Detecting Generated Images

Rajababu Udainarayan Singh
Center for Sensorsystems
University of Siegen
Siegen, Germany
rajababu.singh@student.uni-siegen.de

Shubham Shivaji Suryavanshi
Center for Sensorsystems
University of Siegen
Siegen, Germany
shubham.suryavanshi@student.uni-siegen.de

Abstract

In recent years, Deep-fakes have been a serious threat in many applications, such as face detection, truth on events, etc. For this reason, in this paper, we have proposed methods to detect deep-fakes generated using recent state-of-the-art Spectrally Normalized Generative Adversarial Network (SNGAN) using different upsampling techniques. We are taking advantage of features in the frequency domain to distinguish real and fake images using different machine learning and deep learning techniques. We have described the behaviors of different machine learning models (i.e., Support Vector Machine (SVM), and Logistic Regression) when features are extracted using different methods from a two-dimensional amplitude spectrum. Furthermore, we propose to extract real, imaginary, absolute, and phase components from the complex pixels and simultaneously concatenate them in the third dimension, hence creating a two-dimensional image having four channels. Our proposed convolution neural network (CNN) model has performed well for images generated using other upsampling techniques. Finally, we have exploited the well-known transfer learning approach for classification.

1 Introduction

Generated Images The state-of-the-art generated image using various upsampling techniques [1], raises significant concerns about the safety and security of society. In particular, in recent years some benchmark technologies, such as Super Resolution Generative Adversarial Network (SRGAN), Spectrally Normalized Generative Adversarial Networks (SNGAN), and so on, have shown a dramatic reduction in the rate of detection between real and fake images. The emerging danger of *fake news, manipulated face id, digital scams using face recognition, etc.* has made us work towards finding the most efficient solution to the aforementioned problems.

Our contribution To overcome these problems, in this paper, we propose an in-depth analysis of the following approach:

- Importance of Fast-Fourier Transformed (FFT) images.
- Real and generated image classification when the features are extracted using different methods from a two-dimensional amplitude spectrum.
- Real and generated image classification using deep learning models.

Submitted to Recent Advances in Machine Learning - SS2022 at *University of Siegen*. Do not distribute.

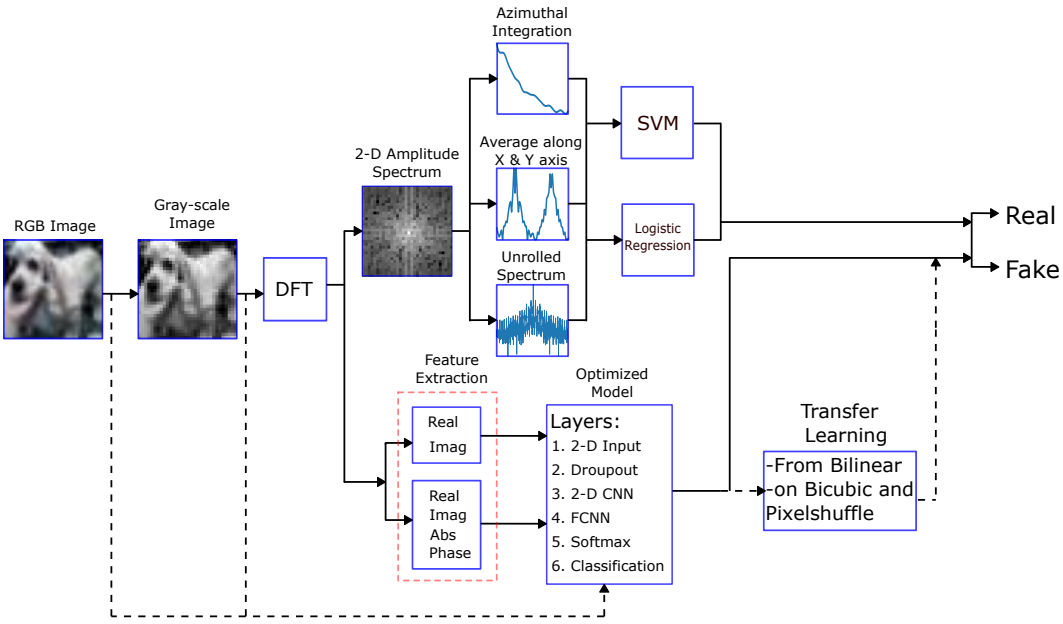


Figure 1: An illustration of our proposed methods. Firstly, we are simplifying a spatially low resolution RGB image by converting it into gray-scale. Secondly, we are making use of Fast Fourier Transform (FFT). Finally, we are classifying real and generated images using extracted features by employing machine learning and deep learning techniques

