Scientific Research Publishing

Journal of Software Engineering and Applications, 2021, 14, 400-417
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

A Modified CNN Network for Automatic Pain Identification Using Facial Expressions

Ioannis Karamitsos, Ilham Seladji, Sanjay Modak

Rochester Institute of Technology, Dubai, UAE
Email: ixkcad1@rit.edu, is1132@rit.edu, ssmcada@rit.edu

How to cite this paper: Karamitsos, I., Seladji, I. and Modak, S. (2021) A Modified CNN Network for Automatic Pain Identification Using Facial Expressions. *Journal of Software Engineering and Applications*, **14**, 400-417.
https://doi.org/10.4236/jsea.2021.148024

Received: June 20, 2021
Accepted: August 20, 2021
Published: August 23, 2021

Abstract

Pain is a strong symptom of diseases. Being an involuntary unpleasant feeling, it can be considered a reliable indicator of health issues. Pain has always been expressed verbally, but in some cases, traditional patient self-reporting is not efficient. On one side, there are patients who have neurological disorders and cannot express themselves accurately, as well as patients who suddenly lose consciousness due to an abrupt faintness. On another side, medical staff working in crowded hospitals need to focus on emergencies and would opt for the automation of the task of looking after hospitalized patients during their entire stay, in order to notice any pain-related emergency. These issues can be tackled with deep learning. Knowing that pain is generally followed by spontaneous facial behaviors, facial expressions can be used as a substitute to verbal reporting, to express pain. In this paper, a convolutional neural network (CNN) model was built and trained to detect pain through patients' facial expressions, using the UNBC-McMaster Shoulder Pain dataset. First, faces were detected from images using the Haarcascade Frontal Face Detector provided by OpenCV, and preprocessed through gray scaling, histogram equalization, face detection, image cropping, mean filtering, and normalization. Next, preprocessed images were fed into a CNN model which was built based on a modified version of the VGG16 architecture. The model was finally evaluated and fine-tuned in a continuous way based on its accuracy, which reached 92.5%.

Keywords

CNN, Computer Vision, Facial Expressions, Image Processing, Pain Assessment

1. Introduction

Humans move their facial muscles, either spontaneously or purposefully, to convey a certain emotional state (e.g., sadness, happiness, fear, disgust, pain) in a nonverbal way. These facial moves are called facial expressions. Facial expressions vary between different species and humans as well; they can be affected by a person's age, gender, psychological state, personality, and social situations. Moreover, they can either be innate or acquired through the influence of someone else's. Humans have the ability to discern hidden feelings and fake emotions in some cases, especially when these are expressed by someone with whom they have strong relationships. However, the automation of such tasks is very laborious and challenging.

Facial expressions can be regarded as an effective alternative to verbal communication. For instance, paralyzed people can communicate through eye contact and eye movements. Therefore, facial expressions are very important and worthy to be interpreted by machines, and one of the applications in which they are involved in the detection of pain.

Pain is an unpleasant feeling which is triggered by an anomaly in the body. This anomaly can either be medical (e.g., an injury), or emotional (e.g., stress and depression which can cause terrible headaches). When nerves detect tissue damage or irritation, they send information through the spinal cord to the brain, thus causing humans to react to that anomaly. Pain is either expressed verbally or physically, through facial expressions.

Pain can vary from being slightly annoying to debilitating. Regardless of its intensity, it gives a strong and reliable message that something within the body is malfunctioning and needs to be cured. Additionally, it can affect a person's behavior, memory, concentration, and intellectual capacities. Hence, it should never be neglected and needs to be taken seriously and treated promptly.

In this regard, many entities around the world work on improving pain relief with the help of researchers and professionals involved in the diagnosis and treatment of pain, because they believe that the relief of pain is a human right; the International Association for the Study of Pain (IASP) [1] is one of them, and it is very active in this sector, one of its main events is the biennial World Congress on Pain.

The main motivation for this research is based on providing quality healthcare services, managing patients' conditions and doing the right diagnoses at the right times is pivotal in life-saving.

COVID-19 has been a paramount motivator for this study. With an unmanageable number of daily cases and the necessity of physical distancing, many countries found the perfect opportunity to switch to tech-enabled and AI-empowered solutions. From online businesses and Chabot-assisted services to robots as assistants to automatic temperature checks in public areas to real-time identification of undisciplined citizens, numerous innovative solutions have emerged to contain the spread of the virus. Having different impacts from each other, each solution proved its worth during these unprecedented times.

