Truth Behind DeepFakes(TBD)

GROUP - 9

CS 776 : Deep Learning For Computer Vision

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur (IITK)

MOTIVATION

1. **Cybersecurity & Identity Fraud.**

- impersonate individuals for **fraud, scams like digital arrests.**

2. **National Security & Law Enforcement.**

- Terrorist organizations create **fake confessions**
- Create **propaganda** during war to gain psychological mileage.

3. **Trust in Digital Media is Declining.**

- people are **losing trust** in videos and social media, making it difficult to distinguish real from fake.
  false statements can cause confusion and political mileage.

4. **Legal and Ethical Challenges**

- Efficient detection system is necessary to enforce laws.

Techniques

1. Image-Based Detection 📸

Detects deepfake images using extracted frames.

Techniques: Pixel & Texture Analysis , CNN-Based Models

Strengths: Works well on deepfake images; fast processing.

Weaknesses: May struggle with high-quality deepfake videos.

2. Video-Based Detection 🎥

Analyzes temporal inconsistencies in videos.

Techniques: Frame Consistency Analysis,Motion Analysis, Frame Extraction + CNNs

Strengths: More reliable than image-based detection.
Weaknesses: Requires more computational power.

3. Audio-Based Detection

Focuses on detecting fake or synthesized voices in deepfake videos.

Techniques: Spectrogram Analysis,Waveform & Pitch Analysis, Lip Sync Analysis
Strengths: Useful for AI-generated voice deepfakes.
Weaknesses: Requires clear audio; background noise can affect accuracy.

4. Physiological Signal-Based Detection

Detects deepfakes using subtle biological signals.

Techniques: Heartbeat Detection,**Eye Blink Rate,Head Pose & Eye Gaze Tracking**

Strengths: Hard for deepfake creators to replicate.
Weaknesses: Requires high-quality videos; sensitive to lighting conditions.

5. Blockchain & Metadata-Based Detection

Tracks **video authenticity** using cryptographic verification.

Techniques:

Blockchain Timestamping,Metadata Analysis,Watermarking: Embeds digital fingerprints into real videos to detect tampering.

Strengths: Useful for verifying original content.
Weaknesses: Doesn’t work for detecting already-existing deepfakes.

Problem Statement

- Rapid advancement of deepfake technology
- Significant risks in misinformation, identity fraud, and digital security.

AIM : To implement and evaluate **multimedia deepfake detection model**

- focus primarily on **image-based detection** using deep learning techniques.
- **benchmark** these models on publicly available deepfake datasets.
- **Try to develop a RNN/ViT based approach** while trying to maintain computational efficiency.

Project Members

Name	RollNo	Papers Read
Divyanshu	241110023	- Deep fake detection: current challenges and next steps - Unmasking DeepFakes with simple Features
Khushwant	241110035	- Mastering Deepfake Detection: A Cutting-edge Approach to Distinguish GAN and Diffusion-model Images - Deepfake detection using convolutional vision transformers and convolutional neural networks
Krishanu	241110037	- Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks - Wavelet-Driven Generalizable Framework for Deepfake Face Forgery Detection
Rajan Kumar	241110087	- Unmasking deepfake faces from videos using cost sensitive appch. - Video face manipulation through CNNS
Rishit	241110056	- DiffusionFake: Enhancing Generalization in Deepfake Detection via Guided Stable Diffusion - Robustness and Generalizability of Deepfake Detection: A Study with Diffusion Models
Senthil Ganesh	241110089	-Towards Solving DeepFake Problem: Improving DeepFake Detection using Dynamic Face Augmentation -DeepFake Detection Method based on Face Edge Bands

DEEP FAKE DETECTION: CURRENT CHALLENGES AND NEXT STEPS

- **There are three major types of DeepFake images/videos**
  - Head Puppetry :
    - Controls a target person head movements using a source person expressions
    - Target appears to mimic the source facial behavior.
  - Face swapping :
    - The target person face is replaced with the source person face while keeping the source person's facial expressions and movements same.
  - Lip syncing :
    - Modifies only the lips of the target person to match new speech.
    - It seem like people are speaking words they didn't actually say.

- **Current Deep Fake Detection methods**
  - Mostly target face swapping videos/images
  - Many of the existing methods are formulated as frame-level binary classification problems.

contd.

- **Deep fake detection method falls in 3 major categories**
  - Physical Inconsistencies-Based Methods
    - Analyzing inconsistencies in human physical and physiological behaviors.
    - Eye Blinking Analysis, Head Pose Analysis, Facial Landmark Patterns.
  - Signal-Level Artifact-Based Methods
    - Detects inconsistencies caused during the video synthesis process.
    - Splicing Artifacts Detection: Identifies blending errors when the fake face is merged with the real video. It uses DNN splicing detection methods
  - Data-Driven Deep Learning Methods
    - Uses DNN trained on both real and DeepFake videos to automatically detect fakes.
    - CNN-Based Models, & LSTM Models.
- **Limitations**
  - Most DeepFake detection methods works on frame-level binary classification problems. There are 2 major issues with this method.
  - Lack of Temporal Consistency Analysis
    - Inconsistencies appear across frames over time.
    - Frame-by-frame classification does not consider the sequence of frames, so it might miss these inconsistencies.

contd.

Extra Step Needed for Video-Level Detection

- Since detection happens at the frame level, to decide if an entire video is fake, we need to combine predictions from multiple frames
- This requires an aggregation method, which adds complexity and may reduce accuracy.