2 Methodology

2.1 From RGB Image to Fast Fourier Transform Image

The state-of-the-art RGB image $I \in \mathbb{R}^{x \times y \times C_{\text{ch}}}$, consists of limited elements (i.e., red, green, and blue channels) for image analysis, where x and y represents the two spatial dimensions and c_{ch} represents the total number of channels. In addition, in certain cases, such as ours, where the spatial resolution is low (see Fig. 1), the model is not able to detect the important features, when the images are in spatial domain. For this reason, we propose the use of two-dimensional Fast Fourier Transform $FFT_{n,m}$ mentioned in [2], [3], [4], [5], which can provide an alternative approach to solve the aforementioned problem. The $FFT_{n,m}$ is an efficient implementation of two-dimensional Discrete Fourier Transform $DFT_{n,m} \in \mathbb{C}^{n \times m}$ with identical results (see (1)).

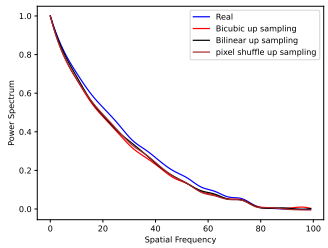
$$DFT_{n,m} = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I^{\text{g}}(x,y) \exp\left\{-\Im 2\pi \left(\frac{nx}{N} + \frac{my}{M}\right)\right\} \tag{1}$$

Where N and M corresponds to the total number of rows and columns of an image. However, due to the complexity of the direct transformation of an RGB image into a $FFT_{n,m}$ image, firstly we propose the transformation of an RGB image into gray-scale image I^{g} (see Fig. 1), which can simplify the image (i.e., reduces the amount of information of the image) by removing the unnecessary elements, simultaneously reduces the processing time, while preserving the important features of the image (i.e., edges, blobs, regions, and so on.) as mentioned in [6].

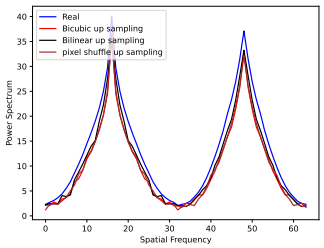
2.2 DeepFake Detection using Machine Learning Techniques

In this subsection, we focus on the methods exploited for image classification in Subsection 2.1. As mentioned in [3], [4], the spectral distribution of an SNGAN generated images differ, when compared to the spectral distribution of the real images. Therefore we analyze the amplitude spectrum of the images to extract simple features.
Following are the methods for extracting simple features from 2d amplitude spectrum.

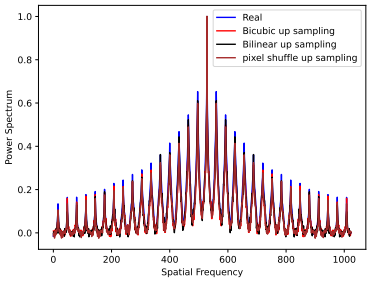
Azimuthal Integration This approach takes mean of all the amplitude with similar frequency in radial direction [4]. So that amplitude of similar frequency of DFT can be compressed into a single



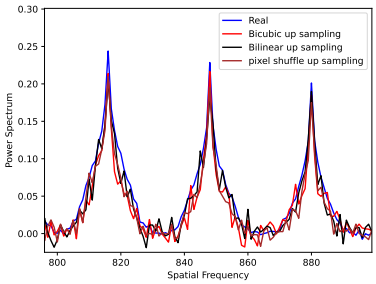
(a) Azimuthal Integration



(b) Average along X & Y axis



(c) Unrolled Amplitude Spectrum



(d) Zoomed view of Unrolled Amplitude Spectrum

Figure 2: Mean plots of extracted features:In figure 2a, we can see that the amplitude of the middle frequencies of real images is higher than that of generated images. In 2b, the amplitude of real image is higher than that of generated images for all frequency range. In figure 2c the amplitude spectrum of real image is higher than the generated images in the low frequency region. These features are used for classification.

