Evaluating the Resiliency of Blink-Based DeepFake Detection Against Adversarial Noise

By Joel Coulon (u2204489)

A Brief Introduction to DeepFakes

- A DeepFake is where a piece of media (usually images and videos) are digitally altered or created by an AI
- Whilst originally created for entertainment purposes
- Can be used for misinformation, scam, and various other nefarious activities
- They are frighteningly realistic:

Image from: https://www.whichfaceisreal.com/

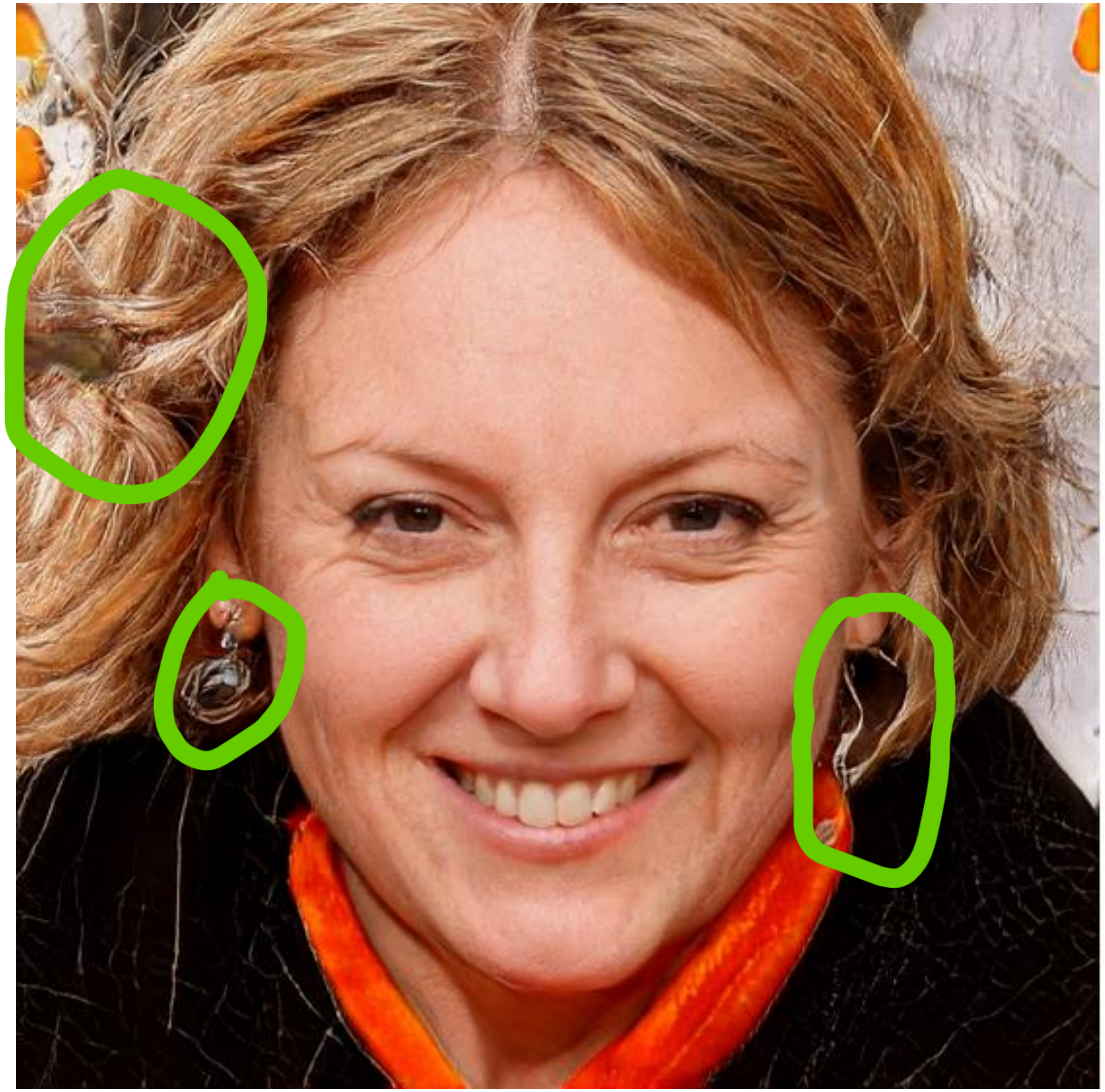


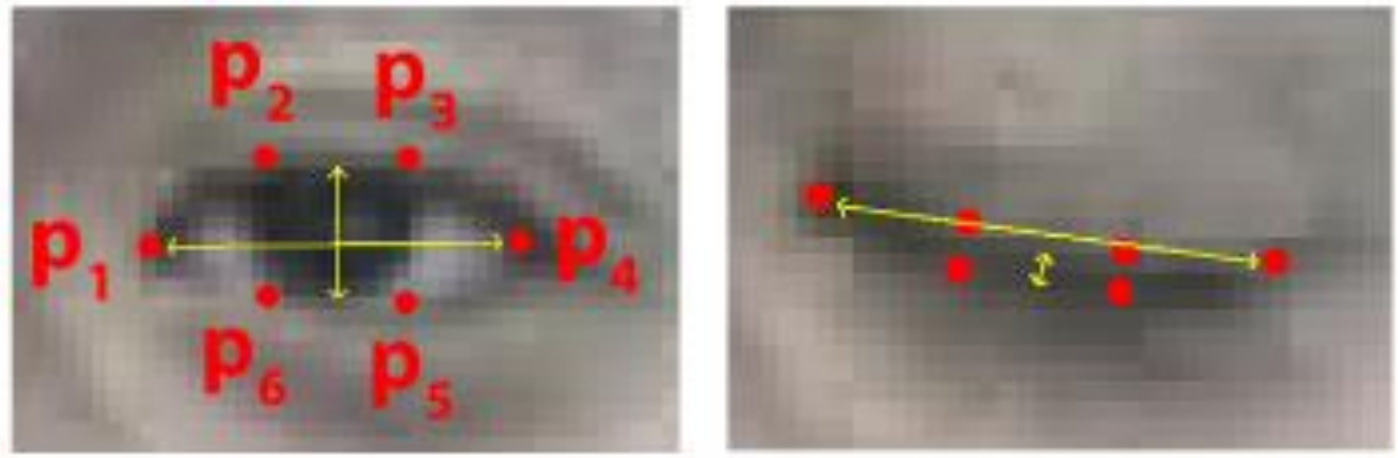
Image from: https://www.whichfaceisreal.com/results.php?r=1&p=1&i1=image-2019-02-17_004111.jpeg&i2=68128.jpeg

