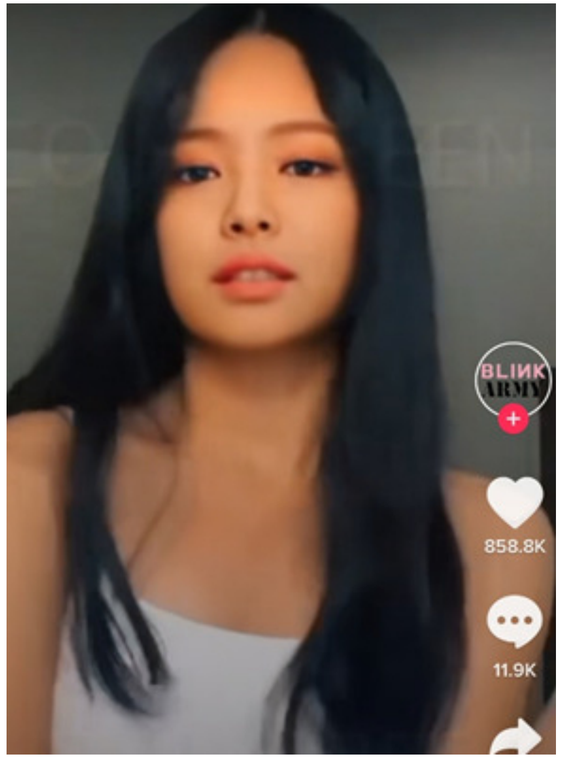
One Detector to Rule Them All

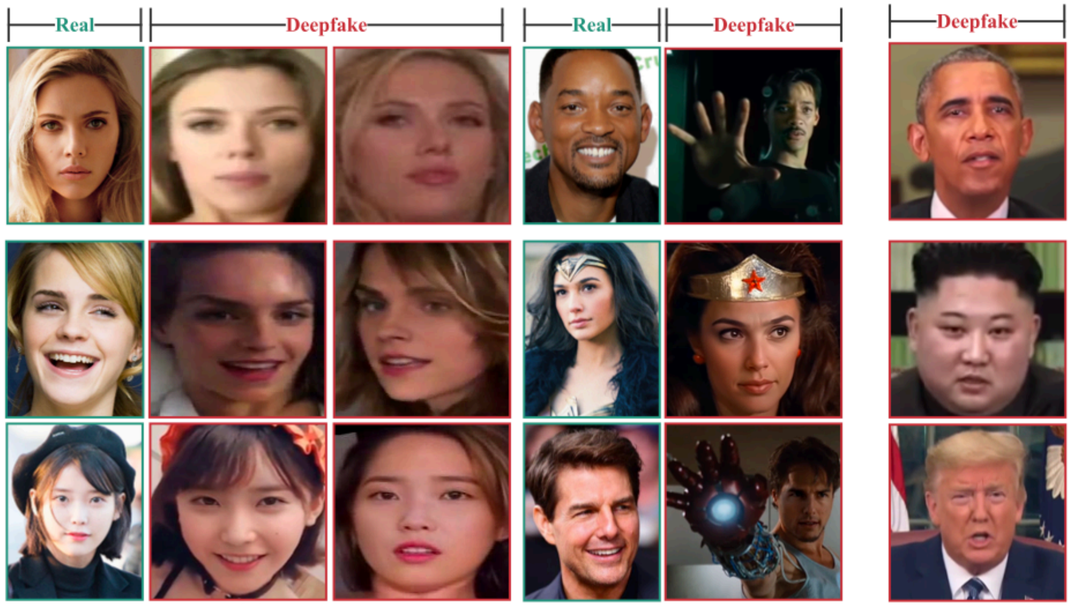
Towards a General Deepfake Attack Detection Framework

Seoul Woman’s University
Hyemin Lee

What are Deepfakes?



What are Deepfakes?



- Deepfake is used in many ways, while the number of potential abuse is increasing.
- A recent study claimed that 96% of the deepfakes originate from porn videos.
- Since the amount of deepfake datasets is small, “Detecting deepfakes” is very difficult !!

Deepfake Model & Dataset

DeepFake generation models

Identity Swap

- Some parts or the entire face of the target is swapped with source face.
- Deepfake Detection (DFD), Face Swap (FS), DeepFake Detection Challenge(DFDC)..

Facial Reenactment

- The source is used to drive the expression, gaze, mouth, and pose of the target.
- Face2Face (F2F), Neural Texture (NT)

Deepfake Model & Dataset

DeepFake generation models

- DeekFake (DF)
- FaceSwap (FS)
- Face2Face (F2F)
- Neural Texture (NT)
- DeepFake Detection (DFD)
- DeepFake Detection Challenge (DFDC)
- DeepFake-in-the-wild (DFW)

