

Detecting Deception through Facial Reading

Hin Ho (Richard) Shum, Zhuyin Lyu,
Christy Yung



Background: Why Lie Detection?

Motivation

- For politicians, detecting dishonestly can foster accountability and trust in governance
- Earnings call and other financial disclosures

Goal

- Develop a face-reading ML model to detect lies from videos
- Less intrusive than traditional lie detection technologies like polygraph

How to Detect Lies: Signs of Deception



Lip Biting

Individuals disclose emotional distress by biting their lips

Micro-Expressions

Uncontrolled emotions like anxiety/fear lasting around $\frac{1}{8}$ second before disappearing

Dilated Pupils

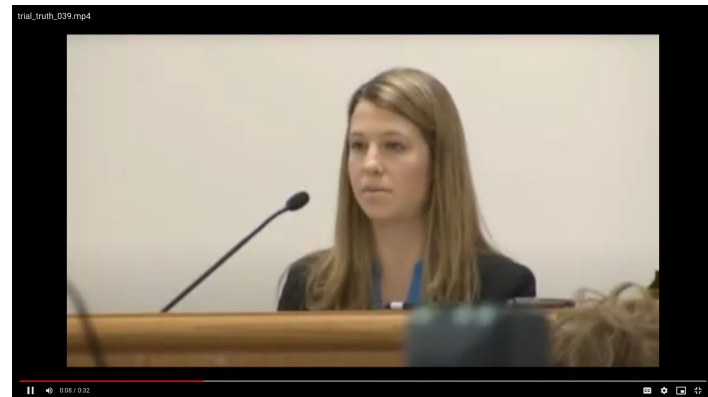
Pupils tend to enlarge when a person is under mental strain

Vocal Tones

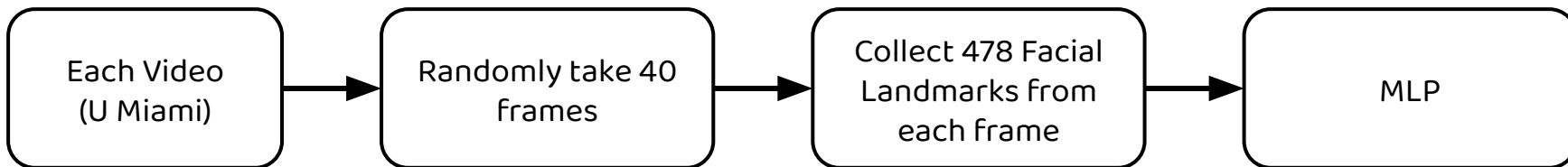
Average pitch and fundamental frequency of their voice go up

Dataset

Dataset	No. of Videos	Stakes	Labels	Availability
UMiami	320	Low	160 Truths, 160 Lies	Special Access
Real Life Court Trial	121	High	60 Truths, 61 Lies	Public



Method1: Naive MLP Approach

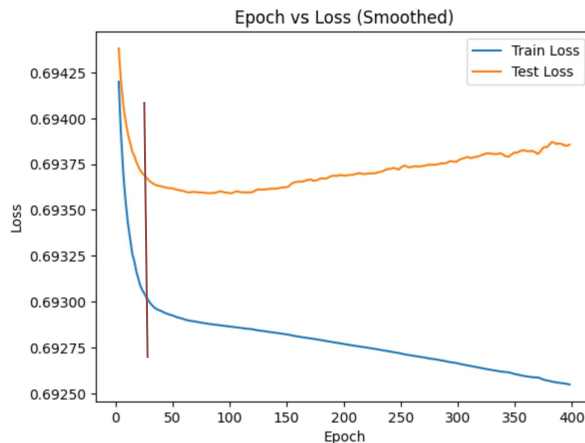


Result:

- Train Accuracy 50%
- Test Accuracy 50%
- Binary Cross Entropy Loss
- Loss decreases very slowly and reaches a limit

Debug:

