Image Operation Chain Detection with Machine Translation Framework

Yuanman Li, *Member, IEEE*, Jiaxiang You, *Student Member, IEEE*, Jiantao Zhou*, *Senior Member, IEEE*, Wei Wang, *Member, IEEE*, Xin Liao, *Senior Member, IEEE* and Xia Li, *Member, IEEE*

***Abstract*—The aim of operation chain detection for a given manipulated image is to reveal the operations involved and the order in which they were applied, which is significant for image processing and multimedia forensics. Currently, *all* existing approaches simply treat image operation chain detection as a classification problem and consider only chains of at most two operations. Considering the complex interplay between operations and the exponentially increasing solution space, detecting longer operation chains is extremely challenging. To address this issue, in this work, we devise a new methodology for image operation chain detection. Different from existing approaches based on classification modeling, we strategically conduct operation chain detection within a machine translation framework. Specifically, the chain in our work is modeled as a sentence in a target language, with each possible operation represented by a word in that language. When executing chain detection, we propose first transforming the input image into a sentence in a latent source language from the learned deep features. Then, we propose translating the latent language into the target language within a machine translation framework and finally decoding all operations, arranged in order. Besides, a chain inversion strategy and a bi-directional modeling mechanism are developed to improve the detection performance. We further design a weighted cross-entropy loss to alleviate the problems presented by imbalance among chain lengths and chain categories. Our method can detect operation chains containing up to seven operations and obtains very promising results in various scenarios for the detection of both short and long chains.**

***Index Terms*—Operation chain detection, image forensics, machine translation, Transformer**

This work was supported in part by the Natural Science Foundation of China under Grant 62001304, Grant 61871273, Grant 61972142 and Grant 61971476; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010645; in part by the Foundation for Science and Technology Innovation of Shenzhen under Grant RCBS20210609103708014; in part by CCF-Alibaba Innovative Research Fund For Young Scholars; in part by Alibaba Group through Alibaba Innovative Research Program; in part by Macau Science and Technology Development Fund under SKLIOTSC-2021-2023, 0072/2020/AMJ, and 0022/2022/A1; and in part by Research Committee at University of Macau under MYRG2020-00101-FST and MYRG2022-00152-FST. (Corresponding author: Jiantao Zhou)

Yuanman Li, Jiaxiang You and Xia Li are with Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China (email: yuanmanli@szu.edu.cn).

Jiantao Zhou is with the State Key Laboratory of Internet of Things for Smart City, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China. (email: jtzhou@um.edu.mo)

Wei Wang is with School of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen 518033, China (email: wangw328@mail.sysu.edu.cn).

Xin Liao is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: xinliao@hnu.edu.cn).

I. Introduction

Operation chain detection plays an essential role in digital image analysis. Many acquired digital images have been degraded by a sequence of image operations, such as JPEG compression, Gaussian noise, resampling, enhancement, etc. Knowing the operation chain applied to a given image could provide more details regarding its degradation history, which are important for many image processing tasks [1]–[3]. Take image restoration with multiple degradations as an example. The degradation history could be used for building a model synthesizing a large number of original-degraded pairs, and subsequently, training a deep network for better image restoration. On the other hand, the information on the operation chain is vital for image forensic purposes as well. Nowadays, image forgeries widely exist in Internet rumors, fake news and dishonest academic literature, detrimentally influencing many aspects of human life. When creating image forgeries, many types of common image operations are usually involved for hiding the manipulation traces or making the resulting images look realistic. Against this background, many image forensic algorithms have been designed to verify the authenticity of images and reveal their operation chain [4], [5].

The majority of previous efforts have focused on the design of forensic methods for identifying a single manipulation. Such methods can be roughly divided into two categories, i.e., targeted operation detection [6], [7] and general-purpose operation detection [8]–[10]. In reality, however, forging an image often involves multiple operations. Thus, investigators have begun to consider operation chain detection. Two examples are shown in Fig. 1, in which an image has been manipulated by means of five operations in different orders. Different from traditional specific and general-purpose manipulation detection approaches, operation chain detection requires simultaneously identifying multiple operations applied to an image and determining their order to reveal the complete image processing history [4]. The task of operation chain detection is beyond the capabilities of traditional forensic algorithms for the following reasons. First, traditional forensic approaches can identify only a single potential operation, and how to effectively accommodate them in revealing multiple operations does not appear to be straightforward. Second, determining the order of the detected operations is still not easy, especially when the number of operations is large. Third, due to the interplay among different image operations, operations applied later may affect and disguise the traces left by previous operations, making some operations difficult to be detected.

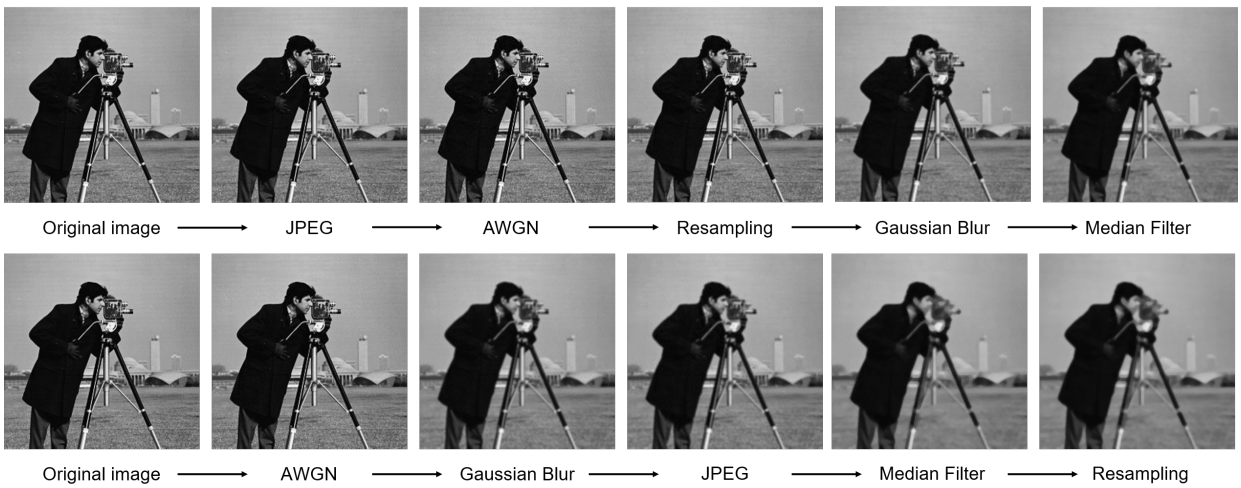


Fig. 1. An image processed by two operation chains of five operations. The chain applied to the top image is JPEG→AWGN→Resampling→Gaussian Blur→Median Filter, while the chain applied to the bottom image is AWGN→Gaussian Blur→JPEG→Median Filter→Resampling.

Considering the above difficulties, limited progress has been achieved in operation chain detection. For example, the work [11] theoretically studied the problem of quantifying the distinguishability between different operator chains using information theoretic measures. The works [12], [13] formulated operation chain detection as multiple hypothesis testing problems. In more recent papers [4], [5], attempts were made to detect operation chains using deep convolutional neural networks (CNNs), achieving promising progress. To the best of our knowledge, all existing algorithms simply treated operation chain detection as a classification problem, and confronted with the following challenges:

- *All* previous approaches consider only chains consisting of at most two operations. As the chain becomes longer, the size of the solution space increases exponentially. For instance, with two operations, there are only $\sum_{i=0}^{2} A_2^i = 5$[1] possible chains, while seven operations yield $\sum_{i=0}^{7} A_7^i = 13700$ possible solutions. Even worse, the interactions between operations are extremely complicated for a long operation chain, with previously created fingerprints potentially being strongly affected or even disguised by subsequent operations. Due to the above challenges, how to detect long operation chains is still an open problem.
- *All* existing approaches model chain detection as a classification problem by directly assigning each chain a unique label. Although this strategy seems very straightforward, it cannot fully exploit the order (or partial order) information of the operations and cannot fully utilize the *strong correlations* among different chains. For example, the relationship between the chains JPEG→AWGN→Resampling and AWGN→Resampling is obscured by unique chain labelling.

Plagued by these issues, image operation chain detection is still in its infancy, and the limited progress achieved through classification-based approaches urges us to consider new methodologies, especially for the detection of long chains.

In this paper, a novel end-to-end deep framework for image operation chain detection is devised, which we name *TransDetect*. Different from previous approaches based on classification modeling, we address the operation chain detection problem within a machine translation framework, and our method inherits the capability of machine translation in decoding very long sentences. Specifically, we model an operation chain as a sentence in a target language, and each image operation is represented by a word in that language. Then, the aim of image operation chain detection is to find a mapping from the processed image domain to the target language domain. To this end, the input image is first transformed into a deep feature space, and the features in this space are then converted into a sentence in a latent source language to describe the fingerprints of the chain. Finally, we propose to translate the latent language into the target language using a machine translation framework, thereby decoding the whole operation chain. We further propose hierarchical feature extraction, chain inversion, bidirectional modeling and weighted cross-entropy loss strategies to enhance the detection performance.

To the best of our knowledge, this is the *first* work to formulate operation chain detection as a generic machine translation problem, and consider chains containing up to seven operations in a unified framework. We summarize the primary contributions and novelties of this work as follows:

- We strategically model an operation chain as a time series (concretely, a sentence in a target language) rather than directly assigning it a simple label. Compared with existing approaches, our chain model preserves the properties of the original chains, such as the order of operations and the strong correlations among similar chains. Based on the proposed chain model, we then devise a novel image operation chain detection method within a machine translation framework.
- Similar to other machine translation tasks, our method decodes the operation chain progressively, i.e., operation

[1]In this work, we adopt the notation A_M^N to count the number of arrangements of N elements taken from a given set of size M, i.e., N-permutations of M.

by operation. This desirable property endows our method with the ability to utilize previously decoded operations as prior information to assist in decoding the next operation and greatly relieves the problem of the large solution space of long chain detection. We further propose two strategies, i.e., chain inversion and bidirectional modeling to improve the detection performance.
- To detect chains of unknown lengths, we design a weighted cross-entropy loss to handle the problem of imbalance among chain lengths and chain categories.
- Extensive experiments demonstrate that our method achieves very promising results in various scenarios for the detection of both short and long chains.

