Deteksi Deepfake Video Berbasis VTT Vision Temporal Transformer

Ikhlasul Amal*
Department Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
ikhlasulamal@mail.ugm.ac.id

Magnolia G. R. Satriorini*
Department Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
magnoliagina@mail.ugm.ac.id

Nilam Mufidah*
Department Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
nilam19mufidah@gmail.com

Sisi Florensia*
Department Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
sisiflorensia@gmail.com

***Abstract*—Penyebaran video *deepfake* yang semakin realistis telah memicu kebutuhan mendesak akan mekanisme deteksi yang andal. Penelitian ini mengimplementasikan dan mengevaluasi arsitektur LIPINC Model yang memanfaatkan VTT untuk mengekstraksi dan menggabungkan fitur spasial dan temporal dari *video frame* dan *residu temporal*. Model ini dirancang untuk mengatasi artefak *deepfake* yang seringkali termanifestasi sebagai inkonsistensi temporal. Selain itu, model ini dioptimalkan menggunakan kombinasi *Loss Categorical Cross-Entropy* untuk klasifikasi dan *Batch Consistency Loss* untuk mendorong kekonsistenan fitur dalam batch. Model ini diuji dalam lima konfigurasi implementasi dan variasi kombinasi dataset. Hasil penelitian menunjukkan bahwa konfigurasi yang menggunakan tiga dataset menghasilkan kinerja terbaik mencapai ROC-AUC 0.9256, AP 0.8889, dan IoU 0.7675. Penggunaan VTT yang diperkuat oleh informasi residu dan *consistency loss* terbukti efektif dalam mendeteksi *deepfake* di lingkungan dataset yang kompleks dan bervariasi.**

***Index Terms*—video classification, deepfake, computer vision, temporal transformer**

I. Pendahuluan

Munculnya jaringan generatif seperti *Generative Adversarial Networks* atau *GANs*, dan teknologi manipulasi wajah telah memungkinkan penciptaan video palsu yang dikenal sebagai *deepfake* dengan tingkat realisme yang mengkhawatirkan. Video-video *deepfake* ini menimbulkan ancaman serius terhadap keamanan informasi, integritas media, dan kepercayaan publik. Oleh karena itu, pengembangan sistem deteksi *deepfake* yang mampu mengidentifikasi artefak halus yang ditanamkan dalam manipulasi video menjadi area penelitian yang sangat penting.

Banyak metode deteksi *deepfake* kontemporer berfokus pada analisis artefak spasial, namun kerentanan terhadap manipulasi wajah sering kali terungkap melalui inkonsistensi domain temporal dan artefak fisik/fisiologis yang gagal dipertahankan antar *frame*.

Penelitian ini menggunakan model kustom LIPINC yang mengadopsi pendekatan berbasis Transformer yang dikenal sebagai *Vision Temporal Transformer (VTT)* untuk secara simultan memodelkan hubungan spasial pada setiap *frame* dan ketergantungan temporal di seluruh urutan *frame*.

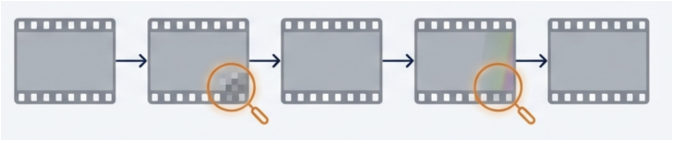

Fig. 1. Inkonsistensi Domain Temporal antar *Frame*

Serangkaian eksperimen ekstensif dilakukan pada berbagai kombinasi dataset forensik video agar dapat dilihat bagaimana skala dan keragaman data pelatihan memengaruhi generalisasi model. Dalam penelitian ini, dilakukan juga uji coba apakah penggunaan teknik pra-pemrosesan video untuk deteksi *deepfake*, seperti Media Pipe, dapat meningkatkan performa model.

II. Metode

Metodologi ini mencakup metode pemrosesan data, perincian arsitektur model Vision Temporal Transformer, dan fungsi *loss* ganda yang digunakan untuk pelatihan. Setelah proses pelatihan, model dievaluasi menggunakan metrik *Average Precision (AP)*, A*rea Under ROC Curve (AUC)*, dan *Intersection over Union (IoU)*.

A. Dataset

Penelitian ini mencakup lima skenario uji coba yang melibatkan empat dataset forensik video: CelebDF V2, FaceForensics++ (FF++), Deepfake Detection (DFD), dan Deepfake Detection Challenge (DFDC). Dalam semua dataset tersebut, jumlah total video palsu jauh lebih banyak dari video asli. Oleh karena itu, maka dilakukan *downsampling* pada data video palsu agar distribusi data video asli dan video palsu setara.

