Expert Systems With Applications 149 (2020) 113305

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Enhanced deep learning algorithm development to detect pain intensity from facial expression images

Ghazal Bargshady[a,*], Xujuan Zhou[a], Ravinesh C. Deo[b], Jeffrey Soar[a], Frank Whittaker[c], Hua Wang[d]

[a] School of Management and Enterprise, University of Southern Queensland, Springfield, QLD 4300, Australia
[b] School of Sciences, University of Southern Queensland, Springfield, QLD 4300, Australia
[c] Nexus e-Care, Adelaide, Australia
[d] Victoria University, Melbourne, Australia

ARTICLE INFO

Article history:
Received 30 July 2019
Revised 29 January 2020
Accepted 10 February 2020
Available online 16 February 2020

Keywords:
Facial expression
Pain detection
Deep neural networks
Expert systems in healthcare
Machine learning

ABSTRACT

Automated detection of pain intensity from facial expressions, especially from face images that show a patient's health, remains a significant challenge in the medical diagnostics and health informatics area. Expert systems that prudently analyse facial expression images, utilising an automated machine learning algorithm, can be a promising approach for pain intensity analysis in health domain. Deep neural networks and emerging machine learning techniques have made significant progress in both the feature identification, mapping and the modelling of pain intensity from facial images, with great potential to aid health practitioners in the diagnosis of certain medical conditions. Consequently, there has been significant research within the pain recognition and management area that aim to adopt facial expression datasets into deep learning algorithms to detect the pain intensity in binary classes, and also to identify pain and non-pain faces. However, the volume of research in identifying pain intensity levels in multi-classes remains rather limited. This paper reports on a new enhanced deep neural network framework designed for the effective detection of pain intensity, in four-level thresholds using a facial expression image. To explore the robustness of the proposed algorithms, the UNBC-McMaster Shoulder Pain Archive Database, comprised of human facial images, was first balanced, then used for the training and testing of the classification model, coupled with the fine-tuned VGG-Face pre-trainer as a feature extraction tool. To reduce the dimensionality of the classification model input data and extract most relevant features, Principal Component Analysis was applied, improving its computational efficiency. The pre-screened features, used as model inputs, are then transferred to produce a new enhanced joint hybrid CNN-BiLSTM (EJH-CNN-BiLSTM) deep learning algorithm comprised of convolutional neural networks, that were then linked to the joint bidirectional LSTM, for multi-classification of pain. The resulting EJH-CNN-BiLSTM classification model, tested to estimate four different levels of pain, revealed a good degree of accuracy in terms of different performance evaluation techniques. The results indicated that the enhanced EJH-CNN-BiLSTM classification algorithm was explored as a potential tool for the detection of pain intensity in multi-classes from facial expression images, and therefore, can be adopted as an artificial intelligence tool in the medical diagnostics for automatic pain detection and subsequent pain management of patients.

1. Introduction

Pain is a complex and a rather an individual experience that poses several challenges in terms of its precise measurements and medical diagnosis (Ashburn & Staats, 1999). However, in current methods, when providing evidence-based treatments for pain management, many clinicians utilize valid and standard pain outcome measures. Currently, there exists no effective and reliable method for objectively quantifying an individual's experience of the pain. Clinicians and health care organizations mainly use a patient's self-report to determine the intensity of pain; however, these approaches may not be effective and could be relatively inaccurate. They may have some degree of limitation, and cannot be used with young children, individuals with certain types of

* Corresponding author at: University of Southern Queensland (USQ) Springfield Campus, 37 Sinnathamby Blvd, Springfield Central, QLD 4300, Australia.
E-mail addresses: ghazal.bargshady@usq.edu.au (G. Bargshady), xujuan.zhou@usq.edu.au (X. Zhou), ravinesh.deo@usq.edu.au (R.C. Deo), jeffrey.soar@usq.edu.au (J. Soar), frank@nexusonline.com.au (F. Whittaker), Hua.Wang@vu.edu.au (H. Wang).

https://doi.org/10.1016/j.eswa.2020.113305
0957-4174/

neurological impairments and dementia, patients in postoperative care, and those with severe disorders requiring assisted breathing, among the other conditions. To deal with this challenge, machine learning algorithms using facial indicators of pain within an automatic pain detection system can provide a more intelligent approach to investigate the pain level (Ashraf et al., 2009; Cootes, Edwards & Taylor, 2001; Khan, Meyer, Konik & Bouakaz, 2013). Such automated mechanisms, especially relying on big data analytics and feature extraction, can be employed to implement modern decision support systems in the diagnosis and management of pain.

Facial expression is one of the most meaningful and natural ways to interpret pain and emotional states. Recognizing expressions on the human face is a very challenging task since the faces of different humans are presented in different poses, varying in respect to the ages of any patient (Ashraf et al., 2009; Kharghanian, Peiravi & Moradi, 2016). The Facial Action Unit Codes (FACS) describe the facial expressions in terms of 46 component movements or action units (AUs), which roughly correspond to the individual facial muscle movements (Ekman & Friesen, 1978). The Prkachin and Solomon Pain Intensity Scale (PSPI) scale is currently the only metric that can define pain on a frame-by-frame basis with the assistance of Facial Action Unit Codes (FACS) (Prkachin & Solomon, 2008). Machine learning algorithms, implemented as intelligent systems, can offer an alternative mechanism for this task, using a camera to collect the relevant data (*e.g.*, a face image), to detect both the pain and its relative intensity level. This can be deduced from the movements in facial muscles and its correspondence with the PSPI scores. One of the challenges, however, is that the accurate development of a robust and versatile method for the four-classifications of facial images have a significantly diverse range of pain-related features to model and measure the actual intensity level in each patient. The second challenge in this regard was that the nature of facial images is generally unbalanced and have a relatively complex data. For example, the unequal instances for different classes of images requires a versatile expert system. Such a system must create a representative model capturing overall inherent features to attain an unbiased classification or prediction of pain from facial images.

