Multi-modal Deepfake Detection and Localization with FPN-Transformer

Chende Zheng, **Ruiqi Suo**, **Zhoulin Ji**, **Jingyi Deng**, **Fangbin Yi**, **Chenhao Lin**
Xi'an Jiaotong University
{zhengchende, srain22, 310449, yi446320, misscc320}@stu.xjtu.edu.cn, linchenhao@xjtu.edu.cn

1 Model Architecture

Our approach treats cross-modal data as unified temporal feature sequences. Figure 1 illustrates the proposed dual FPN-Transformer detection architecture, which comprises three key components: temporal feature embedding and projection module, FPN-Transformer backbone module, and prediction heads. First, during the feature embedding stage, we employ pre-trained self-supervised models (WavLM/CLIP) to encode input temporal data (audio/video) into feature embeddings, generating temporal feature sequences. Subsequently, these embeddings are processed through an encoder composed of lightweight Transformer blocks to obtain the feature pyramid. And finally, classification and regression heads jointly analyze temporal features to predict forgery boundaries at each time step, specifically estimating start times and offset distances of forged segments for precise localization. The following sections will elaborate on these components in detail.

1.1 Feature Embedding and Projection

Problem Definition. Traditional audio or video forgery detection and localization methods exclusively focus on single modal inputs, inherently limiting their cross-modal generalizability. To address this limitation, we propose a unified problem formulation for multimodal temporal data (audio/video). Specifically, given a temporal input sequence $X = \{x_1, \ldots, x_T\}$, our objective is to generate the corresponding output sequence $Y = \{y_1, \ldots, y_N\}$ through a mapping function $f : X \to Y$. Each element $y_n = (p_n, d_n^s, d_n^e)$ of the output Y denotes one potential forged segment, where p_n denotes the forgery probability of the n-th segment, d_n^s denotes the start time offset relative to the input sequence, and d_n^e denotes the end time offset relative to the input sequence.

Notably, for genuine (non-forged) sequences, the expected output Y should be an empty set. This formulation enables precise temporal localization of forged segments while maintaining cross-modal consistency through unified temporal feature representations.

Considering the distinct temporal characteristics across modalities (e.g., audio data exhibits dense temporal resolution with sampling rates reaching kilohertz levels, while video data presents sparse temporal structure with frame rates limited to hertz levels), we implement a unified feature embedding framework through temporal encoding. Specifically, the input sequence $X = \{x_1, \ldots, x_T\}$ is encoded by encoder e into feature representations $Z = \{z_1, \ldots, z_M\}$. For each feature representation z_i, f_L denotes the temporal sequence length corresponding to z_i and f_S denotes the temporal offset between consecutive feature representations. Consequently, the length M of feature sequence Z is given by the following equation:

$$M = \left\lfloor \frac{T - f_L}{f_S} \right\rfloor + 1 \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes integer flooring.

To facilitate batch processing, variable-length sequences are standardized through padding/truncation to a maximum length M_{max}, with masking mechanisms ensuring valid temporal context propagation.

Inspired by previous research [Ojha *et al.*, 2023], we employ pre-trained self-supervised models with frozen weights as encoder e. Specifically, WavLM-LARGE [Chen *et al.*, 2022] is adopted for audio data while CLIP:ViT-B/16 [Radford *et al.*, 2021] is adopted for video data. Compared to conventional feature calculating approaches (e.g., LFCC, MFCC, image DFT), these self-supervised models have been exposed to massive training data, enabling superior capability in capturing low-level features critical for differentiating genuine and forged content [Yang *et al.*, 2021]. The architectural design leverages this property to detect subtle discrepancies inherent in generative artifacts.

Subsequently, we employ a set of masked differential convolutional networks to implement feature projection, which facilitates positional embedding integration and effectively captures local temporal context. Specifically, for a given feature embedding $z \in Z$, the output of the masked 1D differential convolution at timestamp t_0 is formulated as:

$$\text{MDC}(t_0) = \theta \cdot \left(-z(t_0) \cdot \sum_{t_n \in D} w(t_n) \right) \tag{2}$$

$$+ \sum_{t_n \in D} w(t_n) \cdot z(t_0 + t_n) \tag{3}$$

where t_0 denotes the current timestamp, t_n denotes the enumerated timestamps in offset set D, $w(t_n)$ denotes the learnable convolutional weights, and $\theta \in [0, 1]$ is a hyperparameter balancing the contribution between intensity-level and gradient-level information