- **Improvements**
  - Use models that consider multiple frames together rather than treating them independently.
  - Detect unnatural movement patterns in face and background across consecutive frames.
  - Face-swapping DeepFake videos are relatively easier to detect. Our focus will be on detecting two other forms of DeepFake: head puppetry and lip-syncing.

Unmasking DeepFakes with simple Features

- **Proposed Method**
  - It proposes a feature-based method for DeepFake detection using frequency domain analysis.
  - Step 1: Convert Image to Frequency Domain
    - Uses Discrete Fourier Transform (DFT) to analyze high-frequency artifacts.
  - Step 2: Extract Features using Azimuthal Average
    - Converts 2D frequency data into 1D power spectrum for simplicity.
  - Step 3: Classify Real vs Fake Faces
    - Uses simple machine learning classifiers (Logistic Regression, SVM, K-Means).
- **Why Frequency-Based Approach Works?**
  - GAN-generated images have visible artifacts in the frequency domain.
  - Unlike deep learning models, this method works with very few training samples.
- **Limitations of Current DeepFake Detection Methods**
  - Deep Learning-Based Approaches
    - Require large labeled datasets.
    - Can be tricked by advanced Generative Adversarial Network.

contd.

- Frame-Level Classification Issues
  - Ignores temporal consistency in DeepFake videos
  - Needs extra processing for video-level classification.

- **Improvements**
  - Improve detection of low-resolution DeepFakes.
  - Combine frequency-based analysis with deep learning models.
  - Detect DeepFake videos that use realistic voices by combining both video and audio generation in a single tool.

Guarnera et al., 2024 - Mastering Deepfake Detection: A Cutting-edge Approach to Distinguish GAN and Diffusion-model Images

The proposed solution in paper is capable of recognizing whether an image was generated using 9 different GAN engines and 4 diffusion models (DMs) by means of a hierarchical approach.

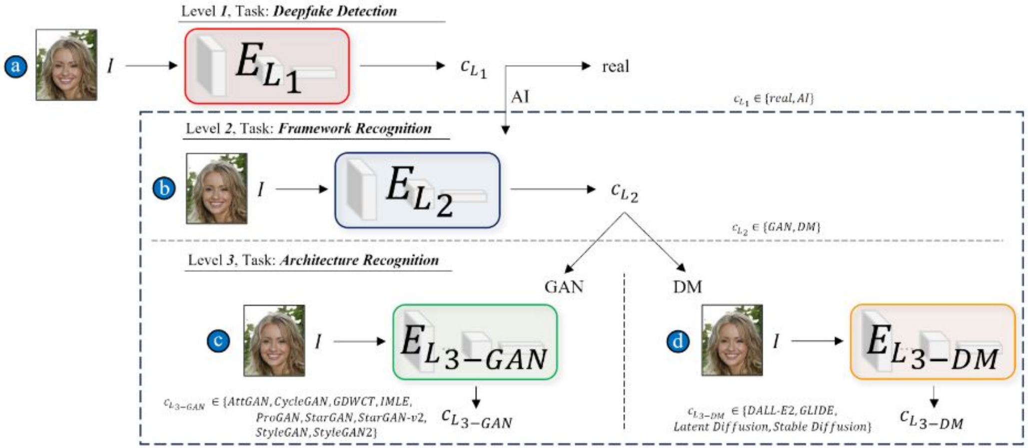
<u>Dataset Overview</u>: 83,000 images: 41,500 real, 41,500 AI-generated using 9 GAN models (2500 images each) and 4 Diffusion Models (5000 images each) resulting into a 14 class labelled dataset (real, 9GANs, 4DMs). A DFT β-statistics spectrum analysis is done to check the shared patterns among models of same category.

Best non-CNN architectures for 14 class flat classification : 95.71% , 13 class flat classification : 96.2%.

Classification accuracy ResNET-101 :

L1 : 98.93%
L2 : 98.45%
L3-GAN : 97.01%
L3-DM : 99.37%

Overall : 97.82%



Hierarchical Multi-level Approach:

<u>Level 1</u>: Classify real vs. AI-generated images.
<u>Level 2</u>: Distinguish between GAN-generated vs. DM-generated images.
<u>Level 3</u>: Identify the specific architecture used to generate the fake image.

The final architecture is defined by four models: E_{L1}, E_{L2}, E_{L3-GAN}, E_{L3-DM}.

ResNET-101 gives the best overall accuracy among different complex models.

https://dl.acm.org/doi/10.1145/3652027

…contd.

Classification accuracy of the whole hierarchical approach :

- If the image from level 1 is misclassified to the AI class, then this error will be counted 3 times. If the image is misclassified to the real class, as previously described, then the error will be counted only once.
- If the image is misclassified from level 2, then this error will be counted twice.
- If the image is misclassified by level 3, then the error will be counted only once.

Robustness and Generalization :

- Robust to JPEG compression, resize attacks, 3x3 Gaussian Blur
- Not Robust to rotation operation and 9x9 Gaussian Blur
- Generalized well on COCOfake deepfake images dataset (All 3 levels can be used)
- Generalized well on FaceForensics++ deepfake video dataset (Only first 10 frames considered per video) , even though the model was never trained on videos.
- Videos are encoded differently than images and different technologies are used for creating deepfake videos. So only L1 can be used for videos.

Cons :

- As the number of generative architectures (GANs and DMs) increases, the classification performance degrades.
- Videos are encoded differently than images. β-statistics extracted from images turn out to be different in videos.so the method could also achieve much lower accuracy values
- The models in this work, on the same ResNET−101 architecture. A combination of the different models can be explored.