Motivation

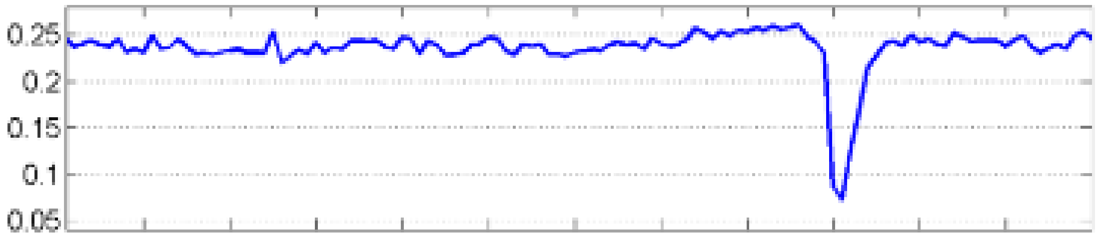
- Wanted to do project related to cybersecurity and AI
- A friend suggested looking into DeepFakes
- Countering Malicious DeepFakes: Survey, Battleground, and Horizon
  - “vulnerable to adversarial noise attacks with imperceptible additive noises”
  - “They do not take physiological signals such as eye blink frequency … into consideration”

Juefei-Xu, F., Wang, R., Huang, Y. et al. “Countering Malicious DeepFakes: Survey, Battleground, and Horizon”, in International Journal of Computer Vision 130, 1678–1734 (2022)