This paper reports on the successful design of an enhanced joint hybrid deep learning approach, utilizing convolutional neural networks (CNN) linked to a joint bidirectional long short term (BiLSTM) neural network algorithm to address existing challenges of pain intensity estimation. The study applies the UNBC-McMaster Shoulder Pain Achieve database (Lucey, Cohn, Prkachin, Solomon & Matthews, 2011), to train the proposed algorithm and subsequently, detect the pain in four different levels efficiently and effectively. To attain this, the popular VGG-Face pre-trainer (Parkhi, Vedaldi & Zisserman, 2015) was fine-tuned and utilized to extract the features from the facial image dataset. To improve the computational efficiency of the algorithm, the full dataset was reduced to only the most significant input features by applying the Principal Component Analysis (PCA), and the outputs of the selected features were then transferred to the CNN-BiLSTM joint network for the classification of pain intensity. The novelty in the study lies in developing a new enhanced joint hybrid CNN-BiLSTM (EJH-CNN-BiLSTM) algorithm that is able to extract and select most prominent features and classify pain levels. The newly proposed system was then conditioned in such a way that the outputs of the machine learning algorithm were further tuned to estimate the pain levels ranging from 0 to 3 [0 = no pain and 3 = strong level of pain]. The contributions of this research work will result in:

1. An efficient and effective algorithm, based on transfer learning, that is designed to extract the most important features by using a fine-tuned VGG-Face and the PCA dimension reduction method.
2. A new enhanced joint hybrid deep learning approach that automatically estimates four-levels of pain from facial expressions is proposed.
3. The newly proposed EJH-CNN-BiLSTM model consists of both the feature extraction and the classification algorithm in a single workflow that can be easily implemented in a real-time online system.

The rest of the paper is organized as follows: In Section 2, a summary of the related works used in the automated pain detection from facial expressions are discussed. In Section 3, an overview of the proposed enhanced deep learning model for pain recognition and its intensity level from facial expression is introduced. Next, the experimental details, obtained results, and validation process of the proposed model are presented and analyzed in Section 4. In Section 5, the discussions and interpretations, limitations, and opportunities for future research works are provided, and finally, the concluding remarks are outlined in Section 6.

2. Related research works

Traditionally, key features representing the human face have been used in machine learning procedure to classify the images of facial expressions, including the detection of pain levels. For example, an Active Appearance Model (AAM) based features combined with the Support Vector Machine (SVM) classifiers have been used to classify pain Ashraf et al. (2009); P Lucey et al. (2011a). In another related work, four-classes were identified using the SVM classifier to estimate the pain intensity (Hammal & Cohn, 2012) whereas another framework based on Relevance Vector Regression models (RVR) was also used to estimate the pain range up to 16 different levels (i.e., numerically organized as 0–15) (Kaltwang, Rudovic & Pantic, 2012). These studies exemplify the importance of machine learning in facial feature extraction and classifications.

Recently, deep learning algorithms have been demonstrated to be an important technique in the image classification field. The results reviewed on these algorithms indicate that deep neural networks have been applied in pain recognition recently since they demonstrated greater effectiveness in both the feature selection and feature extraction such as the enhancement of pre-training methods and transfer learning algorithms. In this paper, we continue this research by applying deep learning algorithms for pain detection using facial expression images.

Several state-of-the-art techniques used to generate results in image classification studies have been based on transfer learning type solutions (Krizhevsky, Sutskever & Hinton, 2012). With the transfer learning technique, instead of starting the learning process from scratch, the process commences from the patterns that have been learned to solve diverse ranges of problems. A comprehensive review of transfer learning showed how to implement a transfer learning solution for image classification problems (Pan & Yang, 2009). One study also demonstrated the extraction of facial features from a pre-trained VGG-Face algorithm, which was then integrated to an LSTM algorithm to utilise the temporal relationships between the video frames (Rodriguez et al., 2017). This model, utilising the UNBC-McMaster Shoulder Pain Expression Archive Database, was seen to outperform the current state-of-the-art AUC metric performance (Rodriguez et al., 2017). A pre-trained CNN (VGG-Face) and LSTM algorithm was also applied to detect pain from the face in the MIntPAIN database dataset (Haque et al., 2018). A fine-tuning process to extract features from pre-trained CNN and combined with Recurrent Neural Networks (RNN) as RCNN was also applied to detect pain intensity from facial expressions (Zhou, Hong, Su & Zhao, 2016). The transfer learning technique was further used to present an algorithm based on CNN

and trained by the Psychological Image Collection at Stirling (PICS) dataset for pain detection from facial expressions (Xu, Cheng, Zhao, Ma & Xu, 2015).

The study of Walecki, Pavlovic, Schuller and Pantic (2017) and Martinez, Rudovic and Picard (2017) proposed different hybrid techniques by combining a deep learning algorithm with the Hidden Markov Model (HMM) for facial expression classification tasks (Martinez et al., 2017; Walecki et al., 2017). By contrast, the study of Xie, Xu and Chuang (2013) proposed a Horizontal Voting, Vertical Voting and the Horizontal Stacked Ensemble method to improve the classification performance of a deep neural network (Xie et al., 2013). They also found that both the linear Horizontal Voting and the Horizontal Stacked Ensemble methods were able to robustly improve the performance of the classification algorithm.

Following earlier works, this study has used the transfer learning technique by applying the customized VGG-Face pre-trained with millions of faces, and then used a hybrid joint deep learning approach including the two-stream CNN-BiLSTM algorithm to classify pain intensity levels. In this study, the BiLSTM algorithm is considered to be a specific type of RNN network and its basic idea is to present each sequence forwards and backwards as two separate hidden states, aimed to capture past and future information, respectively (Dyer, Ballesteros, Ling, Matthews & Smith, 2015; Gers & Schmidhuber, 2001). In the feature extraction stage, we applied PCA onto the extracted features from the customized VGG-Face and then calculated Gaussian noise which was added on the transformed variables in classification part (da Silva & Adeodato, 2011). It should be noted that the PCA method is a mathematical approach that has been seen to perform an orthogonal transformation on the data to convert a set of possibly correlated variables into a set of uncorrelated variables, termed as the principal components (Oja, 1989). This study has applied the Gaussian noise for the proposed framework given that several studies have successfully applied noise in such models proposed over many years. Most of them focussed on the addition of noise during the neural network training as an ancillary tool to improve the generalization capability and the convergence time of a classification algorithm (da Silva & Adeodato, 2011). In the following section, the details and a basic description of the proposed framework is explained.

