Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain - Appendix

Honggu Liu[1*] Xiaodan Li[2] Wenbo Zhou[1†] Yuefeng Chen[2]
Yuan He[2] Hui Xue[2] Weiming Zhang[1†] Nenghai Yu[1]
[1]CAS Key Laboratory of Electromagnetic Space Information,
University of Science and Technology of China
[2]Alibaba Group
lhg9754@mail.ustc.edu.cn, {welbeckz, zhangwm, ynh}@ustc.edu.cn
{fiona.lxd, yuefeng.chenyf, heyuan.hy, hui.xueh}@alibaba-inc.com

1. Detail Proofs of validity of phase spectrum

In this section, we demonstrate the details of theory analysis which aim to prove that the introduction of phase spectrum can help face forgery detection. We first show the phase spectrum contains more information in Section 1.1, and then we demonstrate how the extra information is utilized to CNNs in Section 1.2. Finally we make a conclusion that the phase spectrum help deepfake detection due to the cumulative up-sampling in Section 1.3

1.1. What extra information will we get from phase spectrum compared with amplitude spectrum?

To simplify the calculation, we make all the mathematical derivation based on the one-dimension signal. We first set up the basic notations used in this paper: $x(n)$ and $\mathbf{X}(u)$ denotes a 1D discrete signal and its Discrete Fourier Transform (DFT), where n is the signal location and u represents the frequency. $\mathbf{R}(u)$ and $\mathbf{I}(u)$ respectively denote the real part and imaginary part of $\mathbf{X}(u)$. $\mathbf{A}(u)$ is the amplitude spectrum and $\mathbf{P}(u)$ is the phase spectrum. And we use $c(n)$ and $\mathbf{C}(n)$ denote the convolution kernel and its DFT. And we use $*$ to denote convolutional operation. Besides, $\mathbf{F}(\cdot)$ and $\mathbf{F}^{-1}(\cdot)$ represent the DFT and its inverse. So we have $\mathbf{X}(u) = \mathbf{F}(x(n))$ and $x(n) = \mathbf{F}^{-1}(\mathbf{X}(u))$.

Claim 1. *Phase spectrum will keep more frequency components which tend to zero in amplitude spectrum.*

Proof. For a discrete non-periodic signal $x(n)$, its DFT $\mathbf{X}(u)$ is

$$\begin{aligned}\mathbf{X}(u) &= \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi un}{N}} \\ &= \frac{1}{N}\sum_{n=0}^{N-1} x(n)(\cos\frac{2\pi un}{N} - j\sin\frac{2\pi un}{N}) \\ &= \mathbf{R}(u) + j\mathbf{I}(u)\end{aligned} \tag{1}$$

And the amplitude spectrum of $\mathbf{X}(u)$ is

$$\mathbf{A}(u) = \sqrt{\mathbf{R}^2(u) + \mathbf{I}^2(u)} \tag{2}$$

Axiom 1. *Low-frequency components dominate the frequency domain for a natural image, and many high-frequency components are very small even tend to zero.*

Base on the axiom 1, we define a frequency components set as follows.

Definition 1. $\exists$ U $= \{u_0, u_1, \cdots, u_m\}$, $\forall u_k \in$ U, $\mathbf{A}(u_k) \approx 0$.

According to the definition 1, we get

$$\mathbf{A}(u_k) \approx 0 \iff \mathbf{R}(u_k) \approx \pm 0 \quad \text{and} \quad \mathbf{I}(u_k) \approx \pm 0 \tag{3}$$

For phase spectrum, we know that

$$\mathbf{P}(u) = \arctan\frac{\mathbf{I}(u)}{\mathbf{R}(u)} \tag{4}$$

For every $u_k \in$ U, we get

$$\begin{aligned}\mathbf{P}(u_k) &= \arctan\frac{\mathbf{I}(u_k)}{\mathbf{R}(u_k)} \\ &\approx \arctan \pm 1 = \pm\frac{\pi}{4}\end{aligned} \tag{5}$$

Our claim 1 is proved by Eq.(5). ■

1.2. Why do we need extra abundant frequency components?

For a convolutional neural networks, the basic and key operation is the convolution computation between input and convolution kernel. According to the convolution theorem, we know that $x(n) * c(n) \Leftrightarrow \mathbf{X}(u) \cdot \mathbf{C}(u)$.To simplify the representation, we make a substitution that

$$\begin{aligned}\mathbf{X}(u) &= \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi u n}{N}} \\ &= a_0 + a_1 e^{j\theta_1} + a_2 e^{j\theta_2} + \cdots + a_{N-1} e^{j\theta_{N-1}}\end{aligned} \tag{6}$$

