

# FRANK: FeatuRe Anti spoofing Network

Zih-Ching Chen, Chin-Lun Fu, Lin-Hsi Tsao, James Young Advisor: Yu-Chiang Frank Wang

Graduate Institute of Communication Engineering, National Taiwan University

# Abstract

Face anti-spoofing is an important, yet challenging problem in full-stack face applications. While emerging approaches of face anti-spoofing have been proposed in recent years, most of them focus on developing discriminative models based on the features extracted from images, and do not generalize well to new database. This work propose a novel perspective of face anti-spoofing that disentangles the content features and the spoofed feature from images. In this project, we designed a novel network architecture, feature anti-spoofing networks (FRANK), which focus on improving the generalization ability across different kinds of datasets from the perspective of disentangling spoof traces from the image. We evaluate our method on Oulu and SiW datasets and extensive experimental results demonstrate the effectiveness of our method. Finally, we further visualize some results to show the effect and advantage of disentanglement.

# Related work

## **Domain Adaptation**

Domain Adversarial Neural Networks (DANN) train a feature extractor with a domain-adversarial loss among the source domains. The source-domain invariant feature extractor is assumed to generalize better to novel target domains.

Domain Separation Networks (DSN) decompose the sources domains into shared and private spaces and learns them with a reconstruction signal.

### **Domain Generalization**

Episodic Training for domain generalization trains a single deep network in a way that exposes it to the domain shift that characterizes a novel domain at runtime.

# **Feature Disentanglement**

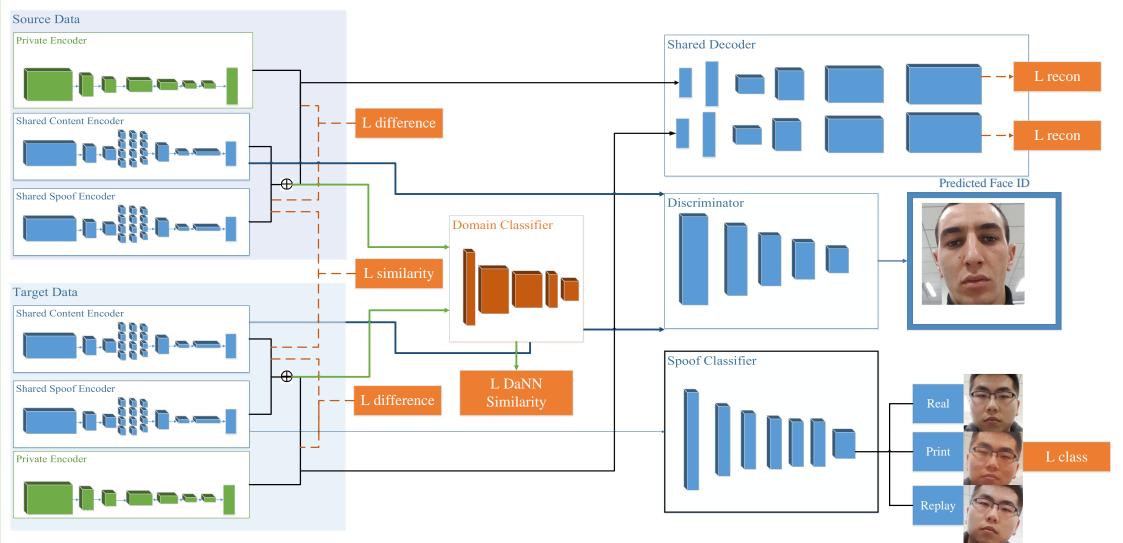
AC-GAN performs supervised feature disentanglement which separates features into disjoint parts representing content information and attribute information.

InfoGAN performs unsupervised interpretable representation disentanglement by maximizing the mutual information between latent features and variation.

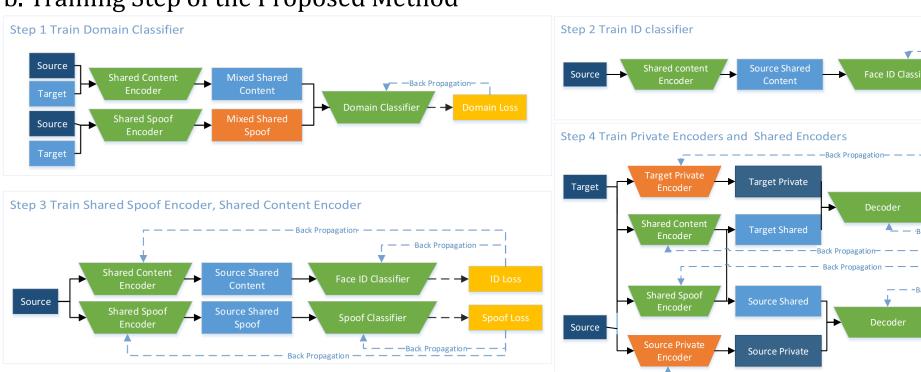
# Methods

### **Model Architecture**

a. Overview of the proposed FeatuRe Anti spoofing NetworK (FRANK)



### b. Training Step of the Proposed Method



### **Loss Function**

a. The domain adversarial similarity loss is used to train a model to produce representations such that a classifier cannot reliably predict the domain of the encoded representation.

