Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

# Pain assessment from facial expression images utilizing Statistical Frei-Chen Mask (SFCM)-based features and DenseNet

Sherly Alphonse[1*], S. Abinaya[1] and Nishant Kumar[1]

**Abstract**

Estimating pain levels is crucial for patients with serious illnesses, those recovering from brain surgery, and those receiving intensive care etc. An automatic pain intensity estimator is proposed in this study that gathers information about pain and intensity from the user's expressions. The faces in the database are first cropped using a 'Chehra' face detector, which performs well even in wildly uncontrolled environments with a wide range of lighting and position fluctuations. The suggested technique extracts the beneficial and distinct patterns from facial expressions using novel Statistical Frei-Chen Mask (SFCM)-based features and DenseNet-based features. As it offers quick as well as accurate pain identification and pain intensity estimation, the Radial Basis Function Based Extreme Learning Machine (RBF-ELM) is employed for pain recognition and pain intensity level estimation using the characteristics. All the data is kept, updated and protected in the cloud because availability and high-performance decision-making are so important for informing physicians and auxiliary IoT nodes (such as wearable sensors). In addition, cloud computing reduces the time complexity of the training phase of Machine Learning algorithms in situations where it is possible to build a complete cloud/edge architecture by allocating additional computational resources and memory in use. The facial expression images from the UNBC-McMaster Shoulder Pain Expression Archive and 2D face dataset are used to test the proposed method. The measurement of pain intensity uses four stages. When compared to the results from the literature, the proposed work attains enhanced performance.

**Keywords** Pain, Intensity, RBF-ELM, DenseNet, Features, Facial expression

## Introduction

Internet of Things (IoTs) has become extremely popular in recent years. IoT has therefore been used in automated driving, smart housing, and traffic congestion monitoring. The Internet of Things (IoT) can be utilized in Artificial Intelligence (AI) with a healthcare framework. The healthcare sector is crucial for creating jobs, funding medical facilities, generating revenue, and contributing to a civilized society in smart cities. Clinical trials, hospitals, telemedicine, outpatient clinics, medical equipment, health insurance, nurses, physicians, and other healthcare professionals are some of these industries [1]. The introduction of mobile health and electronic health facilities in a new technological era has improved the healthcare sectors. Patients in critical condition and those who have experienced surgery require ongoing observation. The self-reports of the patients are used by medical professionals to estimate their pain; however, this information is unreliable. Several techniques for measuring pain include the Visual Analog Scale (VAS) and clinical interviews. Linear Analogue Self-Assessment is another technique for assessing pain (LASA). As VAS performs poorly in comparison to other techniques, it cannot be used as the primary metric for determining the level of pain severity [1]. Doctors must carefully consider the patient's level

*Correspondence:
Sherly Alphonse
sherly.a@vit.ac.in
[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India

of pain while determining the best course of treatment, especially for those with speech disorders. The number of levels used to gauge the severity of the pain varies widely. The self-reporting method has a number of shortcomings, including poor patient recall, consistency issues, and time-based restrictions. Moreover, the self-reporting method varies for each patient and is not appropriate for young children. Moreover, it does not apply to individuals receiving post-operative treatment who have dementia [2]. It is necessary to distinguish between pain that is fake and pain that is real, which cannot be done using the self-reporting approach [3]. In a critically ill patient, an incorrect estimate of pain can have a number of negative effects. To evaluate pain, medical professionals need specialized expertise. Based on the pain scales, there are also variations in the patient's capacity to report [4]. All of these issues can be resolved by an automated system, which is also incredibly dependable and strong. Using an automated method permits the usage of various pain scales. A person's facial expressions are the best way to assess their level of suffering, and they may be used in any healthcare system [5]. The characteristics are extracted using two different sorts of approaches. Model-based methods and pixel-based methods are some techniques among them. A model-based technique is the Facial Action Coding System (FACS) [6]. It is an effective method for assessing pain in people with shoulder pain. FACS can distinguish between painful and uncomfortable circumstances with ease. Also, the outcomes are consistent. This research suggests a system for recognizing pain and estimating pain intensity level from patient's faces. The system calculates the intensity level of pain. In contrast to current methodologies, this uses unique computational intelligence and feature extraction algorithms that are less complex, more resilient to random noise, and report better performance. The contributions of the proposed work are as follows:

- A statistical strategy utilizing Frei-Chen mask filtering and DenseNet features is suggested for feature extraction. The methods for pattern recognition are novel, straightforward, and reliable.
- The suggested model is adaptable to changes in lighting circumstances, facial angles, gender, race, backdrop settings, and morphological look of persons from various geographic places.
- Compared to other filtering methods, Frei-Chen masks (FCM) are more effective in detecting edges.
- DenseNet is very good at reducing some parameters, improving feature map propagation, and solving the gradient vanishing problem.
- The issue of the pain intensity detection is handled as a task for image classification in computer vision processing using both Statistical Frei-Chen Mask (SFCM) -based features and DenseNet-based features.
- For quick and accurate pain identification and pain intensity level estimation, an Extreme Learning Machine and Radial Basis Function-based (ELM-RBF) classifier is used.
- The proposed work is experimented in both Google Colab and distributed computing setups.
- The suggested system can track the expression of pain to gauge the degree of treatment in the context of smart healthcare.

The structure of the entire paper is as follows. The relevant works are discussed in Related works section, the proposed work is illustrated in Proposed method for pain detection and pain intensity estimation section, and the outcomes are discussed in Results and discussion section. The paper is concluded in Conclusion section.

## Related works

Pain recognition from facial expressions plays a crucial role in various fields, including healthcare, biometrics, and human-computer interaction. In recent years, machine learning techniques, particularly deep learning algorithms, have shown promising results in accurately detecting and quantifying pain from facial images and videos. This literature survey provides an overview of different research papers, highlighting their similarities, dissimilarities, and contributions to pain recognition using machine learning. Early research in the domain of pain analysis employed the three machine- techniques as follows: (i) behavioral-focused on non-verbal external signifiers of pain, such as paralinguistic vocalization, body movements, facial expression, and the sound of moaning or crying; (ii) physiological-focused on bioelectric signals, such as ECG, EOG, and EEG; and (iii) multi-modal-based on both physiological and behavioral parameters. Contact-oriented biosensors are utilized in physiologically based pain evaluation to gather the appropriate physiological data. These biosensors are quite susceptible to physiological or physical disturbances, which amplify and tamper with the original signal. To increase the SNR ratio, an extra, effectively constructed filter is necessary. Moreover, it has an issue of patient discomfort during a long-term recording [5].

Since they use non-contact-driven monitoring technologies like the digital camera, behaviorally based pain indicators are preferred. Moreover, behaviorally related parameters may be constantly tracked for the required period without the patient experiencing significant discomfort. In addition, relative to various behavioral markers, facial movements are persistently linked to pain and carry the majority of pain-oriented data [6]. Deep learning algorithms have lately been shown to be a crucial tool in the field of picture classification. The analysis of these results reveals that deep neural networks have just been used in pain recognition because they were more successful in feature selection and feature extraction, as well as in improving pre-training techniques and transfer learning algorithms. In this study, both DenseNet and Extreme Learning Machine has been applied for pain identification.

Multi-modal pain assessment systems are believed to be more accurate. It is inconvenient for patients and more cost-effective than systems based on video-based modalities. Hence, to build an effective pain evaluation method, researchers are concentrating more on behavioral factors. While some studies have demonstrated that it is more accurate to detect pain from facial gestures [7, 8], the task of assessing pain from facial pictures is highly difficult owing to the wide range of facial expressions and head posture. Traditional machine learning techniques use the manual and time-consuming process of carefully choosing and extracting characteristics. The method of active appearance extracts properties from frames and uses a Support Vector Machine (SVM) classifier to construct automatic pain evaluation strategies [9–11]. Later, a comparison between SVM with some additional classifiers such as random forest, decision tree, and Euclidean distance based 2 nearest neighbors was proposed. Their approach involves detecting the face in a chassis, dividing it in the horizontal direction, and employing the respective two partitions as the two inputs. The model then uses the two inputs to create a distinct representation of the face from a Pyramid Histogram of Oriented Gradients (PHOG) and the appearance data from a Pyramid Local Binary Pattern (PLBP). Multimodal pain detection techniques are also available in literature. Some techniques can also interpret pain even from images with facial deformations [12–15]. Long Short-Term Memory (LSTM) [9] is used in few existing approaches for the pain categorization. The Scale-Invariant Feature Transform (SIFT) features and a Classification algorithm are combinedly used after using the Supervised Descent Method (SDM) to extract landmark spots from facial photos. The authors in [16] noticed that there might be a distortion in the features of the face during the discomfort and suggested that non-rigid and rigid deformity of attributes may be differentiated by employing a linear spline approach.