vector feature (15 features) and it is calculated as follows

$$AI(\omega_k) = \int_0^{2\pi} \|DFT_{n,m}(\omega_k \cdot cos(\phi), \omega_k \cdot sin(\phi))\|^2 d\phi, \quad for \quad k \quad = 0, \ldots, {}^{M}/_{2} - 1 \qquad (2)$$

Taking mean along x and y axis This method involves taking mean along x and y axis. That is averaging all the similar x and y frequency components of 2D DFT. Then these two vectors are concatenated to form a single vector. The number of features extracted in this are 4 times $(32 + 32 = 64)$ higher than the azimuthal integration method, which improves overall performance

Unrolling the elements of 2D DFT This method unrolls the 2D DFT into a single vector feature. The number of features in this are 1024(32*32). As the number of features are more, this method outperforms both the previously mentioned methods.
The extracted 1D vector of all the image is fed to support vector machine (SVM) and logistic regression to classify the images. For SVM, the radial basis function (RBF) as a kernel and the regularizer term C=2 is used.

2.3 DeepFake Detection using Deep Learning Techniques

Feature Extraction In this subsection, we focus on the methods exploited for image classification in Subsection 2.1. As mentioned in [5], choosing $\mathbb{R}$ and $\mathbb{I}$ component for complex pixels is giving superior performance, we move one step further and propose to stack the $\mathbb{R}(FFT_{n,m})$, $\mathbb{I}(FFT_{n,m})$, $|FFT_{n,m}|$, and $\angle FFT_{n,m}$ components of the complex pixels in the third dimension featuring four channels.

Deep Learning Model Images consists of low spatial resolution can not perform well if the model consist of batch normalization layer, this is due to the fact that it modifies the important features of an

Support Vector Machine

		Testing		
		Bilinear	Bicubic	Pixelshuffle
Training azimuthal integration	Bilinear	0.7525	0.6475	0.645
	Bicubic	0.66	0.67	0.6325
	Pixelshuffle	0.635	0.685	0.675
Training Average along x and y axis	Bilinear	0.8125	0.755	0.74
	Bicubic	0.6775	0.80	0.7225
	Pixelshuffle	0.6775	0.72	0.785
Training Unrolled 2D DFT	Bilinear	0.875	0.6675	0.665
	Bicubic	0.6275	0.8125	0.66
	Pixelshuffle	0.6375	0.6825	0.8075

Table 1: When compared to other data sets, the bilinear data set in all three approaches has the highest accuracy score. Method 2 provides better generalization over other test data set than the other methods.

image by transforming the data of previous layer close to zero-mean and its standard deviation close to one, as mentioned in [7]. For this reason, we have developed a small two-dimensional convolutional neural network (CNN), which is very good in extracting spatial features of a two-dimensional image. The model consist of six layers, which are 2-D input, droupout, 2-D CNN, Fully Connected Neural Network (FCNN), softmax, and final classification layer. The selection of an adam optimizer with an initial learning rate of 0.2 and a mini-batch size of 75 has yielded superior performance.

Transfer Learning Finally, since transfer learning [8] performs well for problems belonging to similar categories, we propose to exploit this approach, which helps in improving overall accuracy, while reducing the computation time for training our model.

3 Experimental Results

3.1 Introduction to Dataset

We have evaluated our proposed methods on a dataset known as imagewoof consisting of four sets of images (i.e., one set of real images and three sets of generated images using different upsampling techniques in SNGAN), where each set consists of 1000 RGB images, which have been randomly shuffled and distributed between training and validation set, with a ratio kept at $80 : 20$. The used upsampling techniques are bilinear interpolation, bicubic interpolation, and pixel shuffle. Moreover, each image represents a dog with a spatial resolution of 32×32 pixels and a bit resolution of 24 bit.

