

MD-CSDNetwork: Multi-Domain Cross Stitched Network for Deepfake Detection

Aayushi Agarwal¹, Akshay Agarwal², Sayan Sinha³, Mayank Vatsa¹, and Richa Singh¹

¹IIT Jodhpur, India, ²University at Buffalo, USA, and ³IIT Kharagpur

aayushi.agarwal007@gmail.com, aa298@buffalo.edu, sayan.sinha@iitkgp.ac.in, and {richa, mvatsa}@iitj.ac.in

Abstract—The rapid progress in the ease of creating and spreading ultra-realistic media over social platforms calls for an urgent need to develop a generalizable deepfake detection technique. It has been observed that current deepfake generation methods leave discriminative artifacts in the frequency spectrum of fake images and videos. Inspired by this observation, in this paper, we present a novel approach, termed as MD-CSDNetwork, for combining the features in the spatial and frequency domains to mine a shared discriminative representation for classifying *deepfakes*. MD-CSDNetwork is a novel cross-stitched network with two parallel branches carrying the spatial and frequency information, respectively. We hypothesize that these multi-domain input data streams can be considered as related supervisory signals. The supervision from both branches ensures better performance and generalization. Further, the concept of cross-stitch connections is utilized where they are inserted between the two branches to learn an optimal combination of domain-specific and shared representations from other domains automatically. Extensive experiments are conducted on the popular benchmark dataset namely FaceForensics++ for forgery classification. We report improvements over all the manipulation types in FaceForensics++ dataset and comparable results with state-of-the-art methods for cross-database evaluation on the Celeb-DF dataset and the Deepfake Detection Dataset.

I. INTRODUCTION

Manipulating videos or images is not new; earlier seamless manipulation used to be a skillful task requiring a lot of resources, time, and dexterous artists. However, recent growth in the deep learning technology has given anyone the ability to make a convincing fake video, which can also be misused by some people to “weaponize” it for political or other malicious purposes. A Reddit user by the name of ‘deepfakes’ using deep learning to make pornographic videos and posting them online, President Obama using an expletive to describe President Trump and Mark Zuckerberg admitting that Facebook’s true goal is to manipulate and exploit its users are some notable examples of such misuse [4], [5]. Audios can be *deepfaked* as well to create “voice skins” or “voice clones” of public figures. A recent example where audio deepfakes have been exploited was a phone call by a fraudster mimicking the voice of a German energy firm’s CEO. The chief of a UK subsidiary of the firm was tricked, and they paid nearly £200,000 into a Hungarian bank account [5]. Similar scams have reportedly used recorded WhatsApp voice messages. All these instances suggest that developing robust and efficient methods of deepfake detection is of paramount importance.

Most of the existing *deepfakes* alter facial content, and these manipulations can be categorized in four broad groups depending upon the levels of face manipulation [48]:

- identity swap where a face from a source video replaces a face in the target video usually employing computer-graphics or deep learning based methods,
- facial expression reenactment where the expressions in the source video are transferred to the target video while maintaining the identity of the target person,
- complete synthesis of high-quality and realistic new faces using generative adversarial networks (GANs). This is different from manipulating only a certain part (mostly facial region) in the image, and
- attribute manipulation where various attributes such as skin color and hair color are edited.

A. Literature Review

A wide variety of approaches have been proposed for verifying the authenticity of images and videos in the vast area of digital media forensics. Previous research directions have used physiological signals like eye blinking [29], inconsistent head poses [54], and biological signals not preserved in fake videos [15] as well as phoneme-viseme mismatches in videos [8] as the basis for detecting deepfake content. Various deep learning based detection methods [6], [9], [17], [23], [30], [50], [53], [58] have been proposed to mitigate the risks of deepfakes. Some notable examples include [7], where the combination of a static biometric for face recognition along with a temporal behavioral biometric is used for face-swap deepfakes. Also, Face X-Ray [28] uses a more generic approach for detection by identifying whether the input image can be decomposed into a blending of different images or not. [39] uses a capsule network to detect various kinds of deepfake attacks created using deep learning. Further, [57] uses a face classification stream and a patch triplet stream for leveraging features capturing local noise residuals and camera characteristics. A two-stream network is proposed for detecting image manipulations where one of the streams uses RGB image for identifying high-level tampering artifacts like contrast difference and unnatural boundaries [59]. More recently, SSTNet [52] uses a detection framework to detect tampered faces through spatial, steganalysis, and temporal features. Some methods have also used the artifacts produced in the frequency spectrum to mine better discriminative feature representations. F^3 -Net [40] have proposed a two-stream collaborative learning framework that makes use of