Eye Aspect Ratio



$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

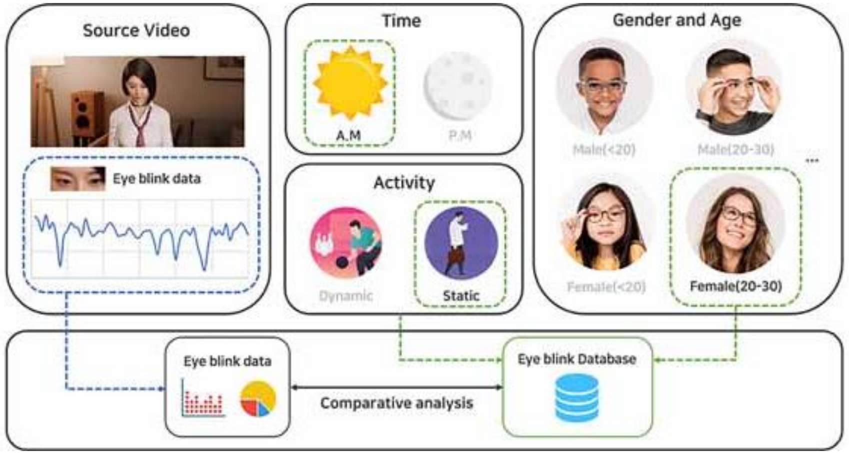


Soukupova, T. and Jan C, "Eye blink detection using facial landmarks“, in 21st computer vision winter workshop, Rimske Toplice, Slovenia, 2016

Existing DeepFake Detection Methods

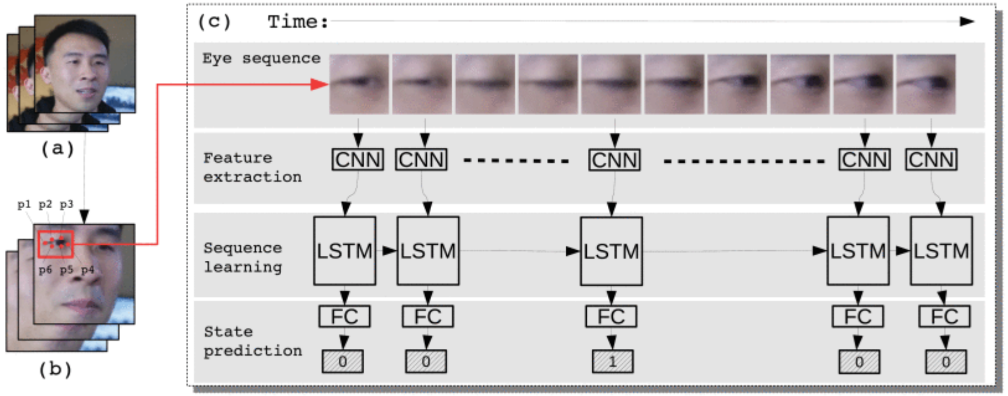
DeepVision[1]

- Gets EAR graph and other metadata
- Extracts number of blinks, length of blinks, period of blinks
- Compares with a database of known "good" data



Ictu Oculi[2]

- Trains an LSTM-based network to determine whether eye is open or not
- Compare number of blinks overtime with known human average (10 blinks/minute) if too low then fake else real



[1]T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern", in IEEE Access 8, 83144-83154 (2020)
[2]Y. Li, M. -C. Chang and S. Lyu, "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking", 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 2018, pp.1-7

Adversarial Noise

CW-L2 attack

- Adds noise that minimises L2 norm of the noise (keeps close to original image)
- But still causes a misclassification

$$\mathbf{x}_{adv} = \tfrac{1}{2}(\tanh(\omega^*) + 1)$$
$$\omega^* = \arg\min_\omega \left\{ \|\mathbf{x}' - \mathbf{x}\|_2^2 + cf(\mathbf{x}') \right\}$$
$$f(\mathbf{x}') = \max\left(\max_{i \neq y} \left\{ \mathbf{Z}(\mathbf{x}')_y - \mathbf{Z}(\mathbf{x}')_i \right\}, -\kappa\right)$$

