran)
install.packages("mxnet",dependencies = T)
library(mxnet)
```

In R3.6

```
install.packages("https://s3.ca-central-
1.amazonaws.com/jeremiedb/share/mxnet/CPU/3.6/mxnet.zip", repos = NULL)
```

Other R versions:

https://cwiki.apache.org/confluence/display/MXNET/MXNet-R+release+process#MXNetRreleaseprocess-Windows

Install mxnet in python

https://mxnet.apache.org/versions/1.4.1/install/windows_setup.html

Q&A

Images Courtesy of Google Image