Deepfake Model & Dataset

Introduction DeepFake Models

- DeekFake (DF)
- FaceSwap (FS)
- Face2Face (F2F)
- Neural Texture (NT)
- DeepFake Detection (DFD)
- DeepFake Detection Challenge (DFDC)
- DeepFake-in-the-wild (DFW)

$$\mathcal{D}_{known} = \{DF, FS, F2F, NT, DFD\}$$

$$\mathcal{D}_{unknown} = \{DFW\}$$

Deepfake Model & Dataset

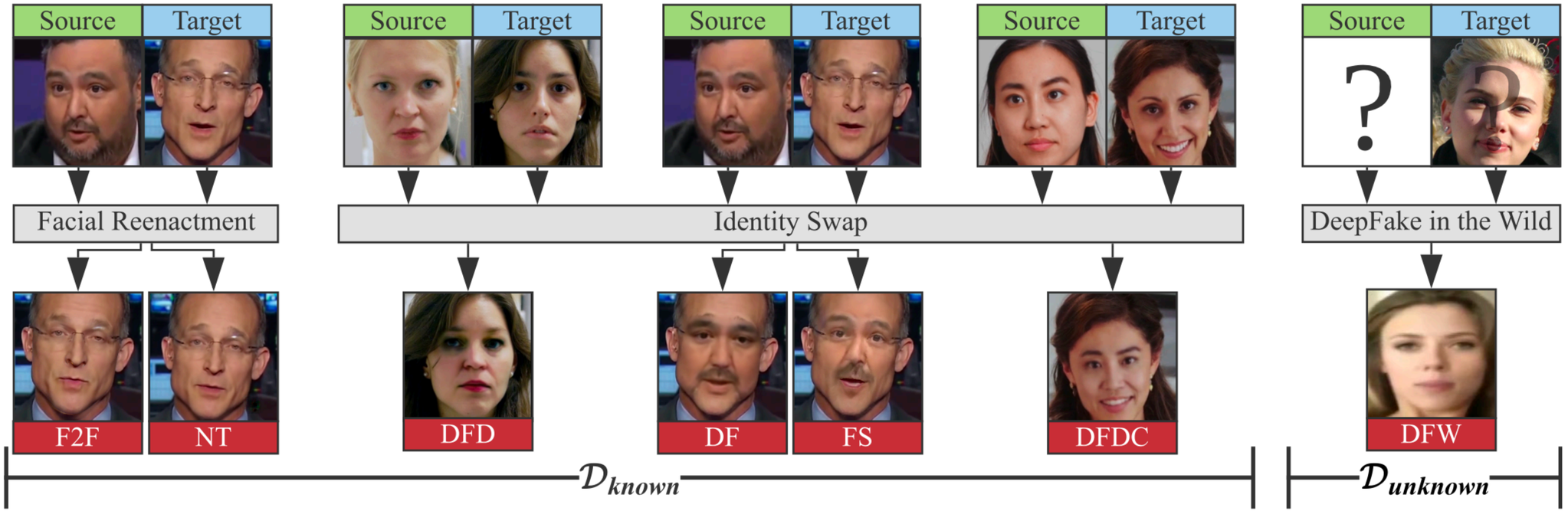


Figure 3: Different deepfake generation methods. Facial Reenactment and Identity Swap are general, high-level categories of deepfakes. The DeepFake-in-the-Wild contains deepfakes whose generation methods are unknown.

Deepfake Detection Method

- Attack Detection with Consecutive Frames

1. Capturing time information of a deep-fake video frame (CNN + RNN)

2. Recognizing temporal information (RNN) by extracting features from a continuous frame (CNN)

→ CLRNet model using Convolutional-LSTM cells

Deepfake Detection Method

- Detection Generalization via Transfer learning

1. It has **difficulties** in collecting and producing **new deepfake samples.**

2. **few-shot transfer learning(TL)** can be used for the detection of deepfakes created **using different methods.**

ex) Using model learned **in one domain** to another domain for **improving** the **generalizability.**

Threat Model

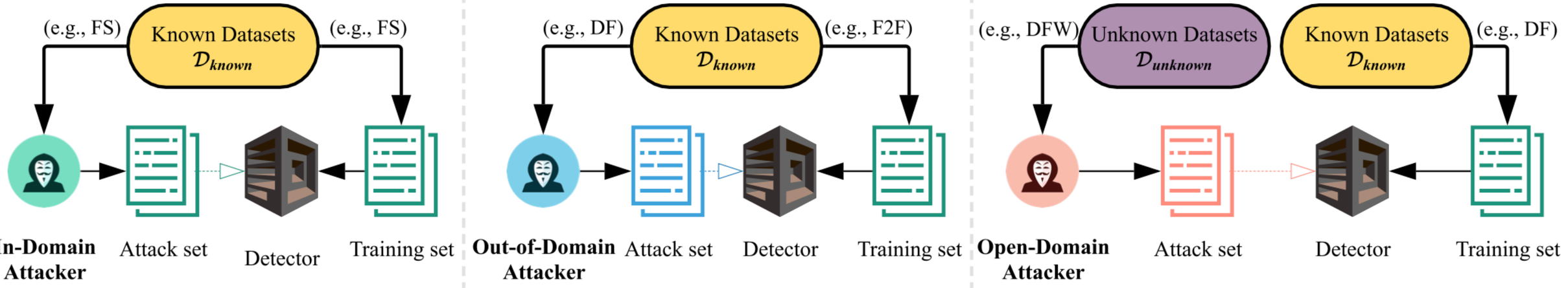


Figure 4: A visual representation of the threat model. We consider three types of attacks, each from an in-domain attacker (left), an out-of-domain attacker (middle) and an open-domain attacker (right).