Soudy et al., 2024 - Deepfake detection using convolutional vision transformers and convolutional neural networks

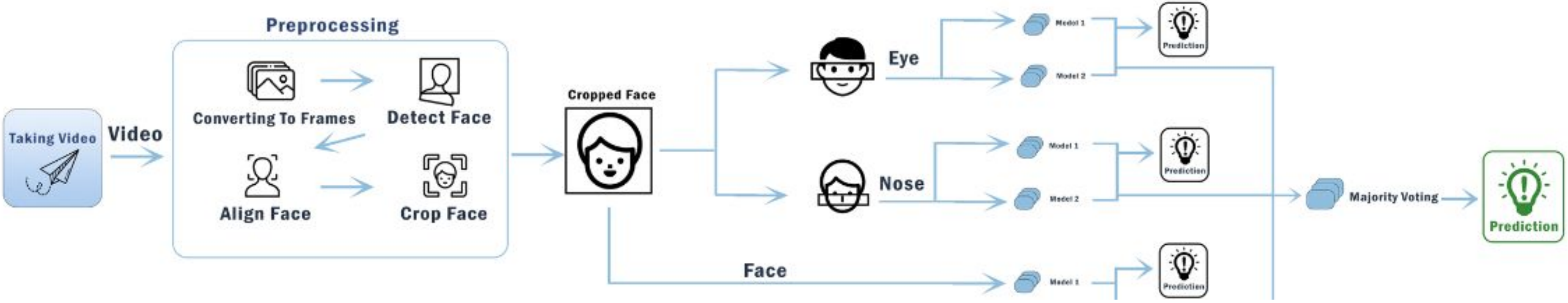
<u>Datasets Used:</u> FaceForensics++ and Deep Fake Detection Challenge (DFDC)

Three main components: Preprocessing, Detection, Prediction

<u>Preprocessing</u>
- Video frame extraction and enhancement
- Face detection using MTCNN
- Face alignment and cropping
- Eye and nose region extraction

<u>Detection Models</u>
- Model A (Deep CNN for Nose) **(97.4%)**
- Model B (Simplified CNN for Eye) **(97%)**
- Model C (CNN + Vision Transformer for Full Face) **(85%)**

<u>Prediction</u>
- Majority voting approach
- Combines results from all three models
- Enhances accuracy and robustness of detection



…contd.

Parameters and Experimentation:

Model A (100 epochs)
- 12 layers (3 convolutional blocks)
- ReLU activation, batch normalization, max pooling, dropout
- Trained on 50x50 pixel images of eye/nose regions
- Eye Region : 95.72% accuracy
- **Nose Region : 97.4% accuracy**

Model B (150 epochs)
- 6 layers (3 convolutional blocks)
- ReLU activation, max pooling, dropout
- Trained on same eye/nose data as Model A
- **Eye Region : 96.98% accuracy**
- Nose Region : 96.76% accuracy

Model C (CViT)
- 224x224 pixel input images
- 17 convolutional layers for 7x7x512 patches
- Embed each patch into a vector
- Pass to Transformer Encoder of depth 6
- 2048 dim MLP head to Softmax for classification
- **Best result: 85% test accuracy (Exp 6)**

References	Algorithms	Features used	Accuracy
Our Proposed	CNNs and CViT	Entire face, Eyes, Nose	97% and 85%
CViT-based Technique [30]	CViT	Entire face	69%
CNN-based Technique [32]	CNN	Eye Region	90%

Cons : - High computational resource requirements
- Not effective in detecting deepfakes that involve changes in parts of the face other than the eyes, nose, and entire face.

https://link.springer.com/article/10.1007/s00521-024-10181-7

<u>Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks</u>
- Romeo Lanzino et al., 2024

- A novel deepfake detection approach on RGB images using Binary Neural Networks(BNNs),FFT and LBP for faster inference and minimal accuracy loss.
- Author believes that modern deepfake detecting algorithms involve training complex neural networks with millions of parameters which cant run on compact hardwares like smartphones.
- Proposed model achieved sota performance on datasets like COCOFake, DMFD, CIFAKE.
- **What are BNNs?**
    - Binarizing weights and activation functions using a sign function.
    - Replacing majority of arithmetic operations with bitwise operations
    - Theoritical speedup of 58x in inference time, 32x less memory
    - Can convert any CNN to a BNN(after handling the quantization loss)
    - Proposed model uses BNext as a backbone which is pretrained on ImageNet dataset, it consists of a binary convolution module with full precision skip connections and a branch with INT-4 precision for information propagation

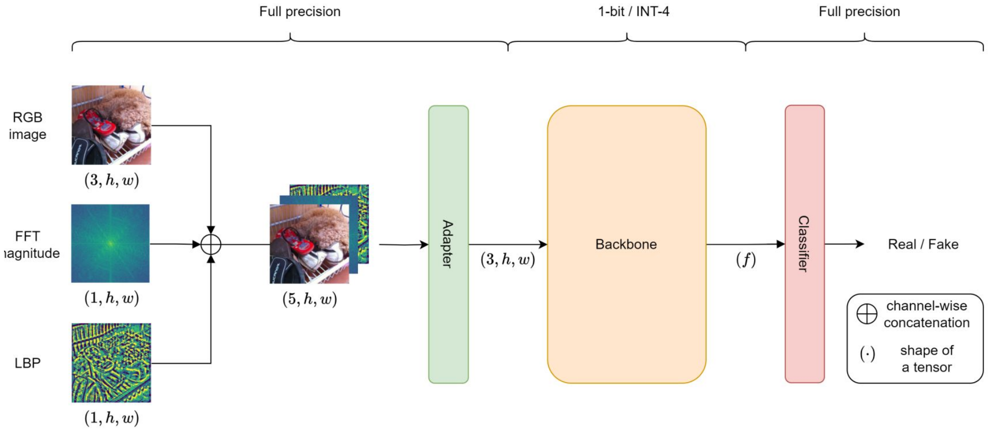
$$sign(x_r) = \begin{cases} +1, & \text{if } x_r \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