3. The proposed model

A block-diagram of the proposed model framework established in this study is illustrated in Fig. 1. Basically, it is divided into three primary components that aim to improve the overall efficacy of the algorithm. In the first step, the original images (captured as video frames) were transferred to the pre-processing stage, which applied procedures related to the cropping, resizing and normalizing techniques required to adjust the original images before being incorporated into the feature extraction phase and the model training stage. Subsequently, in a new proposed framework for feature extraction and selection, we applied the fine-tuned pre-trained CNN framework to extract these features. This is later output into the PCA stage, aimed to reduce the dimensionality of the extracted features from the videos images. Since this study has used video-image based data, additional temporal information was necessary to improve the classification model. It should be noted that the BiLSTM algorithm, used as a specific type of RNN neural net, is especially suited for sequential datasets since their neurons do not only have connections between the next layers but also have connections to themselves, which aim to capture the past and future input features. Therefore, the extracted features in a sequence length were transferred into a newly developed Enhanced Joint Hybrid classifier algorithm, denoted in this study as the EJH-CNN-BiLSTM in order to obtain four distinct pain intensity levels. The details of the proposed model are explained in the following subsection.

3.1. Pre-processing

In a real-world scenario, an image dataset may be taken in a variety of conditions such as different orientations, location, scales, and brightness. Therefore, the conventional image pre-processing technique such as the normalization, cropping, and centralizing are applied in these raw images to improve the identification of the images during any experimental phase. For this purpose, the raw images are applied in the pre-processing part of the proposed model and each original image in the database is cropped, and then centralized. We have therefore resized the images to $224 \times 224 \times 3$ pixels because this representation is the most common input size for most of the deep neural network models after cropping including the VGG-Face. To normalize the pixel values for both the training and testing datasets, these data was rescaled to the range of [0,1]. This involved first converting the data type from unsigned integer to float values, and then dividing the pixel values by the maximum value (Schertler, 2014).

$$Normalize : R \to R : x \to \frac{x}{d}\ d = \max_{x\ \in image} \| x \| \tag{1}$$

3.2. The transferring of parameters and feature extraction

Therefore, in this study, to extract the features prevalent on facial images, a new feature extraction technique denoted as the VGG-Face-PCA method was proposed. It consists of a fine-tuned VGG-Face CNN as a pre-trainer, where the outputs are fed into the PCA stage to reduce the dimensionality of the extracted features. The use of a pre-trained CNN algorithm as a facial feature extractor is expected to be useful as a basis for training the CNN algorithm to accurately detect pain from the facial images. There are a few CNN models that were successfully trained for this face recognition task such as VGG-Face. Its architecture proposed by Parkhi et al. (2015) which is achieved state-of-the-art results in extracting features and is relied on a very deep facial recognition CNN architecture (Parkhi et al., 2015). It consists of 5 convolution blocks and 3 fully connected layers including fc6, fc7, and fc8. Each convolution block comprises of two or three convolutional layers with a max-pooling layer to reduce the size of the output feature map. Achieving an optimal pain detection algorithm is a challenging task. There is a great difference between the target task's image set and the pre-trained image set, regardless of the number of categories or image styles. In the retrieval task of the target image set, the visual features of the image are directly extracted by the pre-trained CNN model. Therefore, to make the pre-trained CNN model parameters more suitable for the feature extraction of the target image set, the VGG-Face pre-trained CNN model was fine-tuned as follows: Therefore, we used VGG-Face and retrain it for pain estimation by keeping the convolution layers of this model unchanged while replacing the fully connected layers with a new fully connected layer. In addition, 5 convolutional blocks and the new replaced fully connected layer were retrained by transfer learning. The replaced, fully connected layer was followed by dropout and the size of it was 1024. The output size of the output layer represents the number of pain levels which is 4. The ADAM-optimizer is one of the most popular gradient descent optimization algorithms and faster than other optimizers. We applied the ADAM-optimizer to retrain fine-tuned VGG-Faces since it is a superior optimizer that can tune the parameter automatically during training (Han, Liu & Fan, 2018).

As discussed previously, the extracted features of fine-tuned VGG-Face as outputs are fed into the PCA stage to reduce the

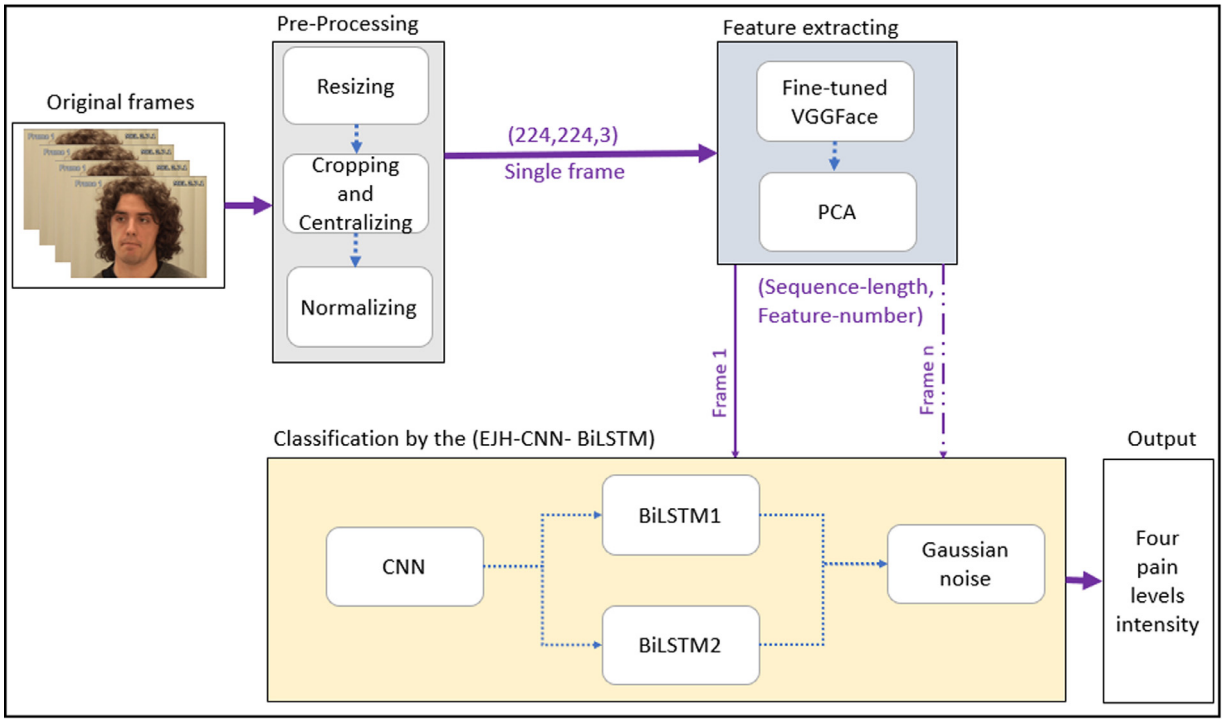

Fig. 1. The proposed EJH-CNN-BiLSTM model designed for pain detection from facial expression image database.