Threat Model

Table 1: Attacker type and deepfake attack generation conditions for attack dataset ($\mathcal{D}_A$) and training dataset ($\mathcal{D}_T$).

Attacker Type	Attack Generation Conditions	Examples of Attack Dataset ($\mathcal{D}_A$)	Examples of Training Dataset ($\mathcal{D}_T$)
In-domain Attacker	$\mathcal{D}_A \subset \mathcal{D}_{known}$ $\mathcal{D}_T \subset \mathcal{D}_{known}$ $\mathcal{D}_A = \mathcal{D}_T$	$\mathcal{D}_A = \{DF\}$ $\mathcal{D}_A = \{FS\}$ $\mathcal{D}_A = \{NT\}$	$\mathcal{D}_T = \{DF\}$ $\mathcal{D}_T = \{FS\}$ $\mathcal{D}_T = \{NT\}$
Out-of-domain Attacker	$\mathcal{D}_A \subset \mathcal{D}_{known}$ $\mathcal{D}_T \subset \mathcal{D}_{known}$ $\mathcal{D}_T \subsetneq \mathcal{D}_A$	$\mathcal{D}_A = \{F2F\}$ $\mathcal{D}_A = \{F2F, NT, DFD\}$	$\mathcal{D}_T = \{DF\}$ $\mathcal{D}_T = \{DF, FS\}$ $\mathcal{D}_T = \{DF, FS\}$
Open-domain Attacker	$\mathcal{D}_A \subset \mathcal{D}_{unknown}$ $\mathcal{D}_T \subset \mathcal{D}_{known}$	$\mathcal{D}_A = \{DFW\}$	$\mathcal{D}_T = \{DF, FS\}$ $\mathcal{D}_T = \{FS, DF, DFD, NT, F2F\}$

1) In-domain Deepfake Attacker

- Same deep fake dataset/generation method as the training of detector.

2) Out-of-domain Deepfake Attacker

- Strictly different deep fake dataset/generation method than the training dataset of detector.

3) Open-domain Deepfake Attack

- A mix of known and unknown generation methods.

- The exact distribution is also unknown.

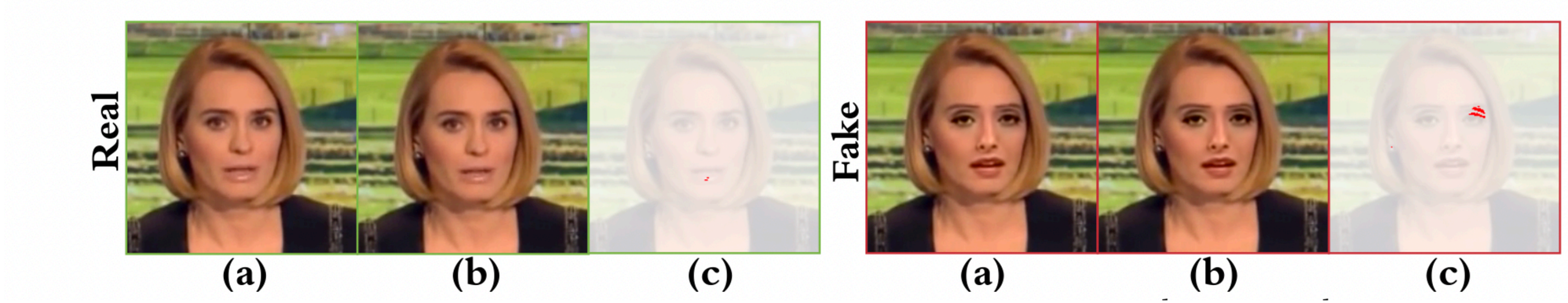
- More practical and real-life scenario.

Method For Generalization

1. **Exploring time information for deep fake detection.**
2. **CLRNet(Convolutional-LSTM based Residual Network) detailed architecture**
3. **Defense Strategy**

Method For Generalization

1. Exploring time information for deep fake detection - Problem

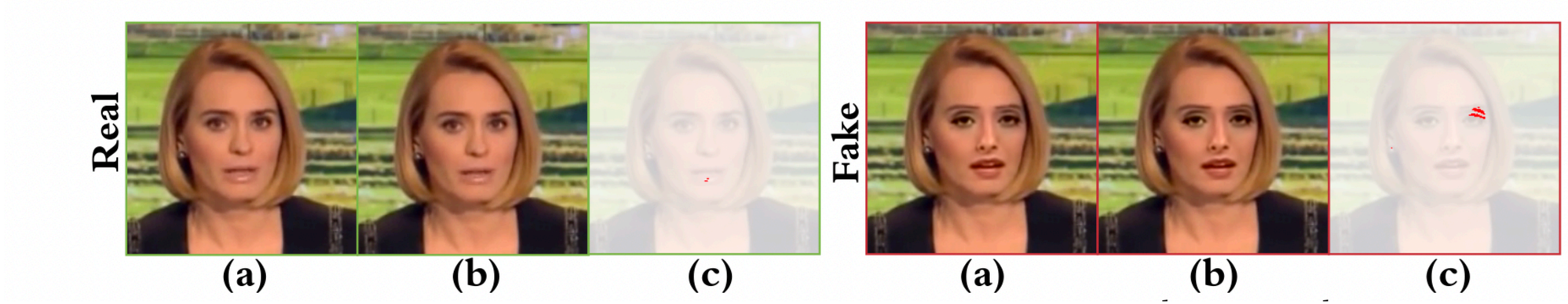


Recent **single frame-based** deepfake detection methods

-> Did not consider the relationship between frame and its **consecutive frame**.

