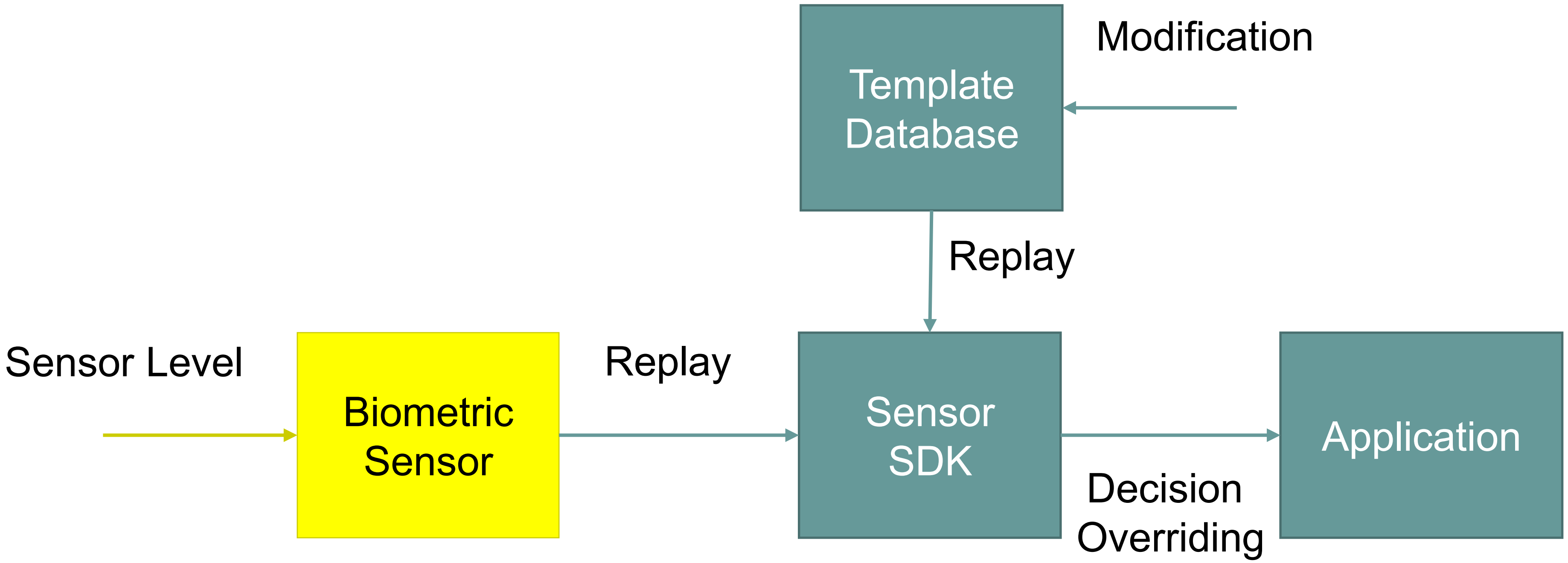
Iris Liveness Detection by Relative Distance Comparisons

Federico Pala, Bir Bhanu
Center for research in intelligent systems
University of California, Riverside

