Supplemental Experimental Procedures

Video data collection

Video data was collected of 45 human participants experiencing genuine pain and faked pain (27 females, all university students). The genuine pain condition consisted of the cold pressor task; this is a task where pain is induced by immersing the arm in ice water at 5^0 Celsius. For the faked pain condition, the water was 20^0 Celsius. For both conditions, participants were instructed to immerse their forearm into the water up to the elbow, and hold it there for 60 seconds. For the faked pain condition, participants were asked to manipulate their facial expressions so that an "expert would be convinced they were in actual pain." Participants' facial expressions were recorded using a digital video camera during both conditions. Examples are shown in Figures 1 and 2 of the main paper. The order of the conditions was kept the same for all participants; faked pain first, followed by real pain. Real pain was always last due to the possibility that one's faked pain expression may be unduly influenced by his or her genuine pain one minute earlier. This influence may be a product of explicit learning or of 'carry over effects' [1] whereby the actual experience of a psycho-physiological state such as pain lingers thus confounding the next observation.

Deception detection by human observers

In Experiment 1, naïve observers viewed a series of pain expression videos individually and judged whether each video contained faked or genuine pain. Observer participants were undergraduates with no explicit training in facial expression measurement. First, observers were shown 25 videos, one of each of the 25 pain participants from either the genuine or faked pain condition. Observer participants then viewed a second set of 25 videos, of the same 25 pain participants but of their other condition (if it was a participant's faked pain expression in video 1, then it was their genuine pain expression in video 2, or vice versa). Therefore, observers did not directly compare genuine and faked expressions within an individual. The videos were counterbalanced and presented sequentially in a randomized order.

In Experiment 2, 35 new observers participated in a training phase where they were provided feedback for the detection of real versus faked pain. Like the computer vision system, the humans were provided training on 24 subjects, including both real and faked pain samples from each subject. During training, the observers viewed 24 video pairs. The same person appeared in both videos in a pair. One video contained an expression of genuine pain, whereas the other contained an expression of faked pain. Both videos were shown sequentially

(the real or fake videos randomly shown as the first or second). Observers judged which one of the pair was real pain or which was faked pain. They were given immediate feedback about their accuracy. After being trained on the 24 pairs, participants saw, in random order, 20 new videos of 20 new models in the test phase. Half of the models were expressing faked pain and the other half were expressing real pain. Without receiving feedback, observers judged whether the presently viewed video contained faked or genuine pain expressions. This design mimicked the machine learning and cross-validation procedure (see below).

Overview of computational methods

The Computer Expression Recognition Toolbox (CERT) [2] was applied to the two one-minute videos of each subject. A set of dynamic features was then extracted from the CERT outputs. These features were passed through a machine learning system allowing the system to learn to predict genuine or faked pain in novel participants. A system overview is shown in Figure 2 in the main text. The components of this system are described below.

The Computer Expression Recognition Toolbox

The Computer Expression Recognition Toolbox is a fully automated system that analyzes facial expressions from video in real-time. CERT automatically detects frontal faces in video and codes each frame with respect to a set of continuous dimensions, including facial actions from the Facial Action Coding System (FACS). FACS is a system for objectively scoring facial expressions in terms of elemental movements, called action units (AUs). FACS identified 46 AUs, each with unique movement and appearance characteristics - which roughly correspond to individual facial muscle movements (see Figure 1 in the main text). FACS was originally developed for manual coding by human experts. Manual coding is laborious, and can take up to 3 hours to manually code 1 minute of behavior. The frame-by-frame CERT output provides information on facial expression intensity and dynamics at temporal resolutions that were previously impractical with human coding.

See [2] for more information on system design and benchmark performance tests. Currently, CERT measures the 20 major AUs from the FACS that have been most strongly associated with emotion. Detection performance by AU is provided in [2]. In addition, CERT estimates of expression intensity correlate with FACS expert intensity codes [2]. The present analysis employed CERT version 4.4. These facial action detectors are available from Emotient, Inc.