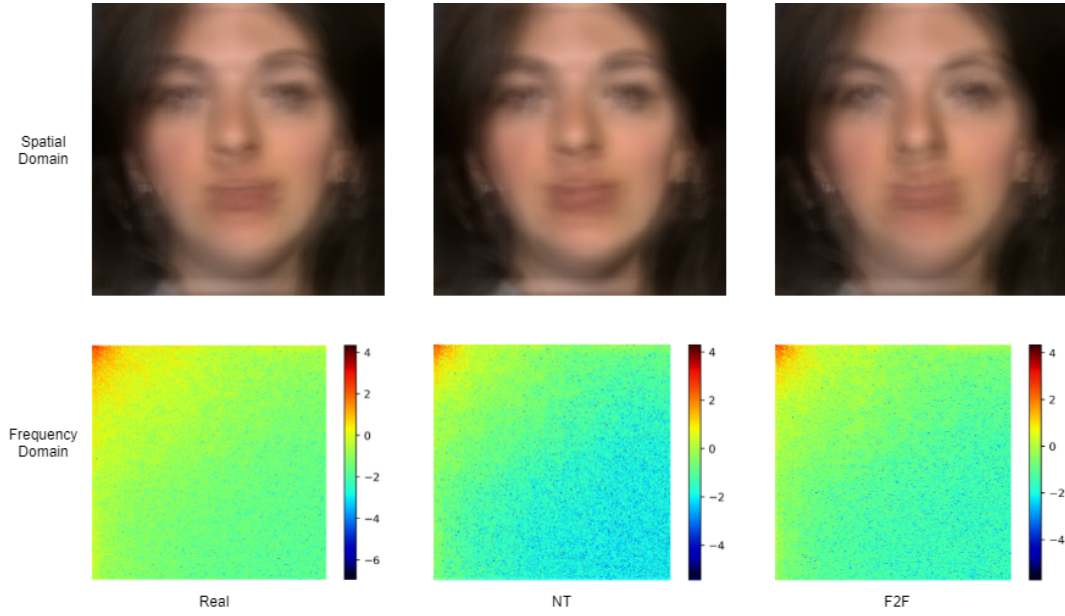


Fig. 1: Comparison of DCT-based frequency spectrum between a real video and a manipulated video. From left to right, row 1 shows the images obtained for a video by averaging the pixel values of all the frames for a Real, Neural Textures [46] and Face2Face [47] videos. There are no observable differences seen in all the three images. From left to right, row 2 shows the heatmaps for the frequency spectrum obtained after averaging the log-scaled DCT coefficients of all the frames for a Real, Neural Textures (NT), and Face2Face (F2F) video. We clearly see the differences in color variations as we move from the low-frequency (top-left) to the high-frequency (bottom-right) region between real vs manipulated videos. Best viewed in color.

frequency-aware decomposed image components and local frequency statistics to mine better forgery patterns. Masi *et al.* [35] have presented a two-branched network structure with one branch propagating the original information in the color domain while the other suppressing the face content and amplifying multi-band frequencies using a Laplacian of Gaussian (LoG). Some recent works [34], [51], [55] focus on identifying special artificial fingerprints common across all GAN synthesized images to detect fake images from their real counterparts. [25], [35], [43], [49] have also exploited the temporal information between the frames in a video to isolate deepfakes through sequence-based models.

Most of the current deepfake creation methods, especially the ones which use generative models (e.g. GANs) to create manipulated faces or completely new synthetic images, rely on convolution based upsampling methods due to which they fail to reproduce the spectral distributions of natural images [19]. This creates artifacts in the frequency spectrum of fake images, which can be used as discriminatory evidence against the real ones. The frequency artifacts are more common and generic across a variety of manipulations, including compression scenarios, where the information in the spatial domain is severely affected.

Recent approaches [20], [21], [56] specifically focus on identifying images synthesized through GANs by using the frequency-related features as inputs to a classifier. In particular, [20], [21] have proposed spectrum-based classifiers that input the frequency features rather than pixel input for

detecting synthetically generated images. However, such a classifier may not be suitable for detecting other kinds of facial manipulations, for example, the manipulations in the widely used FaceForensics++ (FF++) [42] dataset. Here, the artifacts in the spatial domain are also essential and cannot be neglected completely. On the other hand, Face X-Ray [28] proposed a novel image representation to identify any blending boundary artifacts introduced during the manipulation process but uses only spatial domain information. This may not be sufficient for detecting GAN-based manipulations or highly realistic synthetic images that might leave comparatively much fewer artifacts in the spatial domain. Therefore, a combination of features from both the domains is more useful to detect various kinds of manipulation types.