IEEE Workshop on Biometrics CVPR 2017
July 21 2017 – Honolulu, Hawaii, USA

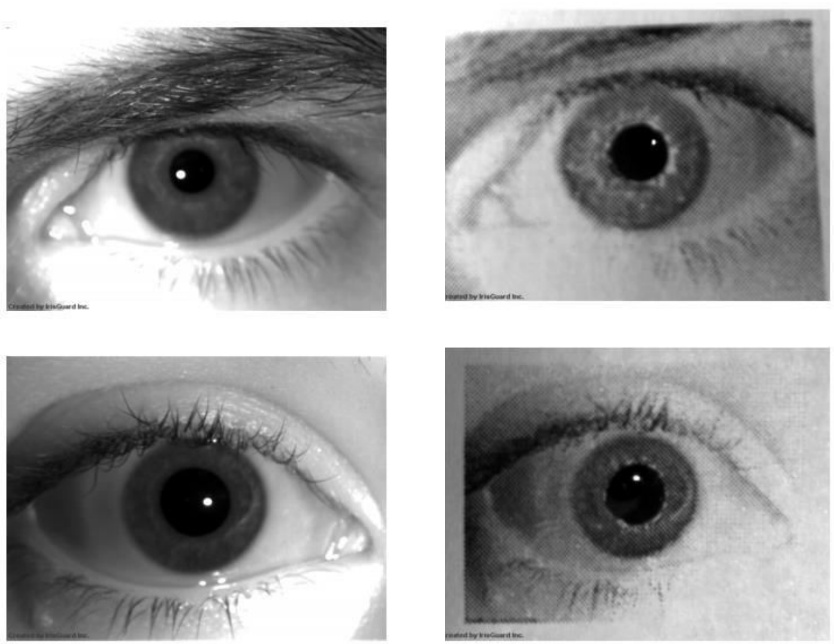
Liveness Detection

- Detect vitality from biometrics signature
- Targeted to **sensor level** attacks



Iris Liveness Detection

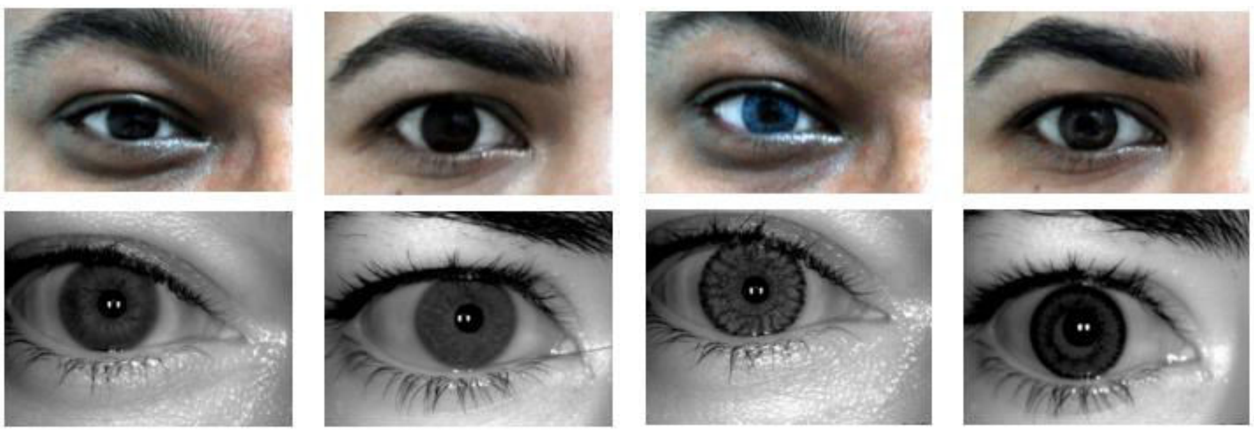
- Training
  - Collection of real and fake iris images
  - Photo attacks:

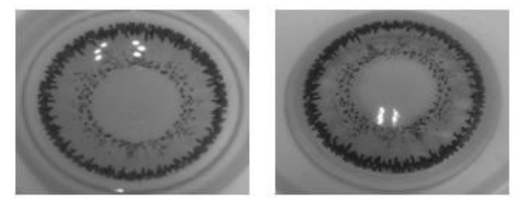


Adam Czajka - **Database of Iris Printouts and its Application: Development of Liveness Detection Method for Iris Recognition**, In: International Conference on Methods & Models in Automation & Robotics (2013)

Iris Liveness Detection

- Training
  - Collection of real and fake images
  - Contact lens attacks:





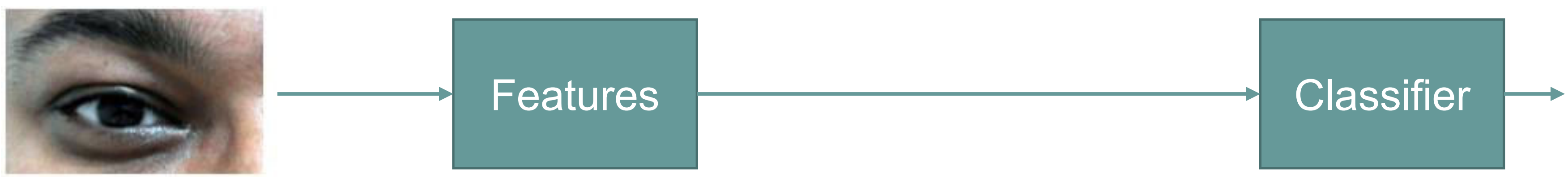
Contact lens

real fake (textured lens)