FGSM

- Finds gradient of model's loss function, adds a small amount of noise to that gradient to offset the model

$$\mathbf{x}_{adv} = \mathbf{x} + \varepsilon\,\mathrm{sign}(\nabla_x J(\mathbf{x}, \mathbf{y}, \theta)).$$

(a) Unperturbed Real Images

(b) Unperturbed Fake Images

(c) Perturbed (FGSM) Fake Images

(d) Perturbed (CW-L_2) Fake Images

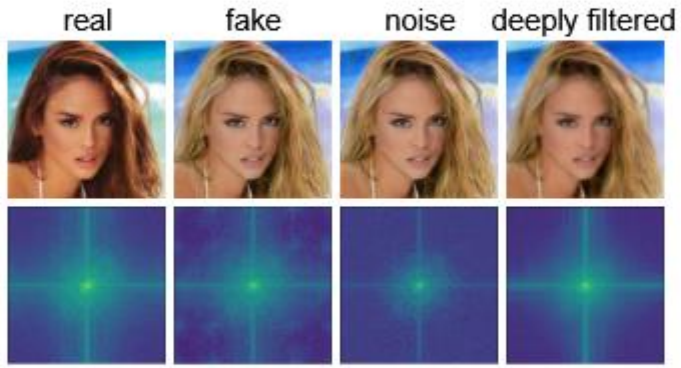
A. Gandhi and S. Jain, "Adversarial Perturbations Fool Deepfake Detectors," *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020, pp. 1-8

Adversarial Noise (cont.)

- Fake Retouch
  - Add gaussian noise to an image based on binary map **A**

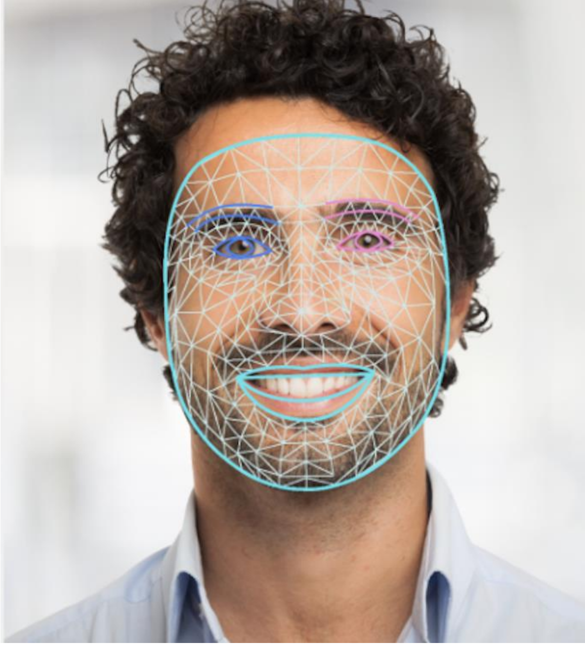
$$\hat{\mathbf{I}} = \mathbf{K} \circledast (\mathbf{I} + \mathbf{A} \odot \mathbf{N}_\sigma) \qquad \arg\max_{\mathbf{A}} J(\mathbf{D}(\mathbf{I} + \mathbf{A}), y) + \|\mathbf{A}\|_1$$

  - Creates Kernels **K** using a neural network
  - Compute noise-map **A** by minimising L1 loss



Huang, Y., Juefei-Xu, F., Guo, Q., Xie, X., Ma, L., Miao, W., Liu, Y. and Pu, G., 2020. Fakeretouch: Evading deepfakes detection via the guidance of deliberate noise

Proof of Concept



- Made over the Christmas holidays
- Uses pre-existing methods where possible
- Google's MediaPipe[1] for eye landmarks
- Number of blinks used as final method
- Threshold for blink set as $\min(EARs) + \sigma(EARs)$
- Traditional detectors represented by VGG19 and ResNet detectors
- Noise using Foolbox[2,3] (FGSM attack)
  - Noise only targeted for VGG19 methods

