International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

True Vision AI: Deepfake Video Detection Using a Hybrid Ensemble of Xception and Video Vision Transformer (ViViT)

Gautham Rishab S[1], Gowtham S[2], Mrs. Sakthi P[3]
[1,2] UG - Department of Information Technology, B.S.A. Crescent Institute of Science & Technology,Chennai, India.
[3]Assistant Professor, Information Technology, B.S.A. Crescent Institute of Science & Technology, Chennai, *India.*
Email ID: *rishabgautham@gmail.com[1], gowthamselvam2579@gmail.com[2], sakthi@crescent.education[3]*

Abstract
Deepfakes have become one of the most pressing issues of our time, and what used to take a team of visual effects experts weeks to do can now be done in minutes using freely available software, with results increasingly indistinguishable from reality. We present True Vision AI, a deepfake video detection system based on a two-stream ensemble approach utilizing both spatial and temporal understanding. Our system combines a fine-tuned Xception network (pre-trained on ImageNet) for detecting subtle visual inconsistencies in individual frames, alongside a Video Vision Transformer (ViViT-B/16x2, pre-trained on Kinetics-400) for detecting motion-level anomalies across frames. Features from both networks are merged into a unified 2,816-dimensional vector fed into a compact classifier to determine whether a video is real or fake. Trained and tested on the Celeb-DF dataset (890 genuine videos and 808 deepfakes), our Xception model achieves 88.5% validation accuracy, ViViT achieves 87.0%, and the ensemble achieves 88.3%, The final system is deployed as a lightweight Flask API that provides a determination, a confidence score, and a frame-level breakdown of where deception is likely occurring.

Keywords: *Deepfake Detection, Xception, Video Vision Transformer (ViViT), Ensemble Learning.*

1. Introduction
Nowadays generative models like GAN and diffusion based techniques can easily create a fake video with ease and the videos are realistic deepfake videos. These videos can swap faces or modify expressions in a way that is often difficult to notice with the human eye. Because of this, deepfakes video have become a serious problematic in places where money and other important political things are happening. Before the advancement of GAN and Diffusion Model detection methods relied on visible artifacts like unnatural facial boundaries or irregular blinking patterns, but modern deepfakes have improved so much you can't relay no older methods. So our Team have researched about this and came to a conclusion. Initial approaches used manually designed features, but more recent work has focused on deep learning models. Convolutional neural networks, such as Xception and similar architectures, are effective at identifying spatial inconsistencies within individual frames. But they do not consider the inconsistencies that change over time in a fake video. And temporal models like LSTMs, 3D-CNNs, Transformers or only focused on capturing the inconsistency over the time. Among these, the Video Vision Transformer has shown strong performance in understanding long range patterns in video data. Our work proposes True Vision Ai Using Ensemble model that combines both spatial and temporal approaches. The system uses a finetuned Xception model to extract features from individual frames and a pretrained Video Vision Transformer to analyze sequences of frames. Then the features are combined using a lightweight ensemble module to

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

improve overall performance. Our Xception model achieved around 88.5% validation accuracy, And the transformer model reached 87.0%. After combining, the system achieved about 88.3% validation accuracy, showing that the integration of both models led to more supportive results. In addition to model development, the system was implemented as a Flask-based API for practical use. It provides predictions along with confidence scores for each frame and a final classification indicating whether the video is real or fake. This makes the approach not only effective but also suitable for real-world deployment.