Portions of this work have previously appeared in [14] as a conference version. In comparison to [14], both the technical and experimental parts have been substantially refined. We summarize the primary improvements as below: First, we reformulate operation chain detection from a generic machine translation perspective, and details the merits of our translation strategy in preserving the strong correlations among similar chains. Second, we propose to address the fundamental imbalance problem caused by the chain lengths and categories, and design a Weighted Cross-Entropy Loss function (WCEL) to tackle this issue, where the parameter selection is also justified. Third, we propose a chain inversion strategy and a bidirectional modeling strategy to mitigate the error propagation caused by early decoded subchains with less confidence, and also propose a hierarchical feature extraction strategy to fuse multilevel features for a richer language representation. Both strategies are justified in the ablation study. Fourth, comparing with the work [14] only providing preliminary results on balanced length experiments with chains containing up to 5 operations, we conduct extensive additional experiments to demonstrate our superiority, including (but not limited to), a new evaluation metric, i.e., Bilingual Evaluation Understudy (BLEU) score, balanced category experiments, cross-dataset validation, comparison with more recent methods, and challenging cases (e.g., unknown parameters and chains consisting of 7 operations). In addition, the performance is significantly improved over [14], e.g., 12.69% ACC improvement over [14] on chain detection of unknown length. Last but not least, we insert a new subsection V-A to discuss the shortcomings and future directions of our work.

The remainder of this paper is organized as follows. Section II reviews related works. Section III details our proposed TransDetect algorithm for operation chain detection. Extensive experimental results are presented in Section IV, and we finally present our conclusion in Section VI.

II. Related Works

In this section, we briefly review some related works on manipulation detection and the popular machine translation framework known as Transformer.

A. Operation Detection

Targeted operation detection The goal of targeted manipulation detection methods is to detect a single specific manipulation. These methods are commonly formulated as binary classification problems by extracting the features of traces left by specific operations, such as resampling [6], [15], median filtering [7], [16], contrast enhancement [17], and JPEG compression [18]. For example, Qiao *et al.* [6] devised a statistical framework for resampling operation detection by analyzing the behavior of the residual noise. Pasquini *et al.* [19] theoretically discussed the detectability of resampling operations on the basis of hypothesis testing theory. Cao *et al.* [17] designed two contrast enhancement detectors based on zero-height gap fingerprints and blockwise peak/gap bins, considering images that had been previously JPEG compressed. Chen *et al.* [16] proposed a median filter operation detector by highlighting the statistical artifacts introduced into the difference domain. Kang *et al.* [7] studied the autoregressive coefficients computed from a median-filtered residual map and then devised an algorithm for the detection of median filtering from compressed images. In addition to the above approaches, there are also many other commonly considered types of manipulation [20]–[25].

General-purpose operation detection In reality, prior information about which operation has been applied is usually not available. In such a case, investigators may run several individual targeted detectors to authenticate an image. However, determining how to fuse the results of different forensic tests while controlling the overall false alarm rate is very challenging in practice. To address this issue, some recent works have studied general-purpose image forensic algorithms, with the aim of revealing the type of manipulation from a pool of potential operations through a single test. Different from targeted manipulation detection, general-purpose image forensic approaches model manipulation detection as a multi-class classification problem [8]–[10], [26]. For example, Qiu *et al.* [10] modeled various image processing operations as steganography and designed a general-purpose image forensic method based on universal steganalytic features, achieving the ability to distinguish six kinds of typical operations. Li *et al.* [9] studied the correlations among adjacent pixels in the residual domain and designed a general-purpose forensic algorithm to differentiate eleven image operations and four anti-forensic operations. Bayar *et al.* [5] designed an end-to-end CNN-based general-purpose image manipulation detector (MISLnet), for which a constrained convolutional layer was devised to learn the features created by manipulations and suppress the content of the manipulated images. Recently, Wu *et al.* [8] proposed an end-to-end fully convolutional network for general-purpose manipulation detection, called ManTra-Net, which considered up to 385 types of image manipulation derived from seven manipulation families. However, the top-1 detection accuracy was only approximately 50%.

Operation chain detection It should be noted that both targeted and general-purpose manipulation detection methods can reveal only a single applied operation from a pool of potential operations. In practice, however, multiple operations are typically sequentially applied to an image. Recently, researchers have begun to attempt to detect operation chains, which requires determining both the operations involved and their order. Boroumand *et al.* [27] proposed a deep-learning-

based image processing history detection algorithm in which an operation is assumed to be inserted between two JPEG compression operations with different quality factors. Chen *et al.* [28] designed an image processing history detection algorithm based on an automated CNN architecture, considering the combination of resizing and four other operations. The authors of [12], [13] formulated the problem of detecting the order of operations as a multiple hypothesis testing problem and then studied the question of when the operation order can or cannot be detected through an information theoretical framework. Bayar *et al.* [5] proposed a constrained CNN to detect image operation chains consisting of up to two image operations. Recently, Liao *et al.* [4] devised a two-stream CNN for detecting a chain of two image operations based on features extracted from both the spatial domain and the transform domain. Note that all existing methods treat operation chain detection as a multiclass classification problem and consider only chains of at most two operations.

TABLE I
OPERATION DICTIONARY.

Word	Operation	Parameter
O_0	Padding Symbol	-
O_s	Start Symbol ($\hat{S}$)	-
O_e	End Symbol ($\hat{E}$)	-
O_1	Gaussian Blur (GB) with $\sigma = 1.1$	kernel size = 5
O_2	Median Filter (MF)	kernel size = 5
O_3	Resampling Using Bilinear Interpolation (RS)	Scaling = 1.5
O_4	Additive White Gaussian Noise (AWGN)	$\sigma = 2$
O_5	JPEG Compression (JPEG)	QF = 70
O_6	Histogram Equalization (HE)	-
O_7	Unsharp Masking (USM) Using a Laplacian Kernel	$\lambda = 1$

B. Transformer

Machine translation aims to convert one (source) language into another (target) language. Considering the sequential property of sentences, many traditional machine translation models are based on recurrent encoder-decoder frameworks, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks [29] and gated recurrent unit (GRU) networks [30]. The Transformer, a simple network architecture based solely on self-attention mechanisms, has recently been proposed to enable the parallelization of the training process [31]. Notably, Transformers eschew recurrence and can model global dependencies without regard to their distance in the input or output sequences. Along with the tremendous success of this architecture in machine translation, researchers have begun to apply Transformers to computer vision tasks. The Vision Transformer (ViT) [32] represents pioneering work on the implementation of a pure Transformer encoder for an image classification task. ViT sequentializes the input image into a series of tokens and then feeds them into a standard Transformer encoder with multihead self-attention. Motivated by this, many other Transformer-based vision architectures have also been devised [33], [34], such as DeiT [33] and Swin Transformer [34].

III. PROPOSED TRANSDETECT ALGORITHM FOR OPERATION CHAIN DETECTION

In this section, we discuss our proposed TransDetect algorithm for operation chain detection. Different from existing algorithms, we formulate operation chain detection as a machine translation problem rather than a classification problem.

A. Problem Formulation

In our work, an operation chain T_i is defined as an ordered sequence of operations consecutively applied to an image [2]. We assume that the manipulations are drawn from a pool of operations $\mathcal{O} = \{O_1, O_2, ..., O_M\}$, and then T_i can be written as

$$T_i = O'_1 \to O'_2 \to ... \to O'_m, \quad O'_1, .., O'_m \in \mathcal{O}. \tag{1}$$

For a chain length of N, the total number of possible chains is A_M^N. Suppose that the maximum length of a chain is N ($N \leq M$); then, there are

$$K = A_M^0 + A_M^1 + A_M^2 + ... + A_M^N, \tag{2}$$

distinct chains. Note that A_M^0 in (2) indicates that an image is not modified, i.e., the chain is of length 0. Let $\mathcal{T}_{chain} = \{T_1, T_2, ..., T_K\}$ contain all possible chains; then, operation chain detection aims to reveal the correct chain from $\mathcal{T}_{chain}$ for a given manipulated image.

It should be emphasized that it is also helpful for image forensic analysis to reveal some long subchains of the complete operation chain. Therefore, in this work, we also define the subchains of a given chain as its consecutive subsequences. For instance, the chain $T_i = O_1 \to O_2 \to O_3$ has subchains O_1, O_2, O_3, $O_1 \to O_2$, $O_2 \to O_3$, and $O_1 \to O_2 \to O_3$.

We can readily derive that the size of $\mathcal{T}_{chain}$ will increase exponentially as M and N increase, which makes the resulting problem extremely challenging. All previous works considered only chains of up to two operations. To handle the long chain detection problem, we conduct operation chain detection within a machine translation framework, where each operation is represented by a word of a sentence in a target language. Compared with existing approaches, our machine translation based method can preserve the order information of original chains and the strong correlations among similar chains. Additionally, our method detects chain progressively, where the prior information of previously decoded operations is used to assist in the next operations detection, thus greatly relieving the problem of the large solution space of long chain detection. Furthermore, as will be shown in Section III-D, by resorting to the sequential modeling of the operation chain, successfully detected subchains can also effectively guide the learning process of our framework.

[2]Similar to [4], [12], we assume that each operation is not applied more than once unless otherwise specified.

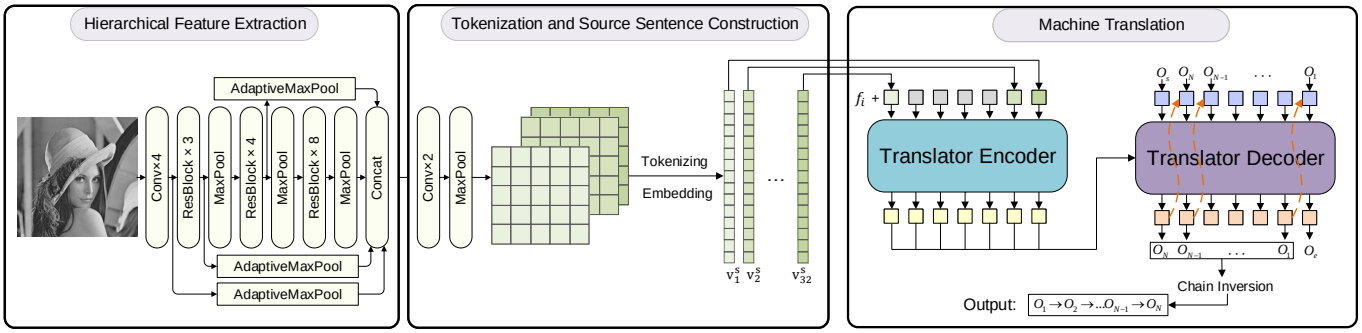


Fig. 2. The framework of our proposed TransDetect algorithm.