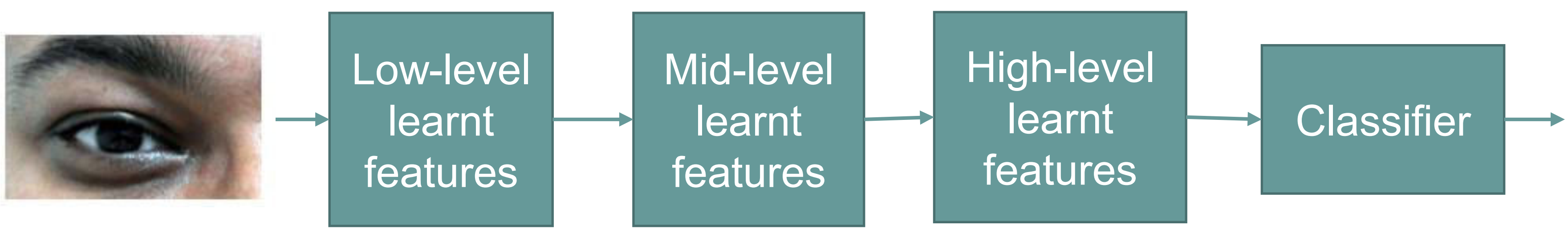
Daksha Yadav, Naman Kohli, James S. Doyle, Kevin Bowyer - **Unraveling the Effect of Textured Contact Lenses on Iris Recognition**, In: IEEE Transactions on Information Forensics and Security (2014)

Artificial Intelligence Approach

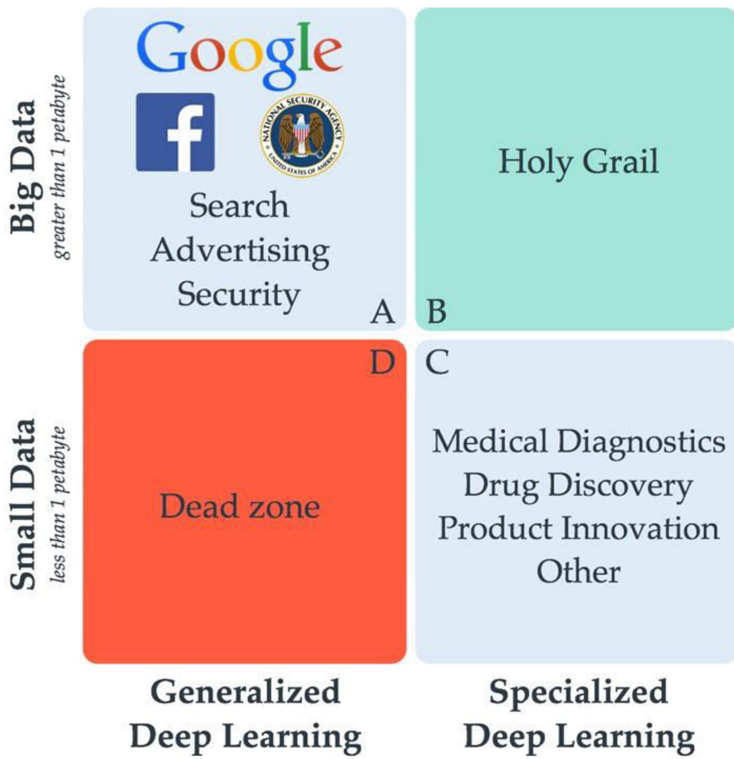
- Let an intelligent agent perceive what are the characteristics that make a biometrics real or fake
- Pattern recognition