Method For Generalization

1. Exploring time information for deep fake detection - Solution



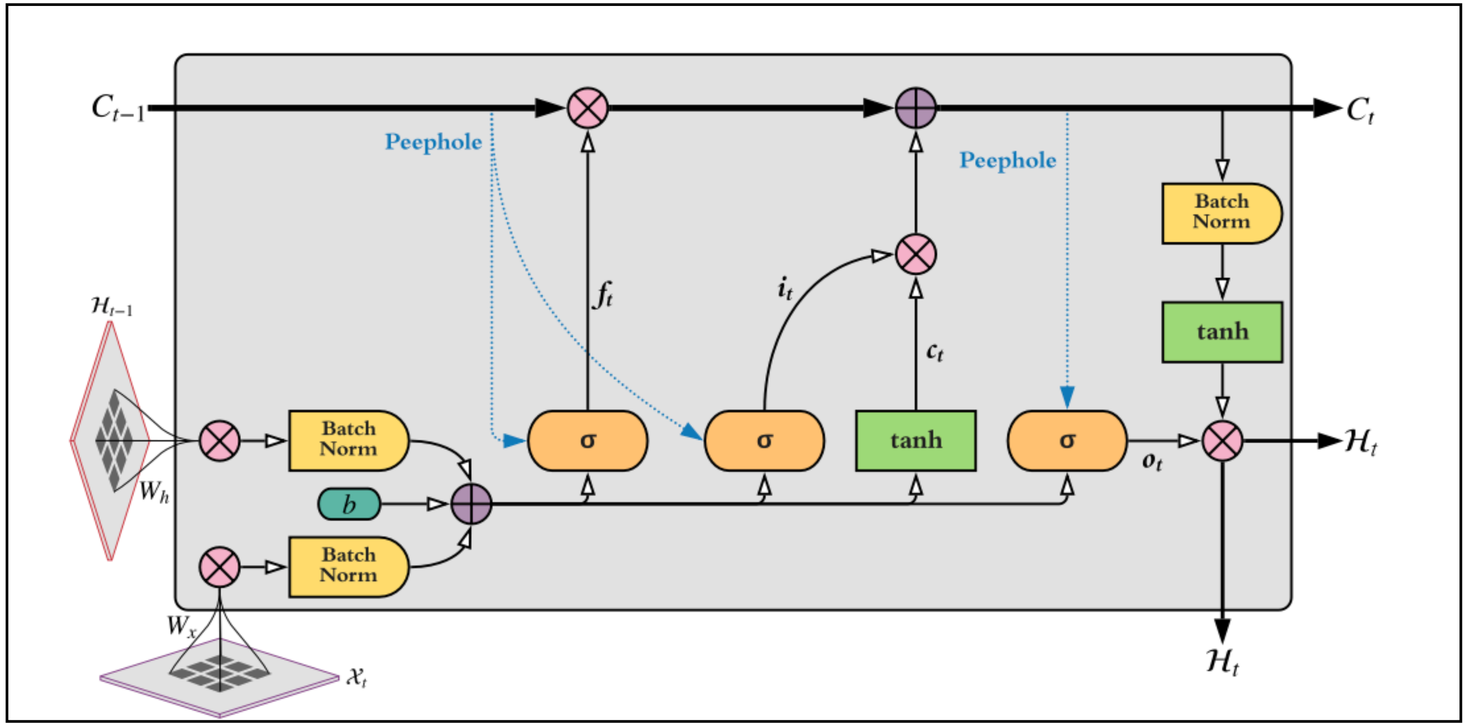
CLRNet can capture the problem better than single-frame in **dual-learning** aspect.

1) change in brightness and contrast on a small region of the face.
2) the size of some facial parts such as eyes, lips, and eyebrows changes between frames.