Similarly,

$$\begin{aligned}\mathbf{C}(u) &= b_0 + b_1 e^{j\theta_1} + b_2 e^{j\theta_2} + \cdots + b_{N-1} e^{j\theta_{N-1}} \\ &= \frac{1}{N}\sum_{n=0}^{N-1} c(n) e^{-j\frac{2\pi u n}{N}}\end{aligned} \tag{7}$$

Then, we make a claim as follows

Claim 2. *Phase spectrum helps CNNs acquire and learn more abundant frequency components which are ignored with convolution calculations of amplitude spectrum.*

Proof. With the derivation in 1.1, we assume that $\mathbf{A}(u_i) \approx 0$ when $i > k$. Thus, we can get

$$\mathbf{X}(u) = a_0 + a_1 e^{j\theta_1} + \cdots + a_k e^{j\theta_k} \tag{8}$$

and

$$\mathbf{X}(u) \cdot \mathbf{C}(u) = (a_0 + a_1 e^{j\theta_1} + \cdots + a_k e^{j\theta_k}) \cdot \mathbf{C}(u) \tag{9}$$

However, when we introduce the phase spectrum $\mathbf{P}(u)$, $\mathbf{P}(u) \neq 0$ with $u > k$. Then we can get

$$\begin{aligned}\mathbf{P}(u) \cdot \mathbf{C}(u) &= (a_0 + a_1 e^{j\theta_1} + \cdots + a_{N-1} e^{j\theta_{N-1}}) \cdot \mathbf{C}(u) \\ &= \mathbf{X}(u) \cdot \mathbf{C}(u) + \mathbf{E}(u) \cdot \mathbf{C}(u)\end{aligned} \tag{10}$$

where

$$\mathbf{E}(u) = a_{k+1} e^{j\theta_{k+1}} + \cdots + a_{N-1} e^{j\theta_{N-1}} \tag{11}$$

■

1.3. Capturing up-sampling artifacts via phase spectrum in face forgery

To detect the observed common artifacts, namely up-sampling, we analyze it in the frequency domain.

Up-sampling will lead to the emergence of new frequency components. And we make a claim as follows

Claim 3. *Phase spectrum is more sensitive to up-sampling artifacts and therefore helps face forgery detection.*

Proof. The increase of spatial resolution in 2D corresponds to the extension of the time domain in 1D. Assume that the input $x(n)$ is up-sampled by factor 2, then

$$\hat{x}(n) = \begin{cases} x(\frac{1}{2}n), & n = 2k \\ 0, & n = 2k+1 \end{cases} \tag{12}$$

where $k = 0, 1, 2, \cdots, N-1$, and

$$\begin{aligned}\hat{\mathbf{X}}(u) &= \frac{1}{2N}\sum_{n=0}^{2N-1} \hat{x}(n) e^{-j\frac{2\pi u n}{2N}} \\ &= \frac{1}{2N}\sum_{n=0}^{N-1} \hat{x}(2n) e^{-j\frac{2\pi u 2n}{2N}} \\ &= \frac{1}{2N}\sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi 2 u n}{N}} \\ &= \underbrace{a_0 + a_1 e^{j\theta_1} + \cdots + a_{2N-1} e^{j\theta_{2N-1}}}_{2N \text{ items}}\end{aligned} \tag{13}$$

Then we have $\hat{x}(n) = x(\frac{1}{2}n) \Leftrightarrow \hat{\mathbf{X}}(u) = \mathbf{X}(2u)$ with the Eq. 13, which leads to the conclusion that the increase of spatial resolution will result in the compression in the frequency domain which is consistent with the property of Fourier Transform (FT). In fact, the essence of DFT is the principle value interval of Discrete Fourier Series (DFS) and thus new frequency components are the duplicate of origin frequency components.