2. Literature Review

What made deepfakes possible in the first place was a fairly clever idea from Goodfellow et al. 2020 [3] train two networks simultaneously, one trying to generate fake content and the other trying to catch it, until the generator gets good enough to fool almost anyone. This adversarial training setup turned out to be surprisingly powerful for face synthesis. Karras et al. 2019 [28] took this further with StyleGAN, producing synthetic faces so visually convincing that even trained observers struggled to identify them as fake. From that point on, the gap between what machines could generate and what humans could detect started closing fast. Detection research picked up pace around the same time generation methods were improving. Güera et al. 2018 [8] put forward one of the earlier video-focused solutions, feeding CNN-extracted frame features into a recurrent network to pick up on unnatural changes between frames over time. Concurrently, Afchar et al. 2018 [6] took a different route and built MesoNet a deliberately small network that still managed decent detection accuracy by focusing on mesoscopic facial properties. What the field also needed was proper evaluation infrastructure, and Rossler et al. 2019 [1] addressed that gap with FaceForensics ++, a structured dataset covering several manipulation techniques that quickly became the go-to benchmark for comparing detection approaches. On the architecture side, Chollet et al. 2017 [4] had already developed Xception for general vision tasks, but its depthwise separable convolution design happened to work well for spotting the kinds of low-level inconsistencies that face manipulation tends to introduce. As detection models improved though, so did the fakes they were tested against. Li et al. 2020 [2] released Celeb-DF to reflect this, a dataset where the manipulated videos were noticeably harder to detect than those in earlier benchmarks, exposing how fragile some previously strong-performing models actually were. More recent work shifted focus toward where and how models pay attention. Zhao et al. 2021 [11] designed a detection system with multiple attention heads targeting specific facial zones — areas like the eyes, mouth, and contour that forgery algorithms tend to handle imperfectly. Separately, Haliassos et al. 2021 [16] found that lip movements carry a surprisingly strong detection signal, largely because current synthesis methods do a poor job of preserving the fine temporal dynamics of natural speech. On the video understanding side, Arnab et al. 2021 [9] introduced ViViT, which applies transformer-style self-attention across both space and time, giving it a meaningful edge over older recurrent and 3D convolutional methods when it comes to modeling how faces move and change across frames. Across all these directions, one pattern keeps showing up: systems that look at video from multiple angles spatially and temporally — tend to hold up better than those relying on a single type of signal. That observation directly motivated the design of the current system, which brings together Xception for frame-level analysis and ViViT for motion-level reasoning, combining their outputs into a unified classifier evaluated on Celeb-DF [2].

3. Methodology

This section describes the end-to-end pipeline of True Vision AI from raw video input through feature extraction and ensemble fusion to final classification and deployment shown in figure 1.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