In [17] supervised techniques are used and pain severity identification is done while part of the labeled frames is employed to provide intensities in a supervised manner. The majority of currently reported research largely makes use of traditional machine-learning techniques. Traditional techniques of machine learning involved the manual and time-consuming process of carefully choosing and extracting characteristics. Deep learning-based methods are more effective in solving these challenges of visual categorization and pattern identification [18]. With this, the authors in ref [13]. extracted temporal characteristics using a classifier based on Recurrent Convolution Neural Networks (RCNN). A pre-trained VGG-16 network with LSTM was used by the authors in ref [20]. to recover both temporal and spatial properties for pain severity recognition. While Convolutional models have successfully been used for pain evaluation [19, 20], these approaches suffer from overfitting [21] and excessive computing costs due to enormous number of learnable parameters. Moreover, LSTM and RNN-based models were additionally been applied recently, which adds to the computational difficulty because weights must be modified over time using backpropagation [22]. A 3D profound convolutional network-based technique for determining pain severity was put out by Tavakolian [23]. The extremely large computational cost of using 3D integration is, however, the main issue [24]. Even though a classifier with just one descriptor might yield good results. The choice of a collection of appropriate descriptors, however, can considerably enhance the efficiency of such a network. The basic concept is that by fusing latent representations that have been recovered, the unique strengths of these various descriptors may be used to complement one another. Deep convolution network-based fusion techniques have been widely employed recently for a variety of image classification applications, and their effectiveness has yielded encouraging results. The faces are often detected from the background before classification [25, 26].

Moreover, the recent tools for assessing pain employ neural networks. For example [5], employed two shallow neural networks like the Spatial Appearance Network (SANET) and Shape Descriptor Network (SDNET) and numerous [6] dual models that integrated CNNs. Different issues develop during the storage and processing of big data due to time constraints and data scalability. The facility for parallel and distributed computing aids in resolving all these issues [27]. Computation offloading is the best way to handle massive data while managing the demanding computations. A system based on brain computer interface has been proposed by Zao et al. [28] to track human cognition. A multi-tier distributed computing system is employed by the system. Near the end, desktop PCs served as the servers and mobile phones served as the user interfaces. The servers at the far-end were made up of a few computer clusters from the Taiwan National Centre. To cut down on time without sacrificing quality, Dong et al.'s [29] parallel processing of image processing methods has been suggested. The processing of static input of photos and handling of dynamic input of images are handled by two different sorts of frameworks. In the former, algorithms are applied to photographs that have already been stored; in the latter, images are received completely fresh. The research found that processing large-scale photos has good stability. The speed of image retrieval and classification is also good. To handle the complex computations required for processing huge data, the overall processing time is decreased. Image processing on distributed platform has been proposed by Liu et al. [30]. It utilises a Hadoop cluster. It achieves the mining of LBP characteristics from photos, and the operations have been greatly sped up as a result. As a result, in the proposed work, the computationally intensive feature extraction, and machine learning methods are handled using a distributed computing strategy. While running the algorithms in parallel, the effectiveness and correctness of the performance are examined.

In line with past studies, the DenseNet, SFCM, and Extreme Learning Machine were employed in this study to categorise different pain severity levels. Both DenseNet and SFCM have been utilised in this work to extract features from the sample photos. In the classification stage, the calculated Gaussian noise is added to the transformed variables. This study introduced Gaussian noise to the framework that was proposed because numerous studies have successfully used noise in similar models that have been proposed over a lengthy period of time. The majority of them concentrated on adding noise to neural network training as an auxiliary tool to increase a classification algorithm's generalization potential and convergence speed.

Here, several approaches are used in literature like (i) statistically based feature representation and machine learning-based classification techniques (ii) denseNet-based feature representation and machine learning-based classification techniques and (iii) extreme learning-based approaches used to extract useful and distinguishing characteristics. These strategies have been used separately. The outcomes of each strategy are combined to determine the final choice for the suggested system. The proposed method's robustness is increased together with performance because of the performance fusion. The specifics and a general explanation of the suggested framework are provided in the section below.

## Proposed method for pain detection and pain intensity estimation

### Cloud and distributed computing setup

Cloud computing updates and services when used with Machine Learning (ML) algorithms have numerous challenges, one of which is efficiently storing, processing, and updating data. Researchers and developers still favor the conventional approach using the local host. The idea of cloud services, enables developers and researchers to access programs and data remotely in a platform with a significant amount of storage space and computation power. This addresses locally distributed host issues. Benefits of storing and processing data in the cloud include:

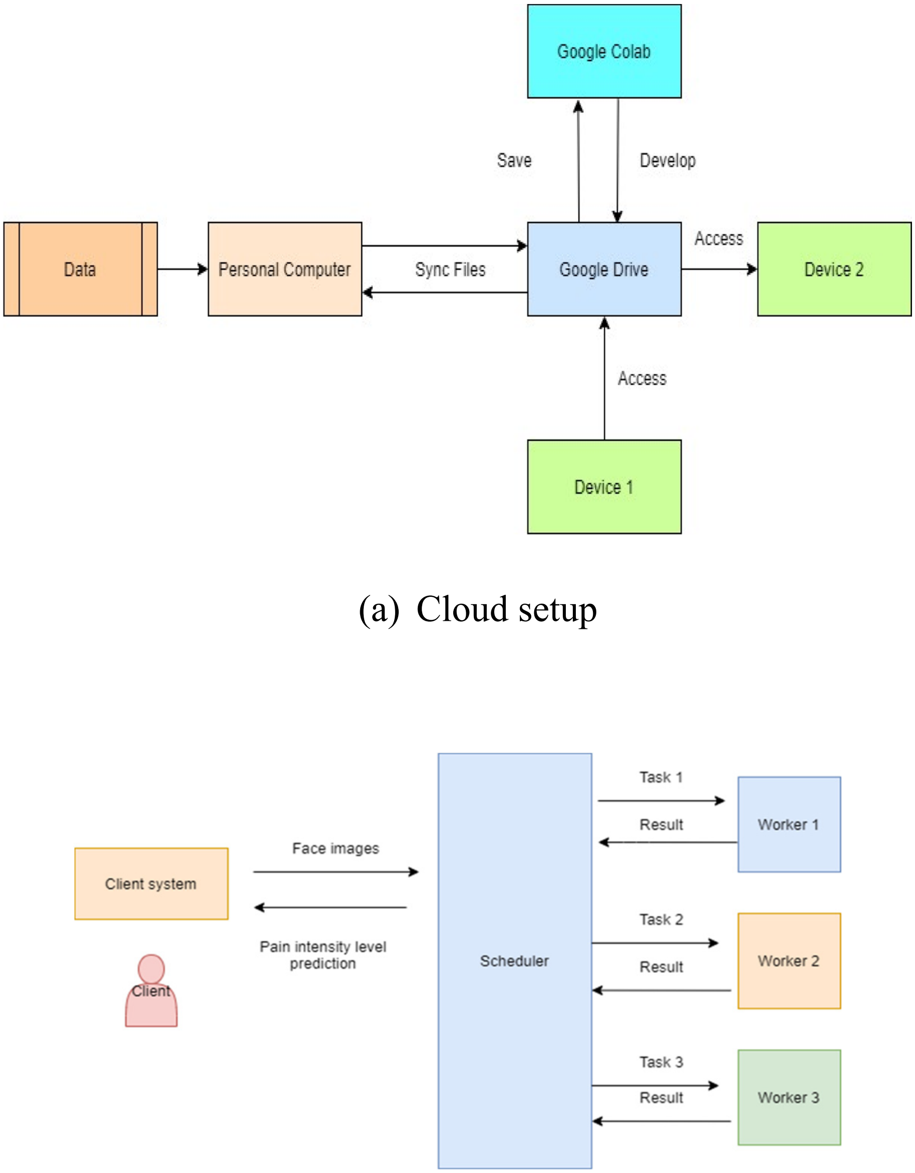
- Increasing the security of code and data and defending them from hacker attacks.
- Making things accessible. In this instance, data can be accessed at any time and from any location.
- Providing additional computational resources, which could lead to reduced time complexity.

The cloud service gives programmers the ability to increase the privacy and flexibility of their code and, in some situations, lower the system's time complexity and power requirements. On the other side, cloud computing can make it easier to process remote end-user data quickly and securely. As an example, intelligent sensors gather health data from users, such as blood pressure and heart rate, and then transfer that data to the cloud for additional processing, where machine learning algorithms are used. The training data can be periodically updated using the cloud-based updates to provide predictions and decisions that are more accurate. With cloud-based control, even a human specialist can be brought in to handle delicate scenarios like the accurate foretelling of a heart attack. In order to run Python code in Jupyter Notebook format, this work employed Google Colab, a free cloud service. In addition to the justification given above, it is another usage of the cloud. Moreover, taking into account its cloud capabilities, Google Cloud ML APIs can be used for additional processing as in Fig. 1(a).

The algorithms are also executed on a cluster of computers with the help of the MATLAB Distributed Computing Server. The MATLAB Parallel Computing Toolbox can be used to run the methods in parallel. The suggested system, which includes one client, one scheduler, and three workers, is depicted in Fig. 1(b).

#### *Client*

The client is a representation of the user who wants to categorize pain and non-pain indications and determine the level of pain from face picture data. In the suggested approach, the client downloads the dataset's face image and handles it to the scheduler. Thus, the client must perform two tasks: (i) classify the symptoms of pain and non-pain; and (ii) determine the degree of pain. These need to be processed, among other things. These involve a number of activities, including feature extraction,



(a) Cloud setup

(b) Distributed computing

**Fig. 1** Cloud setup and distributed computing

preprocessing and ELM-RBF that is used for dimension reduction and categorization. It is able to use up to 12 workers to complete the jobs using the Parallel Computing toolbox. All jobs are password-protected, making it impossible for other people to access them as in Fig. 1.