$$\mathcal{Y} = \mathcal{A}_r \otimes \mathcal{W}_r \approx (\mathcal{A}_b \circledast \mathcal{W}_b) \odot \alpha,$$

- **Role of Fast Fourier Transform and Local Binary Pattern**
    - Deepfake generation introduces minute distortions in frequency domain. Thus introduce FFT channel
    - LBP is a texture descriptor which captures the unique textures of facial features, another channel

Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks
- Romeo Lanzino et al., 2024

- **Key layers**
  - Adapter layer - Convolution layer to convert 5 layers to 3 layers
  - Backbone layer - BNext model giving a tensor of $\{-1,+1\}^f$
  - Classifier layer - Full precision linear layer



- **Results on COCOFake and DFFD**

Model	Pre-training dataset	Accuracy	AUC	Parameters (M)	FLOPs (G)
ResNet50	ImageNet	90.31	-	25.6	4.8
ViT-B/32	ImageNet	87.64	-	88.3	8.56
CLIP-ResNet50	OpenAI WIT	99.07	-	25.6	4.8
CLIP-ViT-B/32	OpenAI WIT	99.11	-	88.3	8.56
OpenCLIP-ViT-B/32	LAION-400M	97.88	-	88.3	8.56
OpenCLIP-ViT-B/32	LAION-2B	**99.68**	-	88.3	8.56
BNext-T with frozen backbone	ImageNet	83.65	81.98	29.8	**0.89**
BNext-S with frozen backbone	ImageNet	93.15	95.19	67.1	1.91
BNext-M with frozen backbone	ImageNet	84.59	82.11	133	3.39
BNext-T	ImageNet	99.25	99.86	29.8	**0.89**
BNext-S	ImageNet	99.28	99.89	67.1	1.91
BNext-M	ImageNet	99.18	**99.91**	133	3.39

Model	Accuracy	AUC	Parameters (M)	FLOPs (G)
Xception	-	99.64	40.0	18.0
VGG16	-	99.67	138.4	15.5
BNext-T with frozen backbone	89.56	87.65	29.8	**0.89**
BNext-S with frozen backbone	89.69	88.58	67.1	1.91
BNext-M with frozen backbone	89.61	86.64	133	3.39
BNext-T	98.95	**99.94**	29.8	**0.89**
BNext-S	**99.01**	**99.94**	67.1	1.91
BNext-M	98.75	99.92	133	3.39

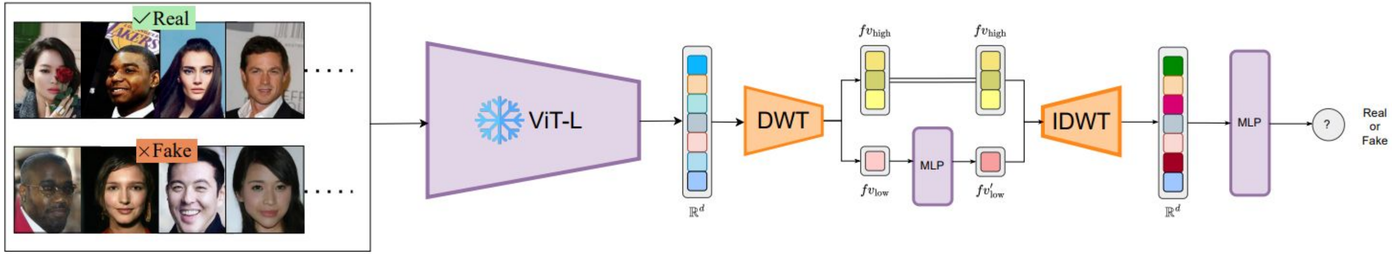
Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks
- Romeo Lanzino et al., 2024

- **Result on CIFAKE –>**
- **Conclusion**
  - Requires up to 5 times fewer FLOPs compared to a ResNet-50 model and nearly 10 times fewer FLOPs than a ViT-B/32 model
  - Capable of matching the performance of their full-precision counterparts with minimal loss in classification accuracy
- **Con**
  - FLOPs reductions are purely theoretical as we need specialized hardware and accelerators to realise the speedups
  - Only trained on ImageNet for pretraining

Model	Accuracy	AUC	Parameters (M)	FLOPs (G)
ResNet-50	95.00	99.00	25.6	4.8
VGG	96.00	99.00	133	7.63
DenseNet	**98.00**	99.00	7.9	5.6
BNext-T with frozen backbone	83.89	91.70	29.8	**0.89**
BNext-S with frozen backbone	80.71	89.25	67.1	1.91
BNext-M with frozen backbone	82.77	90.73	133	3.39
BNext-T	97.29	**99.65**	29.8	**0.89**
BNext-S	96.96	99.55	67.1	1.91
BNext-M	97.35	99.62	133	3.39

Wavelet-Driven Generalizable Framework for Deepfake Face Forgery Detection
- Lalith Bharadwaj Baru et al., 2024

- Modern deepfake detection techniques excel only under same generational family. They fail to generalize well to unseen generations(Because they learn low level artifacts unique to training model).
- Algorithms which uses frequency based statistics generalize well on different generations but fail when training and testing data come from different distributions
- **Authors' approach**
    - Dont explicitly train real vs fake classifiers, instead use feature space of a pretrained vision language model like CLIP-ViT(totally unrelated to our problem domain)
    - Apply Wavelet transformations on extracted features to split features intro frequency components
    - Low frequency components capture granular and nuanced features while high frequency components capture the sharp features. Low frequency components are valuable to us.
    - Pass the low frequency features through an MLP to capture and learn granularity.