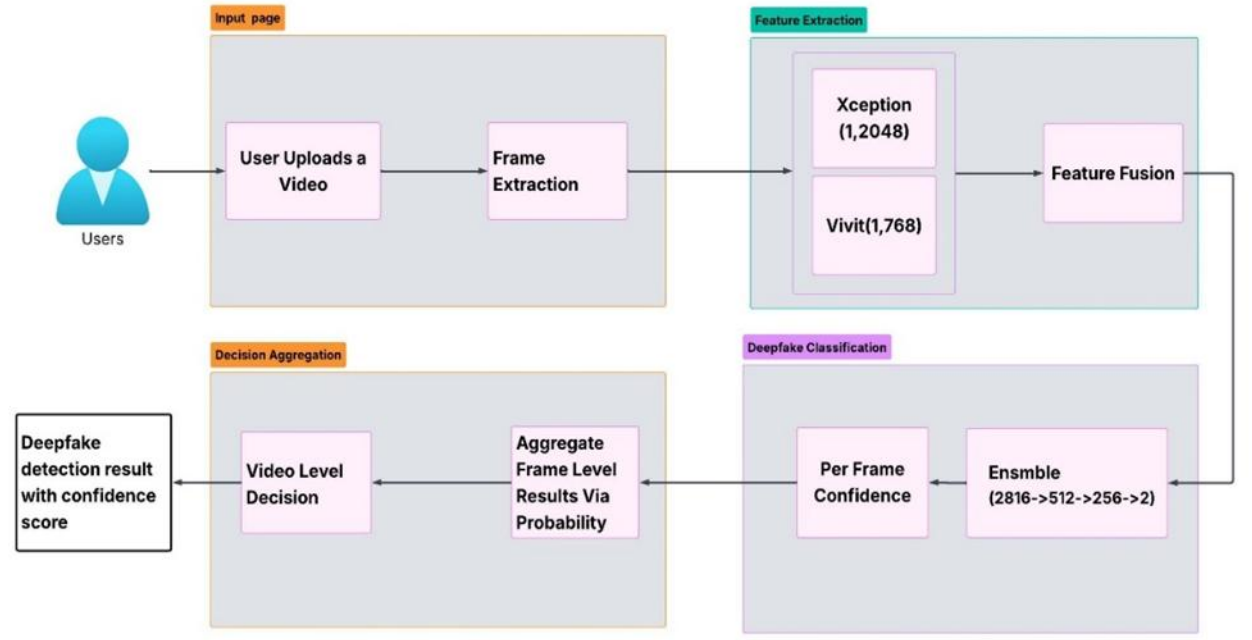

Figure 1 System Architecture of True Vision AI Using Ensemble Model

3.1.Dataset and Preprocessing

The experiments in this work are conducted using the Celeb-DF Dataset version 2 which consists of 890 real videos sourced from celebrity interviews and 808 deepfake videos Compared to older benchmarks like FaceForensics++ Rossler et al. 2019 [1], the synthetic videos in Celeb-DF Li et al. 2020 [2] are considerably more realistic, which makes correct detection harder and the dataset Since both two models used in this system have different input requirements, frame extraction needed to be handled carefully and consistently. Each video is sampled to 16 frames so we could get the fixed step size if a video has shorter frame that is less than 16 frames, all available frames are used and the last frame is repeated to fill the remainder. The extracted frames are resized to 299×299 for Xception and 224×224 for ViViT, we use ImageNet mean and deviation values to normalize the values. To improve generalization during training, augmentations including random horizontal flips, brightness and contrast adjustments, and random cropping are applied uniformly across all frames in a clip rather than per frame individually —so that spatial consistency is preserved throughout the sequence. ViViT additionally uses a randomized temporal offset during sampling to introduce clip-level variability. The full pipeline produces just over 27,000 frame images across the dataset.

3.2.Spatial Stream: Xception

The Xception architecture employs depthwise separable convolutions in an extreme inception layout, achieving an efficient factorization of channel wise and spatial wise convolutions. We represent Xception using the timm library replacing the final classification layer with a binary output for real/fake detection. Training follows a two-phase protocol. In Phase 1 warmup, epochs 1–2 the entire backbone is frozen and only the classifier head is trained allowing the head to stabilize before the backbone weights are perturbed. In Phase 2 epochs 3–20 the backbone is fully unfrozen and trained end-to-end. We use AdamW optimizer with weight decay 0.01, with separate learning rates: 1×10^{-4} for the backbone and 1×10^{-3} for the classifier head. A cosine annealing schedule decays the learning rate smoothly from the initial value to 1×10^{-7} over 20

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

epochs shown in figure 2.