dimensionality of the extracted features. PCA is a dimensionality reduction method that is useful in different applications such as image compression, facial feature extraction, face recognition and finding patterns from large dimensional images (Damale & Pathak, 2018; Li, Zhao, Zhang & Jiao, 2009). It helps to choose the best set of data dimensions that will make the model perform better, and to speed up the algorithm performance (Sun, Chen, Wang & Tang, 2014). Therefore, in this research paper, there are a total of 38,816 features, which have been extracted from the training data set, calculated according to the input shape of the extracted features. For the training data set, these are denoted as (9704, 4) where the number 9704 refers to the number of training images and so, we are able to obtain a product $9704 \times 4 = 38,816$. In addition, the 4 distinct output features (per image) extracted from the fine-tuned VGG-Face are transferred into the PCA algorithm with an aim to reduce the dimensionality of the extracted features and also to speed up the classification algorithm. It would thus be of interest to be able to discover "sparse principal components" such as sparse vectors spanning a low-dimensional space. To achieve this, it is necessary to reduce some of the explained variance and the orthogonality of the principal components. For doing this, the explained variance for each component is calculated by Python software.

The dimensionality reduction process is achieved through an orthogonal, linear projection operation. Without loss of generality, the PCA operation can be defined as (Goodfellow, Bengio & Courville, 2016):

$$Y = XC \tag{2}$$

With $Y \in R^{S \times P}$ is the projected data matrix that contains P principal components of X with $P \leq N$. So, the key is to find the projection matrix $C \in R^{N \times P}$, which is equivalent to find the eigenvectors of the covariance matrix of X, or alternatively solve a singular value decomposition (SVD) problem for X [].

$$X = \cup \sum V^T \tag{3}$$

where $U \in R^{s \times s}$ and $V \in R^{N \times N}$ are the orthogonal matrices for the column and row spaces of X, and Σ is a diagonal matrix containing the singular values, λ_n, for $n = 0, \bullet\bullet\bullet, N-1$, non-increasingly lying along the diagonal. It can be shown that the projection matrix C can be obtained from the first P columns of V with

$$V = [v_1, \ldots, v_N] \tag{4}$$

and

$$C = [c_1, \ldots, c_P] \tag{5}$$

where $v_n \in R^{N \times 1}$ is the n^{th} right singular vector of X, and $c_n = v_n$. In fact, the singular values contained in Σ are the standard deviations of X along the principal directions in the space spanned by the columns of C. Therefore, λ_n^2 becomes the variance of X projection along the nth principal component direction. It is believed that variance can be explained as a measurement of how much information a component contributes to the data representation. One way to examine this is to look at the cumulative explained variance ratio of the principal components, given as (Goodfellow et al., 2016):

$$R_{cev} = \frac{\sum_{n=1}^{P} \lambda_n^2}{\sum_{n=1}^{N} \lambda_n^2} \tag{6}$$

Fig. 2 describes that selecting 2 components is able to preserve majority of the total variance of the input data. A vital part of using PCA in practice is the ability to estimate how many components are needed to describe the data. This can be determined by looking at the cumulative explained variance ratio as a function of the number of components. This graph quantifies how much of the total, 4-dimensional variance is contained within the components. For example, we see that with the first 1 component contain approximately 78% of the variance, while we need around 2 components to describe close to 100% of the variance.

Therefore, if $\rho =$ vectors of length of number of frames, and $f =$ number of features after PCA reduction then $\rho \times f$ used as potential inputs are passed into the EJH-CNN-BiLSTM algorithm. In

Table 1
The structure used for Conv1D-1 and Conv1D-2 used in the EJH-CNN-BiLSTM proposed model.

Input shape	Layer name	Filter size	Kernel size	Layer parameters	Padding
length = 2 feature = 2 input shape = (2,2)	Conv1D-1 Conv1D-2	256 128	10 10	activation = ReLU activation = ReLU	same same

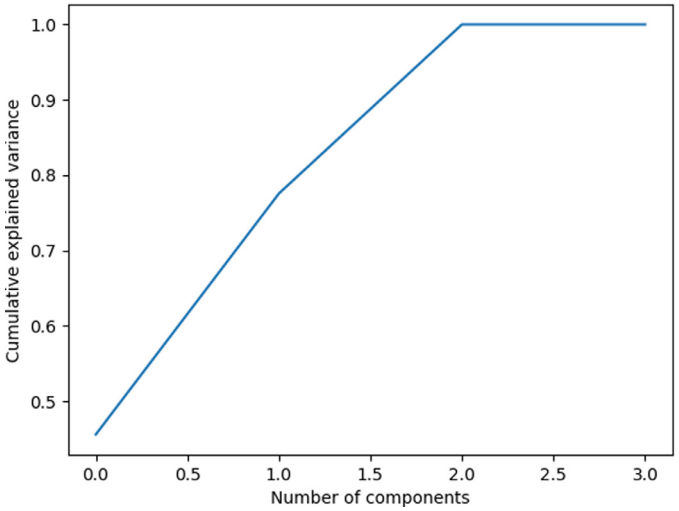

Fig. 2. The Principal Component Analysis (PCA) explained variance ratio for the four components.

this case, based on the above discussion, $f = 2$ and so, we selected length of the number of frames for the ConvD1 layer ($\rho = 2$) with the input shape (2, 2) for the ConvD1 layer. It was also found that $\rho = 20$ worked relatively well for BiLSTM algorithm and the input shape for the BiLSTM algorithm was selected according to (20,2) in order to enter the EJH-CNN-BiLSTM classifier system.