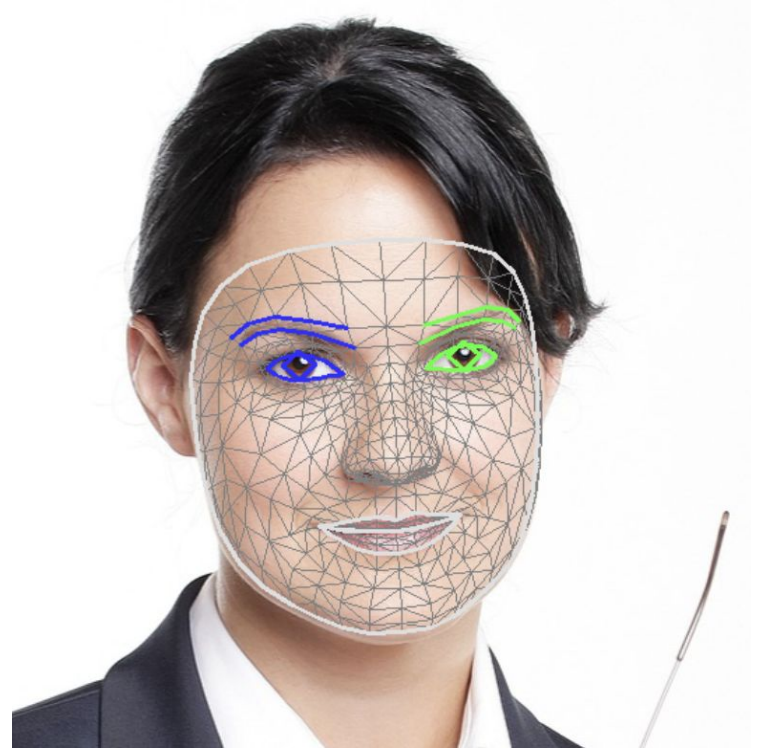
- Microexpressions may appear anywhere in the video so need to use LSTM



```
-----
Epoch 0: train loss 0.6946, test loss 0.6946, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 1: train loss 0.6944, test loss 0.6945, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 2: train loss 0.6941, test loss 0.6943, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 3: train loss 0.6941, test loss 0.6943, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 4: train loss 0.6939, test loss 0.6942, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 5: train loss 0.6938, test loss 0.6941, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 6: train loss 0.6937, test loss 0.6941, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 7: train loss 0.6936, test loss 0.6940, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 8: train loss 0.6935, test loss 0.6940, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 9: train loss 0.6935, test loss 0.6939, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 10: train loss 0.6935, test loss 0.6939, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 11: train loss 0.6934, test loss 0.6939, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 12: train loss 0.6933, test loss 0.6938, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 13: train loss 0.6933, test loss 0.6938, train accuracy 0.5000, test accuracy 0.5000
-----
Epoch 14: train loss 0.6933, test loss 0.6938, train accuracy 0.5000, test accuracy 0.5000
-----
```

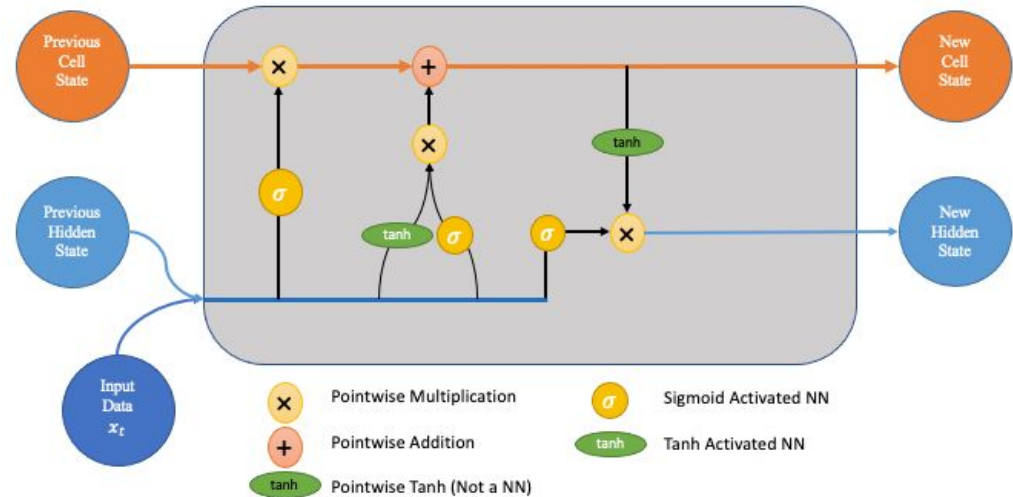
Google MediaPipe

- A python Package for feature extraction
- Facial Landmarks: 2D coordinate that represents important parts of a human face e.g. eyes, nose, mouth
- Each image has 478 facial features which is converted into an array
- XY coordinates are screen coordinates which are normalized to (0-1)