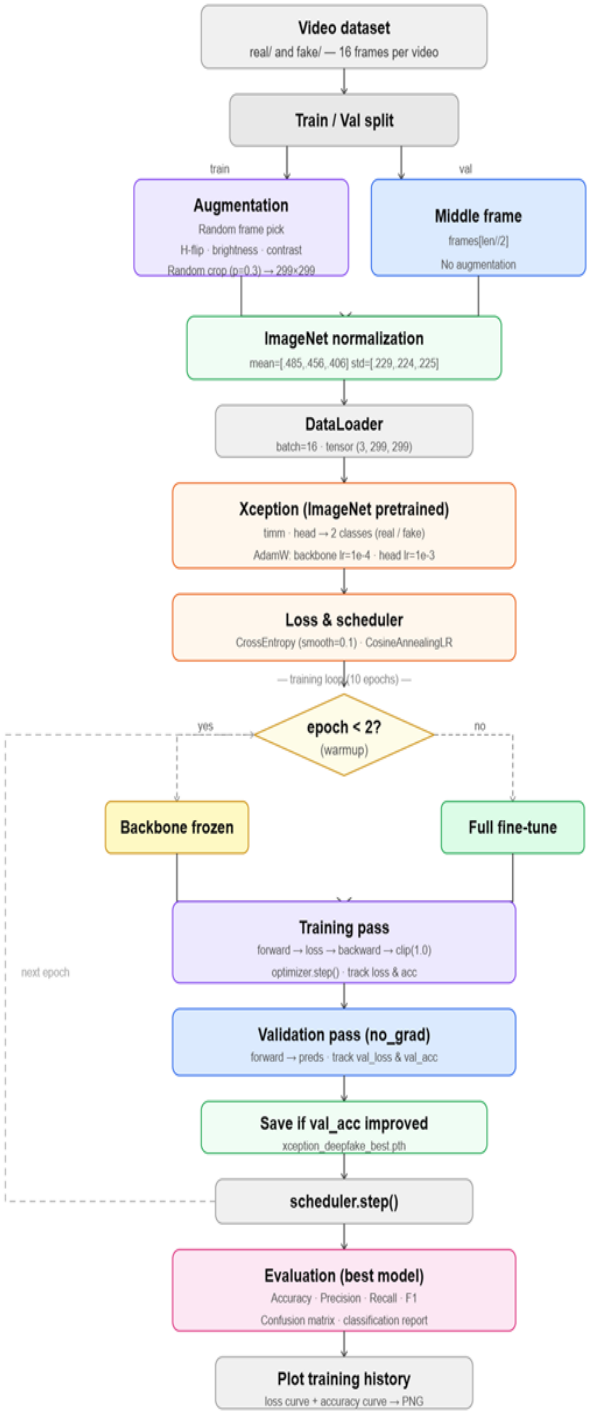

Figure 2 Xception Model Setup

Algorithm: Xception model setup

Nummcls = 2, pretrained ,True

model = timm.create_model('xception', pretrained=True, nummcls=2)
model = model.to(device)
bakbonepprams ← {p | name ∉ {fc, head}}
clasparams ← {p | name ∈ {fc, head}}
optimizer ← AdamW([
{params: bakbonepprams, lr: 1e−4, weightdecay: 0.01},
{params: clasparams, lr: 1e−3, weightdecay: 0.01}
])
criterion = CrossEntropyLoss(labelsmoothing = 0.1)
scheduler = CosineAnnealingLR(optimizer, Tmax=10, ηmin=1e−7)
return model, optimizer, criterion, scheduler

3.3.Temporal Stream: ViViT

ViViT extends the standard Vision Transformer architecture to handle video input by breaking clips into non-overlapping spatio-temporal patches called tubelets and running multi-head self-attention across the full token sequence. The variant used here is ViViT-B/16×2, loaded from a checkpoint pre-trained on Kinetics-400, and reconfigured for binary classification with 16 input frames. Training follows a two-phase approach similar to Xception, though the warmup period is extended to 3 epochs given how sensitive transformer-based models tend to be to early learning rate changes. The optimizer is Adam W with a weight decay of 0.01,we use learning rate of 2×10^{-5} for the backbone and a 2×10^{-4} for the classification head. Our model is trained for 30 epochs with a gradually decreasing learning rate we uses small batches of 4 due to memory limits, applies techniques to keep training stable, and finally uses a 768-dimensional feature from the model to represent each clip during prediction shown in figure 3.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