3.3. THE EJH-CNN-BiLSTM classifier

A new hybrid deep learning classifier denoted as the Enhanced Joint Hybrid CNN-BiLSTM (EJH-CNN-BiLSTM) was developed in this research to classify pain levels in four classifications. The extracted and reduced features from the VGG-Face-PCA were transformed into the new developed EJH-CNN-BiLSTM classifier. The EJH-CNN-BiLSTM consists of two streams hybrid deep learning model which their outputs joint as indicated in the classification section of the Fig. 1. We applied CNN-BiLSTM hybrid deep learning in both the streams. At first, the selected features were transferred into CNNs includes two ConvD1 which outputs of ConvD1-1 were transferred into ConvD1-2 as inputs. Then, the outputs of the ConvD1-2 were entered to both the BiLSTM1 (stream 1) and BiLSTM2 (stream 2) separately. Finally, the outputs of both the BiLSTM1 and BiLSTM2 were merged and the Gaussian noise applied to classify four pain levels. In the following details of each step are described:

3.3.1. Structure of CNNs

Convolutional Neural Networks are extremely powerful models often used in the space of computer vision. CNNs are fast and recently, Conv1Ds have also shown success on sequential learning problems and continue to be explored in this new space. We designed two CNN structure as Conv1D-1 and Conv1D-2. Table 1 shows structure of them.

3.3.2. Structure of RNNs

Since the data type is video and contains video image frames, we applied the RNN in this model to improve classification. RNNs are suited to sequential data since their neurons have connections (weights) between the next layers and keep information from previous inputs. BiLSTM as an RNN type has an elegant solution for each sequence forward and backward as two separate hidden states to capture past and future information, respectively. So, the proposed algorithm is designed based on BiLSTM. The experimental results show incredible improvement in compared with model which used only two streams BiLSTM for this classification problem. Table 2 shows the structure of the BiLSTM1 and BiLSTM2.

Table 2
Structures of the BiLSTM1 and BiLSTM2 used in the EJH–CNN-BiLSTM proposed model.

Layer name	Layer parameters
BiLSTM1	Input shape = (20,2) filter = 256 dense = 4096 activation = ReLU Gaussian noise = PCA-Std dropout = 0.5
BiLSTM2	Input shape = (20,2) filter = 32 dense = 4096 activation = ReLU Gaussian noise = PCA-Std dropout = 0.5

3.3.3. Merging and applying the Gaussian noise

The Gaussian noise calculated from PCA added to outputs of the dense layers during the training of the proposed EJH-CNN-BiLSTM framework. Several studies report the addition of noise during neural networks training as a tool to improve the generalization capability and convergence time. The previous studies focused on creating new input patterns by adding random noise drawn from a Gaussian distribution increased generalization power as long as the amount of noise was kept sufficiently small to have no disruptive effect on the desired output. Therefore, Gaussian noise is added to the PCA components and calculated by *Numpy.STD* of *Keras* library in *Python* and the calculated amount added to outputs of the dense layers. The *Numpy.STD* computes the Standard Deviation (SD) of the given data. SD is measured as the spread of data distribution in the given data set.

$$SD = \sqrt{mean(abs(x - x.mean())^2)} \tag{7}$$

3.4. Overview of the EJH-CNN-BiLSTM

The details of the proposed EJH-CNN-BiLSTM model summarized in Algorithm 1. Five epochs and 48 batches to train and test the proposed algorithm used.

4. Experimental results

Modelling experiments were conducted to validate the effectiveness of the EJH-CNN-BiLSTM proposed model and algorithm implemented under an Intel Core i7 @ 3.3 GHz and 16 GB memory computer. Python software (Sanner, 1999) was used for the model construction and prototyping, since it has freely available

Algorithm 1: EJH-CNN-BiLSTM algorithm.

```
Step 1:   Cropped, resized, normalized data.
Step 2:   Fine tuning the VGGFace
Step3:    For (epoch = 0, epoch = 5, epoch++)
Step 4:       Batches ← 48
Step 5:       Pre-train the fine-tuned VGG-Face
Step 6:   End For
Step 7:   Extract features from fine-tuned VGG-Face
Step 8:   Dimension reducing by PCA
Step 9:   Select features from PCA
Step 10:  calculate the Gaussian Noise by PCA
Step 11:  Modelling EJH-CNN-BiLSTM
Step 12:  Appling Gaussian noise to Dense layer of EJH-CNN-BiLSTM
Step 13:  For (epoch = 0, epoch = 5, epoch++)
Step 14:      Batches ← 48
Step 15:      train, test EJH-CNN-BiLSTM
Step 16:  End For
Step 17:  estimate four classes of pain intensity level (class with the highest output selected).
Step 18:  Evaluate results
```

Fig. 3. The image frame samples of the UNBC-McMaster Shoulder Pain Achieve database (Patrick Lucey et al., 2011) used in this study.

library suits for deep learning such as Keras (Ketkar, 2017), TensorFlow (Abadi et al., 2016), Scikit-learn (Pedregosa et al., 2011), Matplotlib (Hunter, 2007). Keras allows for easy and fast prototyping and supports both convolutional networks and recurrent networks. Matplotlib is a Python 2D plotting library that is used for plotting and statistical analysis of modelling data.

The hypotheses tested in these experiments included: (1) The fine-tuned VGG-Face is more effective in extracting features than the commonly used VGG-Face for this dataset. (2) The designed parameter transferring feature extracting methods, along with PCA, selects features more effectively and efficiently. (3) The new designed EJH-CNN-BiLSTM model effectively classifies four levels of pain from the facial expression dataset.

4.1. Study dataset

The study dataset provides the image's frames of video sequences with each frame coded in terms of the PSPI score. The database provides 200 sequences across 25 subjects, which totals 48,398 images. Fig. 3 shows some of the images indicated by PSPI.