### *Scheduler*

The MATLAB Job Scheduler (MJS) aids in task coordination and execution. All of the duties are divided among the three workers by the scheduler. It carries out the task scheduling in the designated order.

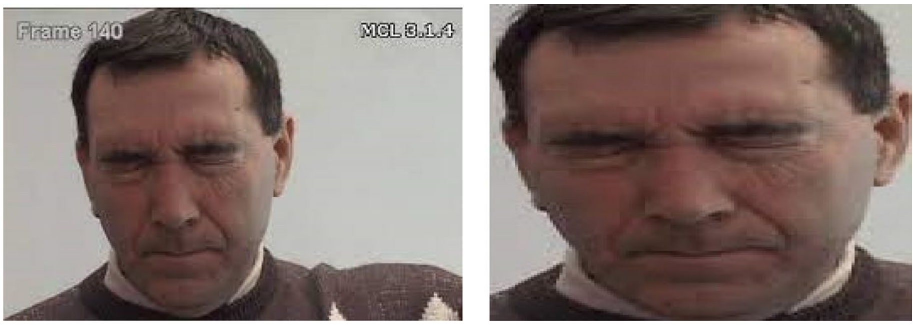
### *Workers*

The employees prepare the training and test photos by preprocessing and feature extraction. One worker is only linked to one MJS at a time. The workers use the ELM-RBF-based classification technique to categorize pain and non-pain symptoms as well as the intensity level of the pain. The Distributed Computing server assesses the tasks among the workers and informs the scheduler of the successes. The client system receives the classification results for pain detection and pain intensity level estimation from the scheduler.

### Proposed work

The client system includes the facial expression images. In order to complete the planned work, only a few frames obtained from the video sequences are used. For face detection in literature, the "Viola Jones" [25] technique is frequently employed. There are classifiers arranged in cascade. When a new sample is received, each classifier is dependent upon the preceding classifier. The existing literature is mostly based on Haar characteristics. In the proposed study, the 'Chehra' [26] face identification technique is employed, and incremental training is carried out on the generic model by means of regression functions as fresh examples are received. When there are many different poses, the "Chehra" face detecting algorithm performs well. Its foundation is SIFT characteristics. The faces have been downsized and the background has been trimmed as in Fig. 2.

Figure 3 shows the overall layout of the suggested facial expression recognition system. It begins by analyzing the



(a) Before pre-processing (b) After Pre-processing

**Fig. 2** Before and after pre-processing using 'Chehra'

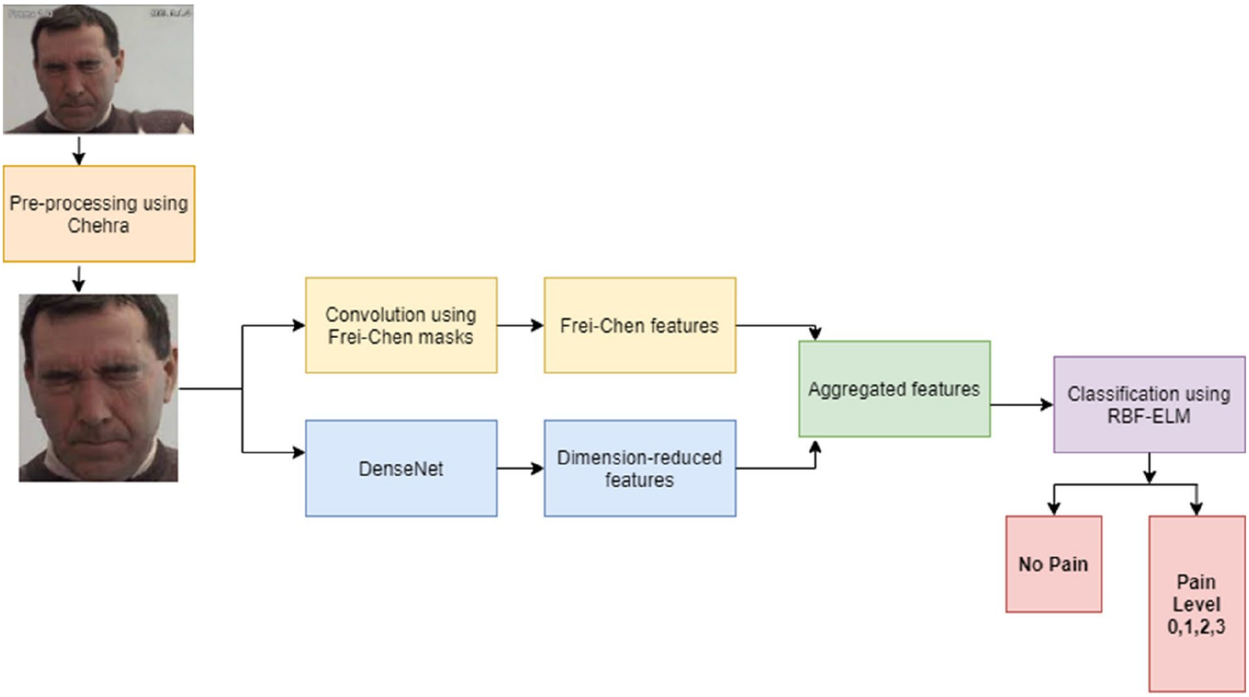

Pre-processing using Chehra
Convolution using Frei-Chen masks
Frei-Chen features
DenseNet
Dimension-reduced features
Aggregated features
Classification using RBF-ELM
No Pain
Pain Level 0,1,2,3

**Fig. 3** Overall view of the proposed work

photos to determine the degree of pain. Deep features are extracted from the various CNN layers through additional processing. They are then used to train an ELM classifier to identify different pain levels together with the proposed FC characteristics. The structure of the ELM is optimized using the structural risk reduction approach, and regularization is applied for precise prediction. The generalization of ELM is improved by this principle. All the operations are experimented on both distributed computing and cloud-based approach [27–30]. The subsequent sections provide more explanations of these steps.

Using feature descriptors, the features are extracted from the cropped images. The suggested feature extraction algorithm uses an edge-based approach [31]. The features in this case are extracted using two innovative feature descriptors. It is based on the edge detection method developed by Frei-Chen (FC) [32] and the features obtained from DenseNet. There are three main steps in both feature descriptors: Filtering with four-directional FC compass masks, creating a code image with the maximum response, and creating a histogram and feature vector are all examples of filtering. Then the DenseNet-derived features are aggregated with SFCM features. The acquired feature descriptors are then used as input by ELM to identify pain and estimate pain intensity.

### *FC features extraction*

There are total of nine convolution masks when using the Frei-Chen edge detector, which likewise operates on a 3×3 texel footprint as in Fig. 4. Unique masks that include all of the basic vectors are called Frei-Chen masks. This indicates that the weighted sum of the nine Frei-Chen masks shown below can represent a 3×3 image area. Four of the Frei-Chen masks listed above are used to represent edges, four more to depict lines, and the last mask is used to represent averages. In the proposed SFCM-based pattern recognition, the colour, texture, and edge-based features are extracted from the facial images. The parameters in the proposed study can be obtained from the RGB colour space and used as input to the classifier to forecast the level of discomfort. All of the RGB colour spaces provide accurate colour differentiation. The SFCM operator can be used to extract both texture data and the RGB colour space. In the suggested method, each colour channel in an RGB image receives a distinct application of the SFCM operator. To collect different colour patterns, several colour channel pairs are used. Pixels for the epicentre and location are selected using a variety of colour channels. The centre pixel in a 3×3 rectangle in $\mathrm{SFCM}_{\mathrm{R,\ G}}$ is constructed with pixels from R and the surrounding pixels from G. In this R-channel image $\mathrm{R}_{\mathrm{c,d}}$ is the pixel at the center and $\mathrm{R}_{\mathrm{c,d}}$, $\mathrm{R}_{\mathrm{c+1,d}}$, $\mathrm{R}_{\mathrm{c-1,d}}$, $\mathrm{R}_{\mathrm{c-1,d+1}}$, $\mathrm{R}_{\mathrm{c,d+1}}$, $\mathrm{R}_{\mathrm{c+1,d+1}}$, $\mathrm{R}_{\mathrm{c-1,d-1}}$, $\mathrm{R}_{\mathrm{c,d-1}}$, $\mathrm{R}_{\mathrm{c+1,d-1}}$ are the eight nearby pixels in a block. In this G-channel image $\mathrm{G}_{\mathrm{c,d}}$ is the center pixel and $\mathrm{G}_{\mathrm{c,d}}$, $\mathrm{G}_{\mathrm{c+1,d}}$, $\mathrm{G}_{\mathrm{c-1,d}}$, $\mathrm{G}_{\mathrm{c-1,d+1}}$, $\mathrm{G}_{\mathrm{c,d+1}}$, $\mathrm{G}_{\mathrm{c+1,d+1}}$, $\mathrm{G}_{\mathrm{c-1,d-1}}$, $\mathrm{G}_{\mathrm{c,\ d-1}}$, $\mathrm{G}_{\mathrm{c+1,d-1}}$ are the pixels present at eight neighboring sides in a block. In this B-channel image $\mathrm{B}_{\mathrm{c,\ d}}$ is the pixel at the center and $\mathrm{B}_{\mathrm{c,\ d}}$, $\mathrm{B}_{\mathrm{c+1,d}}$,$\mathrm{B}_{\mathrm{c-1,d}}$, $\mathrm{B}_{\mathrm{c-1,d+1}}$, $\mathrm{B}_{\mathrm{c,\ d+1}}$, $\mathrm{B}_{\mathrm{c+1,d+1}}$, $\mathrm{B}_{\mathrm{c-1,d-1}}$, $\mathrm{B}_{\mathrm{c,\ d-1}}$, $\mathrm{B}_{\mathrm{c+1,d-1}}$ are the pixels present at eight aneighboring sides in a block. $\mathrm{SFCM}_{\mathrm{R,R}}$, $\mathrm{SFCM}_{\mathrm{G,G}}$, $\mathrm{SFCM}_{\mathrm{B,B}}$, $\mathrm{SFCM}_{\mathrm{R,G}}$, $\mathrm{SFCM}_{\mathrm{R,B}}$ and $\mathrm{SFCM}_{\mathrm{G,B}}$ are the combined channel images. Here each 3×3 block is formed with the subsequent equations.