B. Research Contributions

In this paper, we propose a novel approach, termed as MD-CSDNetwork, of combining information from the two domains, spatial and frequency, by learning a shared representation at both low-level and high-level feature maps. We draw an analogy that features from spatial and frequency domains can act as two related supervisory signals similar to related tasks in a multi-task learning problem. We jointly learn the multi-domain feature representation via the proposed cross-stitched network architecture for learning discriminative features. Specifically, each cross-stitch unit learns an optimal linear combination of domain-specific and shared representations from other domains at various levels in the network. Depending upon the input manipulation type,

the network learns the weights of the activation maps from both the domains as required. In other words, the proposed approach enables learning the contribution of features from both the domains, and the training signals from the two domains help each other to enhance the performance and generalizability. The major contributions of this paper are:

We propose a cross-stitched network architecture for jointly learning features from both spatial and frequency domain information. It gives the network the flexibility to combine features from both the domains at various levels depending upon the input data. We hypothesize that the process of learning a shared representation from two different domain features of the same image can be considered similar to learning a shared representation from multiple related tasks in a multi-task learning setting. Such a representation enables supervision from both feature domains, which helps the model to learn better discriminative representations. We have experimentally evaluated the use of Discrete Cosine Transform (DCT) spectrum over Discrete Wavelet Transform (DWT) or Fast Fourier Transform (FFT) for extracting the frequency-related features in the proposed network. Through extensive experimentation, we show that the proposed model with DCT achieves a considerable improvement in the detection performance.

II. PROPOSED APPROACH

A. Motivation

To understand deepfake artifacts in the frequency domain, Fig. 1 shows plots of the DCT coefficient spectrum of real and fake images for two manipulation types namely Neural Textures (NT) & Face2Face (F2F) in FF++. The horizontal direction corresponds to frequencies in the x -direction, and the vertical direction corresponds to frequencies in the y -direction. The frequencies increase as we move from left to right and top to bottom for x and y directions, respectively. We observe clear distinguishability between the spectrums of real and fake images, especially in the high-frequency regions, which implies that deepfakes are unable to mimic the real spectrum variations.

We further draw the motivation for combining information from spatial and frequency domains by formulating the learning procedure similar to a multi-task learning problem. In multi-task learning, related tasks share the representations learned from multiple supervisory tasks with direct supervision from the primary task and indirect supervision from auxiliary related tasks. This improves generalization by leveraging the domain specific information contained in the training signals of the related tasks. Misra *et al.* [37] explains that deciding how much sharing one needs to enforce among multiple tasks is very cumbersome and highly dependent on the tasks at hand as well as the input data. There can be a spectrum of possible network architectures ranging from all the layers (except the last layers) being shared to no sharing at all. They provide a principled approach to mitigate this problem with the use of cross-stitch units which learn an optimal combination of both task-specific and shared

representations automatically. Hence, the tasks supervise how much sharing is needed for improved classification.

Inspired by the work done in [37], we hypothesize that two input streams, one carrying the spatial information of an image and the other carrying the frequency information of the same image, can be considered as two related supervisory signals. We devise a cross-stitched network that jointly learns a feature representation from both the domains with parallel branches carrying the two input data streams and connected with multiple cross-stitch units at various levels. These units enable direct supervision from themselves and indirect supervision from the other branch and inadvertently allow information flow between both the domains. The cross-stitch units learn an optimal combination of domain-specific and shared representations from other domains for better classification depending upon the input data.

B. Proposed MD-CSDNetwork Architecture

As shown in Fig. 2, the architecture of the proposed method consists of two parallel branches with cross-stitch units inserted after the max-pooling layers. These cross-connections applied at various levels in the network allow sharing between both the domains for both low and high-level semantics. We utilise four cross-stitch units in the proposed MD-CSDNetwork. XceptionNet [14] is used as a backbone for both the branches. The last fully connected layers are concatenated and connected to a classifier. The first branch inputs pixel features in the RGB domain, and the second inputs the DCT spectrum coefficients in the frequency domain. Both the branches are separately normalized. We next present the two ingredients of the proposed network: cross-stitch unit and DCT spectrum.

1) *Cross-Stitch Unit*: Given two activation maps x_R and x_D obtained at a certain layer from the RGB and DCT branches respectively, we learn the linear combinations x'_R and x'_D of both the input activation maps which are then fed as inputs to the next layers (Eq. 1). This linear combination is parameterized using α . Specifically, at any location (i, j) in the activation map:

$$\begin{bmatrix} x'_R \\ x'_D \end{bmatrix} = \begin{bmatrix} \alpha_{RR} & \alpha_{RD} \\ \alpha_{DR} & \alpha_{DD} \end{bmatrix} \begin{bmatrix} x_R \\ x_D \end{bmatrix} \quad (1)$$

α_{RR} , α_{DD} provides the weights to the activations of the same branch (RGB and DCT, respectively), and α_{RD} , α_{DR} provides the weights to the activations of the other branch. The network can decide to make α_{RD} , α_{DR} zero for no sharing at all at a certain layer or choose a more shared representation by assigning higher values to these parameters. Their partial derivatives for loss L with tasks RGB and DCT are:

$$\begin{bmatrix} \frac{\partial L}{\partial x_R^{ij}} \\ \frac{\partial L}{\partial x_D^{ij}} \end{bmatrix} = \begin{bmatrix} \alpha_{RR} & \alpha_{DR} \\ \alpha_{RD} & \alpha_{DD} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial x_R^{ij}} \\ \frac{\partial L}{\partial x_D^{ij}} \end{bmatrix} \quad (2)$$