In any hospital and under any circumstances, for instance during COVID-19, we can have uncontrollable new daily cases, and medical staff can easily become very busy and fail to manage all emergency cases at once in addition to looking after patients who need more attention and care. Additionally, patients and caregivers are sometimes forced to abide by physical distancing rules to avoid contaminations. Furthermore, symptoms associated with the most critical emergencies, such as apneas, are generally very expressive through facial expressions. All these issues have motivated the creation of an automatic pain detection system based on facial expressions.

Being ahead of research topics of the decade, computer vision-empowered pain assessment is in full effervescence. Yet, it is still not widespread in medical centers due to its incompleteness and the necessity to be fused with other disciplines, such as psychology, which makes it a very intriguing and challenging topic that deserves intensive research. This has also been a great motivator to start this research project.

The contribution of this study is to build a reliable pain assessment system based on patients' facial expressions and become a game-changer in healthcare. This study can be broken down into three major contributions, which are:

- Giving the opportunity to all patients, regardless of the language they speak as well as their psychological and physical conditions, to express their pain accurately and receive the right medical care at the right time.
- Constantly looking after patients and notifying medical staff about emergencies while keeping them focused on their main tasks.
- Avoid hiring more medical staff whose sole task is to ceaselessly look after patients and notify doctors only in case of an emergency.

The novelty of this study is the design of a deep learning classification model based on a tailored CNN architecture to be used as a pain assessment tool, in the following way: a camera would be recording patients when they are not under any supervision and sending input frames to the classification model which will process them in real-time and classify them into pain/no-pain images, and ultimately, notify doctors if the pain is detected.

2. Related Work

The field of image processing is in constant development and research. Whether it is applied in medicine, security, satellite imagery analysis or seismic imaging, thousands of researchers are pioneering daily and coming across new methods and approaches to give the advantage of computer-aided imaging. Medical imaging is one such field that requires daily improvements and growth, due to its complexity, diversity of cases and importance of accuracy. Numerous studies on pain detection using facial expressions have been conducted on diverse data sets using different data analytics approaches.

Sourav Dey Roy *et al.* [2] have conducted their research on the UNBC-McMaster Shoulder Pain Expression Archive Database [3] in the following way: First, they converted all image frames into grayscale images. After that, they performed shape alignment using the Generalized Procrustes Analysis (GPA). Next, they applied texture warping, in which the texture of all images is warped with respect to the base shape using an affine warping. Their methodology achieved an accuracy of 82.43% for the pain level estimation.

Xiaojing Xu *et al.* [4] developed an ensemble learning model based on the UNBC-McMaster Shoulder Pain Expression Archive Database. They performed face detection using the cascade DPM face detector and used five metrics to predict the level of pain: the Prkachin and Solomon Pain Intensity (PSPI) score, the Visual Analog Scale (VAS) score, the Observers Pain Rating (OPR) score, the Affective Motivational Scale (AMS) score and the Sensory Scale (SEN) score.

Lijun Yin *et al.* [5] developed their own 3D Dynamic Facial expression database using the Di3D (Dimensional Imaging 3D) face capturing system including six universal expressions: Anger, disgust, fear, happiness, sadness and surprise. They applied a 2D Hidden Markov Model (2D-HMM) to learn the temporal dynamics and spatial relationships of facial regions. The accuracy of their model was 90.44%.

Ghazal Bargshady *et al.* [6] used a hybrid method by combining the CNN and RNN architectures to classify VGGFace images into four classes: Strong Pain, Mild Pain, Weak Pain and No Pain. The VGG Face model was tested on 25 subjects using a leave-one-out cross-validation, and it gave an average accuracy of 75.2%.

One of the methodologies proposed by Laduona Dai *et al.* [7] is an Action Units based method. First, Features/AUs were extracted using software called OpenFace 2.0. Then, an SVM model was trained based on the intensity values of those AUs with a binary output (pain/no-pain). The model was trained and tested using the UNBC-McMaster Shoulder Pain Expression Archive Database using 5-fold cross-validation and gave an accuracy of 85%, but all samples with facial movements were classified as "Pain movements", mainly because all no-pain samples represent neutral faces with no movements.