TABLE I
KONFIGURASI PENGGUNAAN DATASET

Method	V1	V2	V3	V4	V5
[R]Celeb	590	0	590	0	0
[F]Celeb	5639*	0	5639*	0	0
[R]FF++	0	1000	1000	1000	1000
[F]FF++	0	4000*	4000*	0	4000*
[R]DFD	0	0	0	363	0
[F]DFD	0	0	0	3068*	0
[R]DFDC	0	0	0	77	0
[F]DFDC	0	0	0	0	0

*Undersampled for balanced data. [R]Real Video [F]Fake Video

B. Preprocessing Data

Dataset akhir kemudian dibagi menjadi tiga bagian data training/validasi/test dengan proporsi 70/15/15.

Untuk mengatasi keterbatasan memori, data video dimuat secara *on-the-fly* menggunakan *Video Data Generator.* Video diproses dengan memuat 8 *frame* dan setiap *frame* diubah ukurannya menjadi dimensi 64×144 piksel. *Frame-frame* ini kemudian dinormalisasi dan di-padding, kemudian residu dihitung sebagai perbedaan *frame* sekarang dan *frame* sebelumnya.

C. Arsitektur Model: LIPINC

Model yang digunakan dalam penelitian ini adalah variasi dari arsitektur dua aliran yang berfokus pada *frame-frame* video dan gerakan antar *frame* tersebut. Model ini menerima dua jenis input:

1) ***Frame Input*** Urutan video f*rame* berdimensi (8, 64, 144, 3), mewakili 8 frame, masing-masing berukuran 64×144 piksel dengan 3 saluran warna (RGB).
2) ***Residual Input*** Residu temporal yang dihitung sebagai perbedaan *sequential frames* berdimensi (7, 64, 144, 3). Residu ini secara khusus menargetkan dan memperkuat sinyal gerakan dan inkonsistensi transisional antar *frame*.

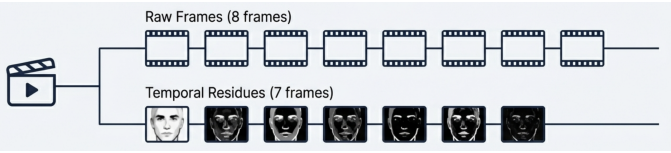

Fig. 2. Dua Aliran Input pada LIPINC

Kedua input ini diproses oleh blok inti yang sama: *Vision Temporal Transformer (VTT)*, yang adalah komponen kunci yang dirancang untuk mengatasi dimensi spasial dan temporal dari data video.

Patching dan Proyeksi Spasial VTT dimulai dengan memecah input video (baik bingkai maupun residu) menjadi patch non-tumpang tindih. Input yang berbentuk (B × F × H × W × C) (*Batch, Frames, Height, Width, Channel*) di reshape menjadi (B . F × H × W × C), dan kemudian dipecah menjadi banyak *patch* dengan ukuran 8×8. Setiap patch kemudian diproyeksikan menggunakan lapisan *Dense*. VTT juga menambahkan *positional embedding* yang dapat dilatih ke representasi *patch* yang diproyeksikan.

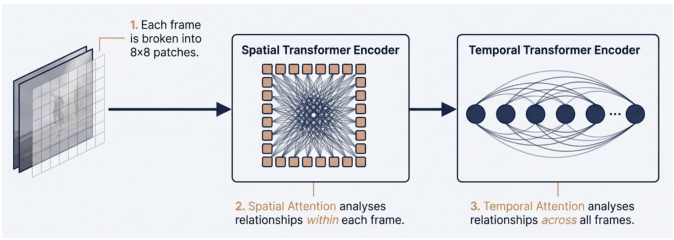

Fig. 3. *Vision Temporal Transformer (VTT)*

Lapisan Transformer Spasial Setelah proyeksi, VTT menerapkan sejumlah lapisan *Spatial Transformer*. Setiap lapisan melibatkan siklus *Multi Head Attention (MHA)*, diikuti oleh *Layer Normalization*, *Feed-Forward Network (FFN)*, dan *Layer Normalization* kedua. Perhatian spasial ini memungkinkan model untuk memodelkan hubungan antar *patch* dalam satu *frame*.