In this paper, we consider up to seven kinds of image operations, which yields 13700 possible operation chains in total. The considered operations and associated parameters are summarized in Table I. For instance, an image processed with the chain MF→GB→HE→USM→AWGN→JPEG can be represented as $O_2 \rightarrow O_1 \rightarrow O_6 \rightarrow O_7 \rightarrow O_4 \rightarrow O_5$. Note that operations O_1–O_5 were also employed in [5]. In addition to the possible image operations, Table I also lists three auxiliary symbols, which will be used for machine translation. O_0 is a padding symbol used in Transformer for parallel training. O_s and O_e are the start and end symbols for decoding.

B. Proposed TransDetect Algorithm

Fig. 2 illustrates the framework of our proposed TransDetect algorithm. Our method primarily consists of three procedures: 1) hierarchical feature extraction, 2) tokenization and source sentence construction, and 3) language translation.

1) Hierarchical Feature Extraction: Due to the complex interactions between operations, learning good feature representations is important for operation chain detection. In our work, we propose extracting hierarchical features of the traces left by mixed operations using a deep architecture. Specifically, our feature extractor is composed of three kinds of layers detailed as follows. Conv: The first layer has 64 filters with a kernel size of 3×3. We apply batch normalization (BN) for feature normalization and use the rectified linear unit (ReLU) activation function for nonlinear mapping. ResBlock: To extract hierarchical features of mixed operations, we stack a set of residual blocks, each of which has two convolutional layers with one skip connection. MaxPool: Three MaxPooling functions are concatenated after each ResBlock to produce more compact features, each reducing the feature maps to one quarter of their original size.

Due to the interplay among different operations, the strengths of the traces left by different operations could differ greatly; some traces could be severely weakened, while others may be of high strength. To extract richer feature representations, we extract both low-level and high-level features, which are concatenated at the end of the hierarchical feature extraction module, as shown in Fig. 2. The AdaptiveMaxPool operation [35] is further applied to features of different levels to produce feature maps of the same size.

For simplicity, we denote the hierarchical feature extraction process by $\mathcal{F}_{fe}(\cdot)$. For a given input image I, its deep representation is computed as

$$F_o = \mathcal{F}_{fe}(I). \tag{3}$$

F_o has dimensions of $C \times H' \times W'$, where C denotes the number of channels; W' and H' respectively represent the width and height of each feature map.

2) Tokenization and Source Sentence Construction: Note that the input to a machine translator is expected to be a time series signal. Thus, how to transform the deep feature representation F_o into a time series to carry the fingerprints of the chain is very important, and doing so is necessary for the translation process. For machine-translation-based image processing tasks, most existing approaches split the input image into a sequence of tokens in the spatial domain [32], [34]. In our case, the manipulations are performed globally and thus are invariant with respect to spatial position. Motivated by this fact, we propose a channelwise tokenization strategy to convert the features into a time series signal. Let the tokenization function be denoted by $\mathcal{T}_{source}(\cdot)$; then, we transform F_o into a time series as follows:

$$S_o = \mathcal{T}_{source}(F_o). \tag{4}$$

For simplicity, we implement the function $\mathcal{T}_{source}(\cdot)$ as two Conv+BN+ReLU layers concatenated with MaxPooling and channel flattening. To trade off the accuracy and complexity, in our experiment, the number of filters of the first and second convolutional layers are empirically set to 64 and 32, respectively. Intuitively, $\mathcal{T}_{source}(\cdot)$ maps the features into *a latent source language space*, where the time series signal S_o can be regarded as a sentence of 32 words in the source language. We should emphasize that the words in the source language space do not one-to-one correspond to the image operations shown in Table I, and the sentence S_o summarizes the complex behavior of the target chain acting on the image.

Denote $S_o = [W_1, ..., W_{32}]$, where for each word, $W_i \in \mathbb{R}^{H'W'}$. Similar to other word embedding strategies [32], we embed each word W_i into a vector of 512 dimensions. Mathematically,

$$v_i^s = Embed_s(W_i), \quad i \in \{1, ..., 32\}. \tag{5}$$

In this paper, we use a fully connected layer to implement the word embedding function.

In summary, by applying the first two procedures, an image can be transformed into an embedded sentence in the latent source language, which is denoted by $V^s = [v_1^s, v_2^s, v_3^s ... v_{32}^s] \in \mathbb{R}^{32\times512}$.

3) Language Translation: Different from the existing methods, where each target chain is assigned a distinct label, our method models a chain as a sentence in a target language space, and each operation is represented by a word in that sentence. The sentence length varies with the length of the operation chain. Fundamentally, our framework is compatible with any machine translation model. Motivated by the great success of Transformer [31] in natural language processing (NLP) and computer vision tasks [32], [36], [37], we employ Transformer [31] for language translation. As shown in Fig. 3, the Transformer architecture is composed of an encoder and a decoder, similar to other machine translation frameworks.

(i) On the encoder side, Transformer adds a positional encoding to each embedded vector v_i^s to encode the relative position of the word in the sentence. Different from the original Transformer, which adopts a hard-coded position embedding strategy, in this paper, we encode the word positions in a learnable way, as proposed in [38]. Specifically,

$$f_i = PositionalEncoding(i), \tag{6}$$

where $PositionalEncoding(\cdot)$ maps the i-th position in the sentence to a vector $f_i \in \mathbb{R}^{1\times512}$. The embedded words with position coding,

$$V_0^s = [v_1^s + f_1, v_2^s + f_2, ...v_{32}^s + f_{32}], \tag{7}$$

are then fed into the Transformer encoder.

The architecture of the Transformer encoder is shown in Fig. 3(a). Specifically, we construct a Transformer encoder with L layers following [31], where each layer contains a feedforward network (FFN) and a multihead self-attention (MSA) block. The encoding procedure can be formulated as follows:

$$\begin{aligned}
&Q_{\ell-1}^{s,i} = FC_{qi}(V_{\ell-1}^s),\\
&K_{\ell-1}^{s,i} = FC_{ki}(V_{\ell-1}^s),\\
&J_{\ell-1}^{s,i} = FC_{vi}(V_{\ell-1}^s),\\
&Head_i = SA(Q_{\ell-1}^{s,i}, K_{\ell-1}^{s,i}, J_{\ell-1}^{s,i}), \qquad i = 1, ..., U,\\
&T_{\ell-1}^s = LN(FC(cat[Head_1, ..., Head_U]) + V_{\ell-1}^s),\\
&V_\ell^s = LN(FFN(T_{\ell-1}^s) + T_{\ell-1}^s), \qquad \ell = 1, ..., L,\\
&[z_1, z_2, z_3 ... z_{32}] = Z^s = V_L^s.
\end{aligned} \tag{8}$$

Here, U denotes the number of heads, $FC_{xi}(\cdot)$ is a fully connected layer for the i-th head, $LN(\cdot)$ represents the Layernorm operation, and $FFN(\cdot)$ is a block of two fully connected layers. $SA(\cdot)$ serves as the self-attention mechanism, which is formulated as

$$SA(Q, K, J) = softmax(\frac{QK^T}{\sqrt{d_k}})J, \tag{9}$$

where Q, K and J correspond to the query, key and value matrices. $Z^s \in \mathbb{R}^{32\times512}$ is the output of the encoder.

(ii) On the decoder side, similar to other translation methods, we associate each sentence with a start symbol (O_s) to control the status of the decoder. For the target operation chain

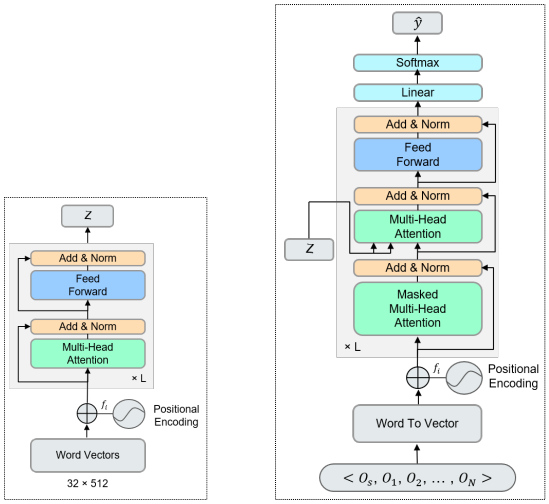


Fig. 3. The framework of Transformer.

$O_1 \to O_2 \to ... \to O_N$, each operation will be embedded into a vector space. We have the following:

$$\begin{aligned}
&v_s^t = Word2Vec(O_s), \quad v_e^t = Word2Vec(O_e),\\
&v_i^t = Word2Vec(O_i^t), \qquad\qquad i = 1, ..., N.
\end{aligned} \tag{10}$$

In our work, we implement $Word2Vec(\cdot)$ by employing the *nn.embedding* function of the PyTorch deep learning framework, where each operation in the target language space will be first indexed from the operation dictionary, and then the corresponding index will be mapped into a vector space.

The architecture of the decoder, which is shown in Fig. 3(b), consists of L layers. Each layer includes a masked MSA block, an MSA block and an FFN block with one skip connection. The goal of the masked MSA block is to force the decoding of the current word not to rely on undecoded words. Similar to the encoder, we adopt a learnable position embedding strategy to encode the relative positions of the words. Note that by means of the Masked-MSA(MMSA) module, Transformer can decode the words in parallel in the training stage, thus simultaneously outputting the probabilities of all operations in the target operation chain. Mathematically, we can formulate the whole decoding procedure as follows:

$$\begin{aligned}
&V_0^t = [v_s^t + f_0, v_1^t + f_1, v_2^t + f_2, ...v_N^t + f_N],\\
&Head = MMSA(Q_{\ell-1}, K_{\ell-1}, J_{\ell-1}),\\
&G_{\ell-1} = LN(FC(Head) + V_{\ell-1}^t),\\
&T_{\ell-1}^t = LN(MSA(FC_q(G_{\ell-1}), FC_k(Z^s), FC_v(Z^s)) + G_{\ell-1}),\\
&V_\ell^t = LN(FFN(T_{\ell-1}^t) + T_{\ell-1}^t), \qquad \ell = 1, ..., L,\\
&\hat{y} = Softmax(FC(V_L^t)).
\end{aligned} \tag{11}$$

We define $Q_{\ell-1}$ as a set containing query matrices for the U heads, where $Q_{\ell-1}^i = FC_{qi}(V_{\ell-1}^t)$. We can similarly define $K_{\ell-1}$ and $J_{\ell-1}$. Then, the procedures of the $MMSA(\cdot)$ operation shown in Eq. (11) can be written as

$$\begin{aligned}
&Head_i = SA_{masked}(Q_{\ell-1}^i, K_{\ell-1}^i, J_{\ell-1}^i)\\
&Head = cat[Head_1, Head_2, ..., Head_U], \ i = 1, ..., U.
\end{aligned} \tag{12}$$

Here, SA_{masked} denotes the masked self-attention mechanism adopted in Transformer. In the inference stage, similar to other

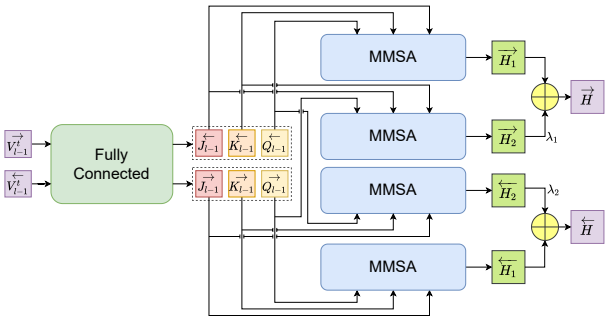


Fig. 4. Illustration of our bidirectional modeling strategy.

machine translation frameworks, Transformer decodes the input sentence progressively, i.e., word by word. This allows our TransDetect algorithm to fully utilize the information of previously decoded operations to facilitate the identification of subsequent operations. More details about Transformer are given in [31]. After obtaining the probabilities of different operations, we then select the operation with the maximum probability.