$$
\begin{array}{r}
\mathrm{SFCM}_{\mathrm{R,R}}\ (\mathrm{p},\mathrm{q},\theta) = <\text{center pixel}(\mathrm{R}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{R}_{\mathrm{c+i,d+j}})\ > \\
\mathrm{SFCM}_{\mathrm{G,G}}\ (\mathrm{p},\mathrm{q},\theta) = <\text{center pixel}\ (\mathrm{G}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{G}_{\mathrm{c+i,d+j}})\ > \\
\mathrm{SFCM}_{\mathrm{B,B}}\ (\mathrm{p},\mathrm{q},\theta) =<\text{center pixel}\ (\mathrm{B}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{B}_{\mathrm{c+i,d+j}})\ > \\
\mathrm{SFCM}_{\mathrm{R,G}}\ (\mathrm{p},\mathrm{q},\theta) = <\text{center pixel}\ (\mathrm{R}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{G}_{\mathrm{c+i,d+j}})\ > \\
\mathrm{SFCM}_{\mathrm{R,B}}\ (\mathrm{p},\mathrm{q},\theta) = <\text{center pixel}\ (\mathrm{R}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{B}_{\mathrm{c+i,d+j}})\ > \\
\mathrm{SFCM}_{\mathrm{R,B}}\ (\mathrm{p},\mathrm{q},\theta) = <\text{center pixel}\ (\mathrm{R}_{\mathrm{c,d}})\ ,\ \text{neighbours}\ (\mathrm{B}_{\mathrm{c+i,d+j}})\ >
\end{array}
\tag{1}
$$

$$
G_1 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}\quad
G_2 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}\quad
G_3 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}
$$

$$
G_4 = \frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}\quad
G_5 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}\quad
G_6 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}
$$

$$
G_7 = \frac{1}{6}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}\quad
G_8 = \frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}\quad
G_9 = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
$$

**Fig. 4** Frei-Chen mask

Where for a 3 × 3 block $1 \geq p \leq 4$ and $1 \geq \mathrm{q} \leq \mathrm{T}$, T is the total number of blocks in each image.

$$(i,j) = \begin{cases} i = 1, j = 1 \;\; where\,\theta \;\; = 45^\circ \\ i = 1, j = -1 \;\; where\,\theta \;\; = 315^\circ \\ i = -1, j = -1 \;\; where\,\theta \;\; = 225^\circ \\ i = -1, j = 1 \;\; where\,\theta \;\; = 135^\circ \\ i = 0, j = -1 \;\; where\,\theta \;\; = 270^\circ \\ i = 0, j = 1 \;\; where\,\theta \;\; = 90^\circ \\ i = 1, j = 0 \;\; where\,\theta \;\; = 360^\circ \\ i = -1, j = 0 \;\; where\,\theta \;\; = 180^\circ \end{cases} \tag{2}$$

The edge magnitude is calculated using the highest value that was discovered following the convolution of the mask and the image. The direction of the edge is determined by the mask that generates the largest magnitude. The feature vector is created using the various channel combinations of the pictures $\mathrm{SFCM}_{\mathrm{R,R}}$, $\mathrm{SFCM}_{\mathrm{G,G}}$, $\mathrm{SFCM}_{\mathrm{B,B}}$, $\mathrm{SFCM}_{\mathrm{R,G}}$, $\mathrm{SFCM}_{\mathrm{R,B}}$ and $\mathrm{SFCM}_{\mathrm{G,B}}$. The feature extraction process for each channel image consists of the following three steps: compass mask image filtering, code image production based on the maximum response, and feature vector construction. The pre-processed photos are taken into account, and then FCM masks are coupled with them to analyse the pattern using the suggested SFCM. The picture is projected onto the required four FCM masks to obtain SFCM features for edge detection, as shown in Fig. 5.

Comparatively to the structural and transformed-based methodologies, the statistical analysis of the texture pattern is more practical and convenient. This method of analysis assesses additional data from the pixel intensity values. The collection and presentation of appearance-based features from an image is helpful. Both regular and irregular patterns could be seen in image. Therefore, statistically based methodologies are better suited to analyse both regular and irregular patterns.

Both global and local to global feature representation techniques have been taken into consideration here during feature computation. The proposed method uses the first four masks on the different channels R, G and B to obtain the four responses (RF) from which the Code Image (CI) is obtained using Eq. (3).

$$\mathrm{CI}(\mathrm{x,y}) = \max\left(\mathrm{RF}_{\theta_1}(\mathrm{x,y}) | 0 \leq \mathrm{i} \leq 4\right) \tag{3}$$

The code image CI obtained from R channel is divided horizontally and then vertically to form four equal halves (horizontally and then vertically) $\mathrm{CI}_1, \mathrm{CI}_2, \mathrm{CI}_3\mathrm{and}\mathrm{CI}_4$. The final feature vector is created by concatenating the normalised histograms obtained for each of the N grids of equal size $\mathrm{g_i}, 1 \leq \ \mathrm{i} \ \leq \ \mathrm{N}$ from that divided code image $\mathrm{CI}_1, \mathrm{CI}_2, \mathrm{CI}_3\mathrm{and}\mathrm{CI}_4$. The Local-Global Feature vector from Red Channel $(\mathrm{FVR}_{\mathrm{LG}})$ is created by aggregation of features obtained from four vertical halves of CI as in Fig. 5. This indicates the features obtained both locally and globally.

The combined histograms of each subregion code image serve as the final feature vectors as in Eq. (4). $\mathrm{FVR}_{\mathrm{LG}}$ is created from CI of R channel after creating grids on the four equal halves of the code image and aggregation of histograms $\mathrm{H}_1, \mathrm{H}_2 \ldots \ldots .\mathrm{H}_\mathrm{N}$.

$$\text{Feature Vector}(\mathrm{FVR}_{\mathrm{LG}}) = < \mathrm{H}_1, \mathrm{H}_2 \ldots \ldots .\mathrm{H}_\mathrm{N} > \tag{4}$$

where N is how many smaller grids there are overall in the code image. The information of the smaller to larger edges and corners of the face can be extracted using this feature vector creation technique. Similarly, feature vector is collected for the remaining channels also as $\mathrm{FVG}_{\mathrm{LG}}$ and $\mathrm{FVB}_{\mathrm{LG}}$.

### *Feature extraction using DenseNet and final feature vector creation*

High, medium, and low-level features of an instructor's face can be extracted using dense deep learning models' layered architecture, which learns features at various layers (hierarchical representations of layered features). Sequential networks and directed acyclic graphs (DAG) are two different forms of networks that are examined [33]. Layers in a serial network are arranged consecutively, similar to AlexNet, which has 8 levels and accepts 227 × 227 2-dimensional input.

A DAG network, on the other hand, contains layers that take the shape of directed acyclic graphs, processing numerous layers concurrently to produce effective results. Examples of DAG models include GoogleNet, DenseNet201, ResNet50, ResNet18, ResNet101, and Inceptionv3 [34–38], each of which has a depth of 22, 201, 50, 18, 101, and 44 layers, respectively. Features are retrieved from the convolution, pooling, and regularisation layers rather than simply the top layer. It is empirically assessed how well different deep network layers performed. Features are retrieved from DenseNet201 using the conv4_block9_1_bn layer.