Temporal Pooling Fitur yang diperkaya secara spasial (B × F × P × d-model), di mana P adalah jumlah *patch* diubah bentuknya kembali menjadi urutan waktu (B × F × P × d-model) dan di-rata-ratakan atau *pooled* sepanjang sumbu *patch*, menghasilkan representasi temporal (B × F × d-model).

Lapisan Transformer Temporal Representasi temporal kemudian melalui sejumlah lapisan *Temporal Transformer*. Lapisan ini menggunakan MHA untuk memodelkan hubungan ketergantungan antar frame sepanjang dimensi waktu. Keluaran akhir dari VTT diperoleh melalui *Global Average Pooling 1D* pada dimensi temporal, menghasilkan vektor fitur padat.

D. Multi Head Attention (MHA)

Lapisan MHA digunakan baik dalam VTT spasial/temporal maupun dalam tahap fusi fitur. MHA kustom diimplementasikan untuk kompatibilitas dengan versi TensorFlow yang lebih lama. Ia melakukan proyeksi linier terhadap *Query (Q)*, *Key (K)*, dan *Value (V)*, membagi head, melakukan *scaled dot-product attention*, dan kemudian menggabungkan head kembali.

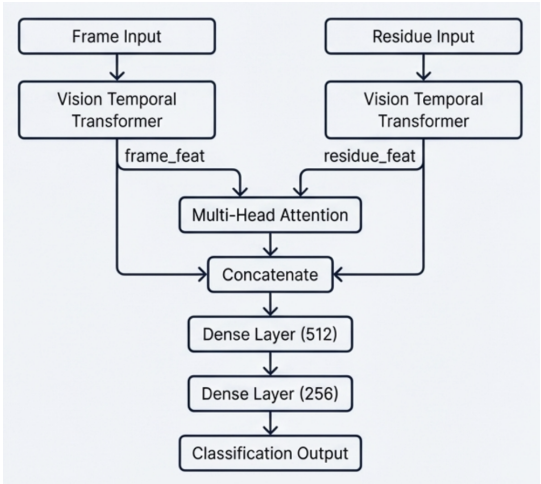

Fig. 4. *Arsitektur Model LIPINC*

E. Fusi Fitur dan Klasifikasi

Fitur yang diekstraksi dari flow *frame* input dan residual input kemudian digabungkan. Fusi ini dilakukan melalui mekanisme *(cross-attention)* di mana fitur *frame* menjadi *Query (Q),* sementara fitur residu berfungsi sebagai *Key (K)* dan *Value (V)*. Keluaran dari *(cross-attention)* ini kemudian digabungkan. Vektor gabungan ini kemudian diproses melalui lapisan *Dense* menggunakan aktivasi ReLU, kemudian diakhiri dengan lapisan *Dense* berikutnya dengan aktivasi softmax untuk menentukan (*class output*).

Baik blok spasial maupun temporal menggunakan sisa koneksi (*residual/skip connection*) dan normalisasi lapisan *(Layer Normalization)* untuk memfasilitasi pelatihan model yang dalam, di mana

$$x = Norm(Input + Attention(Input))$$

dan

$$x = Norm(x + FFN(x))$$

F. Loss Function

Model LIPINC dilatih menggunakan fungsi *loss* ganda yang memproses hasil ekstrasi vektor-vektor fitur dari layer *Dense* terakhir ke tugas berikutnya:

1) ***Categorical Cross-Entropy*** Digunakan untuk menentukan *class output* pada tugas klasifikasi utama untuk membedakan video asli atau palsu.
2) ***Batch Consistency Loss*** Diterapkan pada *features output* untuk memastikan bahwa fitur-fitur yang diekstraksi memiliki tingkat kesamaan yang tinggi dalam satu *batch*. Secara implisit ini memaksa model untuk belajar representasi yang konsisten, membantu mengekspos inkonsistensi yang melekat pada video *deepfake*.

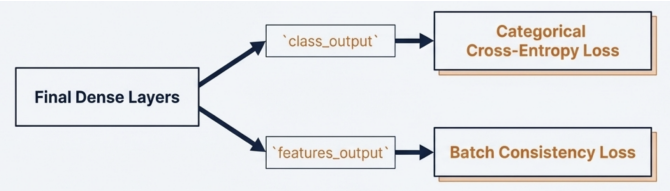

Fig. 5. *Arsitektur Model LIPINC*

Secara teori, fungsi *loss* ini meminimalkan jarak antara fitur-fitur dalam batch, memaksa model untuk mengekstrak fitur yang lebih stabil di berbagai *frame* video. Hal ini sangat berguna dalam deteksi *deepfake*, di mana inkonsistensi yang tidak disengaja sering terjadi akibat manipulasi.

