CNN-based Image Re-scaling Identification

Jia Wei Ngo, Student ID: 717030990005

***Abstract*—The paper covers the review of the implementation details, experimental setup, analyses and experimental results of a proposed CNN model for identifying and classifying true 4K images from 4K images that have been upscaled from a lower resolution.**

***Index Terms*—Convolutional Neural Networks**

I. Introduction

In an increasingly visual digital world, advances in technology such as the prevalence of Ultra-High Definition (UHD) display terminals make UHD videos a major selling point for media consumption. However, due to the limitations of bandwidth and devices used to capture UHD videos, fake 4K videos which have been rescaled/upscaled from a lower resolution plague the internet.

II. Problem Statement

The task is to design one model that can accurately classify real 4K resolution images from false 4K images i.e. images that have been rescaled to 4K resolution from a smaller resolution. A dataset of 200 real 4K images have been provided. The dataset of images have then been downscaled to 1080p, before being upscaled to 4K again with Bicubic [1] and Lanczos [2] interpolation.

In order to visualize the differences between upscaled and original images, we plot the Mean Substracted Contrast Normalization (MSCN) of same 3 images with different rescaling methods in Figure 1.

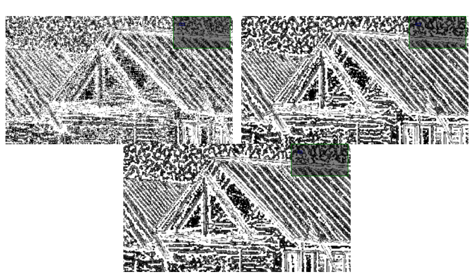

Fig. 1. MSCN of Orginal 4K, Lanczos, Bicubic (left to right)

III. Re-scaling Forensics

There are several state-of-the-art methods that have been proposed to identify transformations that have been conducted on an image. The task at hand is similar to a No-Reference Image Quality Assessment (NR-IQA). NR-IQAs such as the Blind Referenceless Image Spatial QUality Evaluator (BRISQUE) [3] use spatial domain feature extraction strategies to differentiate between natural images and images that have been tampered with. Taking the BRISQUE features of upscaled and original 4K images, a density plot is shown in Figure 2. We can see the clear distinction between original and upscaled images, which shows that BRISQUE features could also be used to classify these images. Natural Scene Statistics (NSS) have also been studied in predicting the up-scaling ratio that an image has undergone from its original resolution [4]. However, for the purpose of this paper, we have opted for a Convolutional Neural Network (CNN).

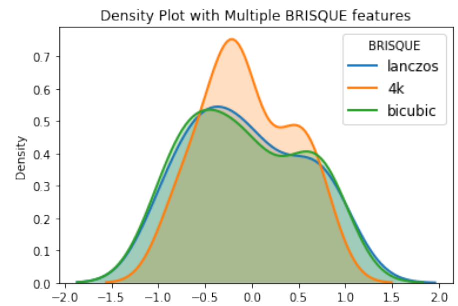

Fig. 2. Density plot of brisque features

IV. Convolutional Neural Networks

CNN architectures have been proven to detect rescaling very reliably [5]. However, it is challenging to work with large images, especially in the 4K dimension (3840x2160). Hence, the proposed solution would be to slice the given images and work on small patches/tiles of 64x64.

A straight-forward CNN design inspired by VGGNet [6] from [5] was chosen. The end-to-end network operates on grayscale images with input patches of 64x64, fed through a total of 25 layers, before batch normalization, leaky ReLu activation and two fully-connected layers into softmax activation. Figure 3 shows the Keras implementation of the CNN model.

A. Experimental Setup

Initially, all 200 4K images have been sliced to non-overlapping 64x64 tiles. However, this caused the dataset to be too large. The proposed solution was to select at random 20 out of the 200 images in the dataset, and slice it near the centre of the image. The hypothesis is that the middle part of images tend to be more focused than the edges, giving us a more reliable dataset.