C. Chain Inversion and Bidirectional Modeling

One primary challenge of operation chain detection is that the traces left by previous operations could be severely weakened by later operations. Then, using decoded early subchains of lower confidence to assist in the decoding of later operations would exacerbate the propagation of error. In this paper, we develop two strategies to remedy this problem, i.e., the chain inversion and the bidirectional modeling.

For the chain inversion strategy, we propose to invert the chain order, i.e., converting the original target chain $O_1 \rightarrow O_2 \rightarrow ... \rightarrow O_N$ (L2R, from the left side to the right side) to $O_N \rightarrow O_{N-1} \rightarrow ... \rightarrow O_1$ (R2L, from the right side to the left side) before calculating the loss. Such inversion forces the latest operations in the chain, which are likely to have left more distinct traces on the image, to be decoded first. By virtue of the progressive decoding nature of Transformer, the operations that are decoded earliest with high confidence will then play a beneficial role as prior information for identifying preceding operations from their relatively weak traces.

Despite the merit of the chain inversion strategy, some prediction errors could still be propagated and make the preceding operations difficult to be detected. Inspired by [39], in this work, we further propose a bidirectional modeling strategy, and let the decoder interactively integrate the information from both L2R and R2L directions. The mutual constraints from opposite directions can greatly alleviate the error accumulation, and thus improve the final detection performance. Fig. 4 illustrates the procedure of our proposed bidirectional modeling strategy. Specifically, similar to the second equation of Eq. (11), we first compute the attention results for L2R and R2L by

$$\begin{aligned} \overrightarrow{H}_1 &= MMSA(\overrightarrow{Q}_{\ell-1}, \overrightarrow{K}_{\ell-1}, \overrightarrow{J}_{\ell-1}), \\ \overleftarrow{H}_1 &= MMSA(\overleftarrow{Q}_{\ell-1}, \overleftarrow{K}_{\ell-1}, \overleftarrow{J}_{\ell-1}), \end{aligned} \tag{13}$$

TABLE II
NUMBER OF SAMPLES IN EACH CATEGORY (N_C) OR OF EACH LENGTH (N_L) USING BCS AND BLS. THE TOTAL NUMBER OF SAMPLES IS 400K.

Operation Set $=\{O_1, O_2, ..., O_7\}$					
Length n	Categories K	BCS		BLS	
		N_C	(N_L)	N_L	(N_C)
n=0	$A_7^0 = 1$	29	(29)	50000	(50000)
n=1	$A_7^1 = 7$	29	(204)	50000	(7143)
n=2	$A_7^2 = 42$	29	(1226)	50000	(1190)
n=3	$A_7^3 = 210$	29	(6131)	50000	(238)
n=4	$A_7^4 = 840$	29	(24526)	50000	(60)
n=5	$A_7^5 = 2520$	29	(73577)	50000	(20)
n=6	$A_7^6 = 5040$	29	(147153)	50000	(10)
n=7	$A_7^7 = 5040$	29	(147153)	50000	(10)
Sum	$A_7^i = 13700$	400000		400000	

where $MMSA(\cdot)$ is the Masked-MSA operation defined in Eq. (12). In our work, we use the notations with $\rightarrow$ for the L2R direction, while the notations with $\leftarrow$ for the R2L direction. Motivated by the cross attention mechanism [40], we propose to interactively integrate the information from two opposite directions as

$$\begin{aligned} \overrightarrow{H}_2 &= MMSA(\overleftarrow{Q}_{\ell-1}, \overrightarrow{K}_{\ell-1}, \overrightarrow{J}_{\ell-1}), \\ \overleftarrow{H}_2 &= MMSA(\overrightarrow{Q}_{\ell-1}, \overleftarrow{K}_{\ell-1}, \overleftarrow{J}_{\ell-1}). \end{aligned} \tag{14}$$

Note that $\overrightarrow{H}_2$ and $\overleftarrow{H}_2$ contain information from both directions. The final attention result is then computed as

$$\begin{aligned} \overrightarrow{H} &= \overrightarrow{H}_1 + \lambda_1 \cdot \overrightarrow{H}_2, \\ \overleftarrow{H} &= \overleftarrow{H}_1 + \lambda_2 \cdot \overleftarrow{H}_2, \\ H &= cat[\overrightarrow{H}, \overleftarrow{H}], \end{aligned} \tag{15}$$

where λ_1 and λ_2 are two parameters learned adaptively.

For the bidirectional modeling, we can directly substitute $Head$ in Eq. (11) with H obtained in Eq. (15) while keeping all the other steps unchanged. As will be shown in the experimental stage, our bidirectional modeling strategy can greatly improve the detection accuracy.

D. Loss Function

When the chain length is known, we directly use the following cross-entropy loss for network training:

$$\mathcal{L} = \frac{1}{KN} \sum_{k=1}^{K} \sum_{i=1}^{N} crossentropy(\hat{T}_{k,i}, T_{k,i}), \tag{16}$$

where K is the number of training samples, $\hat{T}_{k,i}$ is the i-th operation of the estimated chain $\hat{T}_k$, and $T_{k,i}$ denotes the i-th ground-truth operation of the target chain T_k. During the training stage, we pad all chains with the padding symbol O_0 such that they have a fixed length N.

When the chain length is unknown, there are two basic strategies for training our TransDetect in consideration of the imbalance among different chain categories and chain lengths.

1) Balanced Category Strategy (BCS): The first strategy is to assign the same number of samples to each category, which is a natural setting in classification problems. However, BCS can cause serious length imbalance problems in the translation context. The resulting model will favor chains of longer lengths since the number of corresponding categories, $K = A_M^N$, grows exponentially as the chain length (N) increases. As shown in Table II, under the assumption that there are 29 samples in each category, BCS assigns 147153 samples for chains of lengths 7 and 6 and only 204 samples for chains of length 1. We empirically find that a model trained using BCS prefers to decode a long chain, even when the ground-truth chain is very short.

2) Balanced Length Strategy (BLS): The other strategy is to ensure that each chain length is associated with the same number of samples. In this case, the numbers of samples for different categories will be severely imbalanced. As shown in Table II, under the assumption that there are 400k samples and that there are the same number of samples of each chain length, each chain of length 1 has 7k corresponding samples, while there are only 10 samples for each chain of length 7. As a result, the model will tend to be overfitted for short chains and thus work poorly for long chains.

3) Weighted Cross-Entropy Loss (WCEL): To address the tradeoff between category imbalance and length imbalance, we propose a weighted cross-entropy loss function (WCEL) for network training, which is given by

$$\mathcal{L} = \frac{1}{KN}\sum_{k=1}^{K}\gamma(N_k)\sum_{i=1}^{N} crossentropy(\hat{T}_{k,i}, T_{k,i}). \quad (17)$$

Here, $\gamma(N_k)$ is a weighting function with respect to the chain length N_k. In our implementation, we assign samples according to BCS and use $\gamma(N_k)$ to place more emphasis on chains of shorter lengths, thereby eliminating the imbalance problem caused by BCS. Due to the exponentially increasing number of chain categories with respect to the chain length, we define $\gamma(N_k)$ as

$$\gamma(n) = \frac{e^{-p_n}}{\sum_{i=0}^{N} e^{-p_i}}, n = 0, ..., N, \quad (18)$$

where

$$p_n = \alpha \times \frac{A_M^n}{\sum_{i=0}^{N} A_M^i}, \alpha \geq 0. \quad (19)$$

Here, N is the maximum length of a chain, and M is the number of possible operations. We can see that $\gamma(n)$ has a negative relationship with the chain length n, causing the model to pay much more attention to shorter chains. α is a hyperparameter for controlling the slope between $\gamma(n)$ and the chain length; the corresponding curves are illustrated in Fig. 5. How to choose a proper α will be discussed in Section IV.

It should be noted that benefiting from the sequential modeling of the operation chain, the correctly detected subchains can also decrease the losses defined in (16) and (17), thus explicitly guiding network training. For the bidirectional model, Eq. (17) calculate the loss in two directions.

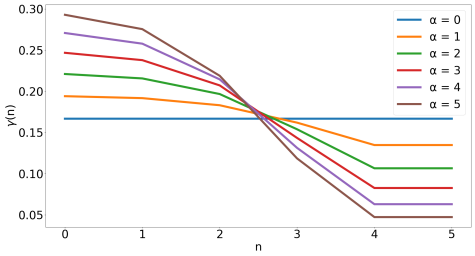


Fig. 5. Curves of $\gamma(n)$ versus the chain length n with different α settings where $M = 5$.

IV. Experimental Results

In this section, we present and analyze a set of experiments conducted to evaluate the effectiveness of our TransDetect method. We implemented our method using the PyTorch deep learning framework. Throughout our experiments, we used the following implementation settings: we set the number of filters in the ResBlock to 64, the number of layers in the Transformer encoder and decoder to 6, and the number of heads in the MSA module to $U = 8$; during the training procedure, the number of epochs was 150 and the batch size was 24. The source code used in these experiments will be released at https://github.com/YuanmanLi/github-TransDetect.