Features are taken from drop7, pool5 drop 7 × 7 s1, activation 94 relu, and avg pool, respectively, for AlexNet, GoogleNet, Inceptionv3, and ResNet50. We choose pool5 for ResNet101 and ResNet18. The goal of the DenseNet architecture was to maximise information flow between network layers. Each layer receives the feature maps created by all earlier levels, which are subsequently transferred into later layers. All layers are physically related to one another [39]. In contrast to ResNets, features are

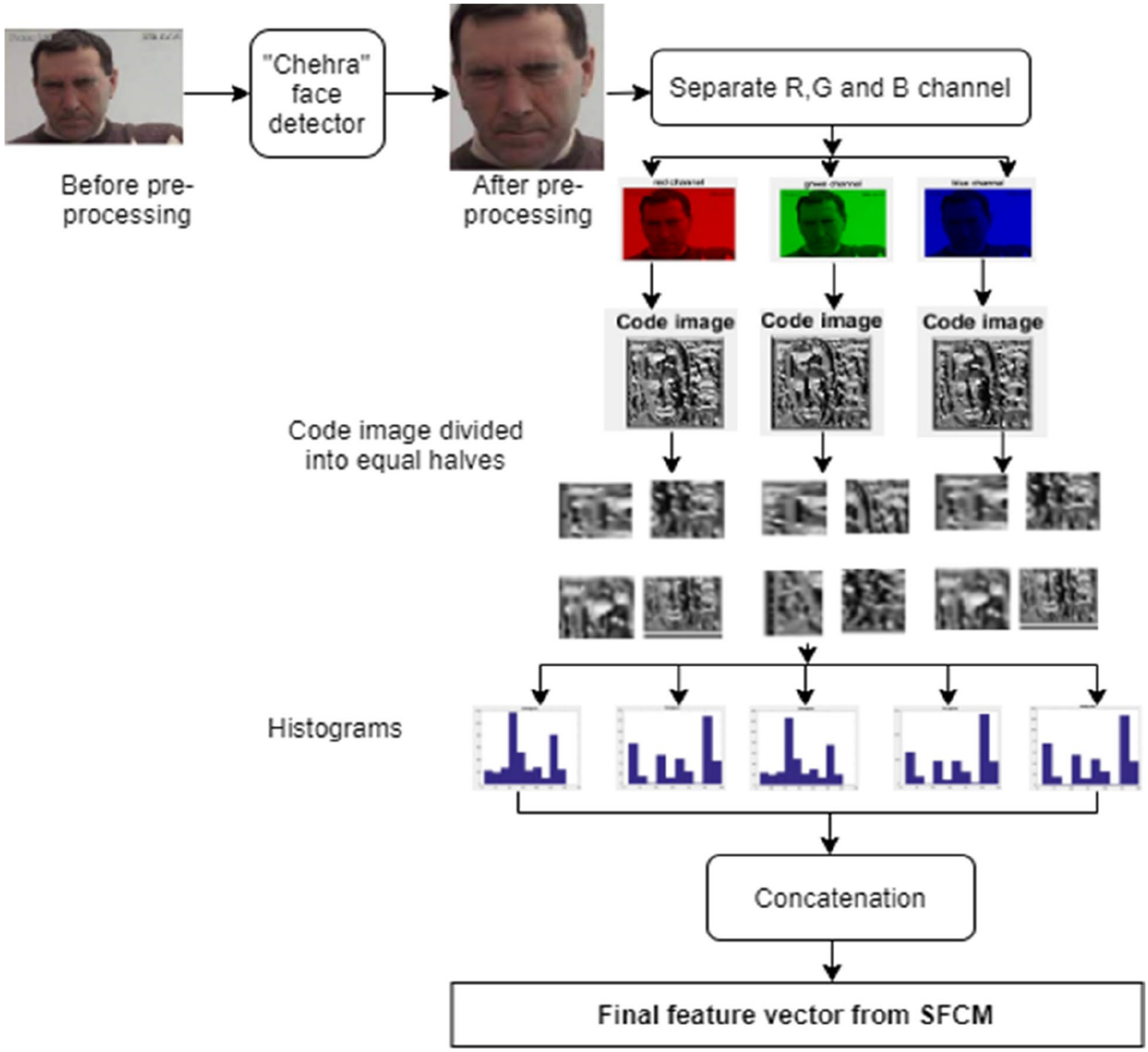


**Fig. 5** Statistical Frei-Chen Mask (SFCM)-based features

concatenated here rather than summarised before being passed into a layer. As a result, the inputs for the l$^{th}$ layer are the feature maps from all of the convolutional blocks before it.

All $L - 1$ layers receive its unique feature maps. In contrast to conventional topologies, which have L connections, an L-layer network has $L(L + 1)/2$ direct connections. The l $^{th}$ layer receives the feature maps x$^{l}$ of all layers that came before it, in the format: $x^0, \ldots, x^{l-1}$. $x^0 = H^L([x^0, \ldots, x^{l-1}])$, where $x^0, x^1 \ldots, x^{l-1}$ are the feature maps concatenated in the 0,., l − 1th layers. Hl (.) is a composite function made up of three operations: a 3×3 convolution, a Rectified Linear Unit (ReLU), and batch normalisation (BN) [40, 41].

The size of the feature maps is cut in half in pooling layers that follow layers in conventional deep CNNs. The change in feature maps would, therefore, cause the concatenation operation employed in Eq. (1) to be incorrect. Convolutional networks, though, must have downsampling layers. Transition layers are introduced and numerous densely connected dense blocks are created using DenseNets to make consistent downsampling possible [38]. Convolution and pooling layers, which are present between dense blocks, make up the transition layers. The feature maps are contained in layer conv4_block9_1_bn, which is related to convolution layer conv4_block9_1_ conv[0][0]. Due to the extensive connection, DenseNet requires fewer parameters than conventional convolutional networks because redundant feature mappings do not need to be re-learned.

In conventional deep CNNs, a pooling layer that cuts the size of the feature maps in half comes after each layer. Convolutional networks must, however, have downsampling layers. The network is divided into numerous densely connected dense blocks using DenseNets, and transition layers are used to enable consistent downsampling [39]. Between thick blocks, transition layers made up of convolution and pooling layers are shown. DenseNet needs fewer parameters because of this dense connectedness because there is no need to memorise it. There are 3 dense blocks and next to next transition layers. The concatenated feature maps are indicated by the solid lines in the Fig. 6. The interconnection between the different layers is represented by the dashed lines. A set of features are finally given as the output from the DenseNet. DenseNet has more diversified characteristics and generally has richer patterns because each layer receives input from all preceding levels. In DenseNet, the classifier employs features of all levels of complexity. It frequently provides more consistent decision boundaries. It also explains why DenseNet performs well in the absence of enough training data. The network can be made smaller and thinner with fewer channels since each layer receives feature maps from all layers that came before it. The growth rate *k* is the additional number of channels for each layer. As a result, its processing and memory efficiency are higher. The concept of concatenation during forward propagation is depicted in the following Fig. 6.

The Final Features (FF) are created by aggregation of SFCM features and the output features of DenseNet (DR) as in Eq. (5).

$$FF =< FVR_{LG}, FVG_{LG}, FVB_{LG}, DR > \tag{5}$$

Subsequently, for the purpose of sensing pain or assessing its intensity level, the SFCM features and the reduced-dimension features from the light CNN are combined. For the classification of testing data, the RBF-ELM classifier usages the intermediate training data results that have been stored. The personnel are divided among the several computations. Figure 2 depicts the general steps in the proposed model. The proposed SFCM-based features are more reliable and straightforward. Using a face detector, the images of the faces are first separated from the backdrop. SFCM-based features use the pixel intensity data from the maximum response after finding the maximum response from four FC masks on the images. While detecting pain and calculating four pain intensity levels, ELM-RBF ensures a quick execution. The next section discusses the classification.

### Classification using RBF-ELM

ELM provides good classification performance in Single Hidden Layer Feed Forward Neural Networks (SLFNs) at a low computational cost as in Fig. 7. Using ELM, the samples are divided into two or numerous categories [39, 40]. Random numbers are used to determine the weights between the input and the buried layer. They learn far faster than back-propagation networks and are good generalizers. In the ELM model.

$$Y = W2\sigma\ (W1X) \tag{6}$$

where W1 is the weight matrix between the input and hidden layer, W2 is the weight matrix between the hidden layer and the output layer, and σ is the activation function. The resultant vector, h (x), is thus equal to h (x)= $[h_1(x)h_2(x) \ldots .h_n(x)]$, where n indicates the total number of neurons present in the hidden layer. The h(x) maps the input "x" onto the ELM feature space. As seen in Fig. 7, the ultimate product is represented as Oi.

### Addition of Gaussian noise

During the training of the suggested SFCM-DenseNet-ELM framework, the Gaussian noise computed from PCA was introduced to the dense layer outputs. According to several research, adding noise to neural network

