GROUP 11: Improved Image Quality Metrics

Zhiwei Tang, Varun Lalwani

April. 10, 2019

Boston University
Department of Electrical and Computer Engineering

Technical report No. ECE-2019-2

BOSTON UNIVERSITY

GROUP 11: Improved Image Quality Metrics

Zhiwei Tang, Varun Lalwani

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

April. 10, 2019

Technical Report No. ECE-2019-2

Contents

1 Introduction

Our goal is to implement two or more modern fidelity metrics, such as SSIM or Wavelet SSIM, which would provide an improved metric for capturing similarities between two images and should perform better than Mean-Square Error (MSE). MSE has shown to be incompetent in accurately measuring image fidelity under human vision for some distortion operators. Using MSE as a metric for fidelity of human perception places an assumption that the spatial and temporal change in the processed signal (e.g. spatial and temporal order of pixels for image signals) doesn't change for the fidelity of the signal, but the order of pixels is critical for an image to be precepted by human.

2 Image Comparison

To further improve the performance of the measurement with the knowledge of computational neuroscience, that human visual system is highly adapted to the natural visual environment, people started to visualize the modeling of natural image source and the human visual system as dual problems. With this knowledge, a new approach named SSIM is introduced. More recently data-driven methods, such as classical regression and neural networks, have been proposed as well. Convolutional Neural Networks have shown to deliver great performance on a wide variety of image of visual information processing applications.

2.1 Principle of the Algorithm

SSIM method reflects the strong neighbor dependencies which is related to the structures of the objects in the visual scene and simulates the structural sensitive nature of human eye. However, for some non-structural distortions that doesn't change much of the fidelity (e.g. relative translation, scaling and rotation of images), the original SSIM shows bad performance. As a result, a wavelet domain of SSIM (CW-SSIM) is introduced. [1] It takes similarity measurement in a complex wavelet domain which is less sensitive to changes in luminance and contrast as well as spatial translation to solve the above problem. One of the principal advantages of deep-learning models are the remarkable generalization capabilities that they can acquire when they are trained on large-scale labeled data sets. Deep Learning models employ multiple levels of linear and non-linear transformations to generate highly general data representations, thereby greatly decreasing dependence on the selection of features, which are often reduced simply to raw pixel values [10], [11].

2.2 Implementation

In this project, we have implemented a Convolutional Neural Network. It composes of 6 convolutional layers, one of which has impulse response matrix of size 6x6 and outputs with Relu as the activation function. Additionally, it also contains 3 Maxpooling layers, which occurs after every 2 Convolutional Layers. It has a dense layer in 7 outputs as the last layer. The dense layer has a SoftMax activation function. The CNN uses an Adam Optimizer, which adjusts the learning rate accordingly. The loss function for the CNN is 'categorical_crossentropy' and the metric for training is accuracy. In total it has 1,734,727 parameters to train.

3 Experimental results

We took 20 outdoor images, resized them to shape of (100x100), then computed three blur version of the original image with three different blurring intensities and we did it for each image. Now we have 4 distinct images for each image and we formed pairs of images by horizontally concatenating 2 images from the 4 images we had, so we 16 pairs of images for each image, so in total we had 320 images to train to check for blur. We computed a prediction on 42 such pair of images, after training our classifier on the 320 images, computed above, with 20 epochs. By looking at the validation accuracy in the last epoch, we can say conclude that we received a 0.7381 accuracy on the testing data. We also noticed that when we tried training with 100 epochs, after about 35 epochs, our model started getting over-trained because our started showing really good training accuracy, but a poor performance in testing accuracy.

4 Conclusions and possible improvements

Although the model is showing a good performance for the amount of data available, it is not as good as other models that have tried reference-based model image quality predictor. This is because they have larger amount of data available. The way to improve our model would to collect larger data. We have to also make our classifier compatible to classify images with noise or shift. We can also try to make our model deeper, i.e. have more convolutional layers and have more dense layers. We can also include the dropout algorithm. We also have yet to implement the SSIM and CW-SSIM algorithm and compare its performance to our Convolutional Neural Network.