- Deep learning



Specialized Deep Learning



	filter			
	random	optimized		
benchmark	AO	*cf10-11*	*spoofnet*	SOTA
Warsaw	**99.84**	67.20	66.42	97.50
Biosec	98.93	59.08	47.67	**100.00**
MobBIOfake	98.63	99.13	**100.00**	99.75

David Menotti et Al. - **Deep Representations for Iris, Face, and Fingerprint Spoofing Detection**, In: IEEE Transactions on Information Forensics and Security (2015)

Shalini Ananda - **An Open Letter to Yann LeCun—Small Data Requires Specialized Deep Learning**, https://tinyurl.com/smalldata

Our Approach

- Learn by comparing **local features**
  - Set of n reference real and fake patches:
    - $R_L = \{\boldsymbol{r}(x_{L_1}), \boldsymbol{r}(x_{L_2}), \dots, \boldsymbol{r}(x_{L_n})\}$
    - $R_F = \{\boldsymbol{r}(x_{F_1}), \boldsymbol{r}(x_{F_2}), \dots, \boldsymbol{r}(x_{F_n})\}$

- Matching problem:
  - Given a query image
    - Extract p patches $Q = \{\boldsymbol{r}(Q_1), \boldsymbol{r}(Q_2), \dots, \boldsymbol{r}(Q_p)\}$
  - Real if $\sum_{j=1}^{p} D(R_L, Q_j) \geq \sum_{j=1}^{p} D(R_F, Q_j)$
  - Fake otherwise