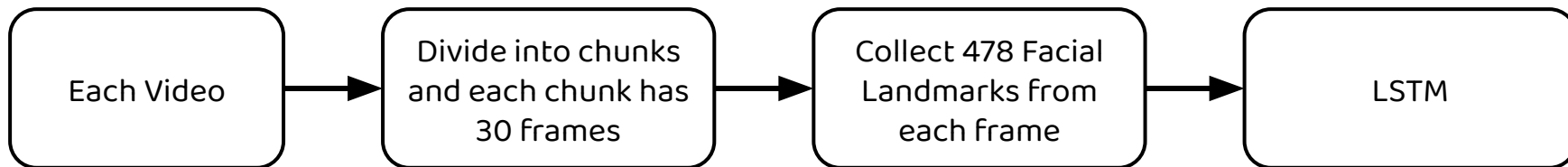


What is LSTM?

- Long short-term memory (LSTM) networks
- It is a type of Recurrent Neural Network (RNN)
- It is particularly useful to process data to do with time
- It allows information to selectively remember or forget information over time



Method2: LSTM with Facial Landmarks

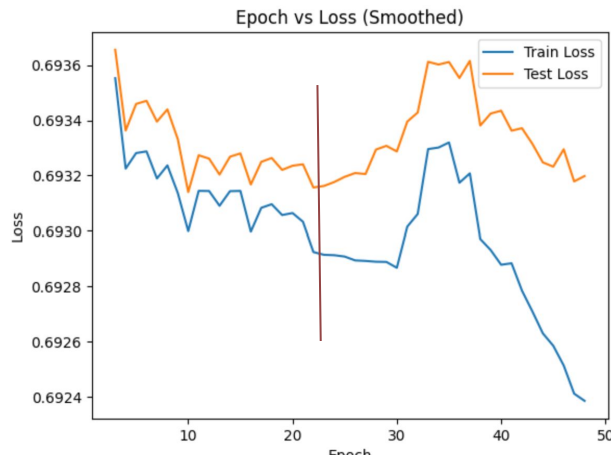


Result:

- Train Accuracy 51%
- Test Accuracy 51%

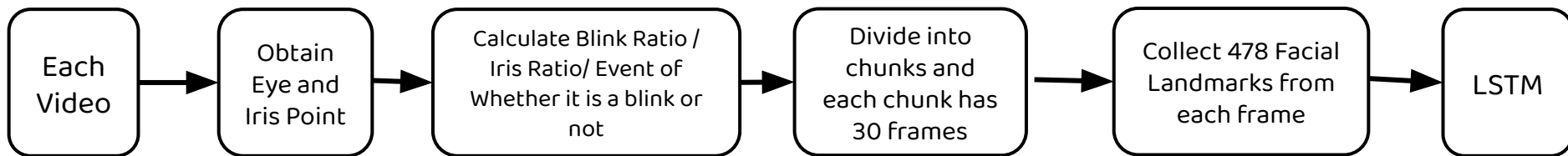
Debug:

- Too many features (noises)
- Need feature reduction



Epoch 0: train loss 0.6950, test loss 0.6950, train accuracy 0.5000, test accuracy 0.5000
Epoch 1: train loss 0.6933, test loss 0.6935, train accuracy 0.5000, test accuracy 0.5000
Epoch 2: train loss 0.6930, test loss 0.6931, train accuracy 0.5078, test accuracy 0.5098
Epoch 3: train loss 0.6934, test loss 0.6935, train accuracy 0.4995, test accuracy 0.5000
Epoch 4: train loss 0.6929, test loss 0.6931, train accuracy 0.5068, test accuracy 0.5039
Epoch 5: train loss 0.6934, test loss 0.6936, train accuracy 0.4990, test accuracy 0.5000
Epoch 6: train loss 0.6936, test loss 0.6940, train accuracy 0.5000, test accuracy 0.5000
Epoch 7: train loss 0.6930, test loss 0.6932, train accuracy 0.5054, test accuracy 0.5176
Epoch 8: train loss 0.6930, test loss 0.6931, train accuracy 0.4985, test accuracy 0.4941
Epoch 9: train loss 0.6932, test loss 0.6933, train accuracy 0.5049, test accuracy 0.5039
Epoch 10: train loss 0.6929, test loss 0.6930, train accuracy 0.5098, test accuracy 0.4883
Epoch 11: train loss 0.6929, test loss 0.6930, train accuracy 0.5132, test accuracy 0.5156
Epoch 12: train loss 0.6938, test loss 0.6938, train accuracy 0.5000, test accuracy 0.5000
Epoch 13: train loss 0.6930, test loss 0.6931, train accuracy 0.5059, test accuracy 0.4902
Epoch 14: train loss 0.6929, test loss 0.6930, train accuracy 0.5161, test accuracy 0.5098