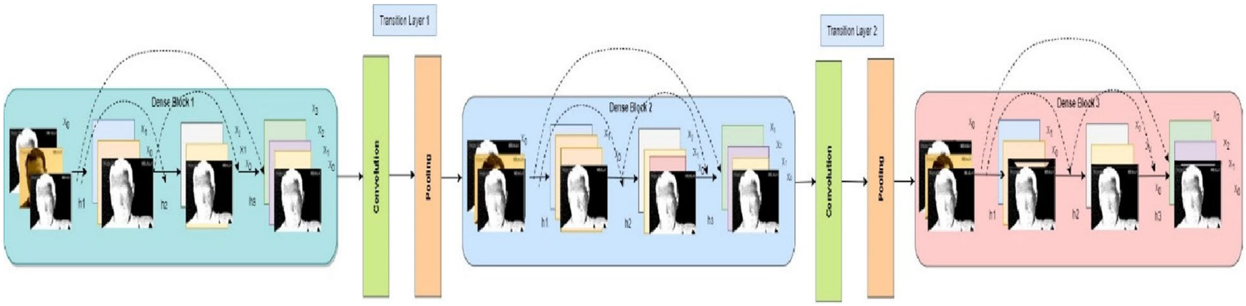


**Fig. 6** Architecture of DenseNet

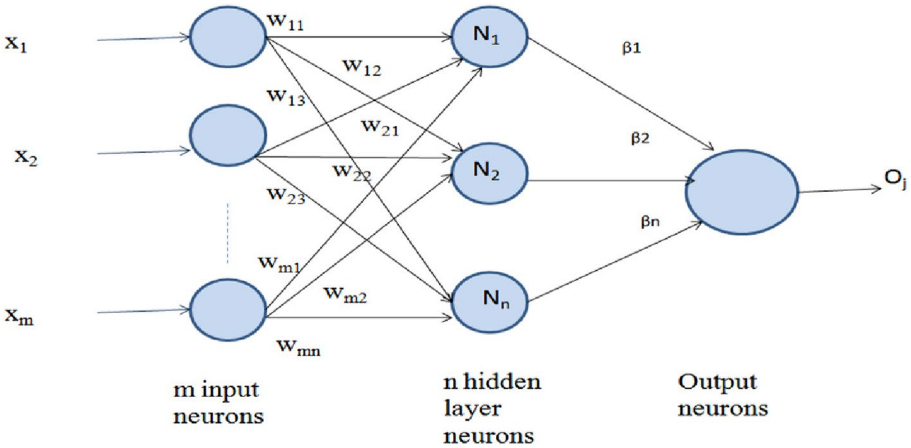


**Fig. 7** General extreme learning machine architecture

training can speed up convergence and increase generalisation capacity. According to earlier research, adding random noise with a Gaussian distribution to new input patterns boosted generalisation power as long as the noise was kept at a level that had no negative effects on the desired output. As a result, the quantity of Gaussian noise added to the PCA components is calculated using Python's Numpy.STD library and then added to the dense layer outputs. The Standard Deviation is calculated via Numpy.STD. The computed quantity and the STD of the Python Keras library were added to the dense layer outputs. The Standard Deviation (SD) of the supplied data is calculated using Numpy.STD. The data distribution in the given data set is used to calculate SD.

$$\text{Here, Standard Deviation} = \sqrt{\text{Mean}(\text{abs}(x - x.\text{mean}())^{2}} \tag{7}$$

**Algorithm 1** SFCM-DenseNet-ELM algorithm

```
Cropped, pre-processed and resized.
Tuning the parameters, the Chehra tool
For (epoch = 0, epoch = 10, epoch++)
    Batch ← 96
    Train the fine-tuned DenseNet
End For
Extraction of features from fine-tuned DenseNet
Setting up convolution with FCM masks
Selecting features from output of convolution
Aggregating the features obtained using both SFCM and DenseNet
Modelling of the SFCM-DenseNet-ELM
Applying Gaussian noise to Dense layer of SFCM-DenseNet-ELM
For (epoch = 0, epoch = 10, epoch++)
    Batch ← 96
    Training, and testing SFCM-DenseNet-ELM
End For
Prediction of the four pain intensity levels
Evaluation of the results
```

Algorithm 1 summarises the specifics of the proposed SFCM-DenseNet-ELM model. 48 batches were used to test and train the proposed algorithm over five epochs. The ELM classifies the features into different pain intensity levels. With the help of the cloud service, programmers can increase the security and adaptability of their code and, in some cases, reduce the system's time and energy requirements. Cloud computing makes

© Jeffrey Cohn

**Fig. 8** Sample images

it possible to process dispersed end-user data quickly and securely. To produce more accurate predictions and judgements, a routine updating of the training data utilising cloud-based updates is done. A cloud-based control can also be used to call in a human expert to handle delicate situations, such as accurately predicting a heart attack.

## Results and discussion

The client and MJS are both installed on a local host that has Windows 7, 8GB of RAM, MATLAB R2014a, an Intel core (TM) i5-4460 CPU running at 3.20 GHz, and a 64-bit operating system. Three systems with similar setups are utilised to set up the workforce. The distributed computing/private cloud setup of one MJS and 3 workers is used to run all of the experiments. To verify the time efficiency attained utilising MATLAB distributed computing, the experiments are also conducted using just one system. To implement the configuration, the MATLAB Parallel Computing toolbox is used. For executing the code in python Google Colab is used in the experiments.

### Dataset

Sixty-six females and 63 males having shoulder pain were included in the tests using the UNBC-McMaster Shoulder Pain expression archive dataset [41]. It contains a big dataset of images. The movies are captured in both active and passive modes. In the active mode, the patient moves his or her own arm, while in the passive mode, a therapist moves the patient's arm. For the investigation, 25 participants, 200 sequences, and 48,398 photos were employed. The video has a resolution of 320 by 240 pixels, with a 140 × 200-pixel face span. The PSPI (Prkachin and Solomon Pain Intensity) values provided with the dataset are taken into account for determining the pain intensity level. 16 different pain levels exist in the dataset [42–47]. The 2D Face Set Database with Pain-expression Set [46] is the second database that is being used. 599 photos from 10 men and 13 women subjects make up this database. This database typically has a 2-class issue. There are 298 photos with the expression "No-pain" and 298 with the expression "Pain".

In Fig. 8, some sample images from the dataset are presented. The count of photos that are available for each degree of pain severity is shown in Table 1. In order to conduct the pain detection studies, the pain detection issue is viewed as a binary classification issue. The 2 classes taken into account for the trials are no pain (PSPI value = 0), and discomfort (PSPI value 1). Images with an intensity level of 0 are classed as having no pain, whereas images with a level of 0 are labelled as having pain for pain detection. The collection contains 8369 photos of pain and 40,029 images of no discomfort. 1/5 of the available photographs in the pain category are involved in the experiment to balance the quantity of images in each class. Also, the trials are carried out using the 10-fold cross-validation technique. The average of the performance measurements is taken into consideration after all experiments are repeated ten times. Thus, only four levels are used in the experiments using the suggested work concerning pain intensity level detection. Level 3 is the

**Table 1** Distribution of the samples in the UNBC dataset

| | Level 0 | Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| **Number of frames** | 12503 | 1263 | 1239 | 1652 |

**Table 2** Confusion matrix for 2 class pain detection on UNBC

| | Pain | No Pain |
|---|---|---|
| Pain | 97.0 | 1.5 |
| No Pain | 0 | 99.0 |

**Table 3** Confusion matrix for 2 class pain detection on 2D face dataset

| | Pain | No Pain |
|---|---|---|
| Pain | 97.0 | 3.0 |
| No Pain | 6.0 | 94.0 |

**Table 4** Performance comparison of proposed features with other descriptors while detecting pain/no-pain on UNBC

| | |
|---|---|
| Local Binary Pattern | 86.5 |
| Local Directional Pattern | 87.5 |
| GABOR | 91.1 |
| Pyramid Histogram of Oriented Gradients | 87.5 |
| Local Directional Texture Pattern | 85.5 |
| Local Directional Number Pattern | 91.0 |
| SFCM | 98.0 |

fourth level, which includes all images with pain intensity levels greater than 3. The features are delivered to ELM after being extracted utilising the suggested patterns and features of DenseNet. There are two phases to the tests. The first stage involves pain detection. There are two levels of pain detection (no pain, pain). The four levels of pain severity are estimated in the second phase. The data distribution is taken as described by Hammal and Cohn [42]. Level 0 indicates no pain, level 1 indicates traces of pain, level 2 indicates weak pain and level 3 indicates a stronger level of pain.

The confusion matrices for the proposed technique that were used to detect pain are shown in Tables 2 and 3. Little expression changes can often lead to confusion between the categories of no discomfort and pain.

The results of the suggested feature descriptors and those already found in the literature are compared in Table 4 for classification accuracy. The resilience and better illustration of structural data of the suggested feature descriptors make them superior to others, as shown in the table. The existing feature descriptors Local Binary Pattern, Pyramid Histogram of Oriented Gradients, and Gabor [35, 47] are affected by noise and lighting artefacts in comparison to the proposed patterns. Table 4 shows an analysis of the suggested feature descriptors' performance with various image sizes and shows that higher resolution can lead to greater accuracy.

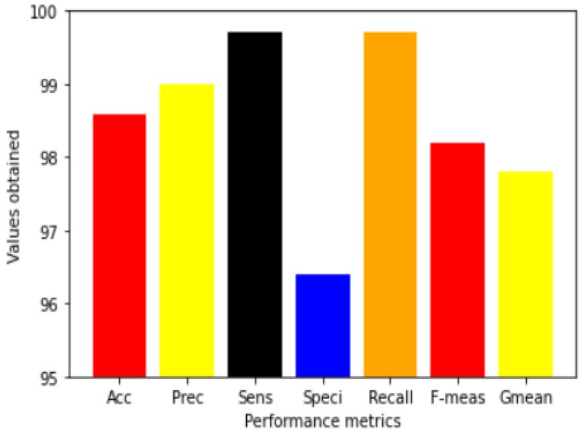


**Fig. 9** Performance metrics for the proposed approach for 4 pain intensity level estimation on UNBC

Four classes are taken into account for pain intensity level estimation (PSPI=0, 1, 2, and 3). In order to prevent findings from being skewed, intensity level 0 has 500 images per subject, whereas the other categories have the same image distribution as the dataset. The size of the images is 162×122. The experiments make advantage of 10-fold cross-validation. There are ten iterations of the experiments. The outcome is then determined by averaging the performance metrics. Consideration is given to different performance criteria as in Fig. 9. P is the quantity of samples from a given class. *N* is the number of samples that did not belong to that specific class.