Base on our claim in Section 1.1. We first assume the amplitude spectrum $\mathbf{X_A}(u)$ and the phase spectrum $\mathbf{X_P}(u)$ of original images $x(n)$. It is

$$\begin{aligned}\mathbf{X}_A(u) &= \underbrace{a_0 + a_1 e^{j\theta_1} + \cdots + a_k e^{j\theta_k}}_{(k+1) \text{ items}} \\ \mathbf{X}_P(u) &= \underbrace{p_0 + p_1 e^{j\theta_1} + \cdots + p_{N-1} e^{j\theta_{N-1}}}_{N \text{ items}}\end{aligned} \tag{14}$$

and the corresponding up-sampling is

$$\begin{aligned}\mathbf{X}_A^{up}(u) &= a_0 + a_1 e^{j\theta_1} + \cdots + a_{2N-1} e^{j\theta_{2N-1}} \\ &= \underbrace{a_0 + \cdots + a_k e^{j\theta_k}}_{(k+1) \text{ items}} + \underbrace{a_N e^{j\theta_N} + \cdots + a_{N+k} e^{j\theta_{N+k}}}_{(k+1) \text{ items}} \\ \mathbf{X}_P^{up}(u) &= \underbrace{p_0 + p_1 e^{j\theta_1} + \cdots + p_{2N-1} e^{j\theta_{2N-1}}}_{2N \text{ items}}\end{aligned} \tag{15}$$

We define that $y_A(n)$ is the output of a convolution layer with an input $x(n)$ and its frequency domain form is $\mathbf{Y_A}(u)$. And we get

$$\begin{aligned} y_A(n) &= x(n) * c(n) \\ &\Updownarrow \\ \mathbf{Y}_A(u) &= \mathbf{X_A}(u) \cdot \mathbf{C}(u)\end{aligned} \tag{16}$$

According to the claim in Section 1.2, we can deduce the frequency domain form is

$$\mathbf{Y}_A(u) = \underbrace{f_0 + f_1 e^{j\theta_1} + \cdots + f_{k+N-1} e^{j\theta_{k+N-1}}}_{k+N \text{ items}} \quad (17)$$

And the corresponding form of the up-sampling is

$$\mathbf{Y}_A^{up}(u) = \underbrace{f_0 + f_1 e^{j\theta_1} + \cdots + f_{2N+k-1} e^{j\theta_{2N+k-1}}}_{2N \text{ items}} \quad (18)$$

In our work, we first take Inverse Discrete Fourier Transform (IDFT) to phase spectrum and acquire the spatial domain form $p(n)$ of phase. And we state a theorem named *the distributive law* as follow,

Theorem 1. $(f(\cdot) + g(\cdot)) * h(\cdot) = f(\cdot) * h(\cdot) + g(\cdot) * h(\cdot)$

Then we consider that we directly concatenate $x(n)$ and $p(n)$ in channel dimension based on theorem 1 and the output $\mathbf{Y}_{A+P}(u)$ is

$$\mathbf{Y}_{A+P}(u) = \underbrace{f_0 + f_1 e^{j\theta_1} + \cdots + f_{2N-2} e^{j\theta_{2N-2}}}_{2N-1 \text{ items}} \quad (19)$$

And the corresponding form of the up-sampling is

$$\mathbf{Y}_{A+P}^{up}(u) = \underbrace{f_0 + f_1 e^{j\theta_1} + \cdots + f_{3N-2} e^{j\theta_{3N-2}}}_{3N-1 \text{ items}} \quad (20)$$

■

According to the derivation in 1.3, we intuitively know that the number of learnable frequency components is N when we leverage the original image and its phase together, but the number just is $N - k$ without the phase spectrum.

2. The Further Analysis of Network Architecture

In this section, we first make a detailed description of our network architecture in Section 2.1. Then we analyzes the technique of shallowing learning. To analyze the shallow network, we mainly exploit the correlation between the performance and the number of convolutional layers of various backbone networks in this section. We respectively conduct two experiments base on the XceptionNet in Section 2.2 and ResNet34 in Section 2.3. We remove the convolutional layers step by step to reduce the receptive field gradually. And all experiments settings remain the same with the main paper. At the same time, the number of convolutional layers and the size of receptive field are also compared. And the way of calculating the size of receptive field is as follows.

