Indian Institute of Information Technology,
Design and Manufacturing, Kancheepuram

Digital Image Processing

Detection of Deep Network Generated Images Using Disparities in Color Components

Author:
Amar Kumar
(CED17I029)
Saumya Prakash
(CED17I043)

Supervisor:
Dr. Masilamani V

June 10, 2020

Contents

1 Abstract

Today in the era of powerful deep learning architectures, we are generating photorealistic images that are fooling human eyes successfully with the help of generative adversarial networks and variational autoencoder architectures. Existing deep networks generate images in RGB color space and have no explicit constraints on color correlations; therefore, DNG images have more obvious differences from real images in other color spaces, such as HSV and YCbCr, especially in the chrominance Components. In this report, we are going to classify real images and DNG images.

2 Introduction

With the rapid development of image processing technology, one can easily create image forgeries without leaving visual artifacts. The spread of fake images may result in moral, ethical, and legal consequences. It is important to identify fake information in order to avoid potential security issues. Therefore, determining the authenticity of images has attracted increasing attention in many applications, such as image forensics and biometric anti-spoofing.

In this, we have analyzed the differences between DNG images and real images and observed some statistical differences between them. We have observed that chrominance components of HSV and YCbCr are different in real images and DNG images. We have also observed that there is a significant difference in DNG and real images when we assemble R,G and B color components together instead of taking it individually.

We have used an effective feature set for DNG image detection. The feature set consists of co-occurrence matrices extracted from the image high-pass filtering residuals of several color components. In order to make the feature dimension compact, binarization or truncation to the residuals is applied, and the elements of co-occurrence matrices are combined based on symmetric property. The proposed feature set is of low dimension, and achieves good detection performance even under the case of a small training set.

3 Theory

In this section, we first analyze some disparities exist between DNG images and real images in different color spaces. Then, we construct a feature set to capture the artifacts of DNG images so as to detect them. Finally, we discuss several detection scenarios and the corresponding detection strategies.

3.1 Investigating DNG Images from the Perspective of Color

Typically, an image generator takes a random latent vector as input, and employs several transpose convolutional layers to gradually expand the spatial size of the random vector to produce an image.

As the image is generated in RGB space, the generator tends to learn the properties of real images in RGB space, while paying less attention to the properties in other color spaces. In this way, although the DNG images may look

like real ones in RGB color space, there may be some differences in other color spaces.

Moreover, a real image is captured from real scene, meaning that the color components are decomposed and digitalized from real world, while the color components in a DNG image are computed by three groups of convolutional weights without putting explicit constrains on their relations. Therefore, it is reasonable to assume that some inherent relations among the color components of the DNG images are different from real ones.

Now we analyze the discernibility of three different color spaces, i.e., RGB, HSV, and YCbCr, in distinguishing between DNG images and real images through analytical experiments. We try to use a metric to examine which color component is more discernible. To this aim, we first obtain the image statistics from different color components. Then we use a metric to evaluate the distance between the statistics from DNG images and those from real images. The larger the distance, the more discernible the color component.

We construct the image statistics as follows-

a) For each color component, extract normalized histograms from some DNG images and real images. Compute the mean histograms by averaging the histograms from these two classes. Denote the mean histograms as $\widetilde{H}^c$ and $\bar{H}^c$ for DNG images and real images, respectively, where c $\in \{R, G, B, H, S, V, Y, Cr, Cb\}$ represent different colour components.

b) Denote the histogram of the i-th as H_i^c. We define a quantity called similarity index (SI) as

$$\lambda_i^c = \frac{d_{x^2}(H_i^c, \bar{H}_i^c)}{d_{x^2}(H_i^c, \widetilde{H}_i^c)} \tag{1}$$

where $d_{x^2}(H_p, H_q)$ is Chi-square distance for evaluating the similarity between two histograms $H_p(x)$ and $H_q(x)$ (x is the bin index) as

$$d_{x^2}(H_p, H_q) = \frac{1}{2}\sum_x \frac{(H_p(x) - H_q(x))^2}{H_p(x) + H_q(x)} \tag{2}$$

c) Compute the histogram of λ_i^c as the image statistics. The Chi-square distance $d_{x^2}(H_{DNG}^c, H_{Real}^c)$

The Chi-square distance $d_{x^2}(H_{DNG}^c, H_{Real}^c)$ as the discernible metric.It is expected that the large the distance, the better the discernibility.

The chrominance components (i.e., H, S, Cb, and Cr) are more discernible than the other components, implying that some statistical feature extracted from these chrominance components would be more effective in distinguishing between DNG images and real images. In addition to extracting features from the chrominance components, it is also helpful for detection if the features are extracted by considering the R, G, and B components together.