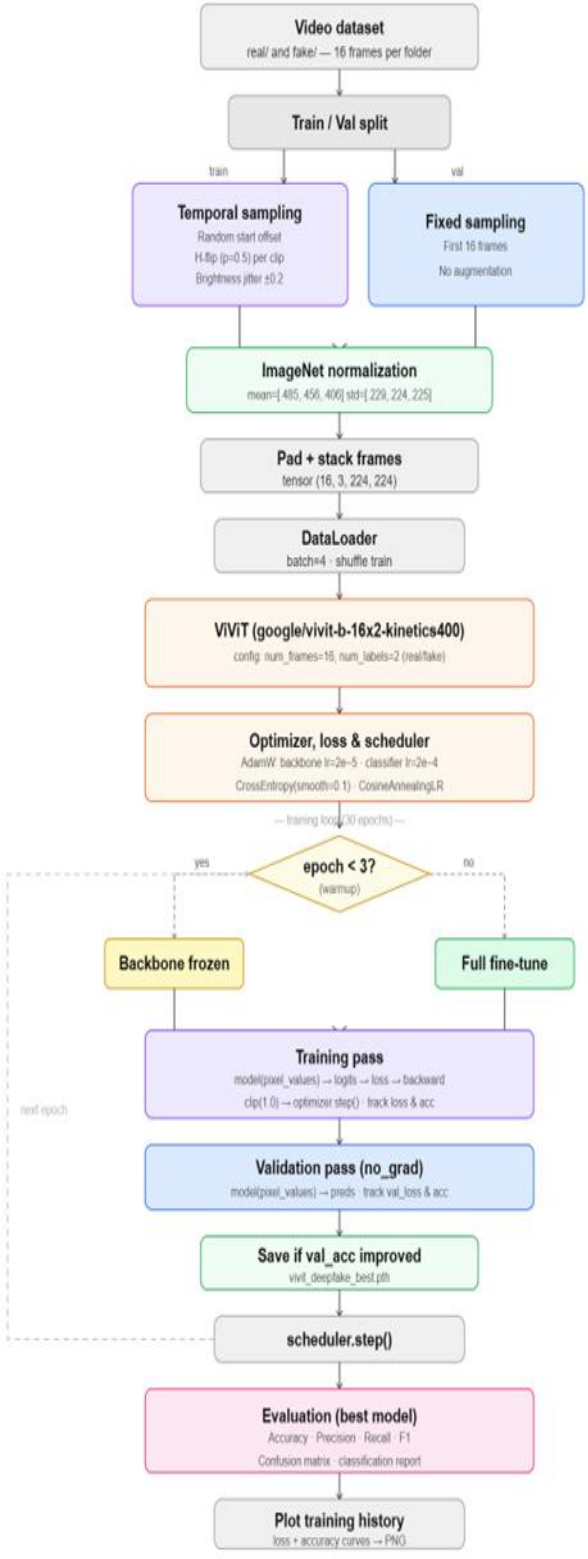

Figure 3 ViViT Model Setup

Algorithm: ViViT model setup

pretrained = 'google/vivit-b-16x2-kinetics400'
num_frames = 16, num_labels = 2
Output: model, optimizer, criterion, scheduler
config ← VivitConfig.from_pretrained(pretrained)
config.num_frames =16
config.num_labels = 2
config.id2label = {0: real, 1: fake}
model=VivitForVideoClassification.from_pretrained(pretrained,config=config,ignore_mismatched_sizes=True)
model =model.to(device)
backbone_params ← {p | name does not contain 'classifier'}
classifier_params ← {p | name contains 'classifier'}
optimizer ← AdamW([{params: backboneparams, lr: 2e−5, weightdecay: 0.01},{params: classifierparams, lr: 2e−4, weightdecay: 0.01}])
criterion=CrossEntroyLoss(label_smoothing= 0.1)
scheduler=CosineAnnealgLR(optimizer,Tmax=30, ηmin=1e−7)
return model, optimizer, criterion, scheduler

3.4.Ensemble Fusion and Classifier Head

We grab a 2048-dimensional vector from Xception for spatial features and mix it with a 768-dimensional CLS token from ViViT. Throw them together and you get one big 2816-dimensional vector. That's what we feed straight into the classifier. The classifier isn't anything fancy. The model pushes that vector through two fully connected layers, chopping it from 2816 to 512, then to 256. After each round we do normalization with ReLU and some dropout. That combo keeps training steady and fights off overfitting. One last linear layer spits out a single value for binary classification real or fake shown in figure 4.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

Figure 4 Ensemble Classifier Head Architecture and Training Setup

3.5.Inference Pipeline and Deployment

Deployment's a breeze. The model sits on a Flask API and takes MP4 videos up to 50 MB. It grabs 16 frames from each video, spaced out evenly, and runs them through the backbone networks. After fusing the features, the classifier head kicks in and gives you a prediction. The API returns a JSON: predicted label, class probabilities, frame-level scores—the whole package. If there's a GPU, things move faster. If not, no sweat, it just runs on CPU. There's nothing to fiddle with, so you can deploy anywhere without hassle.

4. Results And Discussion

4.1. Results

We ran our experiments on Gpu RTX 5050 8 GB VRAM, using Python version 3.12 Our Xception model was loaded through the timm library, while ViViT used Hugging Face Transformers. We used the Celeb-DF (v2) dataset, which contains 890 real and 808 fake videos. From these, were extracted 16 frames per video. To keep results consistent, a fixed random seed of 42 was used. We trained our Xception model using 20 epochs with the extracted frame. While training the loss kept going down, while the accuracy improved from roughly 80% at the start to about 96% by the final epoch. On the validation side, the accuracy increased more slowly and settled around 88–89% towards the later stages. The difference between training and validation accuracy, which is around 7–8%, suggests that the model does overfit a bit, but not in a way that is too concerning. To manage this, we used regularization methods such as dropout, label smoothing, and a learning rate schedule. It helped keep the training stable. Our model was able to pick up meaningful patterns from the data shown in table 1 and figure 5.

Table 1 Xception Training Summary

Metric	Train (Epoch 20)	Best Validation
Accuracy	~96.0%	~88.5%
Loss	~0.26	~0.46

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

Optimizer	AdamW	AdamW
LR Schedule	Cosine Annealing	Cosine Annealing
Epochs	20	20
Batch Size	16	16