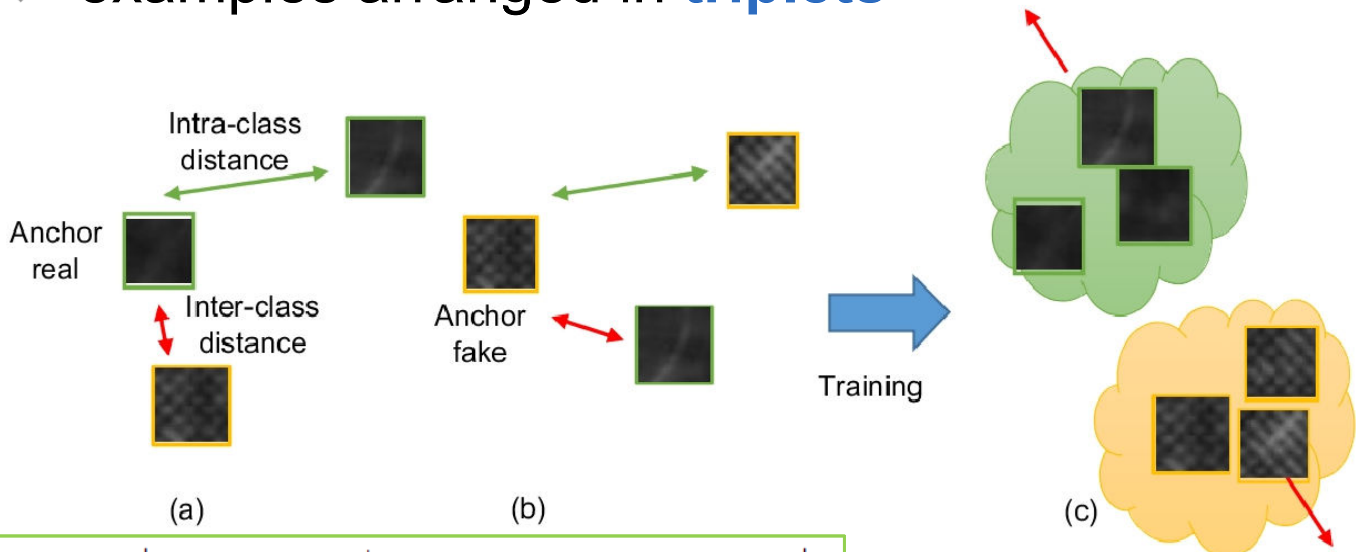
Architecture

Layer description	output
32x32 gray level image	
5x5 conv. filters, stride 1, 1 $\rightarrow$ 64 feat. maps	64x28x28
batch normalization	64x28x28
rectifier linear unit	64x28x28
3x3 conv. filters, stride 2, padding 1, 64 $\rightarrow$ 64 feat. maps	64x14x14
3x3 conv. filters, stride 1, 64 $\rightarrow$ 128 feat. maps	64x12x12
batch normalization	64x12x12
rectifier linear unit	64x12x12
3x3 conv. filters, stride 2, padding 1, 128 $\rightarrow$ 128 feat. maps	128x6x6
3x3 conv. filters, stride 1, 128 $\rightarrow$ 256 feat. maps	256x4x4
batch normalization	256x4x4
rectifier linear unit	256x4x4