3.2 Detection strategy

In practical applications, there are many kinds of generative models, and such models may be trained with different real image sources. As a result, DNG images generated by different models which are trained with different datasets

may more or less exhibit different characteristics, leading to difficulties in distinguishing them from real images. Based on the information that an investigator can access, we divide the detection scenarios into three cases: sample-aware, model-aware, and model-unaware. These scenarios and the corresponding detection strategies are discussed as follows.

- Sample-aware detection - In this case we know the generative model and real images set from which the fake images are generated.We train a binary classifier with DNG images and real images and use this trained classifier to predict the class labels for the given images.
- Model-aware detection - In this case, we know about the generative model but not the real images which was used to train the model. To perform the detection , we first generate the DNG images using alternate real image set with known generative model. With above real and DNG images we train a binary classifier to predict the given images.
- Model-unaware detection - This is the most challenging scenarios where we don't know about both the generative model and real image data sets.

4 Implementation

4.1 Exacting Features from Color Components

- We took the image and separated its color components as iR, iG, iB, iH, iS, iY, iCb and iCr.
- For each color component, we obtained two residuals with the help of High Pass Filters ($[1, -1]$) and ($[[1], [-1]]$). We get $rX1$ and $rX2$ where $X \in \{R, G, B, H, S, Cb, Cr\}$.
- The R, G and B components should be treated as a whole for better discernibility. However, if there are many distinct element values, it would result in a co-occurrence matrix with a huge number of bins. As a result, we first binarize the residual image of each color component R^c(c$\in \{R, G, B\}$) by

$$\hat{R}^c(x, y) = \begin{cases} 1, & \hat{R}(x, y) > 0 \\ 0, & \hat{R}(x, y) \leq 0 \end{cases} \tag{3}$$

  where (x, y) is the position index of an residual image element.Then, we obtain an assembled residual image $\hat{R}^{RGB}$ by

$$\hat{R}^{RGB} = \hat{R}^R.2^0 + \hat{R}^G.2^1 + \hat{R}^B.2^2 \tag{4}$$

- Since H,S,Cb,Cr components represent different chrominance information of an image and thus have few correlations, we process them independently. In order to reduce the number of distinct values, the image residuals are truncated as follow

$$\check{R}^c(x, y) = \begin{cases} \tau, & \hat{R}^c(x, y) \geq \tau \\ R^c(x, y), & -\tau < R^c(x, y) < \tau \\ -\tau, & R^c(x, y) \leq -\tau \end{cases} \tag{5}$$

where c $\in \{H, S, Cb, Cr\}$ and τ is truncation threshold.

- Now find the co-occurrence of $\check{R}^c$ c $\in \{H, S, Cb, Cr\}$ and $\hat{R}^{RGB}$. In total, we have 5 co-occurence matrices named coH,coS,coCr,coCb and coRGB.
- Convert all above matrices into row vector and concatenate them to get required feature vector of image. This feature vector will be used in training the classification model.

4.2 Detection Strategy

About datasets - We used celebA dataset in which we are using 500 real images and 500 fake images that is generated by DCGAN to train the model. These images are of size 128x128.Now we pass all the images to feature_extractor() function in our code to get feature vector.

4.2.1 Sample-aware detection

- We know the generative model (i.e. DCGAN) and datasets used to generate the fake image . So before training the binary SVM classifier we split the sample into X_test and X_train. We train our classifier with X_test and predict using the X_train.

4.2.2 Model-aware detection

- For this scenario we are splitting the real images dataset in 2 halfs and generate 2 sets of fake images. We trained our binary SVM classifier with real image and fake images of first half datasets. For prediction we will use the second half generated fake images.

4.2.3 Model-unaware detection

- For this scenario, we are just training the one class classifier which fits a model to describe the distribution of real images and regard the DNG images as outliers. Hence, by feeding testing images to the one-class classifier, we can identify the DNG images once they are not predicted as real ones.

5 Result

Our trained classifer predicting 1 for fake images and 0 for real images. In case of sample-aware detection our classifer precision is about 80% , and for model-aware detection our classifier precision is about 92%.

6 Conclusion

All models were implemented as stated in research paper. The models provided in research paper is already optimum. Given a real image or fake image which are being generated using deep network generated algorithm, we are using different types of detection strategies as given in research paper to detect an image

whether it is fake or real with the help of features of image on passing it through a binary classifier or one class classifier based on the detection strategies used.

7 Reference

https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python
http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
https://www.youtube.com/watch?v=XDqlyQ46C7M
https://numpy.org/devdocs/reference/generated/numpy.convolve.htm
https://www.sciencedirect.com/topics/engineering/cooccurrence-matrix
https://github.com/cc-hpc-itwm/DeepFakeDetection/blob/master/Experiments_CelebA/dataset_celebA.7z