$$T = P + N \tag{8}$$

$$Tpr = \frac{tp}{P} \tag{9}$$

$$Tnr = \frac{tn}{N} \tag{10}$$

$$Classification\_accuracy(Acc) = \frac{tp + tn}{T} \tag{11}$$

$$Sensitivity(Sens) = tpr \tag{12}$$

$$Specificity(Speci) = tnr \tag{13}$$

$$Precision = \frac{tp}{tp + fp} \tag{14}$$

**Table 5** Obtained confusion matrix for pain intensity level estimation on UNBC

| | Level 0 | Level 1 | Level 2 | Level ≥ 3 |
|---|---|---|---|---|
| Level 0 | 97.0 | 1.5 | 1.0 | 1.5 |
| Level 1 | 0 | 99.0 | 0.5 | 0.5 |
| Level 2 | 0 | 0 | 99.2 | 0.8 |
| Level ≥ 3 | 0 | 0 | 0.9 | 99.1 |

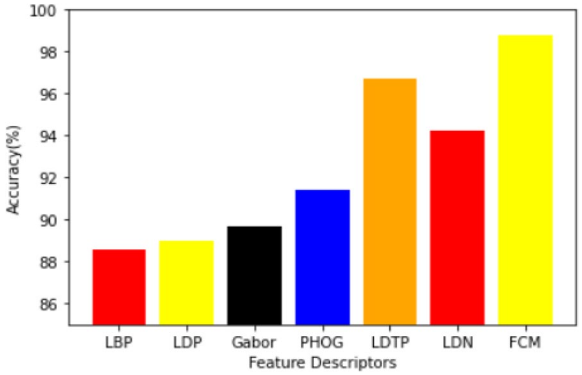


**Fig. 10** Classification accuracy obtained for different pain intensity levels using different descriptors replaced for SFCM-based features on UNBC

$$Recall\ =\ Sensitivity \tag{15}$$

$$F\_measure(F-meas)\ =\ 2*\frac{Precision\ *\ Recall}{Precision\ +\ Recall} \tag{16}$$

$$Gmean\ =\ sqrt(tpr\ *\ tnr) \tag{17}$$

where *fp* stands for false positive and *tp* for true positive. The estimated performance metrics are shown as a graph in Fig. 9. The confusion matrices in Table 4 demonstrate that the images categorised as belonging to pain intensity levels 2 and ≥3 are more accurately classified than the images classified as belonging to levels 0 and 1. Table 5 represents the confusion matrix attained for four pain intensity levels.

A Gaussian white noise with constant mean and variance is aggregated to the facial images, and all the trials were iterated, to ensure the robustness of the suggested patterns to noise. However even with the addition of noise, both of the unique patterns that are proposed still produce good outcomes. This is due to the SFCM's effective structural encoding capability and the convolving stages, which eliminates all noisy edges. Because of its use of direction numbers for encoding, SFCM-based features are also effective for noisy images. The outcomes are shown in Fig. 10.

The efficacy of the suggested technique with proposed feature descriptors [39–42] has been shown in Table 6.

To demonstrate that our model is computationally more effective than other deep learning models, we ran an experiment. The comparison between Densenet-201 and other models is shown in Table 7 in terms of the number of parameters, precision, size, and time complexity. Results reveal that Densenet-201 has the highest accuracy of 98.6% with the fewest parameters. A good accuracy rate was also attained by the DenseNet201.

**Table 6** Comparison with the state-of-the art approaches

| Methods | Technique | Total number of classes | Metric | Value |
|---|---|---|---|---|
| Rathee and Ganotra [47] | Multiview-based Distance learning | 4 | Accuracy | 75.00 |
| P.Rodriguez et al. [48] | Long Short-Term Memory Networks | 2 | Accuracy | 83.00 |
| Zhou et al. [49] | Real-time regression and RCNN | 16 | MSE<br>PCC | 1.54<br>0.65 |
| Semwal and Londhe [50] | Multi-view fusion | 4 | Accuracy | 94.00 |
| Zhao et al. [51] | LBP, Gabor | | ICC<br>MAE<br>PCC | 0.56<br>0.82<br>0.61 |
| Florea et al. [52] | Merged SVR | 16 | AUC | 80.59 |
| Vaish and Sagar [53] | Kaze Features | 4 | Accuracy | 91.87 |
| Hammal and Cohn [42] | SVM | | Accuracy | 83.0 |
| Bargshady et al. [4] | EJH-CNN-BiLSTM | 4 | Accuracy | 85.0 |
| Proposed approach (UNBC) | SFCM-DenseNet-ELM | 4 | Accuracy | 98.58 |
| Proposed approach (2D face) | SFCM-DenseNet-ELM | 2 | Accuracy | 95.52 |

*ICC* Intra-class Correlation Coefficient, *RCNN* Region-based Convolutional Neural Network, *SVM* Support Vector Machine, *PCC* Pearson Correlation Coefficient, *MAE* Mean Absolute Error, *AUC* Area Under Curve, *MSE* Mean square Error, *PCC* Pearson Correlation Coefficient, *SVR* Support Vector Regression

**Table 7** Comparison of existing CNN models on UNBC

| Existing CNN models | MAE | Accuracy | Model size | Parameters |
|---|---|---|---|---|
| NasNet-mobile [54] | | 96.5% | 20 MB | 5.30 M |
| Inception-v3 [54] | | 94.5% | 89 MB | 23.9 M |
| Inception ResNetV2 [54] | | 96.9% | 209 MB | 55.9 M |
| Attention CNN [55] | | 51.1% | 229 MB | 59.9 M |
| Hybrid Neural Network [56] | 0.73 | - | 292 MB | 69.2 M |
| Parallel CNN [57] | | 95.1% | 152 MB | 54.3 M |
| Densenet-201 | | 98.6% | 77 MB | 20.0 M |

**Table 8** Comparison of time (seconds) with different setups for processing a image

| Image size | With distributed computing | Local host | Google colab |
|---|---|---|---|
| 42×62 | 0.20 | 0.51 | 0.23 |
| 102×102 | 0.32 | 0.62 | 0.36 |
| 162×122 | 0.42 | 0.81 | 0.46 |
| 242×242 | 0.55 | 1.00 | 0.62 |

In comparison to other edge detectors, the SFCM-based approach has superior accuracy, lower overhead, the ability to detect corners alongside edges, and is less susceptible to noise. Like the Sobel filter, the SFCM algorithm operates on a 3*3-pixel portion of a picture, but instead of using two masks, it employs four. The four unique masks of this filter can be mentioned as essential attributes. A suitable mask is chosen and applied to the image for edge detection. The limitations of the proposed work are that it had not been applied on multi-channel images and tested and the fact that DenseNet consumes larger execution time. The proposed work will be applied on multi-channels in the future works.

Each image's total processing time for determining the level of pain is calculated. Table 8 shows the overall amount of time spent testing with a single local host. Table 8 also illustrates how employing distributed computing for the computations reduces the amount of time needed. Thus, time efficacy which is crucial in a health care system is increased by job parallelization. The algorithm's time complexity was greatly reduced by assuming Cloud storage and Google Cloud, and, on average, just 40% of the ideal amount of time was needed to perform the techniques. It means that applying the idea of the Cloud enables programmers to design algorithms more effectively, in addition to the fact that the development of approaches is valuable for researchers and medical professionals. As a future enhancement, intelligent sensors can also be used to collect user-provided health information, such as blood pressure and heart rate, and then send that information to the cloud for further processing using machine learning algorithms.

## Conclusion

Remote patient monitoring is handled by an intelligent healthcare system. Smart devices that work with under-the-skin sensors implanted in diabetic patients continuously check their glucose levels. Similar to this, a smart imaging gadget can track several emotions to determine the dose amount, including anxiety, depression, and drug addiction. The facial region is used in this paper in order to predict the sentimental pain intensity level. In this work it is suggested to use a unique pattern based on Frei-Chen masks. When compared to the current patterns, they are more resilient and compact. The suggested work achieves high classification accuracy for a both pain recognition and pain intensity level estimation. The artificial intelligence technique created in this study may be beneficial for the development of automatic pain management techniques for use by physicians and other medical researchers, as well as other fields of medical diagnosis. The suggested model outperformed previous approaches for face photos with distinct orientations, accurately classifying pain intensity levels from frontal face images. Future works will utilise the suggested material on several channels. In the upcoming project, a model will be created that can perform better when evaluating covered faces with occlusions and cross-corpora data. Additionally, it is intended to develop a unique dataset to test the effectiveness of our upcoming technique in real-time.

### Authors' contributions

S.A: Idea, formulation, writing and programming. A.S Formulation, motivation, writing and programming. N.K: Experiments design. All authors read and approved the final manuscript.

### Funding

Open access funding provided by Vellore Institute of Technology. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

## Declarations

### Ethics approval and consent to participate

No ethical approval is required.

### Consent for publication

The authors affirm that consent to publish has been received from all human research participants.

### Competing interests

The authors declare no competing interests.