$$L_{similarity}^{DANN} = \sum_{i=0}^{N_s + N_t} \{d_i \log \hat{d}_i + (1 - d_i) \log (1 - \hat{d}_i)\}$$

b. The classification loss trains the model to predict the output labels we are ultimately interested in. Because we assume the target domain is unlabeled, the loss is applied only to the source domain. We want to minimize the negative log-likelihood of the ground truth class for each source domain sample.

$$L_{task} = -\sum_{i=0}^{N_s} (y_i^s \cdot log\hat{y}_i^s)$$

c. We use a scale—invariant mean squared error term for the reconstruction loss which is applied to both domains.

$$L_{si\_mse(x,\,\hat{x})} = \frac{1}{k} \|x - \hat{x}\|_2^2 - \frac{1}{k^2} ([x - \hat{x}] \cdot 1_k)^2$$

$$\mathcal{L}_{recon} = \sum_{i=1}^{N_s} L_{si\_mse(x_i^s, \hat{x}_i^s)} + \sum_{i=1}^{N_t} L_{si\_mse(x_i^t, \hat{x}_i^t)}$$

d. The similarity loss is applied to align the spoof feature of both domains, which represents the fine feature space for spoof detection

$$L_{similarity} = \sum_{i=1}^{N_S} L_{mse(x_i^S, \hat{x}_i^S)}$$

e. The difference loss is also applied to both domains and encourages the shared and private encoders to encode different aspects of the inputs. We define the loss via a soft subspace orthogonality constraint between the private and shared representation of each domain.

$$L_{difference} = \left\| H_c^{sT} H_p^{s} \right\|_F^2 + \left\| H_c^{tT} H_p^{t} \right\|_F^2$$

f. The human ID classification loss is used to train the ID classifier to discriminate human ID and the content encoder model to produce proper representations for human ID.

$$L_{ID} = -\sum_{i=0}^{N_S} (y_i^S \cdot log \hat{y}_i^S)$$

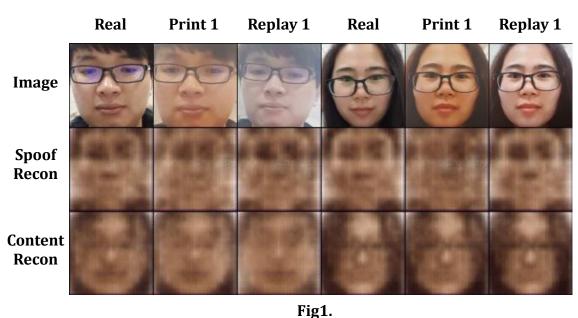
# shar Experiments and Results

## **Quantity Evaluation**

Model	Oulu AUC	SIWAUC	SIW Bonus
DANN	0.9057	0.9604	
DSN	0.9974	0.9872	71.54%
Domain Generalizer	0.9947	0.9690	59.77%
Feature Disentanglement	0.9929	0.9515	59.35%
FRANK (Ours)	0.9985	0.9879	79.24%

# **Quality Evaluation**

Our FRANK can extract features of two different Image spaces: content and spoofed. Content features represent information of a person's face id, which can find differences from people. Spoofed is like noise from the background, which is not the information of a person's face id feature, so spoofed face feature is closely the same between different people. In Fig.1 and Fig.2, we can verify our hypothesis.



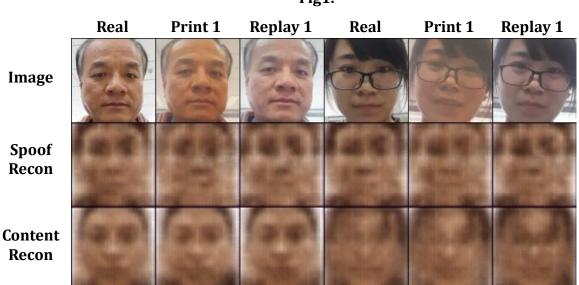


Fig2.

Proposed Model (FRANK)	Oulu AUC	SIWAUC	SIW Bonus
All terms	0.9985	0.9879	79.24%
w/ o face id discriminator	0.9929	0.9515	59.35%
Only session $1 \rightarrow 2$	0.9369	0.7491	54.77%
Only session $2 \rightarrow 1$	0.9720	0.9093	58.05%

# References

- [1] Domain-Adversarial Training of Neural Networks
- [2] Domain Separation Networks
- [3] Episodic training for domain generalization
- [4] Conditional Image Synthesis with Auxiliary Classifier GANs [5] InfoGAN: Interpretable Representation Learning by Information Maximizing Generative
- Adversarial Nets

# **Ablation Study**

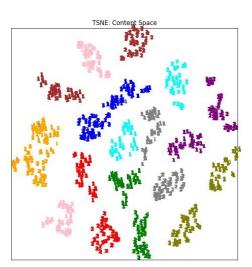
**Visualization Spoof vs Content vs Private** 

We provide three different visualization testing for our model.

spoof encoder respectively are separated, which is our expectation.

and the different number means the domain from session 1 or session 2.

the different number means the class label (0: real, 1: print, 2:, replay)



a. It describes the features extracted from private source encoder, private target encoder, content encoder and

b. It describes the features extracted content encoder. The different colors means different face id from people,

c. It describes the features extracted spoof encoder. The different colors means different face id from people, and