The database used is unbalanced and hence, it has been very challenging to perform the modelling experiments. As shown in Fig. 4, the number of no pain images PSPI score labels are higher than other labels and the number of images with PSPI labels greater than 6 is few in this database. Therefore, based on the specific character of the database it is likely that any model gets biased towards the prediction of no-pain at the cost of missing pain frames. Using imbalance data is basically intentionally biasing data to get an interesting result. To deal with this issue, in this study the database was balanced using under resampling techniques to reduce the majority class (no-pain class). Moreover, full sequences included only no pain (PSPI = 0) frames were removed and some no-pain frames from the beginning and end of sequences which included no-pain frames were removed. A total of 10,783 images were applied in this research. To create the training sequence for the RNN algorithm, the frames are firstly sorted out in time domain, and subsequently, each sequence is set to 20 frames. For classifying pain into four levels, the database was divided into four parts including no-pain (PSPI = 0), weak-pain (PSPI = 1), mid-pain (PSPI = 2 and 3), and strong-pain (PSPI > = 4) as shown in Table 3.

Table 3
Divided levels of pain in the database for four levels based on PSPI codes of images' frames.

PSPI score in UNBC McMaster database	Pain level	Number of images
0	No pain	2483
1	Weak pain	2871
2 and 3	Mid pain	3757
4 and >4	Strong pain	1672

4.2. Performance evaluation metrics

In this study, several performance evaluations measures, including classification accuracy, Mean Absolute Error (MAE), Mean Squared Error (MSE), Area under Curve (AUC), and *F*-measure were used to evaluate the performance of the proposed model. Classification accuracy is the ratios of the number of correct predictions

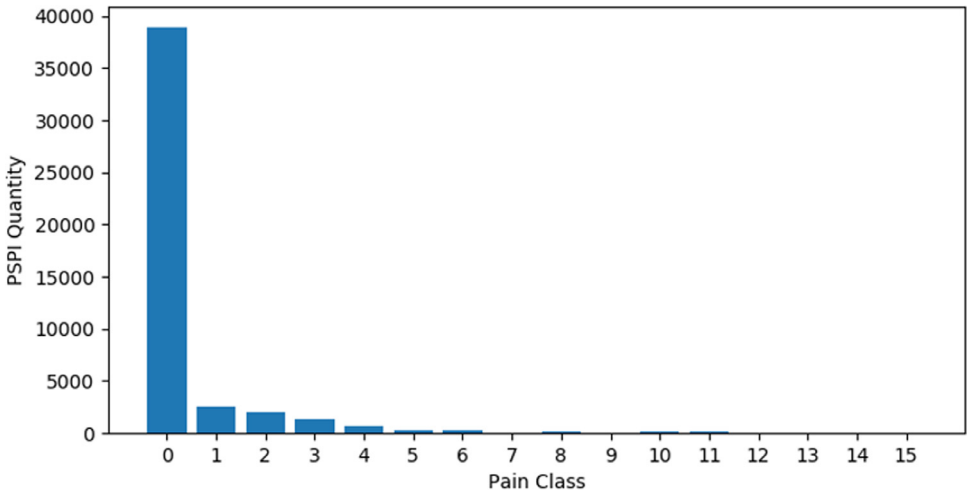

Fig. 4. The amount of PSPI code per each class in the UNBC McMaster Shoulder Pain database.

to the total number of input samples in both training and testing sets. MAE is the average of the difference between the original values and the predicted values. It gives us the measure of how far the predictions were from the actual output. However, they do not give any idea of the direction of the error whether the data is under predicting or over predicting. MSE is quite like MAE, and the only difference is that MSE takes the average of the square of the difference between the original values and the predicted values. The advantage of MSE is that it is easier to compute the gradient, whereas MAE requires complicated linear programming tools to compute the gradient.

The proposed algorithm calculated True Positive Rate (TPR) and False Positive Rate (FPR) metrics and applied them for the measuring of the AUC and F measures. AUC is one of the most widely used metrics for evaluation. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. TPR corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. FPR corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. FPR and TPR both have values in the range [0, 1]. The *F*-measure is used to measure a test's accuracy, and it balances the use of precision and recalls doing it. The F measure can provide a more realistic measure of a test's performance by using both precision and recall. F-measure and precision are calculated based on False Positive (FP), True Negative (TN), False Negative (FN), and True Positive (TP). AUC is the area under the curve of the plot FPR vs TPR at different points in [0, 1] (Powers, 2011). The area under the receiver operating characteristic (ROC) curve (i.e., the AUC) was used as a performance measure for these machine learning algorithms following other works (e.g., Bradley, 1997). The details of the AUC calculation are described in Bradley (1997).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{11}$$

4.3. Results

The obtained results indicated the EJH-CNN-BiLSTM has high performance in detecting pain in four levels. Table 4 shows the average of training MSE, training MAE, training accuracy, test MSE, test MAE, test accuracy, and AUC of the average for 10-fold cross-validation performance measurement. The average accuracy of 91.2% for the training set, 90% accuracy for the testing set, and 98.4% for AUC were achieved.

Moreover, the proposed EJH-CNN-BiLSTM evaluated for each class based on TP, *F*-measure, precision, and AUC in 10-fold cross-validation. Table 5 shows the average performance measuring of the proposed algorithm for each pain levels.

During the experimental tests, different versions of deep learning algorithms were designed and compared with the proposed EJH-CNN-BiLSTM model (See Table 6). The results indicate that the proposed model has significant performance improvement since its AUC and accuracy is higher than other models mentioned in the Table 6. All of them were tested in the same selected balanced dataset with total 10,783 images by applying the 10-fold cross-validation technique for 25 subjects.

The 10-folds cross-validation is not a subject-independent performance measurement since images from the same subject may appear in both the training and testing sets. To control for this limitation, we next performed a leave-one-subject-out cross-validation in which subjects of the training are removed from the testing. This validation allows exploring how the proposed method for pain intensity measurement generalizes to a new set of subjects who were not part of the training set. The leave-one-subject-out cross-validation consists of building 25 classifiers for each one of the four levels of pain intensity and iterating the process. Table 7 shows the achieved results and compared them with state-of-the-art results for the same task.

The comparison results of the proposed EJH-CNN-BiLSTM model with the-state-of-the art results show our proposed framework is significantly more effective. In terms of error measuring, it has fewer errors measured by MAE and MSE in comparison with results obtained by Rodriguez et al. (2017). Furthermore, it achieved the highest accuracy by 85% in comparison with the results of the other hybrid deep learning algorithms done by Rodriguez et al. (2017) and Bellantonio et al. (2016). However, the AUC achieved by Rodriguez et al. (2017) is higher than our proposed model. Also, the comparison results show the EJH-CNN-BiLSTM obtained high performance in accuracy and f-measure in comparison with the results achieved by Hammal and

Table 4
The average performance of the algorithm for 10-fold cross validation.