Method3: LSTM with Eye Landmarks

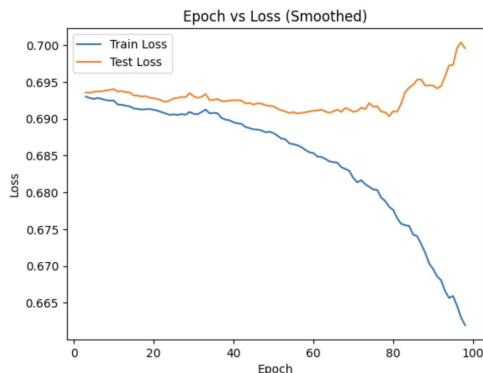


Result:

- Train Accuracy 54%
- Test Accuracy 53%

Debug:

- Not certain if Iris position is correct


















Epoch 0:	train loss	0.6953,	test loss	0.6960,	train accuracy	0.5000,	test accuracy	0.5000
Epoch 1:	train loss	0.6947,	test loss	0.6967,	train accuracy	0.5005,	test accuracy	0.5059
Epoch 2:	train loss	0.6924,	test loss	0.6950,	train accuracy	0.5112,	test accuracy	0.4922
Epoch 3:	train loss	0.6925,	test loss	0.6942,	train accuracy	0.5015,	test accuracy	0.5000
Epoch 4:	train loss	0.6934,	test loss	0.6960,	train accuracy	0.5161,	test accuracy	0.4824
Epoch 5:	train loss	0.6926,	test loss	0.6948,	train accuracy	0.5000,	test accuracy	0.5000
Epoch 6:	train loss	0.6927,	test loss	0.6954,	train accuracy	0.5000,	test accuracy	0.5000
Epoch 7:	train loss	0.6930,	test loss	0.6957,	train accuracy	0.5166,	test accuracy	0.4824
Epoch 8:	train loss	0.6922,	test loss	0.6949,	train accuracy	0.5166,	test accuracy	0.4824
Epoch 9:	train loss	0.6921,	test loss	0.6949,	train accuracy	0.5166,	test accuracy	0.4844
Epoch 10:	train loss	0.6921,	test loss	0.6948,	train accuracy	0.5273,	test accuracy	0.4961

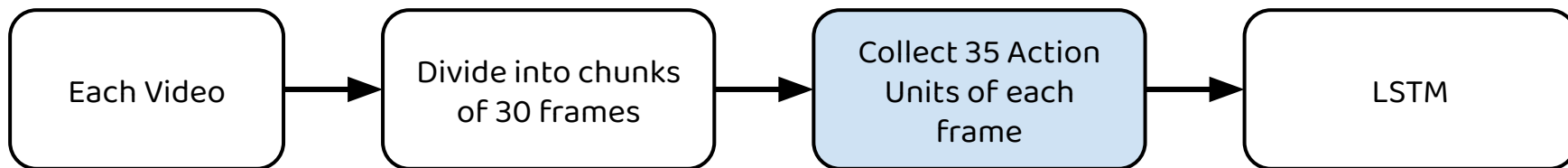
Action Units

- Facial Action Coding System (FACS) is a system developed by Paul Ekman in 1970s
- Action Units (AUs) track the movements of individual or group facial muscles.
- OpenFace is able to detect 35 Action Units
 - Judges existence of 18 AUs
 - Score intensity of 17 AUs
- Action Units are determined using facial landmarks



AU1  Inner brow raiser	AU2  Outer brow raiser	AU4  Brow Lowerer	AU5  Upper lid raiser	AU6  Cheek raiser
AU7  Lid tighten	AU9  Nose wrinkle	AU12  Lip corner puller	AU15  Lip corner depressor	AU17  Chin raiser
AU23  Lip tighten	AU24  Lip presser	AU25  Lips part	AU26  Jaw drop	AU27  Mouth stretch