G. Prosedur Pelatihan dan Evaluasi

Pelatihan dilakukan selama 50 epochs. *Callbacks* digunakan untuk mencegah *overfitting* dan menyimpan model terbaik:

- **Model Checkpoint** memantau *validation accuracy* dan menyimpan bobot terbaik.
- **Early Stopping** memantau *validation loss* dengan parameter *patience* 5, menghentikan pelatihan jika kerugian validasi tidak membaik.

Evaluasi dilakukan pada set pengujian terpisah. Dalam penelitian ini, metode evaluasi yang digunakan mencakup tiga metrik utama, yaitu *Average Precision (AP)*, *Area Under Curve (AUC)* pada kurva *Receiver Operating Characteristic (ROC),* dan *Intersection over Union (IoU)*. Agar tim peneliti dapat menganalisa anomali lebih detail lagi, digunakan juga dua indikator tambahan yaitu skor *recall* dari video asli dan *macro f1*.

Nilai AUC mendekati 1 menunjukkan performa model sangat baik dalam memisahkan dua kelas, sementara nilai mendekati 0,5 menunjukkan performa setara dengan tebak acak. Sedangkan untuk IoU, semakin tinggi nilai IoU, semakin akurat model dalam menemukan lokasi dan durasi manipulasi pada video *deepfake*.

III. Hasil dan Analisa Penelitian

Lima konfigurasi kompleksitas dan skala dataset diuji pada penelitian ini. V1 sampai V4 tidak menggunakan Media Pipe, sedangkan konfigurasi V5 menggunakan teknologi pra-pemrosesan video Media Pipe untuk mengevaluasi jika langkah pra-ekstraksi atau penyesuaian area wajah yang lebih presisi akan dapat menghasilkan akurasi yang lebih baik atau pun tidak.

TABLE II
PERBANDINGAN KINERJA MODEL

Method	V1	V2	V3	V4	V5
Recall*	0.89	0.87	0.41	0.75	0.70
Macro f1	0.46	0.39	0.55	0.85	0.62
AP	0.5564	0.4656	0.5921	0.8889	0.6917
AUC	0.5582	0.4352	0.5921	0.9256	0.6866
IOU	0.1616	0.0859	0.4444	0.7675	0.4225

*Skor recall untuk video asli

A. Kinerja pada Dataset Kecil/Sederhana

Ketika diuji hanya pada dataset tunggal, kinerja model kurang *robust*, terutama pada konfigurasi tanpa Media Pipe.

- Pada uji coba V1 dengan dataset CelebDF, model menunjukkan nilai *recall* 0.89 dari video asli yang cukup tinggi, tetapi nilai ROC-AUC 0.5582 yang mendekati klasifikasi acak. Nilai IOU-nya jusga sangat rendah di 0.1616.
- Saat menggunakan dataset FF++ dengan total 2000 video, *recall* video asli tetap tinggi di 0.87, dan metrik agregat seperti ROC-AUC (0.4352), AP (0.4656), dan macro f1 (0.39) tetap buruk. Nilai IOU-nya semakin jatuh lebih drastis lagi ke 0.0859.
- Dari nilai ini, terlihat sepertinya model dengan dataset tunggal mungkin mendeteksi video asli dengan cukup baik tetapi gagal mendeteksi video palsu, menghasilkan skor f1 makro dan IOU yang rendah.

B. Kinerja pada Kombinasi Dataset Skala Besar

Kinerja model lebih stabil ketika dilatih pada dataset yang lebih besar dan lebih beragam. Konfigurasi tanpa Media Pipe dengan tiga dataset (FF++, DFD, dan DFDC) dengan jumlah total 2,880 video, mencapai metrik deteksi tertinggi dalam semua eksperimen:

- ROC-AUC mencapai 0.9256.
- AP mencapai 0.8889.
- IoU mencapai 0.7675.
- *Recall* video asli agak lebih rendah dibandingkan V1 dan V2 dengan dataset tunggal di 0.75.
- Akan tetapi, *macro f1* meningkat tajam mencapai 0.85.

Peningkatan besar ini dimana model dengan dataset tunggal FF++ ROC-AUC naik menjadi 0.9256 pada model dengan tiga dataset, menekankan pentingnya keragaman dan skala data untuk melatih VTT agar dapat menggeneralisasi perbedaan antara video *deepfake* dan asli di berbagai teknik manipulasi.