Wavelet-Driven Generalizable Framework for Deepfake Face Forgery Detection
- Lalith Bharadwaj Baru et al., 2024

Algorithm 1 Wavelet-CLIP

Input: DATASET $\mathcal{D}$, ENCODER $Enc_{\phi}^{(\text{ViT})}(.)$, ϵ, n;
for ITERATIONS $= 1$ **to** ϵ **do**
  for BATCH $= n$ **do**
    $Z^{(n)} = Enc_{\phi}^{(\text{ViT})}(x^{(n)})$
    $fv_{\text{low}}^{(n)}, fv_{\text{high}}^{(n)} = \text{DWT}(Z^{(n)})$
    $fv'^{(n)}_{\text{low}} = \text{MLP}(fv_{\text{low}}^{(n)})$
    $Z_{\text{new}}^{(n)} = \text{IDWT}([fv'^{(n)}_{\text{low}}, fv_{\text{high}}^{(n)}])$
    $\text{cls}_n = \text{MLP}(Z_{\text{new}}^{(n)})$
  end for
end for
return cls_n

Dataset Name	Train/Test	No. of Samples
FaceForensics++ [30]	Train	114884
Celeb-DF v1 (CDFv1) [23]	Test	3136
Celeb-DF v2 (CDFv2) [23]	Test	16420
FaceShifter (Fsh) [9]	Test	8958

Models	Venue	Backbone	Protocol	CDFv1	CDFv2	Fsh	Avg.
MesoNet [1]	WIFS-18	Custom CNN	Supervised	0.735	0.609	0.566	0.636
MesoInception [1]	WIFS-18	Inception	Supervised	0.736	0.696	0.643	0.692
EfficentNet [32]	ICML-19	EfficentNet B4 [32]	Supervised	0.790	0.748	0.616	0.718
Xception [3]	ICCV-19	Xception	Supervised	0.779	0.736	0.624	0.713
Capsule [26]	ICASSP-19	CapsuleNet [31]	Supervised	0.790	0.747	0.646	0.728
DSP-FWA [22]	CVPR-19	Xception [3]	Supervised	0.789	0.668	0.555	0.677
CNN-Aug [16]	CVPR-20	ResNet50 [19]	Supervised	0.742	0.702	0.598	0.681
FaceX-ray [21]	CVPR-20	HRNet [21]	Supervised	0.709	0.678	0.655	0.681
FFD [5]	CVPR-20	Xception [3]	Supervised	0.784	0.7435	0.605	0.711
F^3-Net [12]	ECCV-20	Xception [3]	Supervised	0.776	0.735	0.591	0.700
SRM [10]	CVPR-21	Xception [3]	Supervised	**0.792**	0.755	0.601	0.716
CORE [27]	CVPR-22	Xception [3]	Supervised	0.779	0.743	0.603	0.708
RECCE [2]	CVPR-22	Custom CNN	Supervised	0.767	0.731	0.609	0.702
UCF [17]	ICCV-23	Xception [3]	Supervised	0.779	0.752	0.646	0.725
CLIP [11]	CVPR-23	ViT [7]	Self-Supervised	0.743	0.750	0.730	0.741
Wavelet-CLIP (ours)	-	ViT [7]	Self-Supervised	0.756	**0.759**	**0.732**	**0.749**

DiffusionFake: Enhancing Generalization in Deepfake Detection via Guided Stable Diffusion

Key Idea:

- Instead of detecting superficial **artifacts**, DiffusionFake focuses on **identity-level feature separation**.
- Uses **Stable Diffusion** to enhance deepfake feature disentanglement.

Step-by-Step Process:

1. **Feature Extraction (CNN-Based Encoder)**
   a. Extracts latent features from a deepfake image x_f as $f=E(x_f)$
2. **Feature Filtering Module (FFM)**
   a. Splits features into:
      i. f_s → **Source features** (motion, expression).
      ii. f_t → **Target features** (identity, texture).
      iii. $f_s=F_s(f)$, $f_t=F_t(f)$ where F_s and F_t are filtering networks.
3. **Weight Module (**Involves weight module loss L_{ws} + L_{wt} **) :** The Weight Module ($W_s(f)$,$W_t(f)$) determines the importance of source vs. target features
4. **Stable Diffusion for Feature Learning** (Involves Reconstruction Loss L_s , L_t)
   a. Uses **frozen Stable Diffusion** to ensure f_s and f_t contain meaningful identity features.
   b. $x_s'=SD(f_s)$, $x_t'=SD(f_t)$
5. **Deepfake Classification**
   a. Classifier predicts **real or fake** using f_s and f_t .
   b. $\hat{Y}=\text{Classifier}(f_s,f_t)$

The total loss function combines: **Cross-Entropy Loss (L_{CE})** for classification. + **Reconstruction Loss (L_s, L_t)** for disentanglement + weight module loss.