Appendix

Below is listed Python source code developed for this project, this code initializes the CNN, and trains it. In this the trainbatches and testbatches variables are used for training and testing”

```
from scipy import signal
import cv2
import numpy as np
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
import tensorflow as tf
from PIL import Image
from sklearn.metrics import confusion_matrix

trainbatches = ImageDataGenerator(rescale= 1./255).flow_from_directory('combos/train',\

target_size=(100,200), \

classes=['0','1','-1','2','-2','3','-3'],\
                                                                                batch_size=16)
testbatches = ImageDataGenerator(rescale= 1./255).flow_from_directory('combos/test',\

target_size=(100,200), \

classes=['0','1','-1','2','-2','3','-3'],\
                                                                                batch_size=7)
classifier = Sequential();
classifier.add(Convolution2D(64,(1,1), input_shape = (100,200,3), activation = 'relu'))
classifier.add(Convolution2D(64,(1,1), input_shape = (100,200,64), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2,2)))
classifier.add(Convolution2D(128,(1,1), input_shape = (50,100,64), activation = 'relu'))
classifier.add(Convolution2D(128,(1,1), input_shape = (50,100,128), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2,2)))
classifier.add(Convolution2D(256,(6,6), input_shape = (25,50,128), activation = 'relu'))
classifier.add(Convolution2D(256,(1,2), input_shape = (20,45,256), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2,2)))
classifier.add(Flatten())
classifier.add(Dense(7,activation = 'softmax'))
#classifier.add(Dense(7,activation = 'sigmoid'))
classifier.compile(optimizer='adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
classifier.fit_generator(trainbatches, steps_per_epoch = 20, validation_data = testbatches, validation_steps
= 6, epochs = 20)
classifier.summary()
```

References

[1] Z. Wang and A.C. Bovik, “Mean squared error: love it or leave it? A new look at signal fidelity measures,” IEEE Signal Processing Magazine, vol. 26, no.1, pp. 98-117, 2009.
[2] J.L. Mannos and D.J. Sakrison, “The effects of a visual fidelity criterion on the encoding of images,” IEEE Trans. Inform. Theory, vol. 20, no. 4, pp. 525–536, 1974.
[3] S. Mallat, A Wavelet Tour of Signal Processing. New York: Academic, 1998.
[4] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger, “Shiftable multiscale transforms,” IEEE Trans. Inform. Theory, vol. 38, no. 9, pp. 587–607, 1992.
[5] E.P. Simoncelli and B. Olshausen, “Natural image statistics and neural representation,” Annu. Rev. Neurosci., vol. 24, pp. 1193–1216, May 2001.
[6] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., Philadelphia, PA, 2005, pp. ii/573-ii/576 Vol. 2.
[7] H.R. Sheikh, M.F. Sabir, and A.C. Bovik, “A statistical evaluation of recent full reference image quality assessment algorithms,” IEEE Trans. Image Processing, vol. 15, no. 11, Nov. 2006, pp. 3449–3451.
[8] Z. Wang, A.C. Bovik, “Reduced-and no-reference image quality assessment,” IEEE Signal processing magazine, vol. 28, no. 6, pp. 29-40, 2011.
[9] Jongyoo Kim, Hui Zeng, Deepti Ghadiyaram, Sanghoon Lee, Lei Zhang, and Alan C. Bovik, “Deep Convolutional Neural Models for Picture-Quality Prediction” IEEE Signal Processing Magazine vol 34, no. 6, pp. 130-141, 2017.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Advances in Neural Information Processing Systems Conf. 2012, pp. 1097–1105.
[11] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell, vol. 35, no. 8, pp. 1798–1828, 2013.
[12] Y. Yuan, Q. Guo, and X. Lu, “Image quality assessment: A sparse learning way,” Neurocomputing, vol. 159, pp. 227–241, July 2015.
[13] H. Sheikh, M. Sabir, and A. Bovik, “A statistical evaluation of recent full reference image quality assessment algorithms,” IEEE Trans. Image Process., vol. 15, no. 11, pp. 3440–3451, 2006.