Training MSE	Training MAE	Training accuracy	Test MSE	Test MAE	Test accuracy	AUC
0.03	0.06	91.2%	0.04	0.07	90%	98.4%

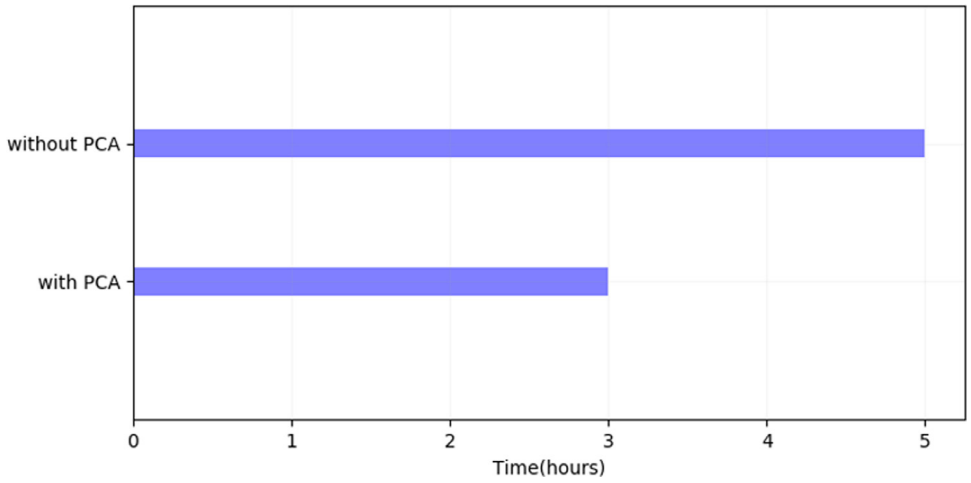

Fig. 5. Comparing the model running time of the algorithm with or without PCA.

Table 5
The average performance of the algorithm per each class.

Class	TP	*F*-measure	Precision
No pain	87.70%	87.51%	87.5%
Weak pain	88.50%	89.10%	90.02%
Mild pain	90.30%	87.93%	86%
Strong pain	93.20%	95%	96.54%

Table 6
Comparing the performance of the proposed model with different versions of the deep learning algorithm designed during experimental test based on average amount of accuracy and AUC for 10-fold cross validation.

No	Model	Accuracy	AUC
1	CNN	54.45%	44.8%
2	VGGFace + CNN	57.3%	52%
3	VGGFace + BiLSTM1	65%	75.3%
4	VGGFace + BiLSTM1 + BiLSTM2	73%	78.5%
5	EJH-CNN-BiLSTM	**91.2%**	**98.4%**

Cohn (2012) tested in four pain levels. The other noticeable point of the results of the proposed model is its ability in the four-levels classification.

The proposed EJH-CNN-BiLSTM algorithm is efficient in terms of running time. The PCA used in feature selection of the algorithm accelerates the algorithm during training and testing. We sped up the running time of the algorithm for 3 h for the whole process by in Core i7 computer with 16 GB RAM. Fig. 5 shows the algorithm running time before and after using the PCA.

5. Discussion

The analyses of the results obtained indicate that the involvement of fine-tuned pre-training and the proposed EJH-CNN-BiLSTM method, when combined with the PCA with additive Gaussian noise can improve the accuracy of the algorithm. By comparing and analyzing the obtained results, we can conclude as follows: In terms of small datasets, CNNs get very low classification accuracy. In addition, they have a huge number of parameters, they have not been fully trained. Moreover, pre-training and fine-tuning are very effective transfer learning techniques for image classification. As the results show, the fine-tuning network can increase the accuracy of the algorithm. In terms of efficiency, PCA reduces the dimensionality of the selected features then accelerates the algorithm's running time. Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserve the maximal data variance. The Principal Component Analysis method with additive Gaussian noise significantly improves the performance of the algorithm.

There are some limitations in terms of the number of pain image datasets from facial expressions in pain detection research study on face images. One of the challenges, however, is that most of the research into facial expressions, especially in the area of fa-

Table 7
Comparison of the proposed framework results with the state-of-the-art results in the UNBC McMaster Shoulder Pain database to detect pain from facial expressions based on leave-one-subject-out.

Ref	Level	Classifier	AUC (%)	Accuracy (%)	*F*-measure (%)	MSE (%)	MAE (%)	Size of data
(Patrick Lucey et al., 2011)	2	SVM	83.9	–	–	–	–	All
(P Lucey et al., 2011)	2	SVM	84.7	–	–	–	–	All
(Rodriguez et al., 2017)	2	CNN-LSTM	93.3	83.1	–	74	50	Down-up
(Bellantonio et al., 2016)	3	CNN-RNN	–	61.9	–	–	–	Down-up
(Hammal & Cohn, 2012)	4	SVM	–	80	60	–	–	16,657 image
Our proposed framework	*4*	*EJH-CNN-BiLSTM*	***88.7***	***85***	***78.2***	***20.7***	***17.6***	***10,783 image***

cial pain detection, currently lacks a standard database. This makes it relatively difficult to train an accurate facial image recognition system to act as a robust platform in recognizing the pain. Future work may study this algorithm in different pain face image or video frames databases or in accelerating feature extracting section for real-time applications.