- $\mathbf{L = L_{CE} + \lambda_s L_s + \lambda_t L_t + L_{ws} + L_{wt}}$
- λ_s λ_t are weighting factors (hyperparameters) for reconstruction loss.
- $\mathbf{L_{CE}}$ ensures correct **real vs. fake** classification.

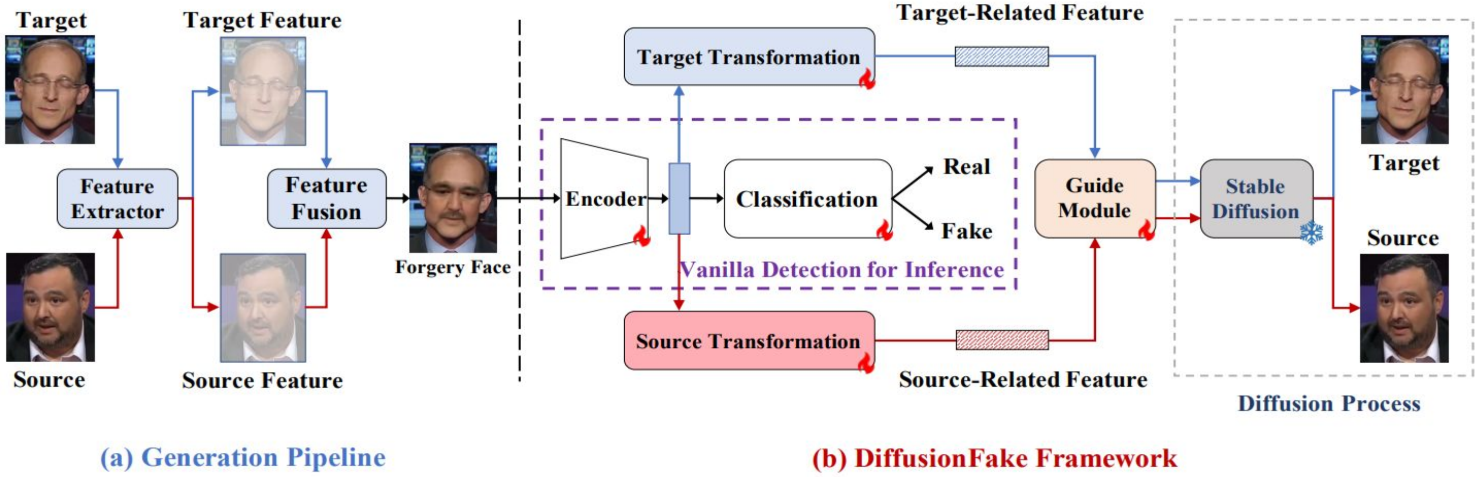


Figure 1: Pipeline of the generation process of Deepfake (a) and our proposed DiffusionFake (b).

Experimental Findings

- Trained on FF++ dataset, tested on Celeb-DF, DFDC-P, WildDeepfake, DFD, and DiffSwap
- Improvement of ~7-10% acc across all
- Stable Diffusion is crucial → Removing it drops acc by 6%.
- Feature Filter is important → Removing it drops acc by 5%.
- Weight Module is essential → Removing it drops acc by 3%.
- No additional parameters or computational overhead during inference.

Why is DiffusionFake More Effective?

CNN-based methods fail on unseen deepfakes → They rely on dataset-specific artifacts.
DiffusionFake generalizes better → Uses identity-based feature separation.

Technical Improvements:

Method	Feature Extraction Approach	Performance on Unseen Deepfakes
CNN-Based Classifiers	Detects artifacts (pixel inconsistencies, blur)	Poor Generalization
Frequency-Based Methods	Detects manipulation in frequency domain	Limited to dataset-specific artifacts
DiffusionFake (Ours)	Disentangles source-target identity features using Stable Diffusion	Strong Generalization

Unmasking deepfake faces from videos using an explainable cost sensitve Deep learning approach (2023)

- Dataset used : Face Forensics++ and Celeb DF V2- 80/10/10(train/test/validate)
- Preprocessing: deletion of corrupt file,face recongnition package to find frames with faces, extract key frames, resize frames- 224x224 and 30 fps.
- Adjustment of cost of datasets- for less sample class
- Four pretrained models (Xceptionnet, inceptionResNet v2, EfficientNetV2s and v2M
- Model training : initial learning rate - .001, lowered if model performed poorely

  Batch size - 16

  Optimization technique - adam optimization

  Global average pooling

  Relu and softmax activation

  Dropout rate- 0.5 to prevent overfitting

Unmasking deepfake faces from videos using an explainable cost sensitve Deep learning approach(2023)

Performance matrix

Model	Accuracy	Precision	Recall	F1-Score
XceptionNet	**98%**	0.98	0.98	0.98
EfficientNetV2S	97%	0.97	0.97	0.97
EfficientNetV2M	97%	0.97	0.97	0.97
InceptionResNetV2	97%	0.97	0.97	0.97

Novelty :

key frame extraction : measure difference between frames rather than frame by frame

cost sensitive neural networks: higher weight to lower class.