A. Dataset

The RAISE-1k dataset [41] [3] was employed in our experiments for performance evaluation. This dataset contains 900 high-resolution images (approximately 3200×4800) and plentiful categories (outdoors, indoors, landscape, nature, people, objects and buildings). The images were captured by different cameras (Nikon D90, Nikon D7000 and Nikon D40). A total of 800 images in the 'train' folder were adopted for training, and the remaining 100 images were adopted for testing. We converted them to grayscale before further processing. For the training set, each image was cropped into subimages of 512×512 with a stride of 256. We finally obtained approximately 26,700 subimages, which were further augmented with rotation or flipping during training. For the test set, we adopt a similar approach to generate approximately 3,000 subimages.

In each training epoch, a varying number of operations were first uniformly selected from Table I and then consecutively applied to a given subimage. We should emphasize that different from the existing Transformer-based image processing approaches, such as ViT [32], the cropped images are fed into TransDetect without further dividing into small patches.

B. Compared Methods and Metrics

We compared our method with two recently published works addressing the operation chain detection problem, i.e., MISLnet [5] and a two-stream CNN [4]. Note that in their original papers, these methods considered only chains consisting of up to 2 operations. In our work, we adapted their

[3] http://loki.disi.unitn.it/RAISE/

TABLE III
CONFUSION MATRICES FOR IDENTIFYING OPERATION CHAINS USING MISLNET [5], A TWO-STREAM CNN [4] AND OUR PROPOSED TRANSDETECT ALGORITHM, WHERE RED INDICATES THE BEST ACCURACY, BLUE INDICATES THE SECOND BEST ACCURACY AND GREEN INDICATES THE THIRD BEST.

MISLnet [5]: ACC = 94.40%										
	OR	MF	GB	RS	MF→GB	GB→MF	MF→RS	RS→MF	GB→RS	RS→GB
OR	99.46%	0.20%	0.00%	0.34%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MF	0.00%	88.46%	0.00%	0.00%	0.00%	0.49%	1.81%	8.96%	0.28%	0.00%
GB	0.00%	0.00%	87.07%	0.00%	0.65%	0.00%	0.00%	0.00%	0.07%	12.20%
RS	0.40%	0.00%	0.07%	99.12%	0.00%	0.00%	0.27%	0.13%	0.00%	0.00%
MF→GB	0.00%	0.00%	0.00%	0.00%	95.97%	1.43%	0.00%	0.00%	1.71%	0.89%
GB→MF	0.00%	0.00%	0.00%	0.00%	0.78%	93.67%	0.92%	3.56%	1.07%	0.00%
MF→RS	0.00%	0.00%	0.00%	0.14%	0.00%	0.07%	99.79%	0.00%	0.00%	0.00%
RS→MF	0.00%	1.32%	0.00%	0.00%	0.00%	10.46%	1.74%	85.22%	1.26%	0.00%
GB→RS	0.00%	0.00%	0.00%	0.00%	0.28%	0.14%	0.28%	2.20%	96.88%	0.21%
RS→GB	0.00%	0.00%	1.71%	0.00%	0.14%	0.00%	0.00%	0.07%	0.41%	97.67%
Two-Stream CNN [4]: ACC = 97.42%										
	OR	MF	GB	RS	MF→GB	GB→MF	MF→RS	RS→MF	GB→RS	RS→GB
OR	99.93%	0.07%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MF	0.00%	95.20%	0.00%	0.00%	0.00%	0.07%	0.42%	4.24%	0.07%	0.00%
GB	0.00%	0.00%	95.28%	0.00%	0.94%	0.00%	0.00%	0.00%	0.00%	3.78%
RS	0.00%	0.00%	0.00%	99.39%	0.00%	0.00%	0.40%	0.20%	0.00%	0.00%
MF→GB	0.00%	0.00%	0.07%	0.00%	98.77%	1.02%	0.00%	0.00%	0.14%	0.00%
GB→MF	0.00%	0.00%	0.00%	0.00%	0.64%	95.95%	0.07%	3.34%	0.00%	0.00%
MF→RS	0.00%	0.00%	0.00%	0.21%	0.00%	0.00%	99.72%	0.07%	0.00%	0.00%
RS→MF	0.00%	0.56%	0.00%	0.00%	0.35%	2.79%	0.98%	94.77%	0.56%	0.00%
GB-RS	0.00%	0.00%	0.00%	0.00%	0.07%	0.35%	0.14%	0.71%	98.72%	0.00%
RS-GB	0.00%	0.00%	2.60%	0.00%	0.62%	0.00%	0.07%	0.00%	0.55%	96.17%
TransDetect: ACC = 98.88%										
	OR	MF	GB	RS	MF→GB	GB→MF	MF→RS	RS→MF	GB→RS	RS→GB
OR	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MF	0.00%	98.76%	0.00%	0.00%	0.00%	0.83%	0.00%	0.41%	0.00%	0.00%
GB	0.00%	0.00%	97.83%	0.00%	0.00%	0.43%	0.00%	0.00%	0.00%	1.74%
RS	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MF→GB	0.00%	0.00%	0.85%	0.00%	97.03%	1.69%	0.42%	0.00%	0.00%	0.00%
GB→MF	0.00%	0.46%	0.00%	0.00%	0.00%	98.15%	0.00%	1.39%	0.00%	0.00%
MF→RS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
RS→MF	0.00%	0.00%	0.00%	0.00%	0.00%	0.80%	0.40%	98.80%	0.00%	0.00%
GB→RS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.85%	99.15%	0.00%
RS→GB	0.00%	0.00%	1.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	98.79%

codes for long chain detection when necessary. It should also be noted that both of these algorithms model chain detection as a classification problem, directly assigning each chain a unique label.

We adopted three metrics to evaluate the performance of different algorithms. The first metric, i.e., the accuracy (ACC), is computed as

$$ACC = \frac{\text{number of correctly predicted chains}}{\text{number of test chains}}. \quad (20)$$

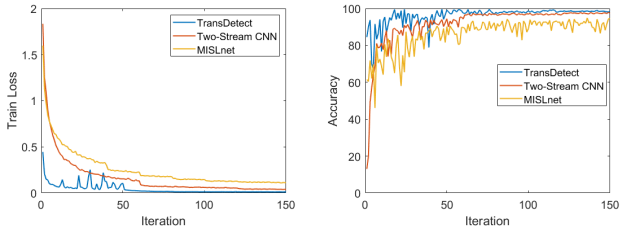
It should be noted that a predicted chain is correct only when all the operations and their orders are exactly the same with the ground-truth.

Practically, detecting some long subchains of the complete chain is also very important for forensics. However, the ACC metric given in (20) can only measure the effectiveness of an algorithm in correctly detecting the whole chain, which fails to tell the differences of the results when the chains are only partially detected. As a result, the ACC metric is insufficient to fully reflect the capability of different algorithms for the long chain detection. Due to the truth that both operation chains and language sentences can be treated as sequential signals, they share some similar characteristics, e.g., both the operations (words) and their orders are important. Motivated by this fact, we adopt another two metrics for sequential data to evaluate how well the method can detect subchains, i.e., the accuracy of the longest matched subchain (ALMS) and the Bilingual Evaluation Understudy (BLEU) score [42]. Mathematically, ALMS is defined as

$$ALMS = Average\Big(\frac{len(\text{longest matched subchain})}{len(\text{target chain})}\Big), \quad (21)$$

where $len(\cdot)$ calculates the chain length, and $Average(\cdot)$ computes the average over all test chains. For instance, given a target operation chain $O_4 \rightarrow O_3 \rightarrow O_6 \rightarrow O_5 \rightarrow O_7$, the predicted result $O_4 \rightarrow O_3 \rightarrow O_6 \rightarrow O_7$ is assigned a score of 0.6 in the ALMS computation but makes no contribution to the ACC metric.

Different from ALMS, BLEU [42] measures the performance of matched subchains of various lengths (not just the



(a) (b)

Fig. 6. Training curves of MISLnet [5], Two-Stream CNN [4] and our proposed method: (a) training losses vs. iterations and (b) ACC vs. iterations on the test dataset.

longest one). Formally, BLEU is defined as

$$BLEU = \begin{cases} exp\Big(\sum_{c=1}^{C} w_c log P_c\Big), & L_p > \quad L_g, \\ e^{1-\frac{L_g}{L_p}} \times exp\Big(\sum_{c=1}^{C} w_c log P_c\Big), & L_p \leq \quad L_g. \end{cases} \tag{22}$$

Here, L_p and L_g are the lengths of the predicted chain and the ground-truth chain, respectively; P_c denotes the c-gram matching score, reflecting the accuracy of predicted subchains of length c; and w_n and C are usually set to $\frac{1}{C}$ and 4, respectively. The BLEU metric measures the performance of matching subchains of various lengths to the target chain, and its value range is $[0, 1]$. Please refer to [42] for more details about BLEU.

C. Detecting Chains of Two Operations

For consistency with previous algorithms [4], [5], we first conducted an experiment on detecting operation chains of up to two operations. To this end, three types of operations, i.e., resampling, Gaussian blurring and median filtering, as adopted in [5], were considered in our experiment; Table I lists the associated parameters. This setting yields $A_3^0 + A_3^1 + A_3^2 = 10$ different possible operation chains in total.

During training, each image patch was processed in accordance with a uniformly selected chain. The training behaviors of the different approaches are plotted in Fig. 6. Note that the methods of [4], [5] simply assign each chain a unique label; thus, the relationships between chains are obscured after labeling. In contrast, TransDetect is based on sequence modeling, and the relationship between a complete chain and its subchains is clear. Successfully detected subchains can also decrease the loss, as shown in (17), thus effectively guiding the network training process. As seen from Fig. 6(a), our proposed TransDetect algorithm converges much faster than MISLnet [5] or Two-Stream CNN [4]. Fig. 6(b) presents the curves of ACC with respect to the number of training iterations, from which it can also be observed that TransDetect performs the best.

Table III reports the detection results in the form of confusion matrices. Overall, MISLnet achieves 94.40% ACC, while Two-Stream CNN and our method obtain ACC values of 97.42% and 98.88%, respectively. Compared with MISLnet, TransDetect achieves higher ACCs for all 10 cases. Specifically, TransDetect outperforms MISLnet by over 10%, 10% and 13% for the chains MF, GB and RS→MF, respectively, in terms of ACC. Furthermore, compared with Two-Stream CNN, TransDetect also performs better in all cases except the chain MF→GB. As will be shown later, the performance gaps among the methods could be more significant when addressing long chains.

TABLE IV
RESULTS OF TRANSDETECT FOR KNOWN-LENGTH OPERATION CHAIN DETECTION.