3x3 conv. filters, stride 2, padding 1, 256 $\rightarrow$ 256 feat. maps	256x2x2
fully connected layer 4x256 $\rightarrow$ 256	256
dropout $p = 0.4$	256
rectifier linear unit	256
fully connected layer 256 $\rightarrow$ 256	256
softmax	256

Jost Tobias Springenberg et Al. - **Striving for Simplicity: The All Convolutional Net**, In: ICLR workshop track (2015)

Metric Learning

- Increase **number of examples**
  - **patch** based representation
  - examples arranged in **triplets**



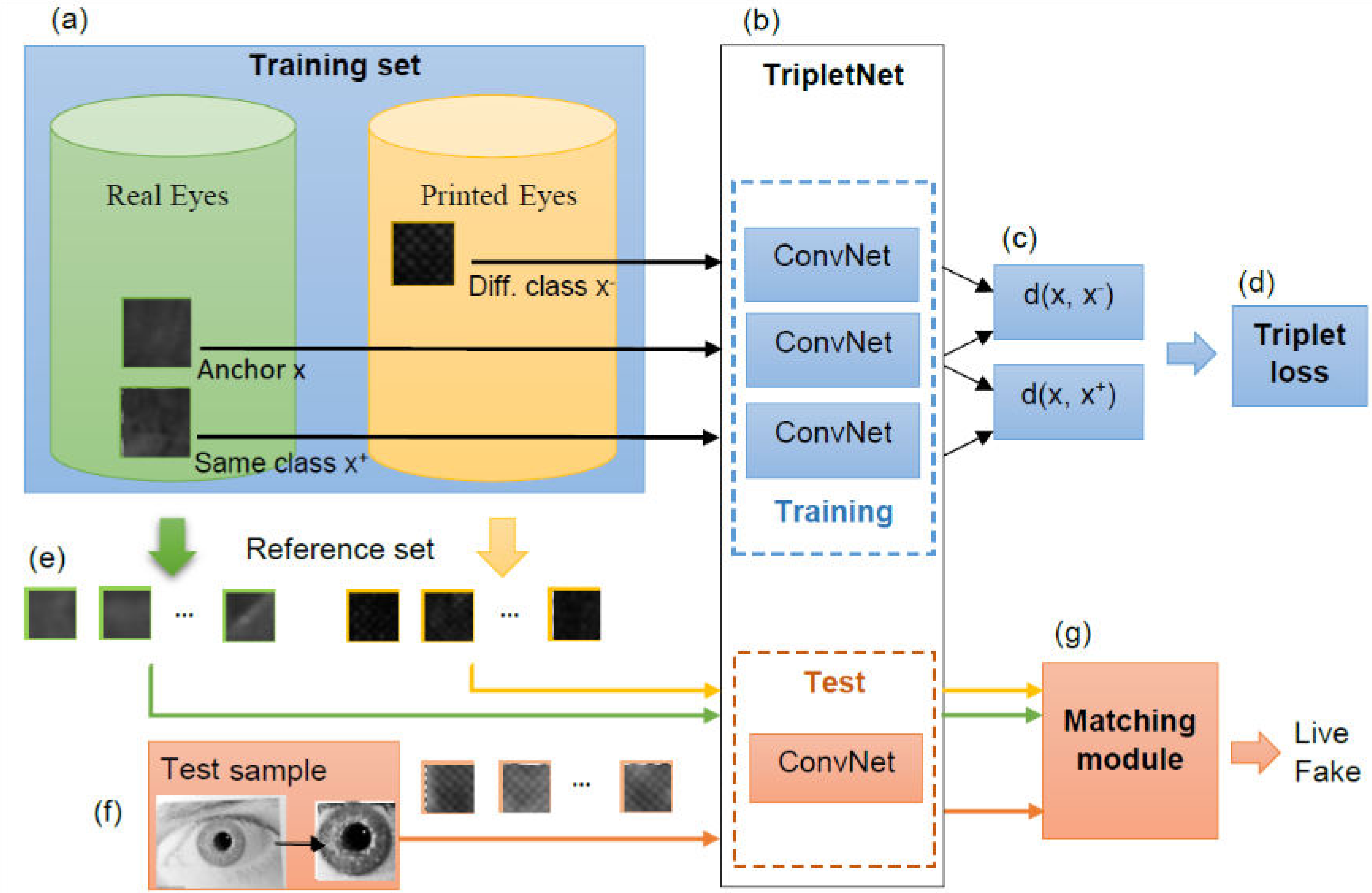
$$c(x_i, x_j^+, x_k^-) = \left| d(\mathbf{r}(x_i), \mathbf{r}(x_j^+)) - d(\mathbf{r}(x_i), \mathbf{r}(x_k^-)) + 1 \right|_+$$