Video Face Manipulation Detection Through CNNs(2020)

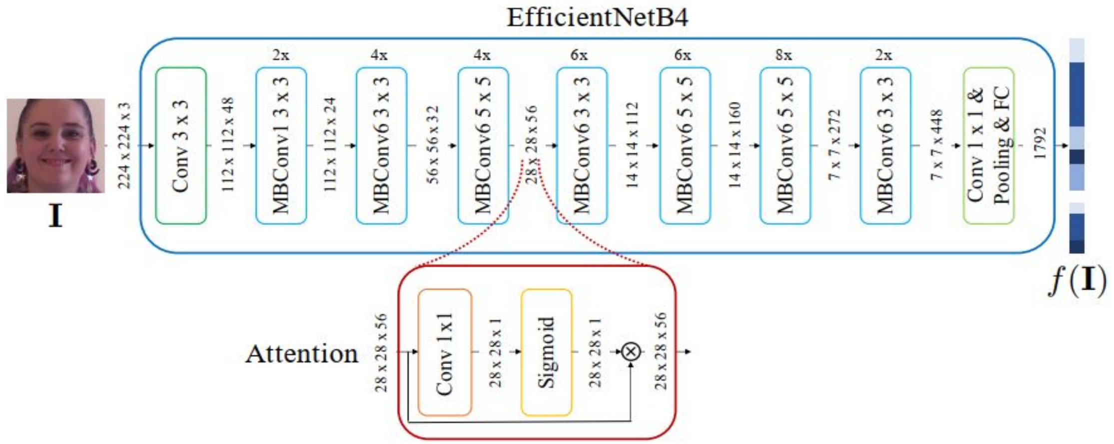
Objective:

Detecting manipulated facial videos (deepfakes, FaceSwap, etc.) using embedding of attention layers Convolutional Neural Networks (CNNs).

Methodology: Operate on a small part of video, non reversible operations leaves peculiar footprint that exposes editing

Attention layer based CNN:

- Uses EfficientNetB4 as a base model : - ImageNet dataset shows an efficiency of 83%, Xceptionnet has 79%
- Incorporates attention layers to highlight key facial regions.



Video Face Manipulation Detection Through CNNs(2020)

Datasets Used:

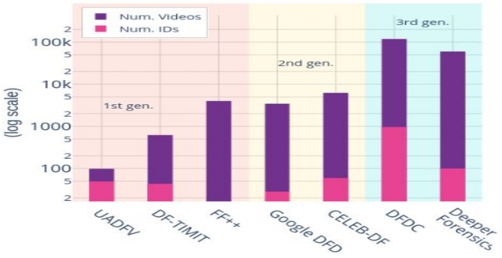
- **FaceForensics++ (FF++)**: Contains 4000 manipulated videos.
- **DeepFake Detection Challenge (DFDC)**: Over 119,000 videos.
-

Results & Performance:

- The attention layer based model outperforms the baseline (XceptionNet).
- Future work is to incorporate temporal analysis.

Sowmen Das et al. *“Towards Solving DeepFake Problem: Improving DeepFake Detection using Dynamic Face Augmentation”*

- Overcomes the problem of overfitting (due to oversampling) in datasets (Memorize)
- Deepfake generation - Variation Auto Encoders (VAE) and GANs
- Face clustering - unique subjects and no. of subjects
- Preprocessing steps to prevent data leak - Split the data based on face clusters

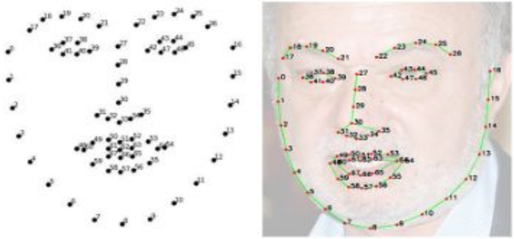


Face-Cutout (Dropout)

- Uses landmark positions to augment training images
- Calculate polygons for face- cutout
- Sensory group removal and convex hull removal

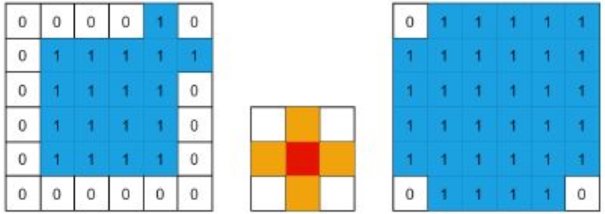
Experimental Setup and Results

- Model Selection - EfficientNet-B4 and XceptionNet
- LogLoss calculation for comparison
- Face-cutout augmentation outperforms