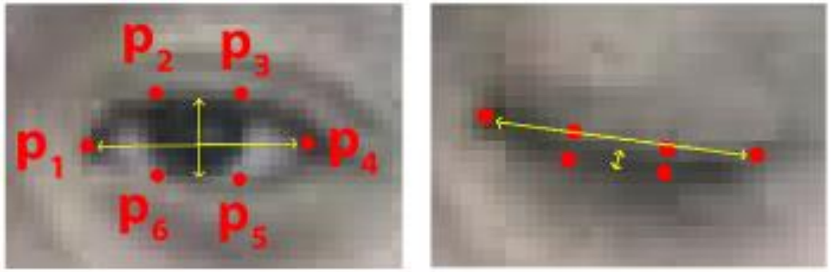
[1] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J. and Chang, W.T., 2019. Mediapipe: A framework for building perception pipelines
[2] Rauber, J., Zimmermann, R., Bethge, M. and Brendel, W., 2020. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, *5*(53), p.2607.
[3] Rauber, J., Brendel, W. and Bethge, M. (2017) 'Foolbox v0.8.0: A Python toolbox to benchmark the robustness of machine learning models', *CoRR*, abs/1707.04131.

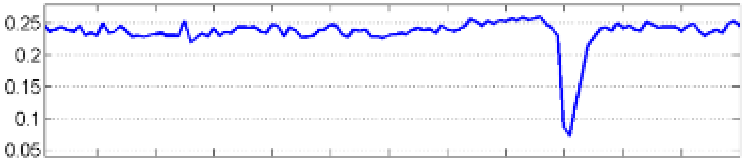
Results of Proof of Concept

	Blink Detection		VGG19		ResNet	
	Original	Noise	Original	Noise	Original	Noise
True Positives (declared real when real)	44	44	48	48	45	45
True Negatives (declared fake when fake)	36	32	50	0	45	50
False Positives (declared real when fake)	14	18	0	50	4	0
False Negatives (declared fake when real)	6	6	2	2	5	5
Overall accuracy (Accuracy on fake videos)	80% (72%)	76% (64%)	98% (100%)	52% (0%)	91% (90%)	95% (100%)

- Noise very specialised to each model
- When varying ε, ResNet would change, Blink Detection would not