$$\mathbf{RF}_{l-1} = s_l \cdot \mathbf{RF}_l + (k_l - s_l) \quad (21)$$

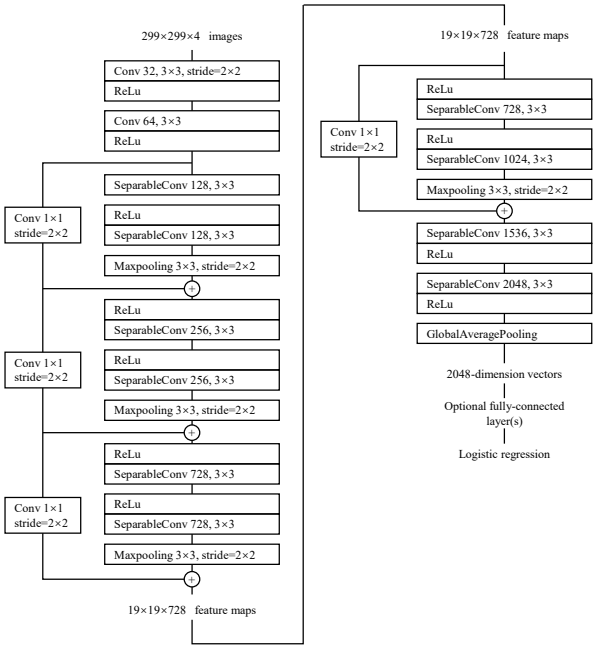

Figure 1: **Network architecture details**. The network architecture is modified based on XceptionNet.

The Equation 21 can be used in a recursive algorithm to compute the receptive field size of the network $\mathbf{RF}_0$. Then it can be simplified as follows.

$$\mathbf{RF}_0 = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1 \quad (22)$$

2.1. Details of the network architecture

To suppress high-level semantic features and extract more texture features, we shallow the backbone XceptionNet by throwing away several blocks, only retain the Xception Block 1-3 and 12. The detailed architecture is shown in Figure 1.

2.2. XceptionNet

We demonstrate the performance on XceptionNet with different blocks. The results are listed in Table 1. By comparing the pristine XceptionNet, the proposed shallow network improves the ACC and AUC scores whether in entirely supervised way or cross-dataset evaluation. We also notice that the performance will dramatically drop when the network is too shallow, and we believe that this phenomenon is due to the overly shallow network is unable to extract sufficient implicit features for detecting forged face.

2.3. ResNet

We also show the results of ResNet34 in Table 2. According to the architecture of ResNet34, we randomly re-

Network	FF++		Celeb-DF		Layers	$\mathbf{RF}_0$
	ACC	AUC	ACC	AUC		
Xcep-B1	78.79	80.20	67.08	63.51	5	19
Xcep-B2	85.87	89.70	70.97	71.34	8	43
Xcep-B3	88.59	92.31	71.94	73.38	11	91
Xcep-B4	89.75	93.74	**73.20**	**75.40**	14	187
Xcep-B5	90.68	93.20	72.63	74.15	17	283
Xcep-B6	90.78	94.61	70.60	72.89	20	379
Xcep-B7	91.44	**95.01**	72.22	74.80	23	475
Xcep-B8	90.22	92.60	72.73	74.17	26	571
Xcep-B9	89.69	91.91	72.85	72.89	29	667
Xcep-B10	91.32	94.42	72.22	74.85	32	763
Xcep-B11	90.72	93.52	72.64	74.98	35	859
Xcep-B12	89.35	92.49	70.81	72.33	38	955
XceptionNet	**92.39**	94.86	71.49	73.67	40	1083

Table 1: Quantitative results (ACC (%) and AUC (%)) on different Xception networks with the proposed SPSL. The corresponding number of convolutional layers and receptive field are also shown. All models are trained on FF++ (HQ) and tested on both FF++ (HQ) and Celeb-DF. The bold results are the best.

Network	FF++		Celeb-DF		Block
	ACC	AUC	ACC	AUC	
Res34-B1	75.57	56.82	63.23	48.44	[0,0,0,0]
Res34-B2	76.64	56.95	61.66	48.59	[1,1,1,1]
Res34-B3	67.51	58.60	64.18	60.53	[3,1,1,3]
Res34-B4	**83.24**	**89.26**	**66.79**	**71.78**	[3,2,3,3]
Res34-B5	74.52	54.24	63.46	48.24	[3,4,0,0]
ResNet34	71.55	81.58	65.19	66.90	[3,4,6,3]

Table 2: Quantitative results (ACC (%) and AUC (%)) on different ResNet34 networks with the proposed SPSL. The corresponding number of four types of basic blocks are shown. All models are trained on FF++ (HQ) and tested on both FF++ (HQ) and Celeb-DF. The bold results are the best.

duce the number of four types of basic block respectively. Compared with the original ResNet34, almost all shallow ResNet34 networks outperform to an extent.

2.4. Discussion

We present the further analysis of the shallow network. Even now the results demonstrate that the shallow network to some extent helps forged face detection, there are still some problems are worthy to study in the future. The most vital thing is to find the best degree of shallow network. And it is also important that if there is a universal strategy which is suitable for all backbone networks.