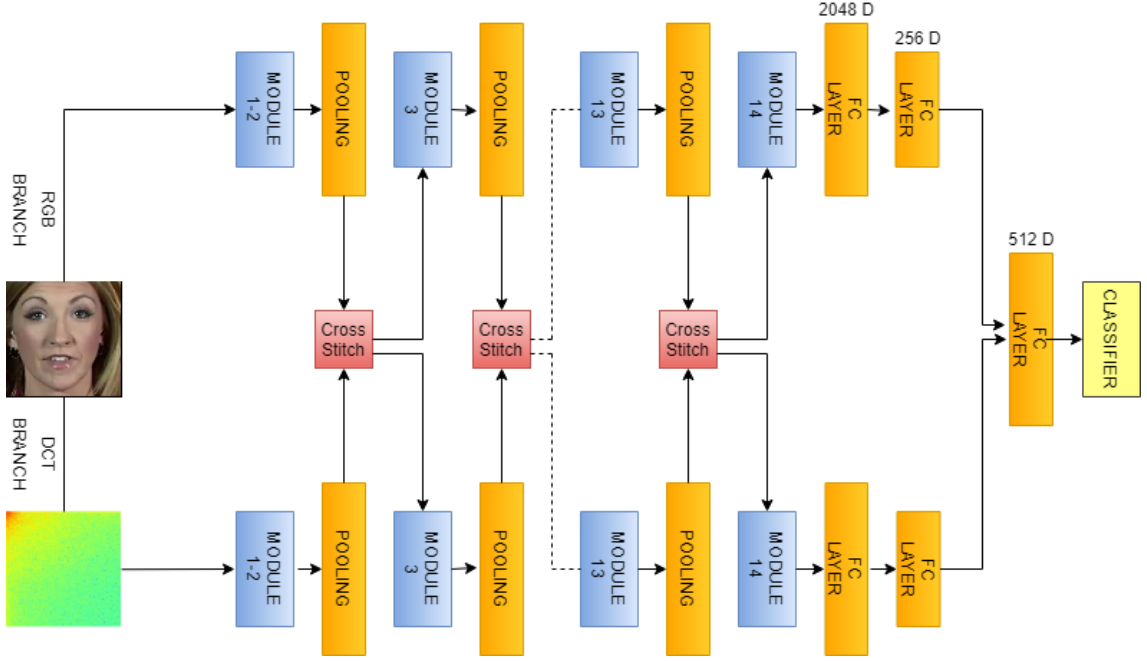


Fig. 2: Network Architecture for the proposed method. The two parallel branches carry the spatial and frequency information, respectively. The top branch inputs the pixel values, and the bottom branch inputs the log-scaled DCT spectrum coefficients. The cross-stitch [37] units are applied after the pooling layers in the backbone XceptionNet [14]. Multiple such units have been represented as dotted lines. In MD-CSDNetwork, four cross-stitch units have been used.

$$\frac{\partial L}{\partial \alpha_{RD}} = \frac{\partial L}{\partial x_D^{ij}} x_R^{ij}; \frac{\partial L}{\partial \alpha_{RR}} = \frac{\partial L}{\partial x_R^{ij}} x_R^{ij} \quad (3)$$

2) *DCT Spectrum*: The magnitude of the DCT coefficients signifies the contribution of a certain frequency in the image. We specifically compute a 2-D DCT as a product of 1-D DCTs across the rows and then across the columns separately for all the channels in the input image. DCT [10] has been widely used in image processing applications, especially in lossy compression, due to its excellent energy compaction properties. Due to this, the coefficients for high frequencies are much lower than low frequencies for natural images, and therefore we apply a natural logarithm to bring the coefficients at the same scale. The formula for 2-dimensional DCT is:

$$\begin{aligned} DCT(i, j) &= \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} pixel(a, b) \\ &\times \cos\left[\frac{(2a+1)i\pi}{2N}\right] \cos\left[\frac{(2b+1)j\pi}{2N}\right] \\ C(x) &= \frac{1}{\sqrt{2}} \text{ if } x = 0, \text{ else } 1 \text{ if } x > 0 \end{aligned} \quad (4)$$

C. Implementation Details

We initialize the spatial branch with ImageNet [18] weights and the frequency branch with random weights. For all the cross stitches, the same branch parameters (α_{RR} , α_{DD}) are initialized with 0.9 and different branch parameters (α_{RD} , α_{DR}) with 0.1 to ensure a convex linear combination

of the parameter values and to make output activation maps of the same order in magnitude as the input activation maps. We train with a base learning rate of 2e-4 and use a slightly higher learning rate of 1e-3 for the cross-stitches. The learning rate is reduced by a factor of 0.2 every time when the validation loss does not decrease for three consecutive epochs. The batch size used is 32, and the model with the highest validation accuracy is used for testing. We use Adam optimizer [27] with default values for the moments ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) and the whole model is trained end-to-end with a cross-entropy loss for binary classification. The network is trained with three V100 GPUs in DGX2 Server under TensorFlow programming environment. At an average, using this computational environment, a single training epoch on FF++ database takes about 4000 seconds.