Mohammad Tavakolian *et al.* [8] used a binary representation of facial expressions, using the UNBC-McMaster Shoulder Pain Expression Archive Database. Their goal was to do a multinomial classification to classify patients' faces into many pain levels. They used a CNN model to extract features from randomly captured frames from each video segment and encoded them to binary, using a deep neural network. They used the Hamming distance for their classification and achieved an accuracy of 98.54%.

Zakia Hammal *et al.* [9] performed a multinomial classification on the UNBC-McMaster Shoulder Pain Expression Archive Database to classify frames into four pain levels, which were defined on the basis of the PSPI score. Then, they trained 4 separated SVM models on each class, which reached the respective accuracies: 97%, 96%, 96% and 98%.

Pau Rodriguez *et al.* [10] used Long Short-Term Memory (LSTM) networks on the UNBC-McMaster Shoulder Pain Expression Archive Database to do a binary classification. First, they used a CNN model to learn facial features from the VGG-Face dataset. Then, they linked it to an LSTM model to perform the classification. The highest accuracy they have achieved was 90.3%.

Zhanli Chen *et al.* [11] came up with a different approach from the ones cited previously, which they implemented based on the UNBC-McMaster Shoulder Pain Expression Archive Database. Instead of relying on pain intensity scores to classify images and video sequences, they detected individual pain-related AUs by themselves and combined them using two different structures (compact and clustered). After that, they used two different frameworks, namely the Multiple Instance Learning (MIL) and Multiple Clustered Instance Learning (MCIL) to train their models based on low-dimensional features. Their classifier achieved an accuracy of 87%.

Reza Kharghanian *et al.* [12] used an unsupervised learning approach to classify unlabeled images. First, they extracted features of shape and appearance separately from faces, using a Convolutional Deep Belief Network (CDBN). Those extracted features were then used to train an SVM model with a linear kernel that has two output classes (pain/no-pain). Their model was tested on the UNBC-McMaster Shoulder Pain Expression Archive Database, and it achieved an accuracy of 87.2%.

3. Proposed CNN Modified Architecture

CNNs have been used previously to solve the automatic pain assessment problem, using the UNBC-McMaster Shoulder Pain dataset, some of which got inspired by the famous VGG16 architecture [13], shown in **Figure 1**.

Based on existing architectures and after many trials and hyperparameter tunings, the modified architecture shown in **Figure 2** has been selected for this study.

This proposed architecture was inspired by the VGG16 architecture and modified to fit the pain detection problem, by replacing the number of neurons in the second fully connected layer (FC2) with 1000 (against 4096 in VGG16) and the number of outputs with 2 for the two classes, pain and no-pain.

In this proposed architecture (**Figure 2**), images are initially preprocessed and resized to 224 × 224 pixels. After that, they are fed into the CNN model which processes them as follows:

- Images go through two convolutional layers, with 64 (3 × 3) filters, a stride of 1 and a padding of 1, followed by a max-pooling layer with a filter size of 2 × 2 and a padding of 2.

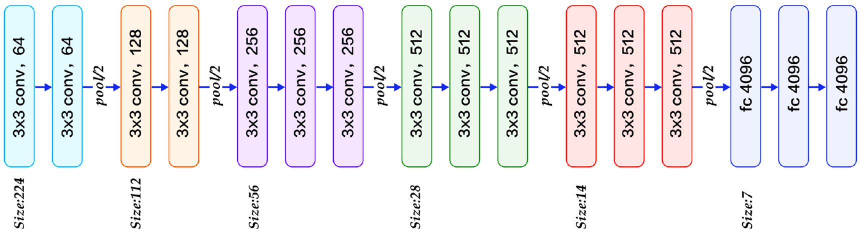


Figure 1. VGG16 Architecture (Source [13]).