Method4: LSTM with Action Units

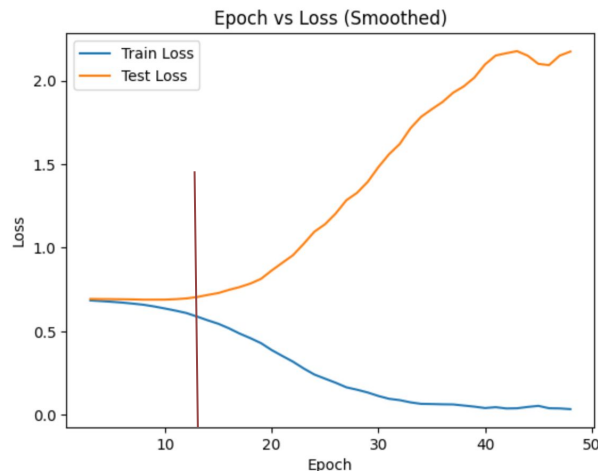


Result:

- Train Accuracy 99%
- Test Accuracy 57%

Debug:

- Dataset has low stakes



Epoch 0: train loss 0.6894, test loss 0.6939, train accuracy 0.5318, test accuracy 0.4925
Epoch 1: train loss 0.6890, test loss 0.6929, train accuracy 0.5470, test accuracy 0.5075
Epoch 2: train loss 0.6827, test loss 0.6926, train accuracy 0.5581, test accuracy 0.5119
Epoch 3: train loss 0.6782, test loss 0.6925, train accuracy 0.5555, test accuracy 0.5040
Epoch 4: train loss 0.6819, test loss 0.6943, train accuracy 0.5682, test accuracy 0.5190
Epoch 5: train loss 0.6695, test loss 0.6899, train accuracy 0.5768, test accuracy 0.5248
Epoch 6: train loss 0.6680, test loss 0.6904, train accuracy 0.5887, test accuracy 0.5155
Epoch 7: train loss 0.6585, test loss 0.6889, train accuracy 0.6057, test accuracy 0.5336
Epoch 8: train loss 0.6472, test loss 0.6882, train accuracy 0.6151, test accuracy 0.5358
Epoch 9: train loss 0.6476, test loss 0.6860, train accuracy 0.6246, test accuracy 0.5504
Epoch 10: train loss 0.6183, test loss 0.6909, train accuracy 0.6486, test accuracy 0.5584
Epoch 11: train loss 0.6074, test loss 0.6910, train accuracy 0.6627, test accuracy 0.5544
Epoch 12: train loss 0.5948, test loss 0.7032, train accuracy 0.6705, test accuracy 0.5566
Epoch 13: train loss 0.5763, test loss 0.7100, train accuracy 0.6928, test accuracy 0.5681
Epoch 14: train loss 0.5413, test loss 0.7282, train accuracy 0.7238, test accuracy 0.5561

Method4: LSTM with Action Units

Real-Life Court
Trial Video

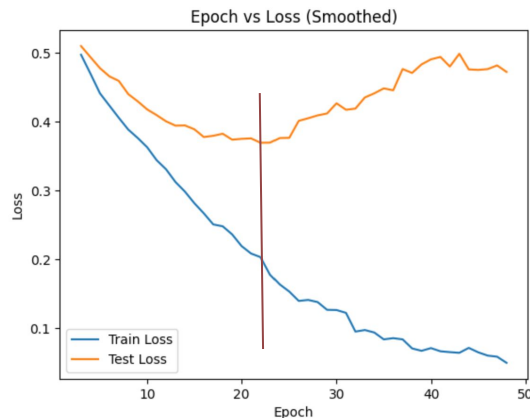
Divide into chunks
of 30 frames

Collect 35 Action
Units of each
frame

LSTM

Result:

- Train Accuracy 99%
- Test Accuracy 86%



Epoch 14: train loss 0.2857, test loss 0.3753, train accuracy 0.8632, test accuracy 0.8111

Epoch 15: train loss 0.2777, test loss 0.4047, train accuracy 0.8632, test accuracy 0.8331

Epoch 16: train loss 0.2446, test loss 0.3775, train accuracy 0.8859, test accuracy 0.8263

Epoch 17: train loss 0.2273, test loss 0.3389, train accuracy 0.8939, test accuracy 0.8583

Epoch 18: train loss 0.2176, test loss 0.4017, train accuracy 0.9027, test accuracy 0.8263

Epoch 19: train loss 0.2721, test loss 0.3901, train accuracy 0.8813, test accuracy 0.8179