Proposed Architecture for Detection





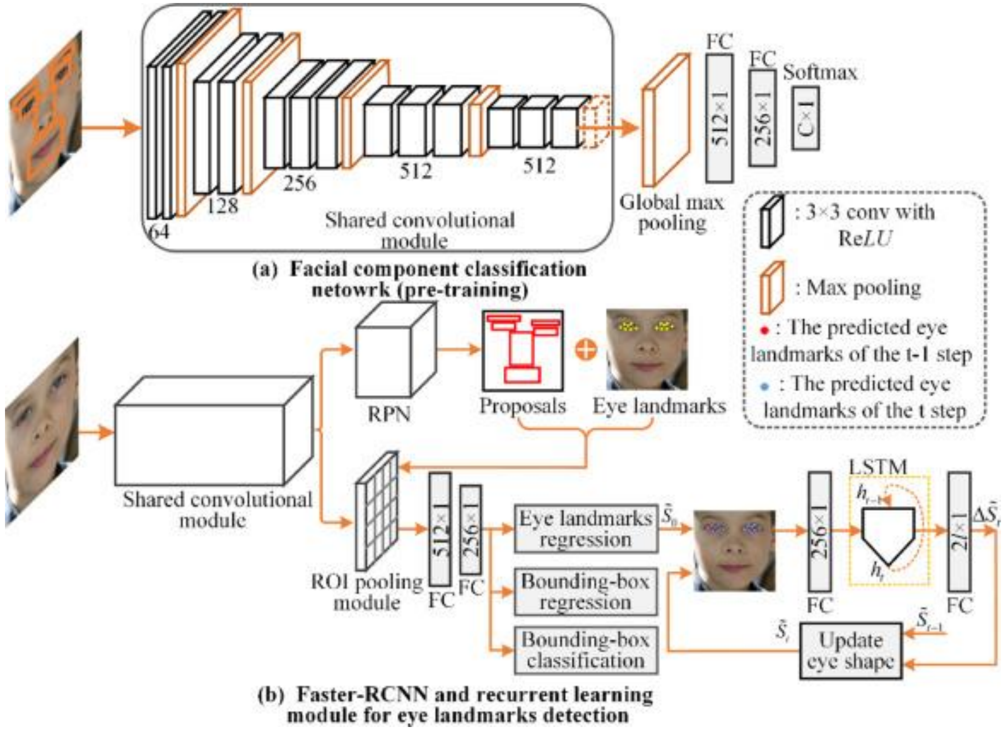
Time-Series Analysis

Selection of Eye Landmark Model

- The vast majority of public facial landmarking models are unsuitable
- They are either optimised for model size or model speed
- Not model accuracy

Facial landmark detection by semi-supervised deep learning

- Most accurate eye landmark detector currently published
- RPN blackbox?
- FC Layer?
- Scrapped development after a month of work

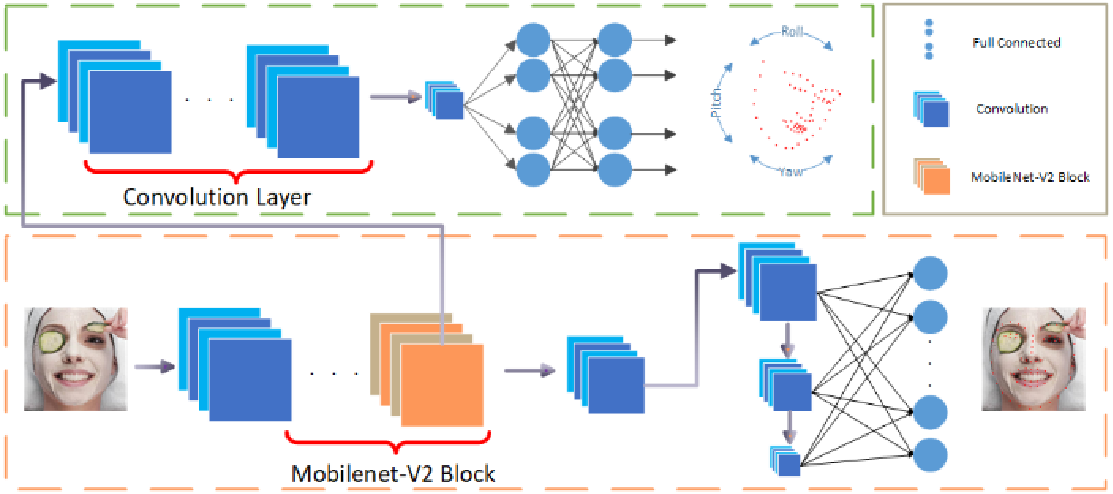


Huang B., Chen R., Zhou Q., Xu W., "Eye landmarks detection via weakly supervised learning", Pattern Recognition, Volume 98, 2020

New Detection Methods

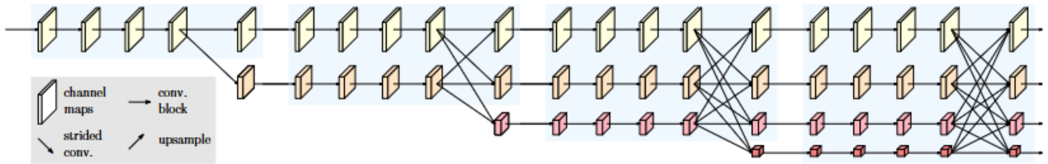
PFLD[1] (yet to be implemented)

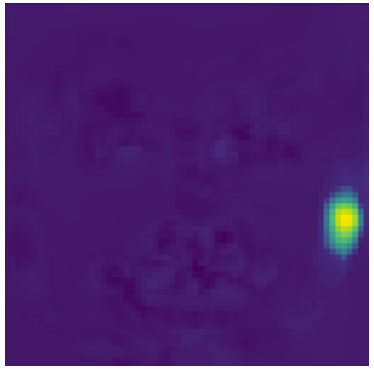
- Uses pre-trained backbone network to do initial predictions
- A second model picks up from an intermediary layer to estimate pitch, yaw, and roll
- Currently uses MobileNetV2 (interchangeable?)