Method For Generalization

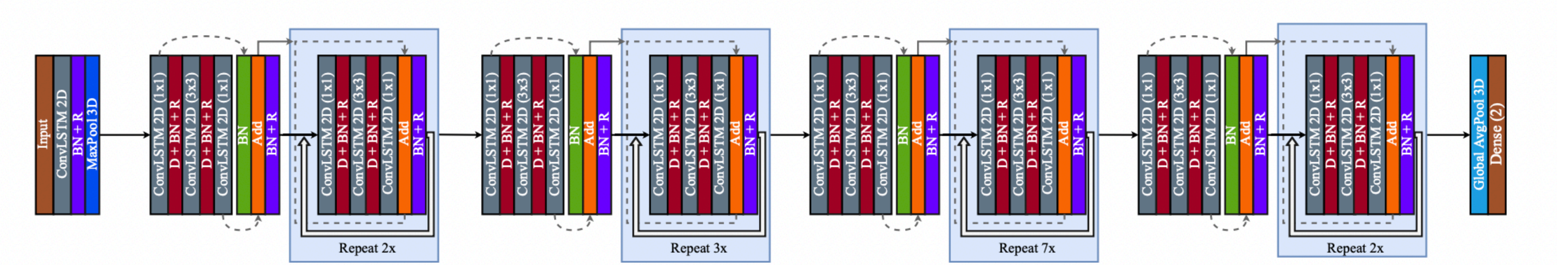
2. CLRNet (Convolutional LSTM Cell)

- Input : (X1, . . . , Xt)
- Cell Input : ($C1$, . . . , Ct)
- Hidden State : (H1, . . . , Ht)
- Gates : (it , ft , ot)



Method For Generalization

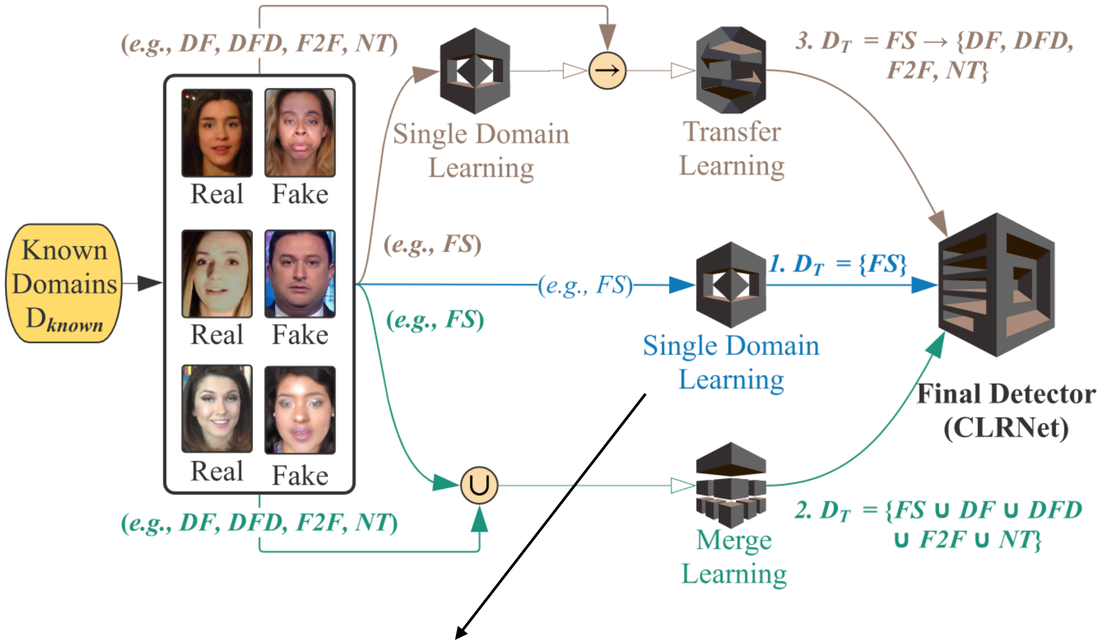
2. CLRNet (Convolutional LSTM Based Residual Network)



- **Input** : Sequence of consecutive images
- **Output** : Real or Fake (Classification Result)