Model	A_5^1	A_5^2	A_5^3	A_5^4	A_5^5
ACC	100.00%	99.51%	97.01%	91.08%	87.81%
ALMS	100.00%	99.76%	98.60%	95.99%	94.11%
BLEU	1.0000	0.9979	0.9904	0.9645	0.9468

D. Detecting Chains of Known Length

We next evaluated the performance of TransDetect in detecting chains of a fixed, known length. In this experiment, we considered lengths from one to five, for which the sizes of the solution space are A_5^1, A_5^2, A_5^3, A_5^4 and A_5^5, respectively. In the training stage, for each length, we manipulated the images in accordance with a chain of operations randomly chosen from the list $\{O_1, ..., O_5\}$ in Table I. Note that these five operation types and associated parameters are consistent with [5].

Table IV summarizes the detection results. TransDetect achieves over 90% ACC when the length of the target chain $L_g \leq 4$, which is quite promising. It can also be seen that the ACC drops gracefully, even though the size of the solution space increases exponentially as L_g increases. Different from previous classification-based algorithms, TransDetect decodes the chain in a progressive manner and thus can fully utilize the information of previously decoded operations to facilitate the detection of subsequent operations. This desirable property endows TransDetect with the ability to reveal many long subchains, even when it fails to detect the complete chain. The quantitative results shown in Table IV also conform to the above conclusion. Moreover, the performance in terms of the ALMS and BLEU metrics degrades much less than the ACC performance. For instance, TransDetect still achieves 94.11% ALMS and a BLEU score of 0.9468 when $L_g = 5$.

E. Detecting Chains of Unknown Length

In this subsection, we verify the effectiveness of our method in detecting operation chains of unknown length. Specifically, we uniformly selected L_g from 0 to 5, and operations from $\{O_1, ..., O_5\}$ list in Table I, resulting in a total of $\sum_{i=0}^{5} A_5^i = 326$ possible chains.

As discussed in Section III-D, when the chain length is unknown, there are two basic strategies for training the framework using the standard cross-entropy loss, i.e., BCS and BLS. In terms of the training and testing settings, there are totally 6 cases, as follows:

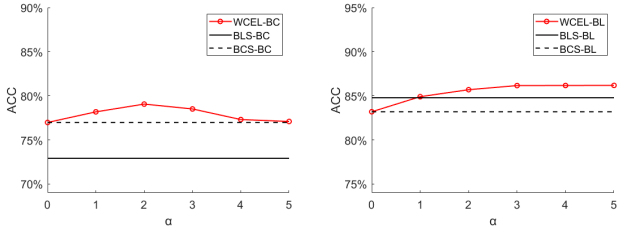
TABLE V
RESULTS FOR UNKNOWN-LENGTH OPERATION CHAIN DETECTION. MODELS TRAINED USING WCEL ($\alpha = 2$).

Category-balanced (BC) testing: ACC = 79.06%, ALMS = 94.23%, BLEU = 0.9223						
L_g \ L_p	0	1	2	3	4	5
0	**97.14%** (0.00%)	2.86%	0.00%	0.00%	0.00%	0.00%
1	0.00%	**90.52%** (0.00%)	9.25%	0.23%	0.00%	0.00%
2	0.00%	0.66%	**87.02%** (0.00%)	11.44%	0.77%	0.11%
3	0.00%	0.00%	2.45%	**84.75%** (1.59%)	10.26%	0.96%
4	0.00%	0.00%	0.04%	8.36%	**74.61%** (4.37%)	12.62%
5	0.00%	0.00%	0.00%	0.45%	11.35%	**78.62%** (9.59%)
Length-balanced (BL) testing: ACC = 85.69%, ALMS = 96.48%, BLEU = 0.9396						
L_g \ L_p	0	1	2	3	4	5
0	**96.59%** (0.00%)	3.29%	0.06%	0.06%	0.00%	0.00%
1	0.27%	**91.53%** (0.00%)	8.10%	0.10%	0.00%	0.00%
2	0.02%	0.58%	**87.14%** (0.19%)	11.23%	0.76%	0.08%
3	0.00%	0.04%	2.32%	**84.67%** (1.83%)	9.79%	1.35%
4	0.00%	0.00%	0.17%	9.09%	**74.03%** (4.59%)	12.12%
5	0.00%	0.00%	0.00%	0.38%	10.47%	**80.03%** (9.12%)

- **BCS-BC**: The model is trained using BCS, and the test set is also category balanced, i.e., the number of samples in each chain category is the same.
- **BLS-BC**: The model is trained using BLS, while the test set is category balanced.
- **WCEL-BC**: The model is trained using our proposed WCEL, while the test set is category balanced.
- **BLS-BL**: The model is trained using BLS, and the test set is also length balanced, i.e., the number of samples in each chain length is the same.
- **BCS-BL**: The model is trained using BCS, while the test set is length balanced.
- **WCEL-BL**: The model is trained using our proposed WCEL, while the test set is length balanced.

Note that the total numbers of training samples and test samples are the same for all the above cases. The results of each of the above cases are plotted in Fig. 7, in which Fig. 7(a) and Fig. 7(b) represent category-balanced (BC) testing and length-balanced (BL) testing, respectively, with the different training strategies. It can be observed that for BCS and BLS, the performance drops when the data distribution of the test set is different from that of the training set. For example, the ACC in the BLS-BL case is 84.87%, while the ACC in the BLS-BC case is only 72.98%, which is much lower than that in the BCS-BC case (76.92%). This phenomenon implies that a model trained using BLS works poorly for detecting long chains. Choosing an appropriate α for our proposed WCEL is crucial to address the tradeoff between category and length imbalance. From Fig. 7(a), we can observe that when $\alpha = 2$, WCEL outperforms BCS and BLS in both BC testing (ACC=79.06%) and BL testing (ACC=85.69%). Hereafter, we set $\alpha = 2$ in our experiments.

Table V reports the detailed quantitative results of our method trained using WCEL. We should emphasize that we are the first to detect operation chains containing up to five operations. Our method achieves an overall ACC of 79.06%, an ALMS of 94.23%, and a BLEU score of 0.9223 for BC testing and an ACC of 85.69%, an ALMS of 96.48%, and a BLEU score of 0.9396 for BL testing. These results are quite promising considering the challenging nature of the problem. To fully demonstrate the effectiveness of our method for different chain lengths, we list the ACCs for each length in Table V. In addition, we show the error rate that an operation chain is wrongly detected as a chain of each other length. L_g and L_p in Table V denote the lengths of the target and detected chains, respectively. Note that the number shown in parentheses represents the error rate that the detected operation chain has only the same length as the target chain but different operation orders or types. For instance, in the case of category-balanced testing and $L_g = 3$, our method correctly detects 84.75% of the operation chains, and 2.45%,10.26% and 0.96%



(a) (b)

Fig. 7. Models trained using the standard cross-entry loss function with BLS and BCS and using WCEL with varying α values (defined in (19)), tested on (a) BC data and (b) BL data.

TABLE VI
COMPARISON OF THE THREE METHODS FOR UNKNOWN-LENGTH OPERATION CHAIN DETECTION.

Model	MISLnet	Two-Stream CNN	TransDetect
ACC	47.94% (53.80%)	54.17% (61.88%)	**79.06% (85.69%)**
ALMS	81.68% (84.86%)	84.75% (87.80%)	**94.23% (96.48%)**
BLEU	0.7917 (0.7591)	0.8216 (0.8108)	**0.9223 (0.9396)**

TABLE VII
CROSS-DATASET VALIDATION OF TRANSDETECT. EACH ROW REPORTS THE TEST RESULTS (ACC) OF TRANSDETECT TRAINED ON THE SPECIFIED DATASET USING WCEL ($\alpha = 2$) AND TESTED ON THE DIFFERENT DATASETS INDICATED IN THE COLUMN HEADERS.

	RAISE	FODB	Dresden
RAISE	**79.06% (85.69%)**	71.86% (79.34%)	72.73% (80.48%)
FODB	77.48% (83.60%)	**75.91% (83.19%)**	75.36% (82.75%)
Dresden	76.72% (82.75%)	72.67% (79.85%)	**75.75% (83.43%)**

of the chains are incorrectly detected as chains of other lengths of 2, 4 and 5, respectively. Moreover, there are 1.59% of the detected chains though are of the same length, they have different operation orders or types. We can observe that due to the interplay between operations in a long chain, TransDetect is prone to decoding one more or one fewer operation for chains with $L_g \geq 3$. It is also shown that the ACC results for $L_g = 5$ are better than those for $L_g = 4$. This interesting phenomenon is caused by the fact that when $L_g = 4$, the chain may be incorrectly detected with either one more ($L_p = 5$) or one fewer ($L_p = 3$) operation, while in the case of $L_g = 5$, the chain cannot be incorrectly detected with one more operation.

Table VI summarizes the quantitative results obtained by MISLnet, Two-Stream CNN and TransDetect in both BC testing and BL testing (shown in parentheses). We observe that our TransDetect method outperforms MISLnet and Two-Stream CNN by large margins. For example, TransDetect surpasses MISLnet and Two-Stream CNN by relative ACC improvements of 64.9% and 45.9% in BC testing, respectively. Compared with the results listed in Table III, these findings further demonstrate the superiority of modeling operation chain detection as a translation problem, especially for detecting long chains.

F. Cross-Dataset Validation

In this experiment, we evaluated the robustness of our TransDetect algorithm in detecting manipulation chains across datasets. This is important in real applications since images may be acquired using cameras of different models. Therefore, we trained and tested TransDetect using two additional image datasets, i.e., FODB [43] and Dresden [44]. Specifically, FODB contains 143 scenes collected by 27 smartphone cameras, and the Dresden image dataset was collected by 73 digital cameras. We adopted 3861 original images from FODB and 7876 authentic images from the Dresden image dataset to perform cross-dataset validation. The images were preprocessed in the same way described in Section IV-A, and 90% of the images were used for training, while the remaining 10% were used for testing. Table VII presents the cross-dataset validation results, where each row reports the test results (ACC for BC (BL) testing) of TransDetect trained on the specified dataset using WCEL and tested on a different dataset. For example, the first row shows that TransDetect trained on the RAISE dataset achieves ACCs of 79.06%, 71.86% and 72.73% on the RAISE, FODB and Dresden datasets, respectively. The diagonal ACC values correspond to the detection results when TransDetect is trained and tested on the same dataset. Results shown in Table VII demonstrate the favorable generalizability of our TransDetect algorithm.

TABLE X
OPERATIONS WITH UNKNOWN PARAMETERS.