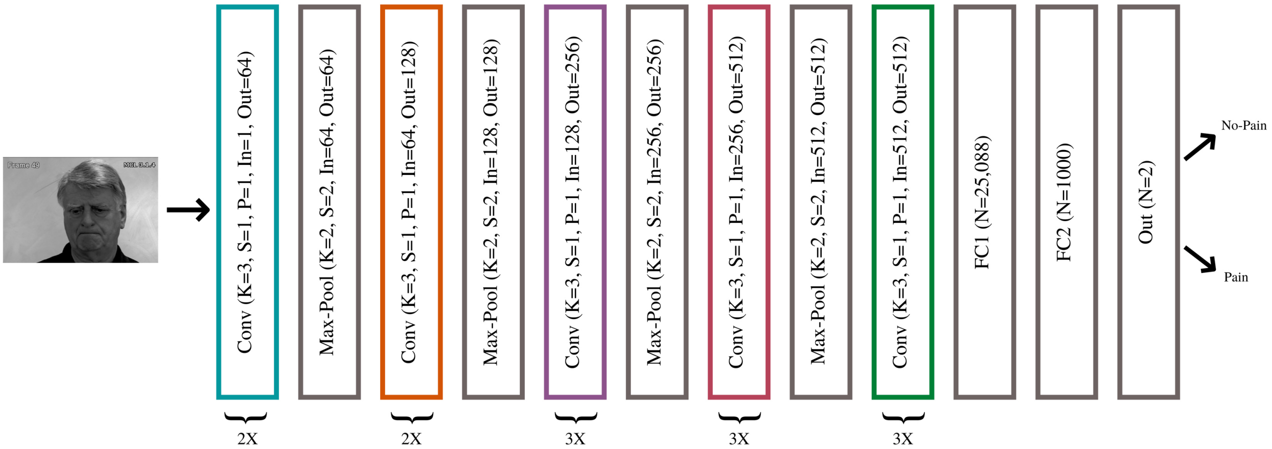

Conv (K=3, S=1, P=1, In=1, Out=64)
Max-Pool (K=2, S=2, In=64, Out=64)
Conv (K=3, S=1, P=1, In=64, Out=128)
Max-Pool (K=2, S=2, In=128, Out=128)
Conv (K=3, S=1, P=1, In=128, Out=256)
Max-Pool (K=2, S=2, In=256, Out=256)
Conv (K=3, S=1, P=1, In=256, Out=512)
Max-Pool (K=2, S=2, In=512, Out=512)
Conv (K=3, S=1, P=1, In=512, Out=512)
FC1 (N=25,088)
FC2 (N=1000)
Out (N=2)
No-Pain
Pain
2X
2X
3X
3X
3X

Figure 2. Proposed modified VGG16 architecture.

- The resulting tensors go through two additional convolutional layers, with 128 (3 × 3) filters, a stride of 1 and a padding of 1, followed by a max-pooling layer with a filter size of 2 × 2 and a padding of 2.
- The resulting tensors go through three more convolutional layers, with 256 (3 × 3) filters, a stride of 1 and a padding of 1, followed by a max-pooling layer with a filter size of 2 × 2 and a padding of 2.
- The resulting tensors go through three further convolutional layers, with 512 (3 × 3) filters, a stride of 1 and a padding of 1, followed by a max-pooling layer with a filter size of 2 × 2 and a padding of 2.
- The resulting tensors go through three final convolutional layers, with 512 (3 × 3) filters, a stride of 1 and a padding of 1.
- The resulting feature maps are flattened and converted to a fully connected layer of 7 × 7 × 512 = 25,088 neurons, which is connected to a second fully connected layer of 1000 neurons, which is finally connected to the output layer of two neurons, corresponding to the two classes (*i.e.*, pain and no-pain).

The model's architecture is illustrated as follows:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_2 (Conv2D)	(None, 224, 224, 64)	36,928
max_pooling2d_1 (MaxPooling)	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73,856
conv2d_4 (Conv2D)	(None, 112, 112, 128)	147,584
max_pooling2d_2 (MaxPooling)	(None, 56, 56, 128)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	295,168
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590,080
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590,080
max_pooling2d_3 (MaxPooling)	(None, 28, 28, 256)	0

Continued

conv2d_8 (Conv2D)	(None, 28, 28, 512)	1,180,160
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2,359,808
conv2d_10 (Conv2D)	(None, 28, 28, 512)	2,359,808
max_pooling2d_4 (MaxPooling)	(None, 14, 14, 512)	0
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2,359,808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2,359,808
conv2d_13 (Conv2D)	(None, 14, 14, 512)	2,359,808
max_pooling2d_5 (MaxPooling)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25,088)	0
dense_1 (Dense)	(None, 1000)	4,097,000
dense_2 (Dense)	(None, 2)	1001
Total params: 121,577,233		
Trainable params: 121,577,233		
Non-trainable params: 0		

The model's hyperparameters have been fixed at the following values:

- Activation function: ReLU + Softmax in the last fully connected layer.
- Loss function: Cross-entropy loss.
- Optimizer: Adam.
- Use of batch normalization: Batch normalization allows the model to be trained on separate mini-batches. As a consequence, weights are updated between batches, so the number of epochs required to train the model is minimized.
- Use of dropout regularization: In our case, since the number of examples is limited, the dropout regularization will be used to reduce overfitting. Dropout drops random nodes during training; thus, a single model can be seen as a number of different simulated architectures, with less computing power required to train actual models.