Method For Generalization

3. Training Strategy

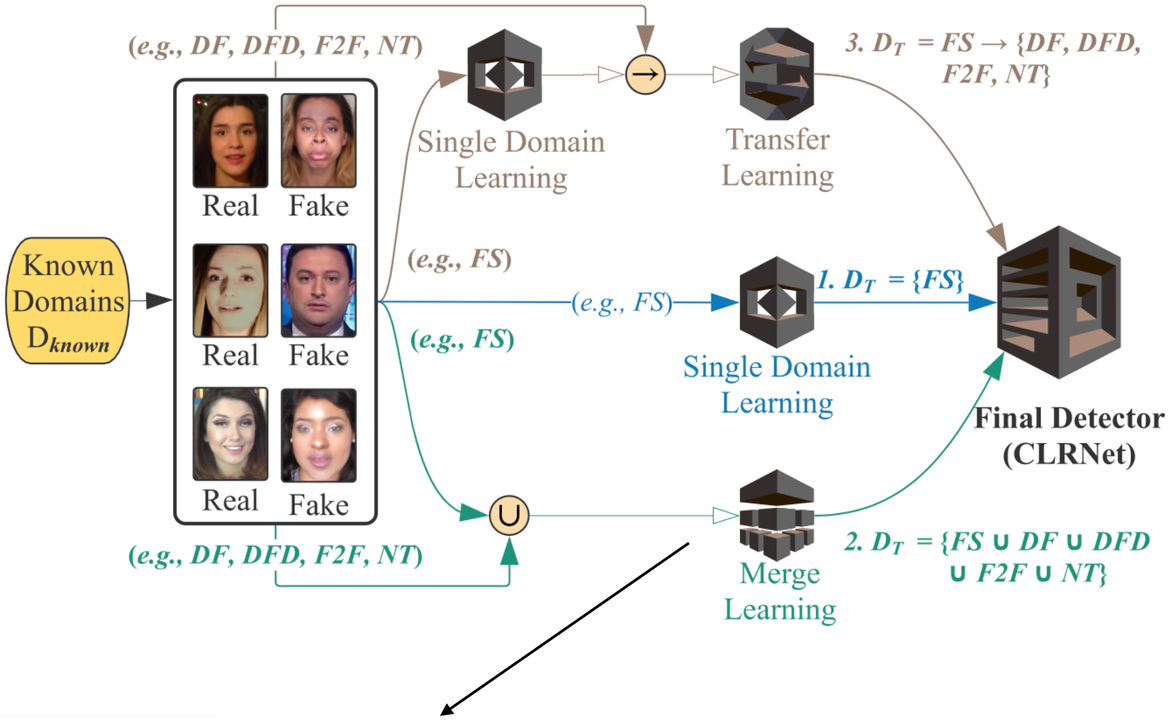


Single-Domain Learning

- It is only one training dataset (e.g., DT = {FS})

Method For Generalization

3. Training Strategy

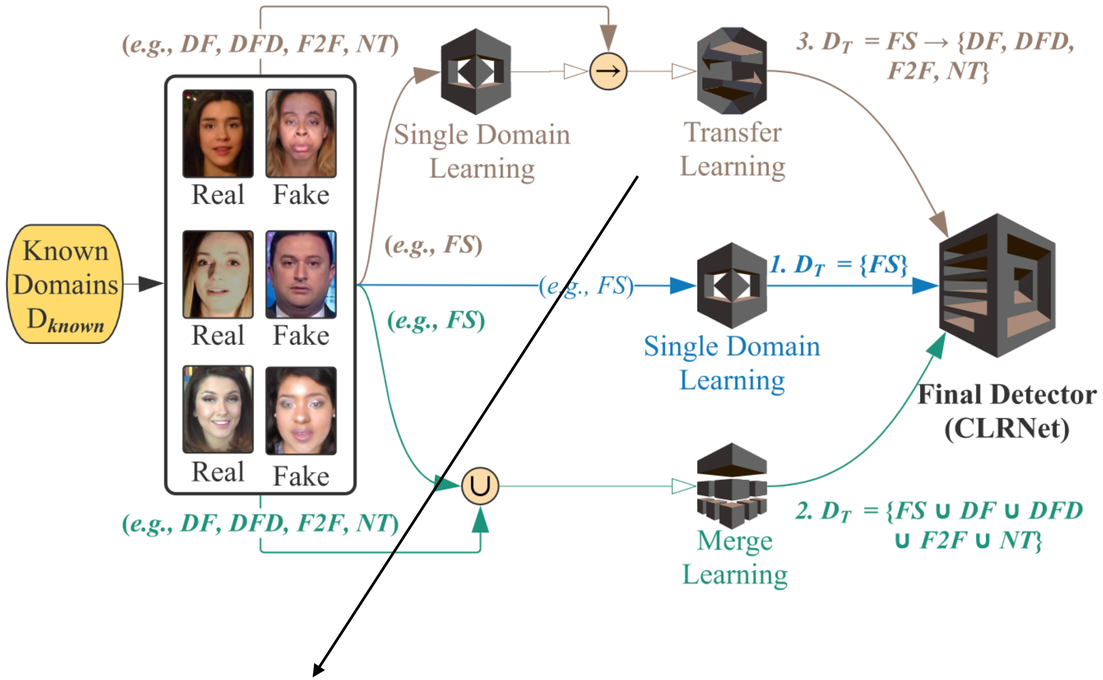


Merge Learning

- It aggregates two or more training datasets (e.g., D_r={FS,DF,DFD,F2F,NT})

Method For Generalization

- Training Strategy



Transfer learning

- It uses 1 dataset for single domain learing and all for transfer learning

Method For Generalization

- Defense Strategy

Algorithm 1: Selecting the best combination for open-domain attack using Restricted Grid Search

Data: $\mathcal{D}_{FacialReenact}$, $\mathcal{D}_{IdentitySwap}$

$\mathcal{C} = \emptyset$
for $i \subset \mathcal{D}_{FacialReenact}$ and $j \subset \mathcal{D}_{IdentitySwap}$ **do**
  $\mathcal{T}$ =TransferLearning(i,j)
  $\mathcal{M}$ =MergeLearning(i,j)
  if $\mathcal{T}$ *performs better than* $\mathcal{M}$ **then**
    Append: $\mathcal{C} \leftarrow \mathcal{T}$
  else
    Append: $\mathcal{C} \leftarrow \mathcal{M}$
  end
end
Result: $\mathcal{C}_{best}$

-$\mathrm{D}_{FacialReenact}$ = {F2F, NT}

-$\mathrm{D}_{IdentitySwap}$ = {DF, FS, DFD}

Table 2: The details of datasets used for training and testing.