III. EXPERIMENTS

A. Datasets and Protocol

For performance evaluation, we use three benchmark datasets, namely FaceForensics++ (FF++) [42], Celeb-DF [31], and Deepfake Detection (DFD) [1].

(a) **FF++** consists of 1000 real videos collected from YouTube and 1000 fake videos for each of the four manipulation types, i.e. FaceSwap [3], Face2Face [47], Deepfakes [2] and Neural Textures [46]. To better simulate real-world scenarios, these videos are also compressed using the H.264 codec to light compression (c23) and high compression (c40) levels. All the videos contain full frontal faces without any occlusions. We used the freely available face detector dlib [26] to extract the largest face in a frame and used an

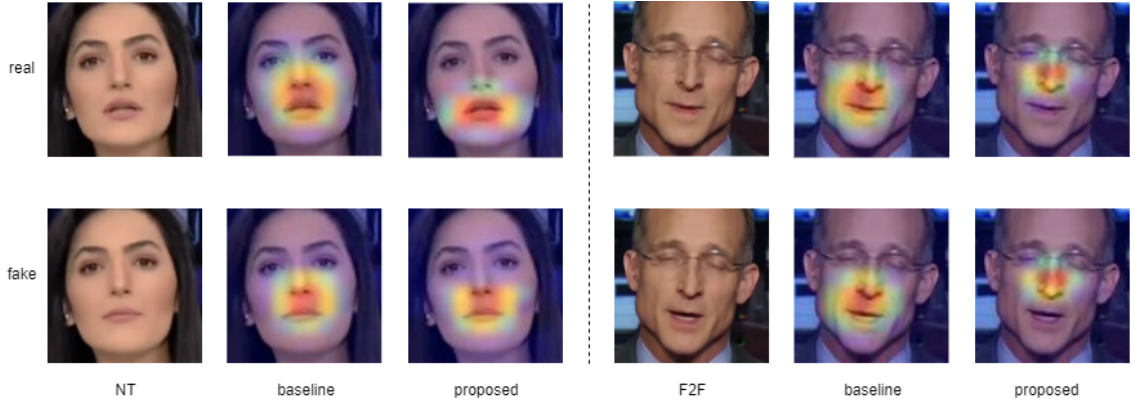


Fig. 3: The Grad-CAM visualization of baseline XceptionNet [14] and the proposed method for two different manipulations in FF++ [42]. Best viewed in color.

enlarged face crop with a scale factor of 1.3. We used 720 videos for training and 140 videos for both validation and testing. We extracted 270 consecutive frames per video for training and 100 consecutive frames per video for validation and testing as per the settings in [42]. We perform two kinds of experiments: (i) train and test on all manipulation types taken together (ii) train and test on specific manipulation types for both low and high compression levels. For training with all the manipulation types, we balance the number of frames for real and fake videos accordingly.

(b) **Celeb-DF** consists of 5639 fake videos and 590 real videos and (c) **DFD** which consists of 3068 fake videos and 363 real videos are recent high-quality datasets used for evaluating cross-database generalisation.

B. Evaluation Metrics

For all the experiments, we used the Accuracy (Acc) and the Area Under Receiver Operating Characteristic Curve (AUC) as the evaluation metrics. (a) **Acc.** Following FF++ [42], we use the accuracy metric as the major evaluation metric in all our experiments. It is the most widely used metric in the face forgery detection task. (b) **AUC.** Following the recent works [28], [31], [35], [40], we use this metric to evaluate the cross-data performance and ablation experiments. For better comparison with recent methods [32], [35], [40], [52], we report the average metric scores for all the frames in a video. For cross-database evaluation on Celeb-DF [31], we extract 25 frames per video and report the frame-level AUC scores as suggested in the original paper.

C. Results

In this section, we discuss the evaluation results for the proposed model and compare it with the recent methods in deepfake detection¹.

1) *Evaluation on FF++:* Tables 1 and 2 show the comparison of the results of the proposed method on the test dataset with other recent methods on FF++ [42] dataset. For both the experiments, significant improvements in both AUC and

accuracy metrics on high-quality HQ (or low-compression c23) levels is observed over the baseline XceptionNet. This shows that a combination of both spatial and frequency domain features can yield better results even for compressed videos, thus implying that frequency artifacts are robust to compression. For experiments with all manipulation types together, the proposed method achieves either comparable or better results when compared with the recent state-of-the-art methods such as [32], [35], [40], [45] on low compression. We observe comparatively lesser results for the high compression scenario, which can be attributed to not exploiting the inter-frame temporal artifacts within a video or patch-based artifacts.