Epoch 20: train loss 0.2187, test loss 0.3617, train accuracy 0.9011, test accuracy 0.8314

Epoch 21: train loss 0.1596, test loss 0.3835, train accuracy 0.9267, test accuracy 0.8516

Epoch 22: train loss 0.1732, test loss 0.3422, train accuracy 0.9385, test accuracy 0.8600

Epoch 23: train loss 0.1913, test loss 0.3701, train accuracy 0.9162, test accuracy 0.8482

Epoch 24: train loss 0.1439, test loss 0.3908, train accuracy 0.9427, test accuracy 0.8634

Epoch 25: train loss 0.1480, test loss 0.3951, train accuracy 0.9385, test accuracy 0.8482

Epoch 26: train loss 0.1077, test loss 0.3850, train accuracy 0.9566, test accuracy 0.8651

Epoch 27: train loss 0.1052, test loss 0.4657, train accuracy 0.9562, test accuracy 0.8499

Epoch 28: train loss 0.1985, test loss 0.3892, train accuracy 0.9162, test accuracy 0.8550

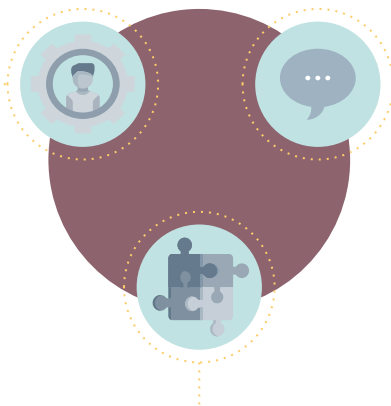
Epoch 29: train loss 0.1283, test loss 0.4113, train accuracy 0.9427, test accuracy 0.8432

Epoch 30: train loss 0.0917, test loss 0.4093, train accuracy 0.9655, test accuracy 0.8651

Challenges

SUITABLE DATA

There are only a few datasets available (lab setting) because people are reluctant to have videos of themselves lying posted online.

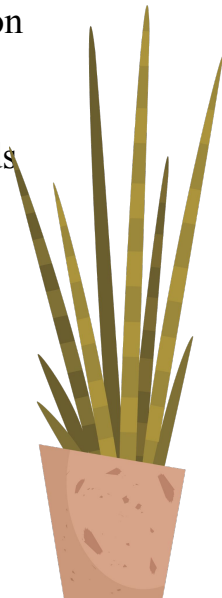


VIDEO PROCESSING

In class, our focus was primarily on image processing. However, since we were working with videos, we had to explore methods for data processing.

LSTM MODEL

LSTM wasn't covered in our class, and we're encountering issues using it with our dataset



Future Plan



Euclidean Distance

Use Euclidean distance to calculate the movement between different frames

Incorporating Text and Audio Data

Audio signals such as pitch, tone, and intonation, may convey emotional states or psychological stressors indicative of dishonesty.

Reduce # of Landmark

Focus on specific landmarks, such as eye movements, to reduce the number of landmarks and assess if there is any change in accuracy

Works Cited

Ahmed Khan, H.U.D., Bajwa, U.I., Ratyal, N.I. et al. Deception detection in videos using the facial action coding system. *Multimed Tools Appl* (2024). <https://doi.org/10.1007/s11042-024-19153-4>

Burgoon, J. K. (2018). Microexpressions Are Not the Best Way to Catch a Liar. *Frontiers in Psychology*, 9, 1672–1672. <https://doi.org/10.3389/fpsyg.2018.01672>

Gallardo-Antolín, A., & Montero, J.M. (2021). Detecting Deception from Gaze and Speech Using a Multimodal Attention LSTM-Based Framework. *Applied Sciences*.

Moi Hoon Yap, Rajoub, B., Ugail, H., & Zwiggelaar, R. (2011). Visual cues of facial behaviour in deception detection. 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 294–299. <https://doi.org/10.1109/ICCAIE.2011.6162148>

Monaro, M., Maldera, S., Scarpazza, C., Sartori, G., & Navarin, N. (2022). Detecting deception through facial expressions in a dataset of videotaped interviews: A comparison between human judges and machine learning models. *Computers in Human Behavior*, 127, Article 107063.

Shen, X., Fan, G., Niu, C., & Chen, Z. (2021, May 17). Catching a liar through facial expression of fear. *Frontiers*. <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2021.675097/full>



Thank You!