Supplementary Material: "Why the face?": Exploring Robot Error Detection Using Instrumented Observer Reactions

MARIA TERESA PARREIRA, RUIDONG ZHANG, SUKRUTH GOWDRU LINGARAJU, ALEXANDRA BREMERS, XUANYU FANG, ADOLFO RAMIREZ-ARISTIZABAL, MANASWI SAHA, MICHAEL KUNIAVSKY, CHENG ZHANG, and WENDY JU

1 STIMULUS VIDEO LIST

Table S1 displays sources and metadata on the videos used as the stimulus dataset.

2 NECKFACE FEATURES USED FOR TRAINING

We list the 55 features generated with the TrueDepth Camera, following Chen et al. [2], also the output from the NeckFace-18 model. These features were used as input for the error detection models.

```
neckface_features = ['browInnerUp', 'browDown_L', 'browDown_R', 'browOuterUp_L', 'browOuterUp_R',
'eyeLookUp_L', 'eyeLookUp_R', 'eyeLookDown_L', 'eyeLookDown_R', 'eyeLookIn_L', 'eyeLookIn_R',
'eyeLookOut_L', 'eyeLookOut_R', 'eyeBlink_L', 'eyeBlink_R', 'eyeSquint_L', 'eyeSquint_R',
'eyeWide_L', 'eyeWide_R', 'cheekPuff', 'cheekSquint_L', 'cheekSquint_R', 'noseSneer_L',
'noseSneer_R', 'jawOpen', 'jawForward', 'jawLeft', 'jawRight', 'mouthFunnel', 'mouthPucker',
'mouthLeft', 'mouthRight', 'mouthRollUpper', 'mouthRollLower', 'mouthShrugUpper', 'mouthShrugLower',
'mouthClose', 'mouthSmile_L', 'mouthSmile_R', 'mouthFrown_L', 'mouthFrown_R', 'mouthDimple_L',
'mouthDimple_R', 'mouthUpperUp_L', 'mouthUpperUp_R', 'mouthLowerDown_L', 'mouthLowerDown_R',
'mouthPress_L', 'mouthPress_R', 'mouthStretch_L', 'mouthStretch_R', 'tongueOut',
'headYaw', 'headPitch', 'headRoll']
```

3 OPENFACE FEATURES USED FOR TRAINING

Below, we list the features included in the dataset. The facial activation features were extracted using OpenFace [1]. While this tool provides a very large set of features, we reduced the feature space by only including the most relevant markers. We obtained a total of 49 facial activation, gaze and pose features:

```
openface_features = ['gaze_0_x', 'gaze_0_y', 'gaze_0_z','gaze_1_x', 'gaze_1_y', 'gaze_1_z',
'gaze_angle_x', 'gaze_angle_y', 'pose_Tx', 'pose_Ty', 'pose_Tz', 'pose_Rx', 'pose_Ry', 'pose_Rz',
'AU01_r', 'AU02_r', 'AU04_r', 'AU05_r', 'AU06_r', 'AU07_r', 'AU09_r', 'AU10_r', 'AU12_r',
'AU14_r', 'AU15_r', 'AU17_r', 'AU20_r', 'AU23_r', 'AU25_r', 'AU26_r',  'AU45_r', 'AU01_c',
'AU02_c', 'AU04_c', 'AU05_c', 'AU06_c', 'AU07_c', 'AU09_c', 'AU10_c', 'AU12_c', 'AU14_c',
'AU15_c', 'AU17_c', 'AU20_c', 'AU23_c', 'AU25_c', 'AU26_c', 'AU28_c', 'AU45_c' ]
```

4 MODEL TRAINING - HYPERPARAMETER TUNING

In Table S2, we leave details of the hyperparameter tested when implementing our models. The final set of hyperparameters was selected based on the best-performing model on the test set, according to the accuracy.

Authors' address: Maria Teresa Parreira; Ruidong Zhang; Sukruth Gowdru Lingaraju; Alexandra Bremers; Xuanyu Fang; Adolfo Ramirez-Aristizabal; Manaswi Saha; Michael Kuniavsky; Cheng Zhang; Wendy Ju.

Table S1. Stimulus video dataset. H: Human, R: Robot, the main actor in the video.