4. Dataset

The data used to fulfill this study is the UNBC-McMaster Shoulder Pain Expression Archive Database [3]. It has been collected by researchers from McMaster University and the University of Northern British Columbia and shared for research purposes in collaboration with Carnegie Mellon University and the University of Pittsburgh [14].

A total of 100 patients who were suffering from shoulder pain caused by arthritis, bursitis, tendonitis, subluxation, rotator cuff injuries, impingement syndromes, bone spur, capsulitis and dislocation, underwent the following range-of-motion tests on both affected and unaffected limbs:

- Abduction: In abduction movements, the arm is lifted forward and up in the sagittal plane (*i.e.*, the plane which divides the body into right and left parts).

- Flexion: In flexion, the humerus (or upper arm) moves forward from the rest of the body.
- Internal (resp. external) rotation of each arm separately: Involves bending the arm 90 degrees at the elbow, abducting it by 90 degrees and finally turning it internally (resp. externally).

All those tests were performed under active and passive conditions. In active tests, patients had to perform movements by themselves to the best of their ability (*i.e.*, until pain would prevent them from doing further movements), whereas in passive tests, a physiotherapist would move patients' limbs until the maximum range would be reached or they would be stopped by patients themselves who would not bear the pain anymore. For each test, a video sequence was recorded with frequent changes in pose, as we can see in **Figure 3**.

The pitch, yaw and roll in frames taken from all sequences range between −40 and 30. According to Lucey *et al.* [14], head movements coincide with painful facial expressions.

A total of 200 video sequences containing spontaneous facial expressions related to genuine pain were recorded for the 100 patients. Those sequences were rated by observers and pain was self-reported by patients using four different types of assessments, to make sure that the rating is as accurate as possible:

1) The Sensory Scale (SEN): Used to reflect the pain intensity. It starts at "extremely weak" and finishes at "extremely intense".

2) The Affective-Motivational Scale (AFF): Used to reflect the unpleasantness incurred by the pain. It starts at "bearable" and finishes at "excruciating".

3) The Visual Analogue Scale (VAS): Gives more flexibility to the patient to rate pain in the most accurate way possible by providing a 10 cm scale anchored at each end with the words "No Pain" and "Worst Pain", on which patients can select the most accurate intensity, even if it ranges between two specific intensities (e.g., between "moderately strong" and "very strong").

4) Observers Pain Intensity (OPI): Trained observers rated patients' pains using a 6-point scale ranging from 0 (no pain) to 5 (strong pain). A number of those ratings were rated by a second rater to assess their reliability, and the Pearson correlation between both ratings was 80%. In addition, the correlation

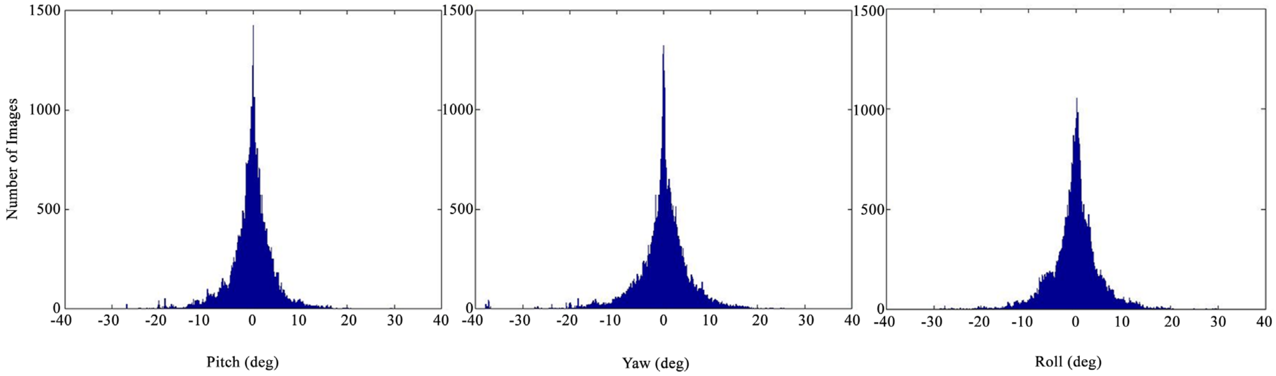


Figure 3. Pitch, yaw, and roll of frames in the UNBC-McMaster dataset (Source [14]).