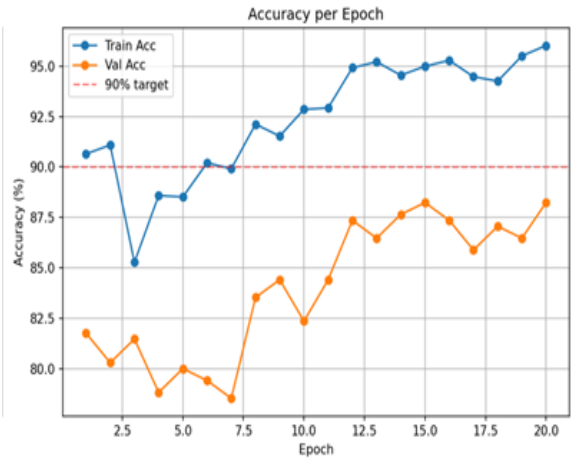

Figure 5 Xception Training and Validation Loss/Accuracy Curves over 20 Epochs

We trained our ViViT model using 30 epochs with 16 - frame clips. In the beginning we froze the backbone for 3 epoch and remaining epoch backbone were unfrozen the loss dropped, and our training accuracy improved and was close to 97% after about 20 epochs. Validation accuracy crossed 80% early on and later settled around 86–87%. To manage this, we used regularization methods such as dropout, label smoothing, and a learning rate schedule. It helped keep the training stable. Our model was able to pick up meaningful patterns from the data shown in table 2 and figure 6.

Table 2 ViViT Training Summary

Metric	Train (Epoch 30)	Best Validation
Accuracy	~97.0%	~87.0%
Loss	~0.20	~0.50
Optimizer	AdamW	AdamW

LR Schedule	Cosine Annealing	Cosine Annealing
Epochs	30	30
Batch Size	4	4

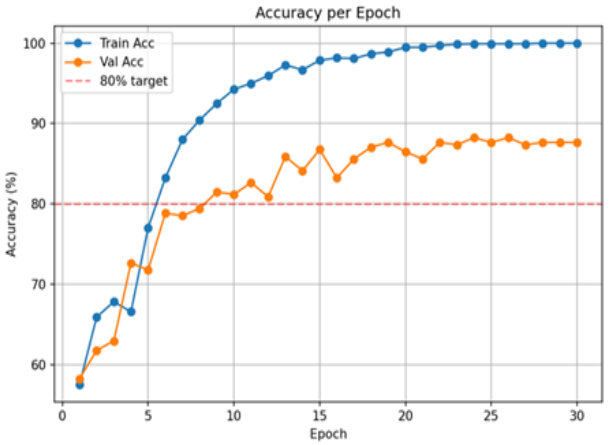

Figure 6 ViViT Training and Validation Loss/Accuracy Curves over 30 Epochs

We trained our ensemble for 20 epochs while keeping both backbone models fixed. The training process was stable. And Our validation accuracy remained consistent between 87 to 88%, and didn't show any sudden drops. Overall, the ensemble seems to combine the features well and gives stable results in yable 3 and figure 7.

Table 3 Ensemble Head Training Summary

Metric	Train (Epoch 20)	Best Validation
Accuracy	>99.5%	~88.3%
Loss	~0.21	~0.51
Fused Dim	2816	2816
Optimizer	AdamW	AdamW
Epochs	20	20
Batch Size	8	8

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

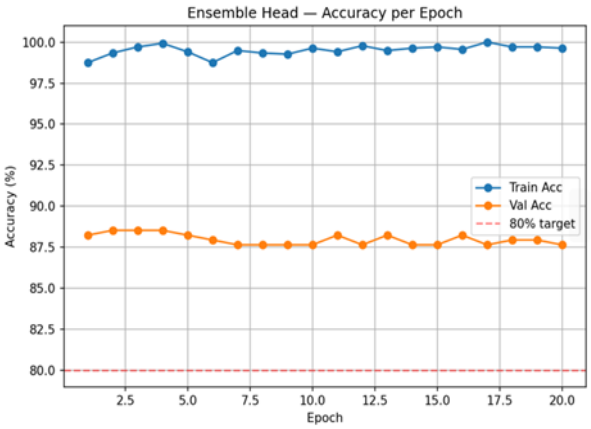

Figure 7 Ensemble Head Training and Validation Loss/Accuracy Curves over 20 Epochs

Table 4 Comparison of All Model Configurations on Celeb-DF Validation Set

Model	Val. Accuracy	Training Epochs	Feature Dim	Modality
Xception (Standalone)	~88.5%	20	2048	Spatial (Frame)
ViViT-B/16×2 (Standalone)	~87.0%	30	768	Temporal (Clip)
Ensemble Head (Fused)	~88.3%	20	2816	Spatial + Temporal

Our overall ensemble achieves validation accuracy comparable to the best individual model Xception while providing the additional benefit of temporal reasoning from ViViT. The ensemble's validation accuracy 88% is consistent with a well-regularized system on a constrained dataset, and the low variance in validation accuracy across epochs demonstrates reliable inference behavior in table 4.