Index	Operation	Parameters
O_1	GB	$\sigma = 0.7, 0.8, 0.9, 1.0$
O_2	MF	kernel size = 5, 3
O_3	RS	Scaling = 1.5, 1.6, 1.7, 1.8
O_4	AWGN	$\sigma = 1.4, 1.6, 1.8, 2$
O_5	JPEG	QF = 70 ,75, 80, 85, 90

G. Ablation Study

In this subsection, we report ablation studies of our proposed TransDetect and analyze how each component (hierarchical features, chain inversion and WCEL) contributes to the detection performance. The results of different variants are shown in Table VIII. We can see that all strategies are useful in improving the detection performance. First, the model with hierarchical features achieves better performance than the baseline using only high-level features. Due to the interactions between different operations, the traces of some operations could be very weak, while others may be very strong. Hierarchical features can provide richer feature representations and thus can more effectively reveal both weak and strong traces. Second, the chain inversion strategy boosts the ACC by 2.8%. This demonstrates that decoding the most recent operations with more distinct traces first can better assist in the decoding of previous operations. Furthermore, we can see that WCEL further noticeably enhances the detection performance by addressing the tradeoff between chain length imbalance and chain category imbalance. Finally, it can be observed that our bidirectional modeling strategy significantly improves all the metrics. This demonstrates that effectively integrating information from both directions is helpful to alleviate the error and improve the detection performance.

H. Performance for More Challenging Chain Detection

To provide a full evaluation of our proposed method, in this section, we study three more challenging cases.

Detecting chains of unknown length and with unknown parameters: In this experiment, we assessed the performance of TransDetect in a more general scenario in which the operation parameters are not fixed. Table X lists the possible parameters of the five considered operations. For each image, we manipulated it by applying a chain of a given length, where the parameters of each operation were randomly selected in accordance with Table X. Compared with the problem investigated in Section IV-E, this case is much more challenging. For example, for a chain length of 5, there are $A_5^5 = 120$ possible combinations of operations when the parameters are known, while there are $A_5^5 \times 4 \times 2 \times 4 \times 5 \times 4 = 76800$ possible combinations of operations with the different possible parameters listed in Table X.

Table IX gives the detection results for chains of unknown length and with unknown parameters. Our method achieves

TABLE VIII
RESULTS OF DIFFERENT VARIANTS OF TRANSDETECT.

Hierarchy	Inversion	WCEL	Bidirection	ACC	ALMS	BLEU
				72.08%	92.25%	0.8943
✓				73.15%	92.41%	0.8980
✓	✓			75.98%	93.92%	0.9097
✓	✓	✓		77.70%	94.11%	0.9170
✓	✓	✓	✓	**79.06%**	**94.23%**	**0.9223**

TABLE IX
DETECTION PERFORMANCE FOR CHAINS OF UNKNOWN LENGTH WITH UNKNOWN PARAMETERS.

BC testing: ACC = 70.15%, ALMS = 90.39%, BLEU = 0.8870 (BL testing: ACC = 80.49%, ALMS = 94.47%, BLEU = 0.9179)						
L_c \ L_p	0	1	2	3	4	5
0	**94.29%** (0.00%)	4.76%	0.95%	0.00%	0.00%	0.00%
1	0.45%	**88.94%** (0.00%)	10.16%	0.45%	0.00%	0.00%
2	0.00%	0.77%	**86.47%** (0.17%)	11.00%	1.21%	0.39%
3	0.00%	0.04%	4.57%	**79.00%** (2.77%)	11.79%	1.83%
4	0.00%	0.00%	0.35%	12.38%	**68.11%** (6.54%)	12.61%
5	0.00%	0.00%	0.02%	1.27%	18.62%	**64.05%** (16.04%)

TABLE XI
RESULTS FOR UNKNOWN-LENGTH (A_7^z) OPERATION CHAIN DETECTION. MODEL TRAINED USING WCEL.

BC testing: ACC = 54.17%, ALMS = 83.17%, BLEU = 0.8149 (BL testing: ACC = 67.36%, ALMS = 89.52%, BLEU = 0.8589)								
L_c \ L_p	0	1	2	3	4	5	6	7
0	**91.94%** (0.00%)	6.45%	1.61%	0.00%	0.00%	0.00%	0.00%	0.00%
1	2.27%	**85.23%** (0.00%)	9.09%	3.41%	0.00%	0.00%	0.00%	0.00%
2	0.00%	2.25%	**71.91%** (1.12%)	20.22%	2.25%	2.25%	0.00%	0.00%
3	0.00%	0.00%	1.34%	**66.22%** (3.36%)	22.15%	4.92%	1.57%	0.45%
4	0.00%	0.00%	0.06%	4.34%	**61.82%** (7.99%)	20.38%	4.51%	0.91%
5	0.00%	0.00%	0.00%	0.13%	8.39%	**57.58%** (14.31%)	16.59%	3.00%
6	0.00%	0.00%	0.00%	0.07%	0.57%	14.03%	**51.94%** (19.46%)	13.93%
7	0.00%	0.00%	0.01%	0.01%	0.12%	1.27%	18.12%	**52.29%** (28.17%)

ACC, ALMS and BLUE scores of 70.15%, 90.39%, and 0.8870, respectively, in terms of BC testing. Compared with the results reported in Table V, the performance degradation is 8.91%, 3.84%, and 0.0353 in terms of the ACC, ALMS and BLEU metrics, respectively, for BC testing, and is only 5.20%, 2.01% and 0.0217 for BL testing. We deem such degradation acceptable considering the challenging nature of the problem. Furthermore, compared with the ACC, the ALMS performance degrades much more gradually, which implies that many long subchains can still be detected.

Detecting chains consisting of up to 7 operations: We further evaluated the effectiveness of our method for detecting chains containing at most 7 operations. To this end, we considered two additional commonly used operations, i.e., histogram equalization (HE) and unsharp masking (USM), as listed in Table I. Note that the solution space for 7 operations (13700 categories) is much larger than that for 5 operations (326 categories), and even worse, the interactions between operations can be much more complex, which makes the problem extremely challenging. In this experiment, we train TransDetect using the chain inversion strategy for the sake of computational efficiency.

Table XI summarizes the detection results. TransDetect achieves ACC, ALMS and BLEU scores of 54.17%, 83.17%, and 0.8149, respectively, in terms of BC testing, and 67.36%, 89.52% and 0.8589, respectively, in terms of BL testing. We can observe that our method attains high detection accuracy when the target chain is short while still maintaining an ACC of over 50% when the chain length is greater than 6. The results are very promising considering the exponentially increased solution space and the complex interplay among the different operations.

TABLE XII
RESULTS FOR DETECTING CHAIN OF UNKNOWN-LENGTH AND REPETITIVE OPERATIONS.

Model	MISLnet	Two-Stream CNN	TransDetect
ACC	39.15%	49.94%	**64.16%**
ALMS	74.80%	81.32%	**88.65%**
BLEU	0.7464	0.8014	**0.8623**

Detecting chains of unknown length and repetitive operations: We now evaluate the effectiveness of different algorithms when the operations could be applied multiple times. Note that such a setting makes our problem very challenging. For example, given 5 operations and the length of the longest chain as 5, under the assumption that one

TABLE XIII
RESULTS FOR ROBUST IMAGE INPAINTING DETECTION.

HP-FCN	chainA	chainB	chainC	chainD	average
	Training with data augmentation				
F1	37.26%	41.69%	40.74%	39.45%	39.79%
AUC	79.25%	83.30%	82.92%	81.90%	81.84%
	Training with processing chain estimation				
F1	**45.06%**	**47.66%**	**46.78%**	**46.67%**	**46.54%**
AUC	**85.29%**	**87.30%**	**86.64%**	**86.30%**	**86.38%**

operation is applied only once, there are $\sum_{i=0}^{5} A_5^i = 326$ possible chains. However, if the operation can be repetitively applied, there are a total of $\sum_{i=0}^{5} 5^i = 3,906$ possible chains, which is much larger than the former case. Even worse, the fingerprints left by repetitive operations could be very similar (even the same) to the fingerprints left by a single operation, thus making the detection extremely challenging.

Specifically, for each image, we manipulate it by a chain, whose operations are uniformly selected from $\{O_1, ..., O_4\}$ list in Table I with replacement. The resulting image is further operated by JPEG compression, i.e., O_5. For simplicity, the maximum length of the operation chain is set as 5. Table XII reports the results obtained by MISLnet [4], Two-Stream CNN [3] and the proposed TransDetect. Our method achieves ACC, ALMS and BLUE scores of 64.16%, 88.65% and 0.8623, respectively. Considering the challenge of this problem, we think that our results are still satisfactory. We can also observe that our performance is much better than the competing algorithms. For example, our method obtains 25.01% and 14.22% performance gains over MISLnet and Two-stream CNN in terms of the ACC metric. Note that the topic of detecting long operation chains is still in its infancy stage, improving the performance when one operation may be applied multiple times still needs more effort in the future.

V. APPLICATION TO IMAGE INPAINTING DETECTION

In this section, we further evaluate the effectiveness of TransDetect through a simple case study of robust image inpainting detection. Image inpainting is a fundamental task in the field of image processing, which can be used to fill in missing regions of incomplete images. It has also become a powerful tool for making forged images, by e.g., altering or removing image contents. During the inpainting process, some postprocessing operations are often applied to weaken the forgery traces. In our experiment, we adopt HP-FCN [45] for image inpainting detection, and we use GC dataset [23] for evaluation, which includes 48K training images and 1K testing images. For simplicity, in this case study, we consider only four kinds of postprocessing chains, i.e., chainA: GB $\rightarrow$ AWGN $\rightarrow$ JPEG, chainB: GB $\rightarrow$ USM $\rightarrow$ JPEG, chainC: GB $\rightarrow$ USM $\rightarrow$ AWGN $\rightarrow$ JPEG and chainD: AWGN $\rightarrow$ GB $\rightarrow$ USM $\rightarrow$ JPEG.

Table XIII reports the inpainting detection results in different settings. In the first experiment, we directly train the model with data augmentation by randomly applying the above four operations chains to the training images. The resulting model achieves 39.79% F1 score and 81.84% AUC score on average. In the second experiment, we first apply our TransDect to estimate the post-processing chain of the given test image, and then fine-tune the model obtained in the first experiment for one epoch using the estimated chain. We can see that the fine-tuned model with the estimated chain through our TransDect improves F1 (AUC) by 6.75% (4.54%). Though the case study is simple, the promising improvement demonstrates that effectively revealing the degradation history of the given image is very helpful in robust forgery detection. Note that there could be many other postprocessing chains in real applications, and how to automatically adjust parameters of the model using the estimated chains can be an interesting work in the future.

A. Discussions

In our experiments, we considered chains containing up to seven commonly used operations to maintain an ACC above 50% for each length, and interesting readers can further apply our method to chains of longer lengths. Note that although our method achieves much better performance than existing approaches, improving the accuracy for longer chains still needs more effort in the future study.