C. Analisis Dampak Media Pipe

Penggunaan Media Pipe menghasilkan peningkatan yang signifikan pada semua metrik agregat (Macro f1, AP, ROC-AUC, IOU) pada FF++ dataset.

- ROC-AUC meningkat dari 0.4352 menjadi 0.6866.
- Macro f1 meningkat dari 0.39 menjadi 0.62.
- IoU meningkat dari 0.0859 menjadi 0.4225.

TABLE III
PERBANDINGAN KINERJA MODEL TANPA DAN DENGAN MEDIA PIPE

Method	V2 tanpa Media Pipe	V5 dengan Media Pipe
Recall*	0.87	0.70
Macro f1	0.39	0.62
AP	0.4656	0.6917
AUC	0.4352	0.6866
IOU	0.0859	0.4225

*Skor recall untuk video asli

Namun, perlu dicatat bahwa skor *recall* dari video asli sedikit menurun dari 0.87 (tanpa Media Pipe) menjadi 0.70 (dengan Media Pipe). Meskipun demikian, peningkatan besar pada Macro f1 dan IOU menunjukkan bahwa Media Pipe membantu menyeimbangkan kinerja klasifikasi secara keseluruhan, mengurangi bias yang terlihat pada konfigurasi tanpa MediaPipe di mana *recall* video asli tinggi tetapi metrik keseimbangan lainnya rendah.

Meskipun demikian, konfigurasi dengan kinerja keseluruhan terbaik ROC-AUC 0.9256 tetap dicapai pada kombinasi menggunakan tiga dataset tanpa menggunakan Media Pipe. Ini menunjukkan bahwa pada data yang cukup besar dan beragam, kemampuan bawaan arsitektur VTT untuk mengekstraksi fitur spatio-temporal, didukung oleh *consistency loss*, mungkin sudah cukup untuk menghasilkan model deteksi *deepfake* yang andal.

D. Hasil Pengujian Inferensi

Penelitian ini juga melakukan pengujian inferensi pada himpunan data dunia nyata yang dikumpulkan secara manual. Untuk video palsu, proses kurasi dilakukan dengan menelusuri artikel klarifikasi hoaks di laman resmi Kementerian Komunikasi dan Digital (Komdigi) serta kanal Cekfakta Tempo menggunakan kata kunci ”deepfake”. Video *deepfake* kemudian diekstraksi dari arsip artikel-artikel tersebut dan dianotasi sebagai video palsu. Untuk video asli, tim peneliti mengumpulkan dua kategori: (1) video sumber asli (sebelum dimanipulasi) yang sesuai dengan pasangan *deepfake* yang ditemukan, dan (2) video lain dengan publik figur yang sama dari berbagai media daring, yang kemudian diverifikasi secara manual agar benar-benar merupakan rekaman autentik dan bukan *deepfake*. Himpunan data ini dirancang untuk merepresentasikan skenario penggunaan riil di mana model harus mengidentifikasi *deepfake* yang telah beredar luas di internet.

Seluruh konfigurasi model V1 hingga V5 kemudian diuji ulang pada himpunan data baru ini tanpa pelatihan ulang, untuk mengukur kemampuan generalisasi di luar distribusi dataset pelatihan awal. Selain metrik yang sama seperti sebelumnya (*recall* video asli, *macro f1*, AP, ROC-AUC, dan IoU), pada pengujian inferensi ini juga diukur latensi rata-rata per video (dalam detik) sebagai indikator efisiensi komputasi saat deployment. Hasil pengujian dirangkum pada Tabel IV. Secara umum, konfigurasi V4 (tiga dataset tanpa Media Pipe) tetap menunjukkan kinerja paling seimbang pada data dunia nyata, dengan ROC-AUC dan IoU yang lebih tinggi dibanding konfigurasi lain serta latensi yang masih kompetitif. Sementara itu, Media Pipe pada V5 memberikan peningkatan IoU yang signifikan dibanding V2, namun dengan kompromi berupa latensi yang lebih tinggi.

TABLE IV
KINERJA UJI INFERENSI PADA DATASET BERITA HOAKS DAN VIDEO PUBLIK FIGUR

Method	V1	V2	V3	V4	V5
Recall*	0.96	0.88	0.94	0.80	0.08
Macro f1	0.3243	0.3985	0.38	0.7494	0.4165
AP	0.5000	0.4955	0.5000	0.6944	0.5208
AUC	0.4800	0.4900	0.5000	0.7500	0.5400
IoU	0.0000	0.0893	0.0566	0.5833	0.5208
Latency (s)	0.1855	0.2054	0.1819	0.1988	0.3072

*Skor recall untuk video asli

IV. KESIMPULAN

Penelitian ini berhasil mengevaluasi arsitektur VTT yang didukung oleh input residu temporal dan *batch consistency loss* untuk tugas deteksi *deepfake* video.