Received: 27 July 2023 Accepted: 8 September 2024
Published online: 23 September 2024

## References

1. Hamzehei S, Akbarzadeh O, Attar H, Rezaee K, Fasihihour N, Khosravi MR (2023) Predicting the total Unified Parkinson's Disease Rating Scale (UPDRS) based on ML techniques and cloud-based update. J Cloud Comput 12(1):1–6
2. Broderick JE, Stone AA, Calvanese P, Schwartz JE, Turk DC (2006) Recalled pain ratings: a complex and poorly defined task. J Pain 7(2):142–149
3. Onyema EM et al (2021) Enhancement of patient facial recognition through deep learning algorithm: ConvNet. J Healthc Eng 2021:5196000
4. Bargshady G, Zhou X, Deo RC, Soar J, Whittaker F, Wang H (2020) Enhanced deep learning algorithm development to detect pain intensity from facial expression images. Expert Syst Appl 149:113305
5. Newman CJ, Limkittikul RK, Chotpitayasunondh KT, Chanthavanich P (2005) A comparison of pain scales in Thai children. Arch Dis Child 90(3):269–270. https://doi.org/10.1136/adc.2003.044404
6. Prkachin KM, Solomon PE (2008) The structure, reliability and validity of pain expression: evidence from patients with shoulder pain. Pain 139(2):267–274. https://doi.org/10.1016/j.pain.2008.04.010
7. Wu CL, Liu SF, Yu TL, Shih SJ, Chang CH, Mao Y, Chao SF, W. C (2022) Deep learning-based pain classifier based on the facial expression in critically ill patients. Front Med 9:851690
8. Li C, Pourtaherian A, van Onzenoort L, a Ten WT, de With PH (2020) Infant facial expression analysis: towards a real-time video monitoring system using r-cnn and hmm. IEEE J Biomedical Health Inf 25(5):1429–1440
9. Rodriguez P, Cucurull G, Gonzalez J, Gonfaus JM, Nasrollahi K, Moeslund TB, Roca FX (2017) Deep Pain: exploiting long short-term memory networks for facial expression classification. IEEE Trans Cybern 52:3314–3324
10. Semwal A, Londhe ND (2021) Computer aided pain detection and intensity estimation using compact CNN based fusion network. Appl Soft Comput 112:107780
11. Peng X, Huang D, Zhang H (2020) Pain intensity recognition via multi-scale deep network. IET Image Proc 14(8):1645–1652
12. Thiam P, Kessler V, Amirian M, Bellmann P, Layher G, Zhang Y, Velana M, Gruss S, Walter S, Traue HC, Kim J, Schork D, Andre E, Neumann H, Schwenker F (2019) Multi-modal pain intensity recognition based on the senseemotion database. IEEE Trans Affect Comput PP 12(3):743–760
13. Craig KD, Prkachin KM, Grunau RVE (2001) The facial expression of pain BT - Handbook of pain assessment. Handb Pain Assess 2:153–169
14. Prkachin KM (1992) The consistency of facial expressions of pain: a comparison across modalities. Pain 51:297–306
15. Neshov N, Manolova A (2015) Pain detection from facial characteristics using supervised descent method. in: Proc. 2015 IEEE 8th Int. Conf. Intell. Data Acquis. Adv. Comput. Syst.Technol. Appl 1:251–256
16. Rathee N, Ganotra D (2015) A novel approach for pain intensity detection based on facial feature deformations. J Vis Commun Image Represent 33:247–254
17. Werner P, Al-Hamadi A, Limbrecht-Ecklundt K, Walter S, Gruss S, Traue HC (2016) Automatic pain assessment with facial activity descriptors. IEEE Trans Affect Comput 8(3):286–299
18. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556
19. Zhou J, Hong X, Su F, Zhao G (2016) Recurrent convolutional neural network regression for continuous pain intensity estimation in video. IEEE Comput Soc Conf Comput Vis Pattern Recognit Work. pp 1535–1543
20. Rodriguez P, Cucurull G, Gonalez J, Gonfaus JM, Nasrollahi K, Moeslund TB, Roca FX (2017) Deep pain: Exploiting long short-term memory networks for facial expression classification. IEEE Trans Cybern 52(5):3314–24
21. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. 15:1929–1958
22. Allen-Zhu Z, Li Y, Song Z (2019) On the convergence rate of training recurrent neural networks. Adv Neural Inf Process Syst 32
23. Tavakolian M, Hadid A (2019) A spatiotemporal convolutional neural network for automatic pain intensity estimation from facial dynamics. Int J Comput Vis 127:1413–25
24. Lu H, Xu H, Zhang L, Ma Y, Zhao Y (2018) Cascaded multi-scale and multi-dimension convolutional neural network for stereo matching. VCIP 2018 - IEEE Int. Conf. Vis. Commun. Image Process
25. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
26. Asthana A, Zafeiriou S, Cheng S, Pantic M (2014) Incremental face alignment in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1859–1866
27. Senger H, Geyer C (2016) Parallel and distributed computing for big data applications. Concurrency Computation: Pract Experience 28(8):2412–2415
28. Zao JK, Gan T, You CK, Chung CK, Wang CK, Mndez CK, Mullen T, Yu T, Kothe C, .Hsiao CT (2014) .Chu, Pervasive brain monitoring and data sharing based on multi-tier distributed computing and linked data technology. Front Hum Neurosci 8:370
29. L.Dong Z, Lin Y, Zhang Q (2016) X.Cao, a hierarchical distributed processing framework for big image data. IEEE Trans Big Data 2(4):297–309
30. Liu T, Liu Y, Li Q, Wang XR, Gao F, Zhu YC, Qian DP (2015) SEIP: system for efficient image processing on distributed platform. J Comput Sci Technol 30(6):1215–1232
31. Chaple GN, Daruwala RD, Gofane MS Comparisions of Robert, Prewitt (2015) Sobel operator based edge detection methods for real time uses on FPGA. In2015 International Conference on Technologies for Sustainable Development (ICTSD). pp 1–4. IEEE.
32. Apdilah D, Simargolang MY, Rahim R (2017) A study of Frei-Chen approach for edge detection. Int J Sci Res Sci Eng Technol 3(1):59–62
33. Li J, Li X, He D (2019) A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. IEEE Access 7:75464–75475
34. Szegedy C, Wei Liu W, Yangqing Jia Y IEEE Conference on Computer Vision and, Recognition P et al (2015) Going deeper with convolutions, in Proceedings of the (CVPR). Boston, MA, USA
35. Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer KD (2014) Implementing efficient convnet descriptor pyramids, http://arxiv.org/abs/1404.1869
36. He K, Zhang X, Ren S, Sun J, IEEE Conference on Computer Vision and, Recognition P (2016) Deep residual learning for image recognition, in Proceedings of the (CVPR).
37. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift, http://arxiv.org/abs/1502.0316
38. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, FL, USA, pp 315–323
39. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, pp 4700–4708
40. Huang GB, Zhu QY, Siew CK (2014) Extreme learning machine: a new learning scheme of feedforward neural networks. in Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541). Budapest, Hungary
41. Lucey P, Cohn J, Prkachin K, Solomon P, Matthews I (2011) Painful data: the unbc-mcmaster shoulder pain expression archive database. in: Automatic Face & Gesture Recognition and Workshops (FG 2011). IEEE International Conference, pp 57–64
42. Hammal Z, Cohn JF (2012) Automatic detection of pain intensity, in: Proceedings of the 14th ACM International Conference on Multimodal Interaction, pp. 47–52
43. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987. https://doi.org/10.1109/TPAMI.2002.1017623
44. Bosch A, Zisserman A, Munoz X (2007) Representing shape with a spatial pyramid kernel, in: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 401–408
45. Bartlett M, Littlewort G, Fasel I, Movellan JR (2003) Real time face detection and facial expression recognition: development and applications to human computer interaction. in: 2003 Conference on Computer Vision and Pattern Recognition Workshop 5:53
46. Hancock P (2008) Psychological image collection at stirling (pics) http://pics.psych.stir.ac.uk
47. Rathee N, Ganotra D (2016) Multiview distance metric learning on facial feature descriptors for automatic pain intensity detection. Comput Vis Image Underst 147:77–86. https://doi.org/10.1016/j.cviu.2015.12.004

48. Rodriguez P, Cucurull G, Gonzalez J, Gonfaus JM, Nasrollahi K, Moeslund TB, Roca FX (2017) Deep Pain: exploiting long short-term memory networks for facial expression classification. IEEE Trans Cybern 52:3314–3324.
49. Zhou J, Hong X, Su F, Zhao G (2016) Recurrent convolutional neural network regression for continuous pain intensity estimation in video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 84–92
50. Semwal A, Londhe ND, MVFNet (2021) A multi-view fusion network for pain intensity assessment in unconstrained environment. Biomed Signal Process Control 67:102537 (https://www.sciencedirect.com/science/article/pii/S17468). Accessed on 10 March 2022
51. Zhao R, Gan Q, Wang S, Ji Q (2016) Facial expression intensity estimation using ordinal information, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3466–3474
52. Florea C, Florea L, Butnaru R, Bandrabur A, Vertan C (2016) Pain intensity estimation by a self-taught selection of histograms of topographical features. Image Vis Comput 56:13–27
53. Vaish A, Gupta SA (2019) Novel Approach for Pain Intensity Detection by KAZE Features. In Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems; Springer: Singapore
54. Radhika K, Devika K, Aswathi T, Sreevidya P, Sowmya V, Soman KP (2020) Performance analysis of NASNet on unconstrained ear recognition. Nature inspired computing for data science, pp 57–82
55. Xin X et al (2020) Pain intensity estimation based on a spatial transformation and attention CNN. PLoS ONE 15(8):e0232412
56. Huang Y, Qing L, Xu S, Wang L, Peng Y (2022) HybNet: a hybrid network structure for pain intensity estimation. Vis Comput 32:871–882
57. Ye X, Liang X, Hu J, Xie Y (2022) Image-based Pain Intensity Estimation using parallel CNNs with Regional attention. Bioengineering 9(12):804

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.