Bags of temporal features

CERT produced a 20 channel time series for each video consisting of a real value for each frame and for each AU (20 facial actions x 1800 frames). These dynamic signals were then characterized using a 'bags of temporal features' algorithm. 'Bags of temporal features' builds upon the concept of 'bags of features' from the computer vision literature [3-4] which provide sensitivity to some aspects of the signal, such as edges of different scales, while providing invariance to aspects of the signal across which we wish to generalize. It provides a rich description of temporal texture within a time window, while also providing invariance to the precise time point within the window at which the expression occurred. In this approach, histograms describe the probability distribution of a set of dynamic descriptors within each time window. Here we define and construct bags of temporal features on the CERT AU detection output.

First, a set of temporal descriptors were defined that represent the local temporal texture at multiple time scales. The 1800 frame output of each AU detector was passed through a bank of eight temporal filters. The temporal filters were a family of Gabor functions, which were cosine functions convolved with a Gaussian envelope, given by the following equation:

$$g(t,k) = e(-\tfrac{1}{2}\,\Delta t^2\, k^2/4) * \cos(kt) \qquad (1)$$
$$\text{where } k = (2^{-4-n/2})\,\pi \text{ for } n=1 \text{ to } 8.$$

This resulted in 8 temporal frequencies, for which the wavelength of the cosine, λ, was calculated using

$$\lambda = 2\pi/k = 1/(2^{-5-n/2}). \qquad (2)$$

This translates to 0.66 - .06 Hz with a frame rate of 30 frames per second. The bandwidth of each Gabor function was related to its temporal frequency according to $\sigma = 2/k = \lambda/\pi$, where σ is the standard deviation of the Gaussian.

The temporal descriptors were derived from the filter output as follows (Figure 3 of the main paper). We first found zero-crossings in the output of each Gabor filter. Curves above and below zero were handled separately. (Negative outputs mean evidence that the facial action is absent, e.g. when the mouth is closed for the 'mouth open' detector.) The filtered outputs were squared to emphasize large values. The area under each curve was then computed as the integral between the zero-crossings.

The distribution of these measures over each one-minute video was then described in a histogram. The histogram employed 6 exponentially spaced bins for outputs ranging from 10^0 through 10^5. Two histograms were generated for each filter output: One for curves above zero (capturing information about the temporal texture of facial actions themselves), and one for curves below zero (capturing information about the dynamic intervals between facial actions). These are referred to as event and interval descriptors respectively. Histograms were computed for each of the 8 temporal filters and each of the 20 facial actions, resulting in a total of 320 histograms per video (8 temporal filters X 20 facial actions X positive/negative).

Machine learning classifier for detecting faked expressions of pain. The bags of temporal features comprised the input representation that was used to train a nonlinear support vector machine (SVM) with Gaussian kernel. More information about SVMs is available in [5-7]. The training signal consisted of a binary value of [1, -1] to indicate whether the expression was real or faked. A hard margin SVM was employed, and the standard deviation of the Gaussian kernel was 1. The SVM was trained using the following equation for the kernel matrix, K:

$$K(x_i,x_j) = \exp(-c||x_i-x_j||^2) \qquad (3)$$

where x_i is training vector i, and c is a normalization constant. The core optimization algorithm was based on non-negative least squares [8]. System output was a continuous value indicating the distance to the separating hyperplane between the classes (the margin).

Training and testing

The computer vision system was tested on each video independently, without within-subject comparison of the real and faked conditions. Because the system is subject to over-fitting, performance was tested on novel data, data that was not used to estimate the parameters. Cross-validation, which provides an unbiased estimate of performance on novel data [9], was employed to test participant-independent deception detection. In this approach, the system is trained on all but one subject's data (48 videos), and all data from the final subject is withheld for testing. The system parameters are then deleted, and the process is repeated, each time holding out a different subject. The set of hold-out subject performances provide a sample from which to estimate system performance.