Datasets	Total Videos	Training Videos	Transfer Learning	Testing Videos
Pristine (Real)	1,000	750	10	250
DeepFake (DF)	1,000	750	10	250
FaceSwap (FS)	1,000	750	10	250
Face2Face (F2F)	1,000	750	10	250
Neural Textures (NR)	1,000	750	10	250
Deepfake Detection (DFD)	300	250	10	50
DeepFake-in-the-Wild (DFW)	200	*DFW is only used for testing*		200

- Python v3.7.5
- Tensorflow v1.13.1
- Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz with 256.0 GB RAM

Experimental Setting

Table 2: The details of datasets used for training and testing.

Datasets	Total Videos	Training Videos	Transfer Learning	Testing Videos
Pristine (Real)	1,000	750	10	250
DeepFake (DF)	1,000	750	10	250
FaceSwap (FS)	1,000	750	10	250
Face2Face (F2F)	1,000	750	10	250
Neural Textures (NR)	1,000	750	10	250
Deepfake Detection (DFD)	300	250	10	50
DeepFake-in-the-Wild (DFW)	200	*DFW is only used for testing*		200

Input Data

- ShallowNet, Xception, and MesoNet - Single frame (1×80 = 80 frames)
- CLRNet and DBiRNN - 5 consecutive frames (5×16=80 frames)
- CNN+LSTM - 20 consecutive frames (20×4=80 frames)

Experiments

Method	Datasets $\mathcal{D}_T = \mathcal{D}_A$ (F_1 score %)						
	Identity Swap				Facial Reenact.		Avg.
	DF	FS	DFD	DFDC	F2F	NT	
CNN+LSTM	78.51±3.5	77.75±1.5	70.31±5.3	86.28±1.5	71.87±3.1	90.54±1.0	77.80
DBiRNN	80.54±2.7	80.56±1.8	82.45±3.4	81.94±3.5	73.12±6.1	94.38±0.3	82.21
ShallowNet	88.97±2.5	93.33±1.3	73.73±4.7	91.75±0.4	75.26±5.1	99.45±0.1	87.08
Xception	99.00±0.2	99.29±0.2	95.53±0.4	96.50±0.1	87.62±2.4	99.46±0.1	96.18
MesoNet	99.01±0.1	99.26±0.1	95.37±0.1	95.69±0.1	99.01±0.1	99.27±0.1	98.38
CLRNet(Ours)	**99.20**±0.1	**99.50**±0.2	**96.00**±0.1	**96.76±0.2**	**99.20**±0.1	**99.50**±0.1	**98.61**

1) Performance on In-domain Attack

- CLRNet model shows **outperformed** the baseline models.

- On Average, **CLRNet** showed the **best performance (98.61%)**, MeosoNet showed the 2nd performance (98.38%).

Experiments

Method	$\mathcal{D}_T$	Attack Datasets $\mathcal{D}_A$ (F_1 score %)				$\mathcal{D}_{unknown}$
		Identity Swap			Facial Reenact.	DeepFake in the Wild
	F2F	**DF**	**FS**	**DFD**	**NT**	**DFW**
CNN+LSTM	*71.87±3.1*	51.12±0.1	50.45±0.4	50.25±0.1	46.96±0.6	49.14±0.5
DBiRNN	*73.12±6.1*	53.65±0.3	50.12±0.3	50.86±0.5	48.45±0.3	49.34±0.1
ShallowNet	*75.26±5.1*	55.57±0.1	51.37±0.6	51.58±0.4	47.29±0.1	50.08±0.1
Xception	*87.62±2.4*	52.40±0.2	50.10±0.2	50.25±1.3	50.52±0.2	50.06±0.1
MesoNet	*99.01±0.1*	51.90±0.5	50.43±0.3	49.73±0.2	50.36±0.3	49.77±0.2
CLRNet(Ours)	***99.20±0.1***	**64.18**±0.1	**52.32**±0.1	**56.75**±0.2	**50.60**±0.1	**50.59**±0.1

2) Performance on Out-Of-Domain Attack

- The detector's performance of an Out-of-domain attack will be lower than that of an in-domain attack.
- D_T shows the **high performance** results of **CLRNet** for **F2F(Face2Face)**.
- All methods performed poorly with an F1 score of **about 50%, CLRNet performed better.**

Experiments

Method	**Open-Domain Attack $\mathcal{D}_A = \{DFW\}$ (F_1 score %)**			
	Single Domain Learning	Merge Learning	Transfer Learning	**Best** (RGS)
	$\mathcal{D}_T = \{FS\}$	$FS \cup DF \cup DFD \cup F2F \cup NT$	$NT \rightarrow \{DF, FS, DFD, F2F\}$	
CNN+LSTM	45.23±1.2	53.75±3.1	55.10±4.7	63.64
DBiRNN	47.15±1.0	51.97±4.6	58.12±6.1	65.71
ShallowNet	48.36±2.1	57.84±7.7	60.35±8.3	70.72
Xception	49.12±3.2	71.45±5.9	61.41±7.2	72.51
MesoNet	49.45±1.3	78.12±4.8	75.56±6.6	80.25
CLRNet(Ours)	**50.65**±0.9	**84.95**±0.2	**93.86**±0.2	**95.92**

3) Performance on Open-Domain Attack

- Detector model trained for D*known* was used using **Merge learning** and **Transfer learning.**

- **CLRNet** showed the **highest performance** in merge learning and transfer learning (84.95%, 93.86%)

Experiments

Table 5: Performance comparison of defense strategies (merge and transfer learning) against out-of-domain attack.