For experiments on a single manipulation type, the network performs better than recent methods such as SSTNet [52] which extracts spatial, steganalysis, and temporal features for detecting tampered videos, SRMNet [59] which uses noise features extracted from SRM filters and comparable results with state-of-the-art results by F^3 -Net [40]. Among all four manipulation types, Neural Textures (NT) is the most challenging on both low and high compression due to highly realistic image synthesis. For a better understanding of the results, we visualize the regions of an image on which the network relies for classification using Grad-CAM [44]. As shown in Fig. 3, we observe that the proposed method focuses on a more localized region in both real and fake images as compared to the baseline method. This might be the result of better discriminative representations learned from both spatial and frequency domains.

2) *Evaluation on Cross Databases:* It is necessary to evaluate the generalizability of the algorithm to reflect real-world scenarios. We used two large-scale datasets - Celeb-DF [31] and DFD [1] for evaluating the transferability of our framework trained on FF++ [42]. This setting is more challenging since the testing set shares much less similarity with the training set. As shown in Table III, the proposed method shows better results when compared to several other previously published methods. Though, we still lag behind some state-of-the-art results [28], [32], [35], we obtain higher in-database results when compared with methods like [28],

¹The two best results have been marked in the tables with bold face and underline.

TABLE I: Quantitative results (Acc %) for both high-quality HQ (or low compression c23) & low-quality LQ (or high compression c40) settings on all four manipulation types i.e FaceSwap (FS) [3], Face2Face (F2F) [47], Deepfakes (DF) [2] and Neural Textures (NT) [46] in the FF++ [42] dataset.

Methods	FS		F2F		DF		NT	
	HQ	LQ	HQ	LQ	HQ	LQ	HQ	LQ
Steg. Features [22]	-	68.93	-	73.72	-	73.64	-	63.33
Cozzolino <i>et al.</i> [16]	-	73.79	-	67.88	-	85.45	-	78.00
Rahmouni <i>et al.</i> [41]	-	56.31	-	64.23	-	85.45	-	60.07
Bayar and Stamm [11]	-	82.52	-	73.72	-	84.55	-	70.67
MesoNET [6]	-	61.17	-	56.20	-	87.27	-	40.67
SRMNet [59]	-	89.10	-	92.70	-	91.90	-	-
FF++ [42]	<u>98.23</u>	92.17	<u>98.36</u>	88.95	<u>97.51</u>	93.46	<u>93.59</u>	78.89
SSTNet [52]	-	94.09	-	90.48	-	95.33	-	-
SPSL [32]	-	92.26	-	86.02	-	93.48	-	76.78
F^3 -Net [40]	-	96.53	-	95.32	-	97.97	-	83.32
Proposed MD-CSDNetwork	99.10	<u>94.64</u>	99.19	<u>93.57</u>	98.82	<u>97.34</u>	94.55	<u>81.78</u>

TABLE II: Quantitative results (Acc % and AUC) on the FaceForensics++ (FF++) dataset with high-quality HQ (or low compression c23) and low-quality LQ (or high compression c40) settings.

Methods	HQ		LQ	
	Acc	AUC	Acc	AUC
Steg. Features [22]	70.97	-	55.98	-
Cozzolino <i>et al.</i> [16]	78.45	-	58.69	-
Rahmouni <i>et al.</i> [41]	79.08	-	61.18	-
Bayar and Stamm [11]	82.97	-	66.84	-
Face X-ray [28]	-	0.8740	-	0.6160
DSP-FWA [30]	-	0.5749	-	0.6234
MesoNET [6]	83.10	-	70.47	-
Xception-ELA [24]	93.86	0.9480	79.63	0.8290
FF++ [42]	95.04	0.9509	80.32	0.8176
Two-branch [35]	96.43	0.9912	86.34	0.9110
EfficientNet-B4 [45]	96.63	0.9918	86.67	0.8820
F^3 -Net [42]	97.52	0.9810	90.43	0.9330
SPSL [32]	91.50	0.9532	81.57	0.8282
Proposed MD-CSDNetwork	<u>97.28</u>	0.9929	<u>87.57</u>	0.8901

[32] (refer Table II). The cross-database results on the DFD dataset are reported in Table IV. The table shows that it performs significantly better than the baseline and comparable results with recent state-of-the-art methods.

D. Ablation Study

To understand the effectiveness of the proposed network, we quantitatively determine the importance of each component of the model. We evaluate the results for (a) baseline XceptionNet using only features from the spatial domain, (b) XceptionNet using only features from the frequency domain, (c) Combination of both without any cross stitches, (d) Combination of both with a single cross stitch, and (e) Combination of both with all the cross stitches. Table V shows the AUC results for each of these scenarios. We observe that using features only from the frequency domain does not perform well as compared to using features only from the spatial domain. Therefore, we cannot lose all the spatial information and rely completely on frequency domain features for classification. Instead, a combination of both the domains improves the performance significantly. A simple hard combination of features from both the domains by concatenating the last fully connected layers improves the

TABLE III: Cross-dataset evaluation (AUC) on Celeb-DF [31]. The model is trained on FF++ and tested on the Celeb-DF dataset. The results in the first column report the AUC values when tested only on the deepfake class in FF++. Our method outperforms most of the listed methods in cross-generalization. Results for other methods are directly cited from [32].