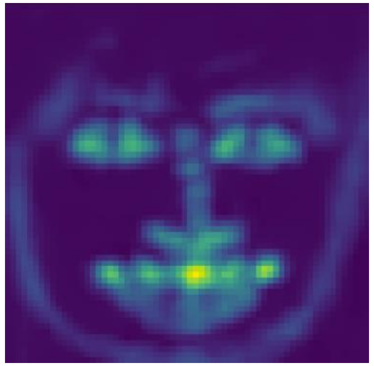


HRNet[2]

- Multiple modular high to low fusion blocks in parallel
- Lower blocks are downsampled to focus on finer features
- Output is a heatmap per landmark, max value of heatmap is landmark

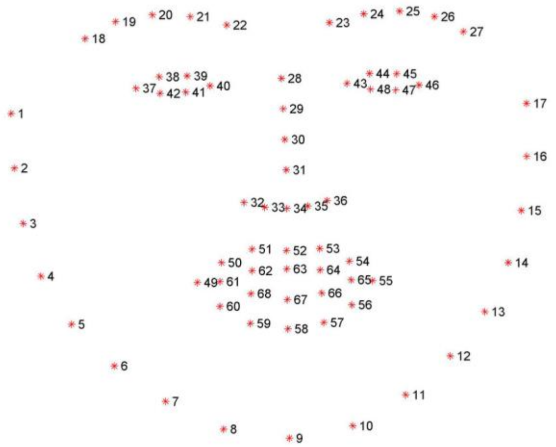






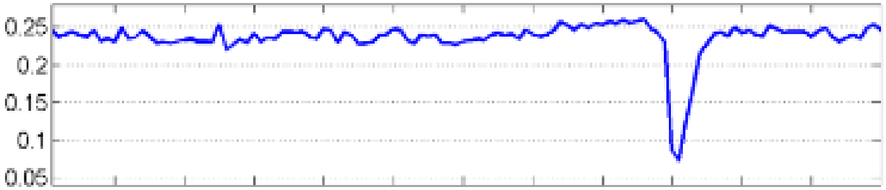
[1] Guo, X., Li, S., Yu, J., Zhang, J., Ma, J., Ma, L., Liu, W. and Ling, H., 2019. PFLD: A practical facial landmark detector
[2] Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W. and Wang, J., 2019. High-resolution representations for labeling pixels and regions.

Datasets – Facial Landmarks



- Various datasets exist for facial landmarking
- Vast majority use 68-landmarks
- Datasets used are: 300W[1], AFW[2], COFW[3], HELEN[4], iBug[5], LFPW[6], WFLW[7]
- Chosen for wide variety of facial poses, situations, and occlusions

Time Series Analysis



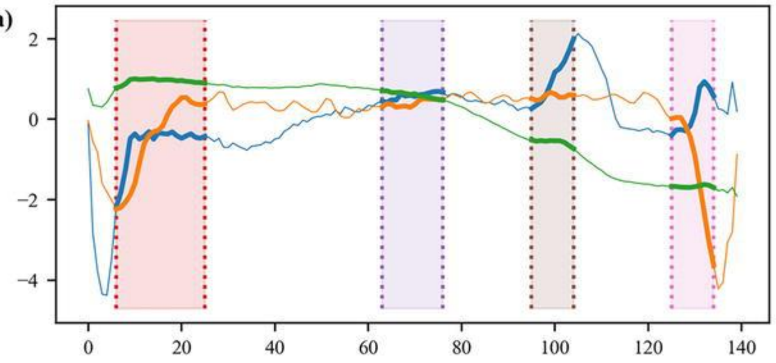
- Now have an EAR-time graph
- Can be abstracted to univariate timeseries
- A variety of analysis methods exist, both classical and neural-network-based
  - Time Series Classification: A Review of Algorithms and Implementations[1]
  - LSTM Fully Convolutional Networks for Time Series Classification[2]
  - Deep learning for time series classification: a review[3]

[1] Faouzi J. Time Series Classification: A Review of Algorithms and Implementations [Internet]. Time Series Analysis - Recent Advances, New Perspectives and Applications. IntechOpen; 2024
[2] Karim F, Majumdar S, Darabi H, Chen S. LSTM fully convolutional networks for time series classification. IEEE access. 2017 Dec 4;6:1662-9.
[3] Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. Data mining and knowledge discovery. 2019 Jul;33(4):917-63.