Because of the post-hoc nature of sequential feature selection, the 5-AU classifier was tested with *double* cross validation. Double cross validation approximates the use of two hold-out sets to test performance, one for choosing the features, and a separate one for testing the final model. In this approach, we first employ single cross validation testing where data from 23 of the 25 participants are used for training, the 24th is held out for testing, and the process is repeated, each time holding out a different one of the 24 subjects. Feature selection was based on the single cross-validation performance. After the features were selected, the final model was then tested on the 25th participant. The entire feature selection process was then repeated for all 25 participants. The five selected features were consistent across the 25 hold-out cases.

Evaluation

Expression detection performance was assessed using a measure from signal detection theory, called the area under the ROC (receiver operating characteristic) curve [10]. The ROC curve is obtained by plotting true positives against false positives as the decision threshold ranges from one extreme (0 detections and 0 false positives) to the other (100% detections and 100% false positives).The area under the ROC (A') has a range of 0.5 (chance) to 1 (perfect discrimination). In order to estimate the area under the ROC curve from the 25 cross-validation cases, the outputs for each pain participant were concatenated prior to computing the ROC. For this to hold, the real-valued predictions for each participant obtained from the application of 25 different classifiers were assumed to be comparable. The SVM algorithm includes a normalization mechanism (the unity product of weights and outer support vectors) that allowed for the comparison of outputs from different classifiers. A measure of percent correct was also provided for a decision threshold at equal error rate, where the rate for false positives and false negatives are equal. This was estimated from the A' score by assuming unit normal distributions for target and nontarget populations, with means separated by a distance that produces a given A' score. The threshold was then set to the equal error rate, and the probability of correct response was computed from the distributions.

For statistical significance tests, percent correct was compared using a Z-test to compare two proportions. Comparisons to chance were 1-tailed. Comparisons between two classifiers were 2-tailed.

Evaluating dynamics: Classification performance was tested for temporal integration window sizes ranging from 100 frames (3.3 seconds) to 60 seconds.

Performance for a temporal window of length N was computed by dividing the 60 second video into segments of length N with 50% overlap, computing bags of temporal features for each segment, and then pooling the samples across temporal position for training an SVM. Generalization to novel subjects was tested using cross-validation.

A measure of the duration of mouth openings as well as the intervals between mouth openings was extracted as follows. The CERT output for mouth opening was first smoothed with a Gaussian filter (σ^2=10 frames). We label this y. An estimate of the duration of mouth openings, τ, was obtained by assigning a threshold, θ, at the 75th percentile for each subject, and measuring the duration of continuous frames for which $y-\theta > 0$. Similarly, an estimate of the temporal intervals between mouth openings consisted of the compliment of τ.

Supplemental References

1. Trochim, W.M.K. (2005). Research Methods: The Concise Knowledge Base. (Cincinnati, OH: Atomic Dog Publishing).
2. Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., and Bartlett, M. (2011). The computer expression recognition toolbox (CERT). Proc. In IEEE International Conference on Automatic Face and Gesture Recognition, 298 - 305.
3. Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior Recognition via Sparse Spatio-Temporal Features. In International Conference on Computer Vision, Workshop on Visual Surveillance, 65-72.
4. Demirdjian, D. and Wang, S. (2009). Recognition of Temporal Events using Multiscale Bags of Features.In IEEE Workshop on Computational Intelligence for Visual Intelligence.
5. Vapnik, V. (1995). The Nature of Statistical Learning Theory. (New York: Springer-Verlag).
6. Christianini, N., and Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and other kernel-based learning methods. (Cambridge University Press).
7. Gunn, S.R. (1998). Support Vector Machines for Classification and Regression. Technical Report. Image Speech and Intelligent Systems Research Group, University of Southampton.
8. Gunn, S.R. (2001). MATLAB support Vector Machine Toolbox. Available from: http://www.isis.ecs.soton.ac.uk/resources/svminfo/.

9. Tukey, J.W. (1958). Bias and confidence in not-quite large samples. Ann. Math. Statist. *29*, 614.
10. Green, D.M., and Swets, J.A. (1966). Swets, Signal Detection Theory and Psychophysics. (New York: Wiley).