Method	FF++	Celeb-DF
Two-stream [58]	0.7010	0.5380
Meso4 [6]	0.8470	0.5480
MesoInception4 [6]	0.8300	0.5360
HeadPose [54]	0.4730	0.5460
FWA [30]	0.8010	0.5690
VA-MLP [36]	0.6640	0.5500
Xception-raw [42]	0.9970	0.4820
Xception-c23 [42]	0.9970	0.6530
Xception-c40 [42]	0.9550	0.6550
Multi-task [38]	0.7630	0.5430
Capsule [39]	0.9660	0.5750
DSP-FWA [30]	0.9300	0.6460
Face-XRay [28]	<u>0.9912</u>	0.7420
F^3 -Net [40]	0.9797	0.6517
Two-Branch [35]	0.9318	<u>0.7341</u>
EfficientNet-B4 [45]	0.9970	0.6429
SPSL [32]	0.9691	0.7688
Proposed MD-CSDNetwork	0.9970	0.6877

TABLE IV: Cross-dataset evaluation (AUC) on DFD [1]. The model is trained on FF++ and tested on the DFD dataset. The results for other methods are directly cited from [33].

	Method	DFD
Trained on FF++	Xception [14]	0.8310
	Face X-Ray [28]	0.8560
	High Frequency [33]	0.9190
	Proposed MD-CSDNetwork	0.8951

AUC score. However in this case, there is no cross-talk between both the branches, and no layer parameters are shared. This gives limited room for information flow between both the domains.

To see the effect of sharing any parameters at the intermediate layers, we first inserted only a single cross-stitch after the first max-pooling layer to connect the two branches. The applied stitch enables gradient flow at low-level feature maps and allows both direct and indirect supervision from

TABLE V: Ablation study and component analysis of the proposed method on high quality (HQ) setting in FF++. We add each component step-by-step and compare the obtained results in terms of the AUC values.

ID	RGB Branch	Freq Branch	No Stitches	One Stitch	All Stitches	AUC
1	✓					0.9509
2		✓				0.8114
3	✓	✓	✓			0.9789
4	✓	✓		✓		0.9900
5	✓	✓			✓	0.9929

TABLE VI: Ablation study with different frequency transformations for extracting the frequency related features for our network. The table lists AUC values.

Transform	AUC
DWT [12]	0.9665
FFT [13]	0.9800
DCT [10]	0.9929

the branches. An improved result as compared to the case of a simple concatenation of fully connected layers verifies our initial hypothesis of features from both the domains functioning as related supervisory signals helping one another for better generalization and performance. Next, we apply the remaining cross stitches after all the pooling layers in the network and observe further improvement in classification performance. This allows sharing at multiple levels in the network and ensures learning a better optimal combination of shared representations from both feature domains.

In another set of ablation experiments, we study the effect of using a different transformation in place of DCT for extracting the frequency domain features. We use the Discrete Wavelet Transform (DWT) [12] and the Fast Fourier Transform (FFT) [13] as other types of image transformations. In FFT, the mathematical transformations applied to the input signal (or raw image) completely lose spatial information like DCT, whereas DWT offers a time-frequency representation of the image. For FFT, we use the amplitude spectrum as input frequency features in the network. We calculate the magnitudes of the series of complex numbers obtained as a result of the mathematical operations in FFT. As shown in Table VI, the best results are observed with DCT. Using FFT, which outputs pure frequency domain features, yields lower but comparable results with DCT. Interestingly, DWT (Haar Wavelet Transform) gives the lowest results even though it contains both spatial and frequency information.

IV. CONCLUSION AND FUTURE WORK

The paper presents MD-CSDNetwork, a novel approach of combining information obtained from spatial and frequency domains for improved face forgery detection. To learn this shared representation, we draw an analogy from the multi-task learning paradigm where multiple related supervisory tasks assist each other to learn better representations. We show that the features from two different domains (spatial and spectral) can also be treated as related supervisory

training signals and an optimal shared representation can be learned via a cross-stitched network architecture. We obtain benchmark results for both intra-database and cross-database evaluation on challenging datasets.

Generation and detection of deepfakes is like a cat and mouse game, and it requires continuously evolving detection methods as more efficient ways to create realistic media emerge. Current generation methods do not impose any constraints in the temporal dimension and may provide useful cues for improving the classification performance. In the future, we plan to explore inter-frame temporal artifacts to further improve the detection of deepfake videos.

ACKNOWLEDGEMENTS

M. Vatsa is partially supported by the Department of Science and Technology, Government of India through Swarnajayanti Fellowship.