Methods evaluated

- Long-Short-Term Memory (63%)
- Multi-Layer Perceptron (60%)
- Fully Convolutional Neural Network (51%)
- Convolutional Neural Network (64%)
- ResNet (70%)
- K-Neighbours
  - Dynamic Time Warping
  - DTW (63%), Sakoechiba-DTW (68%), Itakura-DTW (70%), Fast-DTW (67%)
- Learning Shapelets (55%)
- **Time Series Forest (73%)**
- Time Series Bag-of-Features (68%)

(a)



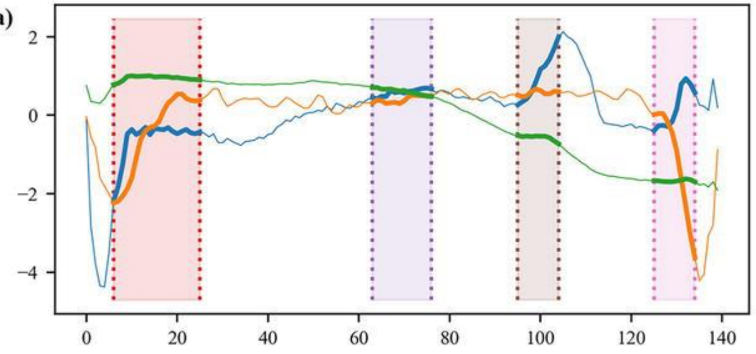
(b)

	Interval 1			Interval 2			Interval 3			Interval 4		
	Mean	SD	Slope	Mean	SD	Slope	Mean	SD	Slope	Mean	SD	Slope
Time series 1	-0.61	0.504	0.052	0.58	0.086	0.02	1.004	0.572	0.197	0.189	0.504	0.156
Time series 2	-0.467	0.977	0.158	0.403	0.084	0.017	0.568	0.061	0.011	-1.204	1.305	-0.431
Time series 3	0.944	0.061	0.002	0.604	0.075	-0.018	-0.57	0.065	-0.017	-1.671	0.024	0.003

Time Series Forest

- Random subsequences of a minimum and maximum length are generated
- From each subsequence the mean, standard deviation, and slope are calculated
- A random forest classifier is trained on these values

(a)



(b)

	Interval 1			Interval 2			Interval 3			Interval 4		
	Mean	SD	Slope	Mean	SD	Slope	Mean	SD	Slope	Mean	SD	Slope
Time series 1	-0.61	0.504	0.052	0.58	0.086	0.02	1.004	0.572	0.197	0.189	0.504	0.156
Time series 2	-0.467	0.977	0.158	0.403	0.084	0.017	0.568	0.061	0.011	-1.204	1.305	-0.431
Time series 3	0.944	0.061	0.002	0.604	0.075	-0.018	-0.57	0.065	-0.017	-1.671	0.024	0.003

Noise & DeepFake Detectors

- The same CW-L2, FGSM, and Fake Retouch were used
- Custom implementation to account for custom (non-tensorflow-based models)

- DeepFake Detectors
  - FACTOR
  - EfficientNetB4

Datasets - DeepFakes

- FaceForensics++
  - A subset of 100 videos for training (50 real, 50 fake)
  - A subset of 100 videos for testing (50 real, 50 fake)
  - The entire dataset to be used for report
- More to be added in final report

Results

- Results go here

Project Management

- Progress tracked in a central document
- Buffer weeks were used
- Switched to eye landmark models with pre-existing implementations
- Analysis was sped up thanks to pre-existing library