$$c(x_i, x_j^+) = d(\mathbf{r}(x_i), \mathbf{r}(x_j^+))$$

$$L = \sum_{i,j,k} \left\{ c(x_i, x_j^+, x_k^-) + \beta c(x_i, x_j^+) \right\} + \lambda \|\theta\|_2$$

Elad Hoffer, Nir Ailon - **Deep Metric Learning Using Triplet Network**, International Workshop on Similarity-Based Pattern Recognition (2015)

Framework



- Validation: Classification error leads to **underfitting**
  - **Criterion**: how many patches closer to the respective reference set?
  - **Stop**: when the number of violating triplets does not decrease for five epochs.

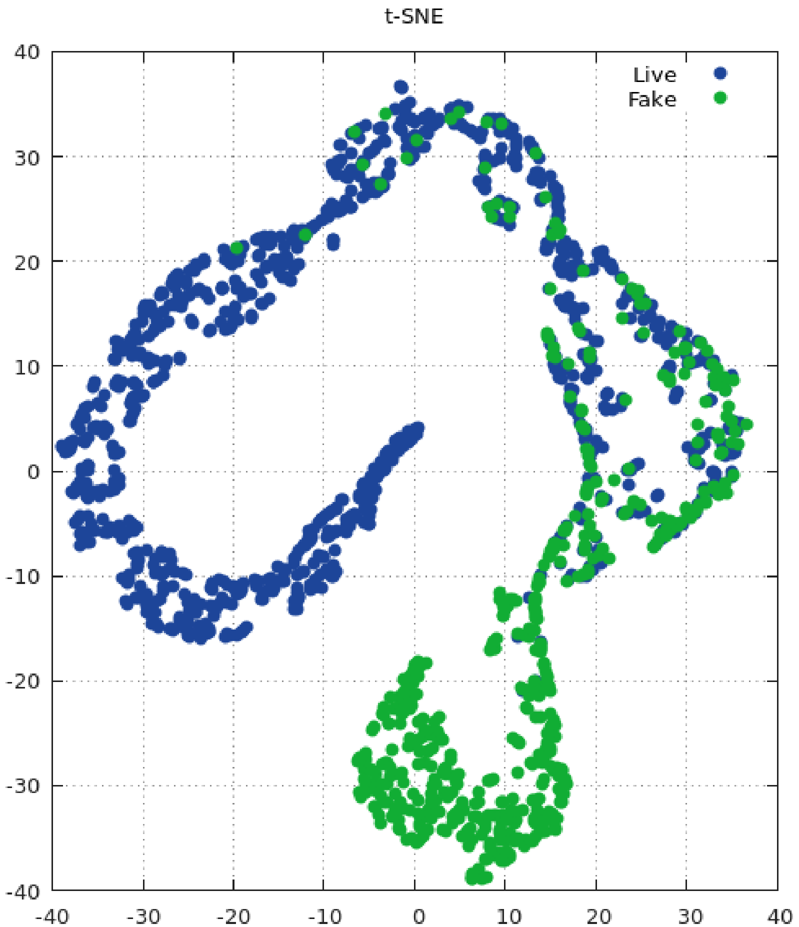
Results

- Average Classification Error %

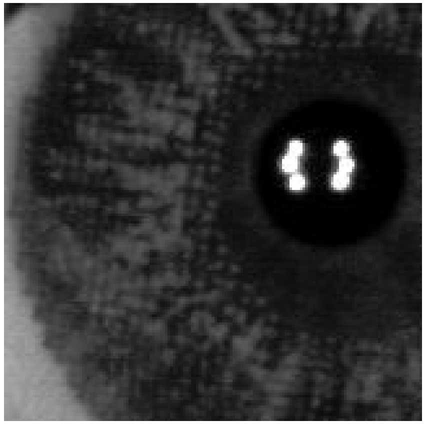
Dataset	Triplet Net	SID	CNN	Dense SIFT	DAISY	LCPD
Iris-2013-Warsaw	**0.0**	0.0	0.2	0.5	0.9	7.1
Cogent	**5.5**	6.2	-	13.9	17.2	11.0
Vista	**0.7**	3.5	-	2.5	8.8	3.1

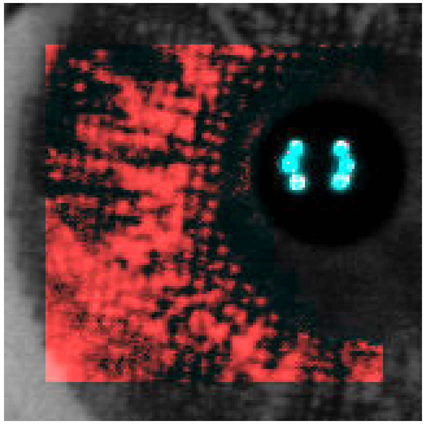
- Computational time: 0.2ms + 1ms (GPU)
- Average between
  - Attack Presentation Classification Error Rate (APCER)
  - Bona Fide Presentation Classification Error Rate (BPCER)

T-SNE Representation

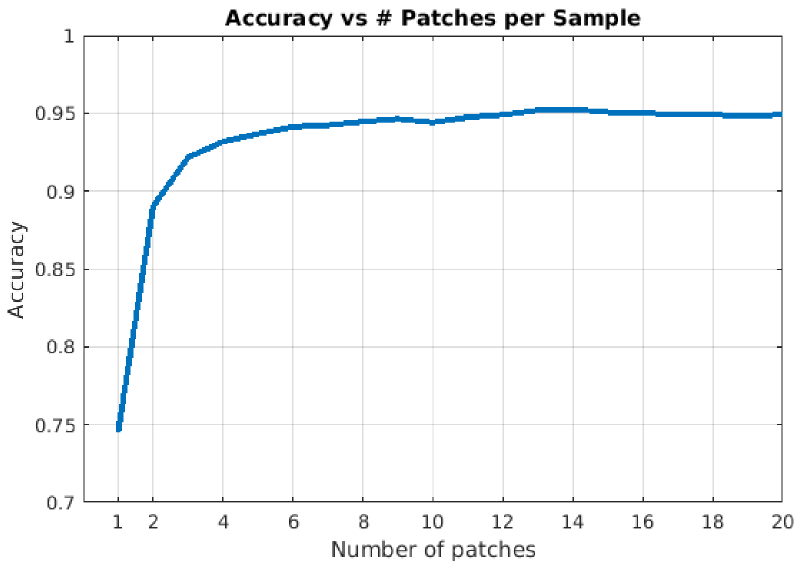


Visualization





In red, patches classified as Fake (sliding window)



Accuracy wrt the number of patches to evaluate a test iris image

Conclusions

- We proposed a **software system** for Iris liveness detection

- We overcome the **scarcity of examples** by:
  - **Patch** based representation
    - Full scale image analysis
  - Metric learning
    - Examples arranged in a multitude of **triplets**

- State of the art or better performance