3.2 Results

3.2.1 DeepFake Detection using Machine Learning Techniques

The SVM and logistic regression are used for classifying real and fake image. As per the table 1 and 2, the accuracy score on the bilinear data set has increased from 0.7525 for method 1 (azimuthal average) to 0.875 for method 3 (Unrolling 2D DFT). The reason for the improvement is that the number of features have increased from 15 (method 1) to 1024 (method 3). As some of the spectrum of the real image over lap with that of the generated image, we are unable to achieve a score 1.

3.2.2 DeepFake Detection using Deep Learning Techniques

We have evaluated our proposed deep learning methods on the dataset described in Section 3.1.As observed from Tab. 3, our model classify the real and fake images for the sets consist of bilinear and real images with a validation accuracy of 95%. When detecting the images on sets consist of bicubic and real images, our model is able to attain 93%. Despite performing poor (i.e. 67%) on the raw RGB images (see Tab. 1), our model is able to attain a validation accuracy of 92%, when image samples has been pre-processed using our proposed methods.Based on the result depicted in Tab. 1,

Logistic Regression

		Testing		
		Bilinear	Bicubic	Pixelshuffle
Training azimuthal integration	Bilinear	0.72	0.635	0.62
	Bicubic	0.6825	0.6725	0.6225
	Pixelshuffle	0.6475	0.695	0.67
Training Average along x and y axis	Bilinear	0.7725	0.725	0.685
	Bicubic	0.6875	0.7875	0.6725
	Pixelshuffle	0.68	0.725	0.7875
Training Unrolled 2D DFT	Bilinear	0.8675	0.655	0.6275
	Bicubic	0.605	0.8025	0.585
	Pixelshuffle	0.625	0.6475	0.81

Table 2: Similar to SVM bilinear data set has highest score and method 2 provides better generalization.

Table 3: Result based on Deep Learning Techniques

Up sampling Techniques	**Validation accuracy using developed deep learning model in %**							
	RGB		*Gray-scale*		*DFT: [$\mathbb{R}$,$\mathbb{I}$]*		*DFT: [$\mathbb{R}$,$\mathbb{I}$,abs,Phase]*	
Bilinear	85	*T.L.*	82	*T.L.*	84	*T.L.*	**95**	*T.L.*
Bicubic	76	78	78	77	82	84	**93**	93
Pixelshuffle	67	68	73	72	84	84	**92**	89

we can see that our developed model performs well for images generated by SNGAN using other upsampling techniques.

4 Conclusion

In this work, we have exploited several state-of-the-art machine learning and deep learning techniques. Our results have depicted that the spatial low-resolution deep-fake generated using SNGAN based on different upsampling techniques can be classified using our developed methods with a maximum accuracy of 95%. Additionally, we can notice that images generated using the bilinear upsampling technique have been classified most accurately. The future work can be an analysis of our developed models on images that are wavelet-based transformed. Furthermore, an attempt can be made on exploiting the behavior of our proposed models, when the RGB images have been directly transformed into FFT images, hence featuring twelve channels, which is four for each RGB channel.

Appendix

- Our machine learning methods has been implemented using Python.
- Our deep learning methods has been implemented using MATLAB.

Acknowledgement

We would like to express our sincere thanks to **Prof. Dr.-Ing Margret Keuper** for providing an interesting project and supporting us through her valuable feedbacks during the entire course of time.

References

[1] I. Sanchez and V. Vilaplana, "Brain mri super-resolution using 3d generative adversarial networks," 04 2018.

[2] H. Guo, G. Sitton, and C. Burrus, "The quick discrete fourier transform," in *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. iii, pp. III/445–III/448 vol.3, 1994.

[3] R. Durall López, M. Keuper, and J. Keuper, "Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions," 03 2020.

[4] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, "Unmasking deepfakes with simple features," *ArXiv*, vol. abs/1911.00686, 2019.

[5] M. Heredia Conde, "A material-sensing time-of-flight camera," *IEEE Sensors Letters*, vol. 4, no. 7, pp. 1–4, 2020.

[6] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing, A practical approach with examples in MATLAB*, ch. 1, pp. 10–14. Wiley-Blackwell, 2011.

[7] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[8] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, 09 2020.