REFERENCES

- [1] Deepfakedetection. <https://ai.googleblog.com/2019/09/contributing-data-to-deepfakedetection.html>. accessed: 2020-05-10.
- [2] Deepfakes github. <https://github.com/deepfakes/faceswap>. accessed: 2018-10-29.
- [3] Faceswap. <https://github.com/marekrowalski/faceswap/>. accessed: 2018-10-29.
- [4] <https://www.forbes.com/sites/robtoews/2020/05/25/deepfakes-are-going-to-wreak-havoc-on-society-we-are-not-prepared/?sh=64cc902d7494>.
- [5] <https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them>.
- [6] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.
- [7] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim. Detecting deep-fake videos from appearance and behavior. In *2020 IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2020.
- [8] S. Agarwal, H. Farid, O. Fried, and M. Agrawala. Detecting deep-fake videos from phoneme-viseme mismatches. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2814–2822, 2020.
- [9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. Protecting world leaders against deep fakes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [10] N. Ahmed, T. Natarajan, and K. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.
- [11] B. Bayar and M. C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, page 5–10, 2016.
- [12] P. Bentley and J. McDonnell. Wavelet transforms: an introduction. *Electronics & Communication Engineering*, 6(4):175–186, 1994.
- [13] R. N. Bracewell and R. N. Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.
- [14] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [15] U. A. Ciftci, I. Demir, and L. Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [16] D. Cozzolino, G. Poggi, and L. Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, page 159–164, 2017.
- [17] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva. Forensicttransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.
- [18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

- [19] R. Durall, M. Keuper, and J. Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [20] R. Durall, M. Keuper, F.-J. Pfrendt, and J. Keuper. Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*, 2019.
- [21] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning*, pages 3247–3258, 2020.
- [22] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
- [23] L. Guarniera, O. Giudice, and S. Battiato. Deepfake detection by analyzing convolutional traces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2841–2850, 2020.
- [24] T. Gunawan, S. Hanafiah, M. Kartiwi, N. Ismail, N. Za’bah, and A. Nordin. Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis. *Indonesian Journal of Electrical Engineering and Computer Science*, 7(1):131–137, 2017.
- [25] D. Güera and E. J. Delp. Deepfake video detection using recurrent neural networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6, 2018.
- [26] D. E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, 2009.
- [27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, Conference Track Proceedings*, 2015.
- [28] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [29] Y. Li, M.-C. Chang, and S. Lyu. In icu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.
- [30] Y. Li and S. Lyu. Exposing deepfake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [31] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3213, 2020.
- [32] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu. Spatial-phase shallow learning: Rethinking face forgery detection in frequency domain. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [33] Y. Luo, Y. Zhang, J. Yan, and W. Liu. Generalizing face forgery detection with high-frequency features. *arXiv preprint arXiv:2103.12376*, 2021.
- [34] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi. Do gans leave artificial fingerprints? In *IEEE Conference on Multimedia Information Processing and Retrieval*, pages 506–511, 2019.
- [35] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed. Two-branch recurrent network for isolating deepfakes in videos. In *European Conference on Computer Vision*, pages 667–684. Springer, 2020.
- [36] F. Matern, C. Riess, and M. Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE Winter Applications of Computer Vision Workshops*, pages 83–92, 2019.
- [37] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [38] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2019.
- [39] H. H. Nguyen, J. Yamagishi, and I. Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2307–2311, 2019.
- [40] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European Conference on Computer Vision*, pages 86–103. Springer, 2020.
- [41] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *2017 IEEE Workshop on Information Forensics and Security*, pages 1–6, 2017.
- [42] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [43] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019.
- [45] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning*, volume 97, pages 6105–6114, 2019.
- [46] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph.*, 38(4), 2019.
- [47] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104, 2018.
- [48] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020.
- [49] L. Trinh, M. Tsang, S. Rambhatla, and Y. Liu. Interpretable and trustworthy deepfake detection via dynamic prototypes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1973–1983, 2021.
- [50] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu. Fakespotter: A simple yet robust baseline for spotting ai-synthesized fake faces. pages 3444–3451. International Joint Conferences on Artificial Intelligence Organization, 2020.
- [51] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020.
- [52] X. Wu, Z. Xie, Y. Gao, and Y. Xiao. Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2952–2956, 2020.
- [53] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9544, 2019.
- [54] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8261–8265, 2019.
- [55] N. Yu, L. Davis, and M. Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE/CVF International Conference on Computer Vision*, pages 7555–7565, 2019.
- [56] X. Zhang, S. Karaman, and S.-F. Chang. Detecting and simulating artifacts in gan fake images. In *IEEE International Workshop on Information Forensics and Security*, pages 1–6, 2019.
- [57] P. Zhou, X. Han, V. Morariu, and L. Davis. Two-stream neural networks for tampered face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.
- [58] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-stream neural networks for tampered face detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1831–1839, 2017.
- [59] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018.