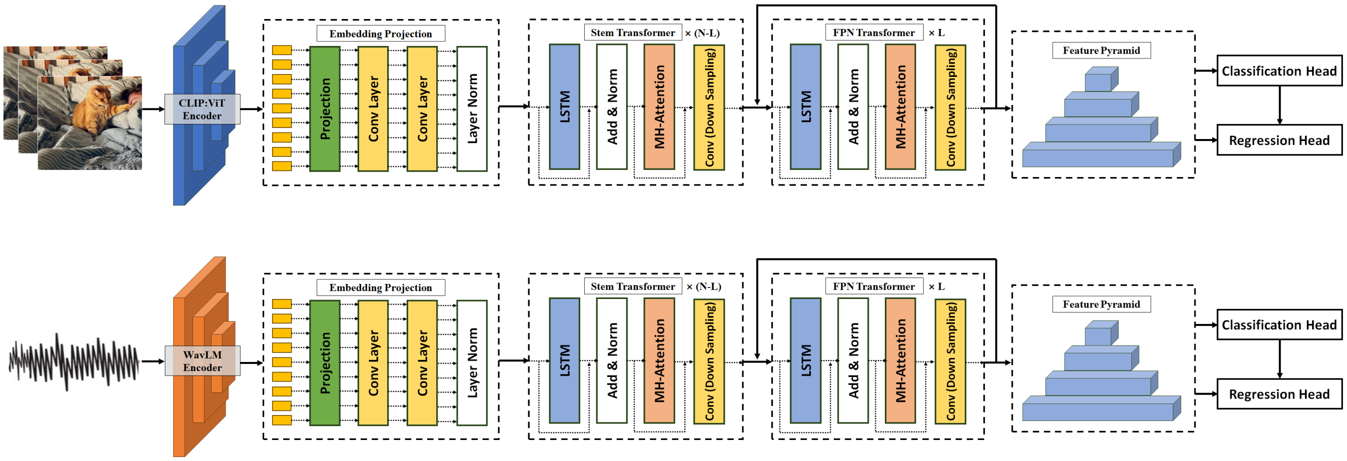

Figure 1: Framework of proposed dual FPN-Transformer detection method. First, we employ pre-trained self-supervised models (WavLM/CLIP:ViT) to encode input temporal data into feature embeddings. Then, these embeddings are processed through an encoder composed of Transformer blocks to obtain the feature pyramid. And finally, prediction heads jointly analyze temporal features to predict forgery boundaries. We train two models separately for audios and videos, and combine the output results.

1.2 FPN-Transformer Architecture

We employ N layers of R-TLM blocks [Sun *et al.*, 2021] to perform deep feature encoding. Compared to standard Transformer architectures, R-TLM incorporates additional LSTM and Fusion layers to explicitly model cross-context representation interactions. The multi-head self-attention (MSA) layer in R-TLM integrates temporal context across the sequence. Notably, we apply localized attention masking to constrain computations within sliding windows, motivated by two factors: (1) forged segments exhibit localized temporal characteristics, and (2) this design significantly reduces computational complexity.

To capture hierarchical temporal features at multiple scales for constructing a feature pyramid, we integrate R-TLM with strided 1D depthwise convolutions. Specifically, we introduce a strided depthwise 1D convolution after each MSA layer. By aggregating outputs from multi-level R-TLM structures, we obtain a hierarchical feature pyramid $\mathcal{F} = \{F^{(1)}, ..., F^{(L)}\}$ with L levels.

A critical component involves temporal alignment between feature sequence timestamps τ and original input timestamps t. Given a virtual timestamp τ_i at the i-th pyramid level and its cumulative stride factor s_i, we map τ_i to the corresponding physical timestamp t in the raw input domain through:

$$t = \left\lfloor \frac{s_i}{2} \right\rfloor + \tau_i \cdot s_i \qquad (4)$$

1.3 Prediction Head

We employ a dual-branch prediction head to decode the feature pyramid $\mathcal{F}$ into the desired output Y. The decoder consists of:

Classification Head. Given feature pyramid $\mathcal{F}$, the classification head evaluates all L pyramid levels at each timestamp t to predict the forgery probability $p(t)$. This is implemented through lightweight 1D convolutional networks attached to each pyramid level, with parameters shared across levels. Specifically, the classification network comprises 3 convolutional layers (kernel size=3), layer normalization (applied to first two layers), and ReLU activation. A final sigmoid function is applied to output dimensions to produce probabilistic forgery predictions.