Merge Learning (FS ∪ DF ∪ DFD ∪ F2F ∪ NT)						
Method	Attack Datasets $\mathcal{D}_A$ (F_1 score %)					**Avg.**
	Identity Swap			Facial Reenact.		
	FS	DF	DFD	F2F	NT	
CNN+LSTM	57.45±5.1	63.12±2.3	55.54±4.1	53.23±7.5	80.12±4.1	61.89
DBiRNN	59.42±2.1	61.75±2.1	59.85±5.2	55.42±1.3	82.10±3.8	63.71
ShallowNet	55.86±7.1	65.16±5.4	50.92±1.2	58.84±5.2	86.03±1.8	63.36
Xception	73.29±5.8	78.57±6.2	54.35±4.2	74.17±6.3	82.45±2.3	72.57
MesoNet	80.17±2.6	84.21±1.2	86.27±0.2	82.52±3.1	87.60±1.9	84.35
CLRNet(Ours)	**87.20**±0.8	**85.72**±0.1	**86.52**±0.1	**89.20**±0.1	**89.28**±0.1	**87.58**

Vs

Transfer Learning (FS → {DF, DFD, F2F, NT})						
Method	Attack Datasets $\mathcal{D}_A$ (F_1 score %)					**Avg.**
	Identity Swap			Facial Reenact.		
	FS	DF	DFD	F2F	NT	
CNN+LSTM	70.21±1.0	52.75±5.1	53.52±4.5	50.73±3.1	63.34±4.1	58.11
DBiRNN	67.45±2.5	55.95±1.0	50.43±7.2	51.79±8.5	66.08±5.0	58.34
ShallowNet	87.51±3.7	50.29±1.4	42.38±9.9	53.83±7.2	67.24±8.2	60.25
Xception	73.90±7.6	65.04±5.7	57.60±8.1	55.43±9.2	76.79±7.0	65.75
MesoNet	93.36±3.1	69.60±9.3	64.58±7.9	83.58±7.2	81.51±9.1	78.52
CLRNet(Ours)	**98.70**±0.2	**97.23**±0.3	**97.13**±0.1	**97.50**±0.2	**97.30**±0.1	**97.57**

Best

Merge Learning vs Transfer Learning

- Merge learning requires much more data on the train than transfer learning.
- In the experiment, the difference in data set size took 4 times longer for transfer learning.

Visualization using CLRNet

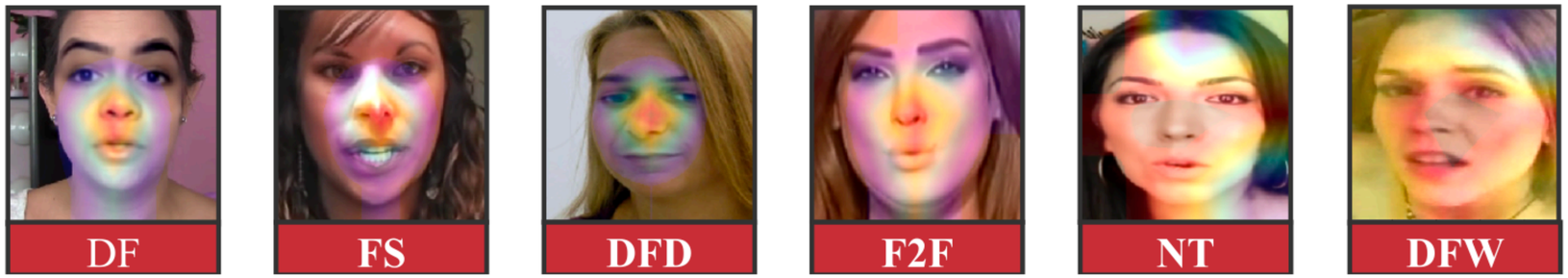


Figure 8: *Class Activation Map from CLRNet*: Artifacts that distinguishes real and deepfakes are present near or around the nose region for DF, FS, F2F, and DFD. The CAM of DFW looks different for each video.

- **DF, FS, DFD, and F2F**: Activaon are very concentrated in the center of the face (around the nose).
- **Natural Textures (NT)**: Activation is randomly distributed around the face, and the area around the nose has no activation.
- **DeepFake-in-the-Wild (DFW)**: It indicates that the dataset is mixing various deepfake methods.

  - It is better to generalize and detect other deepfake attacks with one model using CLRNet (93.86%).

Conclusion

- Convolution LSTM used Residual Network -> CLRNet
- Spatial and temporal information is necessary -> SOTA Detector
- Transfer Learning Strategy using both source and target -> Best Generalization
- Evaluation on DeepFake-in-the-Wild dataset -> 93.86%

Thank you