4.2.Discussion

The Looking at how the models trained, a few things stand out. Freezing the backbone at the start and then unfreezing it later seemed to help both models learn more smoothly. After unfreezing, the improvements were more consistent. The learning rate schedule also helped keep things stable, especially in later epochs where the loss curves became smoother. Label smoothing played a role too, since it prevented the model from becoming too confident, which is why the validation loss didn't blow up even when training accuracy was very high. For the ensemble head, the behavior was a bit different. It reached very high training accuracy almost immediately, while validation accuracy stayed steady. This makes sense because the backbone models were already extracting strong features, so the head just needed to learn how to combine them properly, which it did quite quickly.

Conclusion

Our paper presents True Vision Ai Using Ensemble model our system is a detection system that combines spatial and temporal learning approaches. This model uses a fine-tuned Xception network to analyze individual frames and a Video Vision Transformer to capture patterns across video sequences. Our idea behind this combination is that a single model cannot fully capture all types of manipulation and we thought CNNs are effective at detecting visual inconsistencies within frames and transformer models help in understanding how these inconsistencies change over time, so by bringing both models together and using an ensemble strategy, our system is able to make more balanced predictions. Our model achieved around 88.3% validation accuracy on the Celeb-DF dataset. A Flask-based pipeline was also developed to make the system more practical, providing detailed outputs such as frame-level confidence scores and probability distributions, which can be useful in real-world applications like media verification.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

Acknowledgements

The authors are highly obliged to all the staff members at their institution for their constant guidance and constructive suggestions during the development of this paper. They owe much credit to them for their indispensable assistance in the development of this paper. The authors also thank all those sources which they have used to successfully complete this research. No financial assistance from any outside source was taken while conducting this research.

References

[1]. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Seoul, Korea, 2019, pp. 1–11.

[2]. Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, 2020, pp. 3207–3216.

[3]. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 27, 2020.

[4]. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1251–1258.

[5]. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. Int. Conf. Machine Learning (ICML), Long Beach, CA, 2019.

[6]. D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," in Proc. IEEE Int. Workshop Information Forensics and Security (WIFS), Hong Kong, 2018, pp. 1–7.

[7]. R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection," Information Fusion, vol. 64, pp. 131–148, Dec. 2020.

[8]. D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in Proc. IEEE Int. Conf. Advanced Video and Signal-Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1–6.

[9]. A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A Video Vision Transformer," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Montreal, Canada, 2021, pp. 6836–6846.

[10]. Z. Liu, X. Qi, and P. H. S. Torr, "Global Texture Enhancement for Fake Face Detection in the Wild," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, 2020.

[11]. H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-Attentional Deepfake Detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Nashville, TN, 2021.

[12]. Y. Li, M. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking," in Proc. IEEE Int. Workshop Information Forensics and Security (WIFS), Hong Kong, 2018.

[13]. S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-Generated Images Are Surprisingly Easy to Spot... For Now," arXiv preprint arXiv:1912.11035, 2019.

[14]. H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a Capsule Network to Detect Fake Images and Videos," arXiv preprint arXiv:1910.12467, 2019.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0317
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2197 – 2206

[15]. E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos," arXiv preprint arXiv:1905.00582, 2019.
[16]. A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips Don't Lie: A Generalisable and Robust Approach to Face Forgery Detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Nashville, TN, 2021.
[17]. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4489–4497.
[18]. C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Seoul, Korea, 2019, pp. 6202–6211.
[19]. I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, "Two-Branch Recurrent Network for Isolating Deepfakes in Videos," in Proc. European Conf. Computer Vision (ECCV), Glasgow, UK, 2020, pp. 667–684.
[20]. H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the Detection of Digital Face Manipulation," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, 2020.
[21]. B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection," arXiv preprint arXiv:2101.01456, 2021.
[22]. B. Dolhansky, "The DeepFake Detection Challenge (DFDC) Dataset," arXiv preprint arXiv:2006.07397, 2020.
[23]. I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2017.
[24]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770–778.
[25]. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015.
[26]. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR).
[27]. W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. C. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," arXiv preprint arXiv:1705.06950, 2017.
[28]. T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, 2019, pp. 4401–4410