Regression Head. Distinct from the classification head, the regression head predicts temporal boundaries only when timestamp t lies within forged segments (During inference, we utilize the classification head's output to determine whether a timestamp lies within forged segments). For each pyramid level, we predefine an output regression range to model the start offset d_t^s and end offset d_t^e. The regression head employs 1D convolutional networks with ReLU activation to ensure precise distance estimation. Specifically, the most probable forgery span $[s_t, e_t]$ corresponding to timestamp t is determined by:

$$c_t = \operatorname{argmax} p(c_t), \quad s_t = t - d_t^s, \quad e_t = t - d_t^e \qquad (5)$$

2 Implement Details

2.1 Loss Function

Our prediction task involves dual objectives: (1) binary classification of forgery probability at each timestamp t, and (2) temporal boundary regression for forged segments (start/end offsets). We design a composite loss function combining two components:

Classification Loss. We employ focal loss [Lin *et al.*, 2017] to address class imbalance between forged and genuine segments. For timestamp t, the classification loss ($\mathcal{L}_{\text{cls}}$) is formulated as:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{T^+}\sum_{t=1}^{T} 1(t \in \Omega^+) \cdot [\log(p_t) + \gamma \cdot (1 - p_t)^{\alpha}] \qquad (6)$$

where p_t denotes predicted forgery probability, Ω^+ represents forged regions, $T^+ = |\Omega^+|$ is the total number of positive samples, and α and γ are hyperparameters balancing hard mining effects.

Regression Loss. For timestamps $t \in \Omega^+$, we minimize DIoU loss ($\mathcal{L}_{\text{reg}}$) [Zheng *et al.*, 2020] between predicted boundaries $\hat{s}_t, \hat{e}_t$ and ground-truth s_t^*, e_t^*:

$$\mathcal{L}_{\text{reg}} = \frac{1}{T^+} \sum_{t \in \Omega^+} \left(1 - \text{DIoU}(\hat{s}_t, \hat{e}_t; s_t^*, e_t^*)\right) \tag{7}$$

Final Loss. The overall objective combines both components:

$$\mathcal{L}_{\text{total}} = \frac{1}{T^+} \sum_{t=1}^{T} \left[\lambda \mathcal{L}_{\text{cls}} + \mathcal{I}(t \in \Omega^+) \mathcal{L}_{\text{reg}}(t)\right] \tag{8}$$

where $\lambda \in [0, 1]$ is the balancing ratio between classification and localization tasks, and $\mathcal{I}(t \in \Omega^+)$ is an indicator function (1 if timestamp t lies in forged regions Ω^+, 0 otherwise).

2.2 Preprocessing

Audio data are resampled to 16 kHz and processed using WavLM-LARGE to extract 1024-dimensional feature vectors at 20 ms intervals (50 FPS). Video data undergoes frame extraction at 25 FPS, followed by resizing to 224×224 pixels and normalization. Per-frame feature extraction employs CLIP:ViT-B/16, generating 768-dimensional embeddings. All implementations utilize PyTorch on NVIDIA L40 GPUs.

2.3 Training

We adopt the AdamW optimizer with mini-batch processing, incorporating a 5-epoch warmup phase and cosine decay for learning rate scheduling. The initial learning rate is 1×10^{-3} and the weight decay is 1×10^{-2}. Variable-length sequences are standardized through padding/truncation to a maximum length of 1024, with masking mechanisms ensuring valid temporal context propagation. The number of R-TLM blocks N is set to 6, and the number of FPN levels L is set to 5. θ is set to 0.6 and the balancing ratio λ of the loss is set to 0.01. The model is trained for fixed epochs (up to 95) with a batch size of 64, and separate models are trained for audio and video modalities.

2.4 Inference

During inference, the complete sequence is input to the model with a batch size of 1. Non-Maximum Suppression (NMS) [Neubeck and Van Gool, 2006] is applied to refine predictions by eliminating highly overlapping and inefficient instances, yielding the final forged segment outputs. For unimodal temporal data, the maximum forgery confidence among all predicted segments is treated as the overall confidence score for the entire sequence. The final outputs are obtained by containing both audio and video modalities, where the higher confidence score between the two modalities is selected as the final sample-level forgery confidence, while the union of predicted audio and video forged segments forms the complete output set of forged regions.

3 How to Reproduce and Run the Code

Please refer to README.md in our submitted materials, which includes the details about how to run our codes.

References

[Chen *et al.*, 2022] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[Neubeck and Van Gool, 2006] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[Ojha *et al.*, 2023] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[Sun *et al.*, 2021] Guangzhi Sun, Chao Zhang, and Philip C Woodland. Transformer language models with lstm-based cross-utterance information representation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7363–7367. IEEE, 2021.

[Yang *et al.*, 2021] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*, 2021.

[Zheng *et al.*, 2020] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.