Similar to all the existing operation chain detection algorithms, such as [4], [5], we assume that the operations are conducted globally for a given image patch. Sometimes, an image is only manipulated locally. In this case, we could first identify the possible manipulation areas using a universal image forensic method, and then apply our algorithm to detect the image operation chains locally. Further, similar to the existing approaches, we assume that operations of the chains are from a known set. In reality, there could be some other standard and customized operations. How to minimize their influences on the detection methods is challenging, and also an interesting topic in the future.

Practically, one can directly apply our method to reconstruct the image processing history when one or several operations were applied, thus obtaining sufficient information for a reliable decision in determining the authenticity and origin of an image.

In addition, in reality, some common image operations are widely used to weaken or disguise the fingerprints of complex digital forgeries, such as copy-move, splicing and inpainting images. Such unknown operations would heavily degrade the performance of those forgery detectors [46]. In this scenario, one may first apply our framework to reveal the postprocessing history. Then, the forgery detectors with such information can make adaptive adjustments to achieve high robustness against the negative effect introduced by the postprocessing operations. How to effectively incorporate our method with other forgery detection algorithms still needs more efforts in the future.

VI. CONCLUSION

In this paper, we have proposed a new methodology for image operation chain detection. This is the first work to consider chains containing up to seven operations, a scenario that yields more than 10k possible solutions. Specifically, we

conduct operation chain detection within a machine translation framework, an approach that is completely different from the existing algorithms, which are built upon classification models. In our framework, a chain is treated as a sentence in a target language, where each operation in the chain is represented by one word. Then, the goal of image operation chain detection is to map the original image space to the target language space. To this end, the original image is first transformed into a latent source language space, sentences in which can properly describe the fingerprints left by different operation chains. Then, the chain is decoded by translating the corresponding sentence in the source language into a sentence in the target language in a progressive manner. Furthermore, we have proposed three strategies, i.e., hierarchical feature extraction, chain inversion, bidirectional modeling and a weighted cross-entropy loss, to enhance the detection performance. Extensive experiments have been conducted to demonstrate the advantages of our scheme.

REFERENCES

[1] X. Sun, X. Li, L. Zhuo, K. M. Lam, and J. Li, "A joint deep-network-based image restoration algorithm for multi-degradations," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2017, pp. 301–306.

[2] J. He, C. Dong, and Y. Qiao, "Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 53–68.

[3] Y. Li, J. Zhou, J. Tian, X. Zheng, and Y. Y. Tang, "Weighted error entropy-based information theoretic learning for robust subspace representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4228–4242, 2022.

[4] X. Liao, K. Li, X. Zhu, and K. J. R. Liu, "Robust detection of image operator chain with two-stream convolutional neural network," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 955–968, 2020.

[5] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2691–2706, 2018.

[6] T. Qiao, R. Shi, X. Luo, M. Xu, N. Zheng, and Y. Wu, "Statistical model-based detector via texture weight map: Application in re-sampling authentication," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1077–1092, 2019.

[7] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

[8] Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9543–9552.

[9] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 31–45, 2016.

[10] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensic strategy based on steganalytic model," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2014, p. 165170.

[11] P. Comesaa, "Detection and information theoretic measures for quantifying the distinguishability between multimedia operator chains," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2012, pp. 211–216.

[12] M. C. Stamm, X. Chu, and K. J. R. Liu, "Forensically determining the order of signal processing operations," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2013, pp. 162–167.

[13] X. Chu, Y. Chen, and K. J. R. Liu, "Detectability of the order of operations: An information theoretic approach," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 823–836, 2016.

[14] J. You, Y. Li, J. Zhou, Z. Hua, W. Sun, and X. Li, "A transformer based approach for image manipulation chain detection," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 3510–3517.

[15] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensic detection of resampled images," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 536–545, 2012.

[16] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain based approach," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4699–4710, 2013.

[17] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 3, pp. 515–525, 2014.

[18] X. Jiang, P. He, T. Sun, F. Xie, and S. Wang, "Detection of double compression with the same coding parameters based on quality degradation mechanism analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 170–185, 2018.

[19] C. Pasquini and R. Bhme, "Information-theoretic bounds for the forensic detection of downscaled signals," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1928–1943, 2019.

[20] P. Kakar, N. Sudha, and W. Ser, "Exposing digital image forgeries by detecting discrepancies in motion blur," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 443–452, 2011.

[21] L. Su, C. Li, Y. Lai, and J. Yang, "A fast forgery detection algorithm based on exponential-fourier moments for video region duplication," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 825–840, 2018.

[22] Y. Li, J. Zhou, and A. Cheng, "SIFT keypoint removal via directed graph construction for color images," *IEEE Trans. on Inf. Forensics and Security*, vol. 12, no. 12, pp. 2971–2985, 2017.

[23] H. Wu and J. Zhou, "IID-Net: Image inpainting detection network via neural architecture search and attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1172–1185, 2022.

[24] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y.-Q. Shi, "A serial image copy-move forgery localization scheme with source/target distinguishment," *IEEE Trans. Multimedia*, vol. 23, pp. 3506–3517, 2021.

[25] Y. Li, J. Zhou, J. Chen, J. Tian, L. Dong, and X. Li, "Robust matrix factorization via minimum weighted error entropy criterion," *IEEE Trans. Comput. Social Syst.*, pp. 1–12, 2021.

[26] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1–6.

[27] M. Boroumand and J. Fridrich, "Deep learning for detecting processing history of images," *Electronic Imaging*, vol. 2018, no. 7, pp. 1–9, 2018.

[28] Y. Chen, Z. Wang, Z. J. Wang, and X. Kang, "Automated design of neural network architectures with reinforcement learning for detection of global manipulations," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 997–1011, 2020.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neu. comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neu. Inf. Process. Syst.*, 2017, pp. 1–11.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[33] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[36] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 213–229.

[37] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 299–12 310.

[38] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2021.

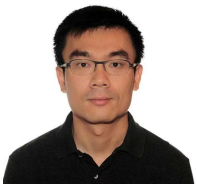
[39] L. Zhou, J. Zhang, and C. Zong, "Synchronous bidirectional neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 91–105, 2019.

[40] P. Li, J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, V. Manjunatha, and H. Liu, "Selfdoc: Self-supervised document representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5652–5660.

[41] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: A raw images dataset for digital image forensics," in *Proc. of ACM multimedia syst. conf.*, 2015, pp. 219–224.
[42] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. annual meeting Assoc. Comput. Linguist.*, 2002, pp. 311–318.
[43] B. Hadwiger and C. Riess, "The forchheim image database for camera identification in the wild," *arXiv preprint arXiv:2011.02241*, 2020.
[44] T. Gloe and R. Böhme, "The'dresden image database'for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1584–1590.
[45] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8301–8310.
[46] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1307–1322, 2019.

Yuanman Li (M'20) received the B.Eng. degree in software engineering from Chongqing University, Chongqing, China, in 2012, and the Ph.D. degree in computer science from University of Macau, Macau, 2018. From 2018 to 2019, he was a Post-doctoral Fellow with the State Key Laboratory of Internet of Things for Smart City, University of Macau. He is currently an Assistant Professor with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia security and forensics, data representation, computer vision and machine learning.

Jiaxiang You (Student Member, IEEE) received the B.Eng. degree in communication engineering from Shantou University, Shantou, China, in 2020. He is pursuing the M.S. degree in Shenzhen University, Shenzhen, China. His research interests include multimedia security and forensics, and computer vision.



Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, in 2009. He held various research positions with University of Illinois at Urbana-Champaign, Hong Kong University of Science and Technology, and McMaster University. He is an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, and also the Interim Head of the newly established Centre for Artificial Intelligence and Robotics. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence and big data. He holds four granted U.S. patents and two granted Chinese patents. He has co-authored two papers that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is serving as the Associate Editors of the IEEE TRANSACTIONS on IMAGE PROCESSING and the IEEE TRANSACTIONS on MULTIMEDIA.

Wei Wang (Member, IEEE) is currently an Associate Professor with School of Intelligent Systems Engineering, Sun Yat-sen University, China. Before this, he had been the UM Macao Research Fellow at University of Macau, Macau SAR. He received PhD degree in software engineering from Dalian University of Technology in 2018. His research interests include computational social science, data mining, internet of things, and artificial intelligence. He has authored/co-authored over 50 scientific papers in international journals and conferences, e.g., IEEE Transactions on Computational Social Systems, IEEE Internet of Things, IEEE Transactions on Industrial Informatics, IEEE Transactions on Big Data, IEEE Transactions of Emerging Topics in Computing, IEEE Transactions on Human-Machine Systems, The Web Conference, etc. He is the Leading Guest Editor of International Journal of Distributed Sensor Networks, Computer Communication, and Wireless Communication & Mobile Computing. He is the guest editor of ACM Transactions on Internet Technology, IEEE Journal of Biomedical and Health Informatics, and IEEE Sensor Journal. He is a PC member of International Conference on Smart Internet of Things 2019 and regular reviewer of IEEE Communications Magazine, Future Generation Computer Systems, IEEE Transactions on Industrial Informatics, IEEE Transactions on Big Data, and IEEE Transactions of Emerging Topics in Computing.

Xin Liao (senior Member, IEEE) received the B.E. and Ph.D. degrees in information security from Beijing University of Posts and Telecommunications in 2007 and 2012, respectively. He is currently an Associate Professor and a Doctoral Supervisor with Hunan University, China. He worked as a Post-Doctoral Fellow with the Institute of Software, Chinese Academy of Sciences, and also a Research Associate with The University of Hong Kong. From 2016 to 2017, he was a Visiting Scholar with the University of Maryland, College Park, USA. His current research interests include multimedia forensics, steganography, and watermarking. He is a member of Technical Committee (TC) on Multimedia Security and Forensics of AsiaPacific Signal and Information Processing Association, TC on Computer Forensics of Chinese Institute of Electronics, and TC on Digital Forensics and Security of China Society of Image and Graphics. He is serving as an Associate Editor for the IEEE Signal Processing Magazine.

Xia Li (Member, IEEE) received her B.S. and M.S. in electronic engineering and SIP (signal and information processing) from Xidian University in 1989 and 1992 respectively. She was later conferred a Ph.D. in Department of information engineering by the Chinese University of Hong Kong in 1997. Currently, she is a member of the Guangdong Key Laboratory of Intelligent Information Processing. Her research interests include intelligent computing and its applications, image processing and pattern recognition.