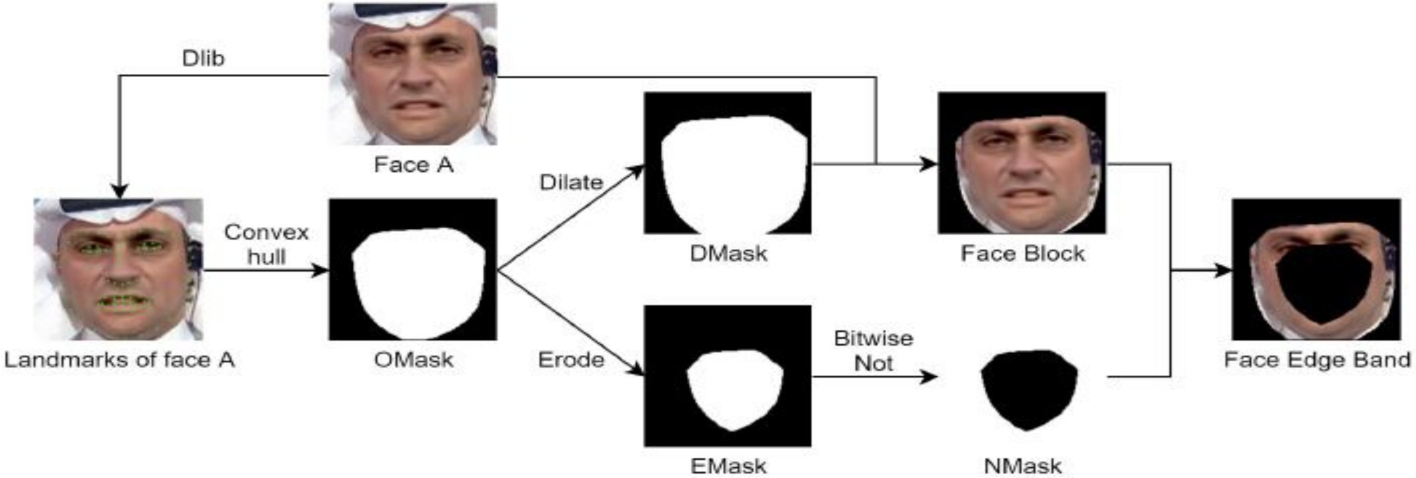


Zhengjie Deng et al. *"DeepFake Detection Method based on Face Edge Bands"*

- Synthetic forgery traces found at the edges of faces
- Uses only edge bands of faces for deepfake detection
- EfficientNet - B3 is the training network used
- Dilation, Erosion and Bitwise Not Algorithm
- Smaller no of pixels used for AUC of over 99.8%



a) Original imgae b) Structuring Element c) Dilated imgae



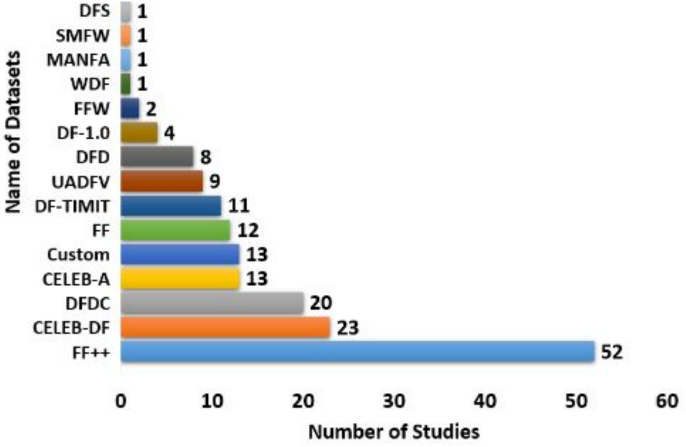
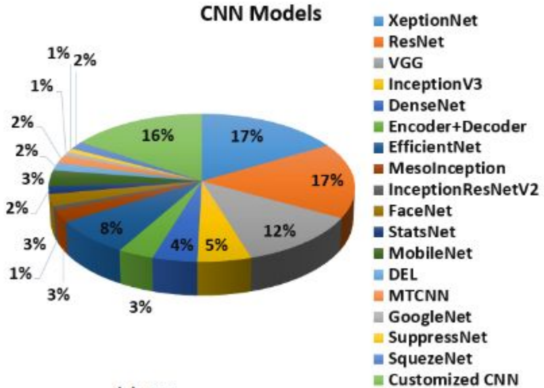


FIGURE 7. List of datasets used in Deepfake related studies.



(a) CNN

TABLE 5. Distribution of used models.

Category	Model	#Studies
Deep Learning	CNN	71
	RNN	12
	RCNN	2
Machine Learning	SVM	11
	k-MN	4
	LR	3
	MLP	3
	BOOST	2
	RF	1
	DT	1
	DA	1
	NB	1
	MIL	1
Statistical	EM	1
	TV, KL, JS	1

A Recurrent Neural Network (RNN) is a type of neural network designed to process sequential data. Unlike traditional feedforward neural networks, RNNs have loops that allow information to persist across timesteps.
Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs) are their more advanced variants.

Expected Outcome and deliverables

- Taking computational capacity into account, we will design a Streamlit-based frontend for the detection of Deep Fake images and videos.

- **Phase 1: Deep Fake Image Detection**
  - Utilize a CNN and train a classifier on the publicly available 140k Real and Fake Faces dataset from Kaggle. (DenseNET/VGGFace/XceptionNet/Custom CNN Architectures) and provide Heatmap Visualization.
  - Conduct robustness analysis to further evaluate the model.

- **Phase 2: Deep Fake Video Detection**
  - Initially employ the mini_face_forensics dataset from Kaggle for model training and evaluation.
  - Explore potential model architectures:
    - CNN + LSTM/GRU combination.
    - ViTs, such as CViT/Swin Transformer/FasterViT
    - Model Ensembling
  - Extract features using state-of-the-art pretrained models such as InceptionV3 or XceptionNet for transfer learning. The pretrained model will be used to obtain a feature vector, further the LSTM/ViT layers will be trained using these features and create a baseline model.

- Implementation Strategy:
  - Integrate various techniques from research papers studied to improve baseline model performance.
  - If feasible, apply the finalized model to a state-of-the-art Faceforensics++ dataset for further validation.
- Deliverables:
  - A web-based interface for Deep Fake detection.
  - Jupyter Notebooks documenting the implementation.

Timeline

Literature review	Ongoing
Dataset Acquisition	In Progress
Preprocessing	30 Mar
Training Models	5 APR
Trained models apply on test dataset, Robustness analysis etc	10 APR
Performance Analysis	16 APR

Presentation and code review as per alloted schedule


Thank you!