Description	Source	Video Type	Error Time
Human falls trying to imitate Pixar intro scene	Youtube	Failure (H)	3.3s
Falling off of a trampoline	OOPs Dataset	Failure (H)	3.4s
Human crashes into another person on a lawnmover	OOPs Dataset	Failure (H)	2.8s
Human failure in using gym equipment	OOPs Dataset	Failure (H)	3.3s
Human crashes into a pine tree on a hover board	OOPs Dataset	Failure (H)	5s
Human crashes into inventory with fork lift	Youtube	Failure (H)	6s
Human fails at stocking wines and causes wine bottles to break	Youtube	Failure (H)	0.8s
A human falls into the swimming pool	OOPs Dataset	Failure (H)	2.9s
Human fails at performing box jumps	OOPs Dataset	Failure (H)	6.3s
Crashing into one another on bicycles	OOPs Dataset	Failure (H)	5.3s
Humanoid Robot moving back and forth	Youtube	Failure (R)	7.8s
Robotic arm feeding cheetos to a mannequin	Youtube	Failure (R)	4.8s
Robotic armhits baby doll trying to feed it milk	Youtube	Failure (R)	6s
Boston Dynamics Atlas robot fails to jump across	Youtube	Failure (R)	4.7s
Humanoid robot losing balance while playing soccer	Youtube	Failure (R)	5.5s
Robotic arm placing pieces on top of one another	Youtube	Failure (R)	8s
Robotic arm spilling coffee	Youtube	Failure (R)	8s
Humanoid robot losing balance while standing	Youtube	Failure (R)	17s
Humanoid robot failing at climbing stairs	Youtube	Failure (R)	2.3s
Pressure washing on a slab of stone	Youtube	Control (H)	
Removing a sticker from a 4x4 using a pressure washer	Youtube	Control (H)	
Robot climbs down a staircase	Youtube	Control (R)	
Robot jumps over a table	Youtube	Control (R)	
Robot picks up clothes from the floor	Youtube	Control (R)	
Robot turns on a light switch	Youtube	Control (R)	
Robot places dishes in the dishwasher and throws coke can in the trash	Youtube	Control (R)	
Robot fails at handing over a coke can successfully	Youtube	Failure (R)	6s
Human avoids being drenched in water from a wave	Youtube	Control (H)	
Lawn Mowing	Youtube	Control (H)	
Pottery	Unknown source	Control (H)	

REFERENCES

[1] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. 59–66. https://doi.org/10.1109/FG.2018.00019

[2] Tuochao Chen, Yaxuan Li, Songyun Tao, Hyunchul Lim, Mose Sakashita, Ruidong Zhang, Francois Guimbretiere, and Cheng Zhang. 2021. NeckFace: Continuously Tracking Full Facial Expressions on Neck-mounted Wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 58 (Jun 2021), 31 pages. https://doi.org/10.1145/3463511

[3] Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. 2021. MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3447548.3467231

[4] Ignacio Oguiza. 2023. tsai - A state-of-the-art deep learning library for time series and sequential data. Github. https://github.com/timeseriesAI/tsai

[5] Maia Stiber, Russell H. Taylor, and Chien-Ming Huang. 2023. On Using Social Signals to Enable Flexible Error-Aware HRI. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (Stockholm, Sweden) *(HRI '23)*. Association for Computing Machinery, New York, NY, USA, 222–230. https://doi.org/10.1145/3568162.3576990

Table S2. Hyperparameters that were used for hyperparameter tuning, for each model type.

Model	HyperParamaters tested
GRU, LSTM (Keras)	'use bidirectional': [$True, False$], 'num gru layers': [1, 2, 3], 'gru units': [64, 128, 256], 'dropout rate': [0.0, 0.3, 0.5, 0.8], 'dense units': [32, 64, 128, 256, 512], 'activation function': [$'tanh', 'relu', 'sigmoid'$], 'optimizer': [$'adam', 'sgd', 'adadelta', 'rmsprop'$], 'learning rate': [0.001, 0.01, 0.005, 0.1], 'batch size': [32, 64, 128], 'recurrent regularizer': [$'l1', 'l2', 'l1_l2'$], 'loss' : 'binary crossentropy', 'categorical crossentropy', 'sequence length' : [1, 5, 12, 24, 48]
ml-DNN [5]	'dropout rate': [0.0, 0.3, 0.5, 0.8], 'activation function': [$'tanh', 'relu', 'sigmoid'$], 'optimizer': [$'adam', 'sgd', 'adadelta', 'rmsprop'$], 'learning rate': [0.001, 0.01, 0.005, 0.1], 'batch size': [32, 64, 128], 'recurrent regularizer': [$'l1', 'l2', 'l1_l2'$], 'loss' : 'binary crossentropy', 'categorical crossentropy', 'sequence length' : [1]
Transformer (pytorch)	'num heads': [1, 7, 25, 50, 100], 'num layers': [2, 3], 'hidden dim': [64, 128, 256.32, 512], 'dropout rate': [0.0, 0.3, 0.5, 0.8], 'activation function': [$'tanh', 'relu', 'sigmoid'$], 'optimizer': [$'adam', 'sgd', 'adadelta', 'rmsprop'$], 'learning rate': [0.001, 0.01, 0.005, 0.1], 'batch size': [32, 64, 128], 'recurrent regularizer': [$'l1', 'l2', 'l1_l2'$], 'loss' : 'binary crossentropy', 'categorical crossentropy', 'sequence length' : [1, 5, 12, 24, 48]
Minirocket [3] and tsai [4]	'model': ['LSTM_FCN','GRU_FCN', 'InceptionTime', 'InceptionTimePlus','MiniRocket', 'gMLP'], 'dropout LSTM FCN': [0, 0.8, 0.2, 0.5], 'fc dropout LSTM FCN': [0, 0.2, 0.5, 0.8], 'n estimators': [40, 20, 80], 'stride train': [1, 5, 10, 30, 80], 'stride eval': [1, 5, 10, 30, 80], 'lr': [$2e-4$, 0.01, 0.001], 'focal loss': [$False, True$], 'interval length': [1, 5, 12, 25, 40, 80], 'context length': [0], 'batch size': [128, 256]