From slicing the images from original 4K images and upscaled images, we end up with a total of 23,100 64x64 patches. 11,500 original and 11,500 upscaled patches.

We train the network on a total of 18480 patches of size 64x64, and a validation dataset of 4620 patches of size 64x64.

```
model = Sequential()

model.add(Conv2D(16, (3,3), input_shape = (64, 64, 1)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))
model.add(Conv2D(16, (3,3)))

model.add(BatchNormalization())
model.add(LeakyReLU())

model.add(MaxPooling2D(pool_size=(4,4)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))

model.add(Dense(2))
model.add(Activation("softmax"))

# Compiling the model using some basic parameters
model.compile(loss="sparse_categorical_crossentropy"
              ,optimizer="adam"
              ,metrics=["accuracy"])
```

Fig. 3. Keras implementation of CNN model

```
=============== Train dataset metrics ===============
578/578 [==============================] - 69s 120ms/step - loss: 0.1107 - accuracy: 0.9601
[0.11066196460895575, 0.96006495]
=============== Validation dataset metrics ===============
145/145 [==============================] - 16s 109ms/step - loss: 0.0343 - accuracy: 0.9924
[0.03434816428684983, 0.99242425]
=============== Test dataset metrics ===============
19/19 [==============================] - 2s 118ms/step - loss: 0.1039 - accuracy: 0.9633
[0.1039439278997873, 0.9633333]
```

Fig. 4. Results of CNN model on train, validation and test dataset

These patches were selected only from 20 random images out of the dataset.

In the test dataset, we sliced one 64x64 patch from the middle of the image from the 200 4K images given in the dataset, 200 Lanczos upscaled to 4K from 1080p and 200 Bicubic upscaled to 4K from 1080p, with the same assumption that the middle part of the images tend to be more focused and provide us with reliable data for testing if the image has been upscaled. For the purpose of this model, we have also assumed that the whole image has been upscaled, and not just certain parts of the image/video. Hence, we may then only use one 64x64 patch to test for the whole image.

The test dataset contains a minimum of 580 64x64 patches that have not been seen by the model during training/validation out of 600, since the 20 original 4K images and 20 upscaled 4K images selected were from the same images.

B. Experimental Results

The results of the model, trained up to the 9th epoch with a batch size of 32 are shown in Figure 4

As seen in the results, we obtained **96.33%** accuracy on test images that the model has not seen before, showing that the model is capable of identifying images that have been upscaled reliably.

V. Conclusion

Our experiments with a CNN-based Image Re-scaling identifier have demonstrated its ability to reliably classify images between being true 4K images and images that have been upscaled to 4K from a lower resolution.

The file *CNN.ipynb* as well as python functions used in slicing the images have been attached in the submission. In *CNN.ipynb*, the last cell includes a function *slice_image* which can be used in order to slice 64x64 patches from images in the secret dataset at one go, so that it can be input in the proposed trained CNN model.

Further implementations could be added in the proposed CNN model, for example reading all 64x64 patches in a 4K image input, and pooling the predictions together to make a final prediction, which might further improve our prediction accuracy.

References

[1] R. E. Carlson and F. N. Fritsch, "An algorithm for monotone piecewise bicubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 26, no. 1, pp. 230–238, 1989.

[2] K. Gallivan, G. Grimme, and P. Van Dooren, "A rational lanczos algorithm for model reduction," *Numerical Algorithms*, vol. 12, no. 1, pp. 33–63, 1996.

[3] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[4] T. R. Goodall, I. Katsavounidis, Z. Li, A. Aaron, and A. C. Bovik, "Blind picture upscaling ratio prediction," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1801–1805, 2016.

[5] C. Liu and M. Kirchner, "Cnn-based rescaling factor estimation," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec'19. New York, NY, USA: ACM, 2019, pp. 119–124. [Online]. Available: http://doi.acm.org/10.1145/3335203.3335725

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.