6. Concluding remarks

Appropriate pain management strategies and automated detection of pain intensity, based on facial images by means of an expert system, is a growing area of research in health informatics. In this study, a novel hybrid joint CNN-BiLSTM deep learning approach for four level pain recognition on facial images is proposed. To achieve satisfactory results in terms of pain intensity estimation, the fully connected layer of the VGG-Face was improved for this task by adding an extra fully connected layer and the dimensionality of the extracted features reduced by PCA to increase the overall computational efficiency of the proposed algorithm. The reduced extracted features, which were the most useful patterns for pain intensity estimation, feed to the classification section of the newly developed EJH-CNN-BiLSTM model. Moreover, experimental results demonstrated that the proposed EJH-CNN-BiLSTM method significantly improves the performance achieved by using the conventional approach. The enhanced algorithm obtained an AUC of 98.4% and test accuracy of 90% on the balanced UNBC-McMaster Shoulder Pain database. Furthermore, for generalizing the proposed algorithm and comparing it with other similar research works, we applied the leave-one-subject-out performance measuring technique as well, and obtained results indicate the effectiveness of the proposed algorithm for unseen data. The artificial intelligence method developed in this study can have useful implications for the medical diagnostic areas, particularly, supporting the implementation of automatic pain management practices for clinicians and other medical researchers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Ghazal Bargshady: Conceptualization, Methodology, Software, Validation. **Xujuan Zhou:** Conceptualization, Methodology. **Ravinesh C. Deo:** Conceptualization, Investigation, Writing - review & editing. **Jeffrey Soar:** Writing - review & editing. **Frank Whittaker:** Writing - review & editing, Investigation. **Hua Wang:** Writing - review & editing.

Acknowledgement

This study was funded by the Australian Research Council (ARC) (grant number LP150100673). The authors declare no conflict of interest.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. *Paper presented at the 12th symposium on operating systems design and implementation savannah*.

Ashburn, M. A., & Staats, P. S. (1999). Management of chronic pain. *The Lancet, 353*(9167), 1865–1869.

Ashraf, A. B., Lucey, S., Cohn, J. F., Chen, T., Ambadar, Z., Prkachin, K. M., et al. (2009). The painful face–pain expression recognition using active appearance models. *Image and Vision Computing, 27*(12), 1788–1796.

Bellantonio, M., Haque, M. A., Rodriguez, P., Nasrollahi, K., Telve, T., Escalera, S., et al. (2016). Spatio-temporal pain recognition in CNN-based super-resolved facial images. *Paper presented at the video analytics. Face and facial expression recognition and audience measurement*.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 23*(6), 681–685.

Damale, R. C., & Pathak, B. V. (2018). Face recognition based attendance system using machine learning algorithms. *Paper presented at the 2018 second international conference on intelligent computing and control systems (ICICCS)*.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. *Paper presented at the ACL 2015*.

Ekman, P., & Friesen, W. V. (1978). *Facial action coding system: Investigator's guide*. Palo Alto, CA: Consulting Psychologists Press.

Gers, F. A., & Schmidhuber, E. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks, 12*(6), 1333–1340.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hammal, Z., & Cohn, J. F. (2012). Automatic detection of pain intensity. *Paper presented at the proceedings of the 14th ACM international conference on multimodal interaction*.

Han, D., Liu, Q., & Fan, W. (2018). A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems with Applications, 95*, 43–56.

Haque, M. A., Bautista, R. B., Noroozi, F., Kulkarni, K., Laursen, C. B., Irani, R., et al. (2018). Deep multimodal pain recognition: A database and comparison of spatio-temporal visual modalities. *Paper presented at the 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90.

Kaltwang, S., Rudovic, O., & Pantic, M. (2012). Continuous pain intensity estimation from facial expressions. *Paper presented at the international symposium on visual computing*.

Ketkar, N. (2017). Introduction to keras. In *Deep learning with python* (pp. 97–111). Berkeley, CA: Springer.

Khan, R. A., Meyer, A., Konik, H., & Bouakaz, S. (2013). Pain detection through shape and appearance features. *Paper presented at the 2013 IEEE international conference on multimedia and expo (ICME)*.

Kharghanian, R., Peiravi, A., & Moradi, F. (2016). Pain detection from facial images using unsupervised feature learning approach. *Paper presented at the 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Paper presented at the advances in neural information processing systems*.

Li, J., Zhao, B., Zhang, H., & Jiao, J. (2009). Face recognition system using SVM classifier and feature extraction by PCA and LDA combination. *Paper presented at the 2009 international conference on computational intelligence and software engineering*.

Lucey, P., Cohn, J. F., Matthews, I., Lucey, S., Sridharan, S., Howlett, J., et al. (2011). Automatically detecting pain in video through facial action units. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 41*, 664–674.

Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., & Matthews, I. (2011). Painful data: The UNBC-McMaster Shoulder Pain expression archive database. *Paper presented at the face and gesture 2011*.

Martinez, L., Rudovic, O., & Picard, R. (2017). Personalized automatic estimation of self-reported pain intensity from facial expressions. *Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems, 1*(01), 61–68.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). *Deep face recognition*. Swansea, UK: Paper presented at the bmvc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*(Oct), 2825–2830.

Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol., 2*(1), 37–63.

Prkachin, K. M., & Solomon, P. E. (2008). The structure, reliability and validity of pain expression: Evidence from patients with Shoulder Pain. *Pain, 139*(2), 267–274.

Rodriguez, P., Cucurull, G., Gonzàlez, J., Gonfaus, J. M., Nasrollahi, K., & Moeslund, T. B. (2017). Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE Transactions on Cybernetics*, (99), 2168–2267.

Sanner, M. F. (1999). Python: A programming language for software integration and development. *Journal of Molecular Graphics and Modelling, 17*(1), 57–61.

Schertler, N. (2014). Improving jpeg compression with regression tree fields. *(Master)*. Dresden Germany: Technische Universität Dresden Retrieved from https://tu-dresden.de/ing/informatik/smt/cgv/ressourcen/dateien/lehre/ergebnisse_studentischer_arbeiten/masterarbeiten/nico_schertler_ss14/files/Thesis.pdf?lang=en.

da Silva, I. B. V., & Adeodato, P. J. L. (2011). PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets. *Paper presented at the 2011 international joint conference on neural networks*.

Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. *Paper presented at the 27th international conference on neural information processing systems*.

Walecki, R., Pavlovic, V., Schuller, B., & Pantic, M. (2017). Deep structured learning for facial action unit intensity estimation. *Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition*.

Xie, J., Xu, B., & Chuang, Z. (2013). Horizontal and vertical ensemble with deep representation for classification. *Paper presented at the ICML 2013*.

Xu, M., Cheng, W., Zhao, Q., Ma, L., & Xu, F. (2015). Facial expression recognition based on transfer learning from deep convolutional networks. *Paper presented at the 11th IEEE international conference on natural computation (ICNC)*.

Zhou, J., Hong, X., Su, F., & Zhao, G. (2016). Recurrent convolutional neural network regression for continuous pain intensity estimation in video. *Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition workshops*.