Temuan utama meliputi:

1) **Superioritas Skala dan Keragaman Data** Kinerja optimal model dicapai ketika dilatih pada dataset yang besar dan beragam, yang mencakup FF++, DFD, dan DFDC. Konfigurasi ini menghasilkan metrik tertinggi: ROC-AUC 0.9256, AP 0.8889, dan IoU 0.7675. Hal ini menggarisbawahi pentingnya generalisasi model terhadap berbagai teknik manipulasi yang berbeda.
2) **Efek Positif Konsistensi/Residu** Peningkatan kinerja yang signifikan pada dataset gabungan, yang melampaui kinerja klasifikasi acak yang diamati pada dataset

tunggal, mengindikasikan efektivitas model LIPINC yang memanfaatkan residu dan *consistency loss*, dalam menangkap inkonsistensi *spatio-temporal*.

3) **Dampak Media Pipe** Penggunaan Media Pipe pada dataset FF++ tunggal sebagai lapisan pra-pemrosesan wajah secara signifikan meningkatkan metrik agregat seperti Macro f1, AP, ROC-AUC, meningkatkan kinerja dari buruk menjadi moderat. Ini menunjukkan bahwa pemotongan dan penyesuaian wajah yang akurat dapat menjadi substitusi efektif untuk keragaman data ketika menggunakan data pelatihan yang terbatas.

Secara keseluruhan, VTT menawarkan solusi yang kuat untuk deteksi *deepfake*, terutama bila didukung oleh mekanisme *loss* yang mendorong kekonsistenan fitur dan dilatih pada corpus data yang luas dan bervariasi.

V. Rekomendasi

Berdasarkan hasil eksperimental dan batasan metode yang digunakan, berikut beberapa rekomendasi untuk penelitian di masa depan.

1) **Eksplorasi Efek MediaPipe dalam Skala Besar** Konfigurasi tiga dataset yang menggunakan Media Pipe tidak berhasil dilakukan dalam penelitian ini. Penelitian lanjutan dapat membandingkan secara langsung kinerja *best-case* (tiga dataset tanpa Media Pipe) dengan hasil tiga dataset menggunakan Media Pipe. Tujuannya adalah untuk memahami apakah pra-pemrosesan yang presisi tetap memberikan nilai tambah pada model yang sudah dilatih pada data yang banyak dan sangat beragam.
2) **Analisis Visualisasi VTT** Meskipun VTT dirancang untuk menganalisis spasial dan temporal, penelitian di masa depan sebaiknya difokuskan pada visualisasi mekanisme atensi MHA untuk mengonfirmasi bagian mana dari bingkai (wajah, leher, atau latar belakang) atau bingkai waktu mana yang paling berkontribusi terhadap keputusan deteksi *deepfake*.
3) **Optimasi *Hyperparameter* VTT** Model yang diuji menggunakan konfigurasi VTT yang sederhana. Menguji konfigurasi yang lebih dalam, misalnya 3 atau 4 lapisan Transformer, dan dimensi model yang lebih tinggi dapat mengoptimalkan kemampuan ekstraksi fitur lebih lanjut.
4) **Keterbatasan Residu** Metode residue yang digunakan adalah perbedaan *frame* sederhana. Mengimplementasikan metode estimasi gerakan yang lebih canggih, seperti *optical flow*, dapat memberikan sinyal gerak yang lebih bersih dan mungkin meningkatkan performance model, terutama dalam mendeteksi anomali gerakan yang halus.

References

[1] S. K. Datta, S. Jia, and S. Lyu, "Detecting Lip-Syncing Deepfakes: Vision Temporal Transformer for Analyzing Mouth Inconsistencies," *arXiv*, 2025. doi: 10.48550/arXiv.2504.01470.

[2] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[3] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 3163–3172.

[4] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video Swin Transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 3202–3211.

[5] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, 2005.

[6] Z. Yan, Y. Luo, S. Lyu, Q. Liu, and B. Wu, "Transcending forgery specificity with latent space augmentation for generalizable deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 8984–8994.

[7] P. Kwon, J. You, G. Nam, S. Park, and G. Chae, "KoDF: A large-scale Korean deepfake detection dataset," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10744–10753.