between the OPI and patient self-reported VAS was 74%, which is higher than the high concurrent validity threshold (70%) [14]. Thus, sequence-level ratings can be considered trustworthy. A total of 48,398 frames were captured from each sequence and coded into Action Units (AUs) by certified Facial Action Unit System (FACS) coders.

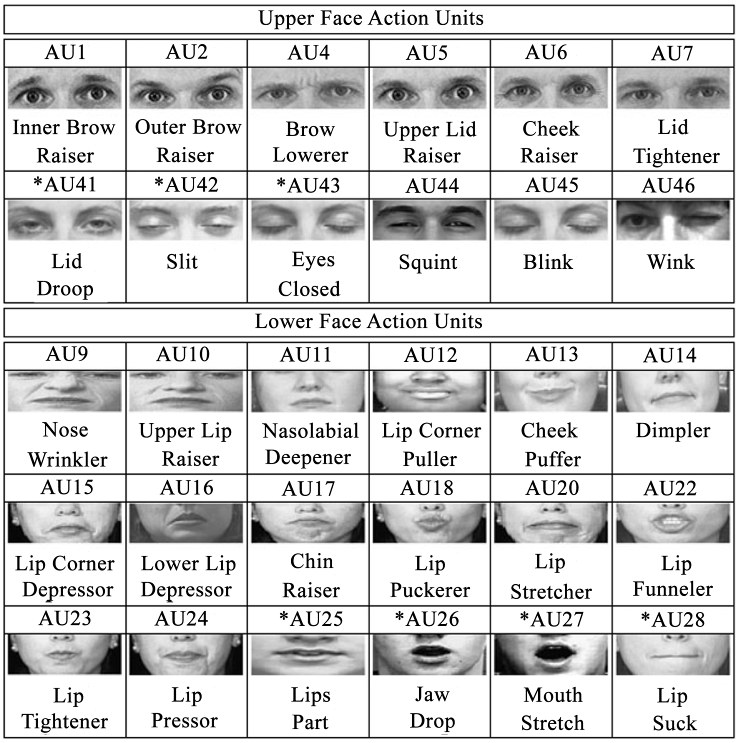
4.1. Pain Assessment

We need to introduce three basic components for pain assessment, namely action units (AUs) and Facial Action Coding system (FACS), and the Prkachin and Solomon Pain Intensity (PSPI) score [16].

Action Units are the fundamental actions involving one or multiple muscles in response to a certain feeling, such as cheek raising, lip stretching and head left turning. AUs are encoded using the FACS.

Facial Action Coding System (FACS) is a system used to taxonomize facial actions or movements. It is used by psychologists and animators to categorize a person's emotions based on facial actions. The FACS assigns to every AU a unique code. For instance, AU number 6 corresponds to cheek rising, AU number 20 corresponds to lip stretching and AU number 51 to head left turning. A typical example of AUs is depicted in **Figure 4**.

Prkachin and Solomon [16] found that the most representative AUs for pain are brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43). Based on that, they defined the following pain formula:



Upper Face Action Units					
AU1	AU2	AU4	AU5	AU6	AU7
Inner Brow Raiser	Outer Brow Raiser	Brow Lowerer	Upper Lid Raiser	Cheek Raiser	Lid Tightener
*AU41	*AU42	*AU43	AU44	AU45	AU46
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU9	AU10	AU11	AU12	AU13	AU14
Nose Wrinkler	Upper Lip Raiser	Nasolabial Deepener	Lip Corner Puller	Cheek Puffer	Dimpler
AU15	AU16	AU17	AU18	AU20	AU22
Lip Corner Depressor	Lower Lip Depressor	Chin Raiser	Lip Puckerer	Lip Stretcher	Lip Funneler
AU23	AU24	*AU25	*AU26	*AU27	*AU28
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck

Figure 4. Example of action units [15].

$$\text{Pain} = \text{AU}(4) + (\text{AU6 or AU7}) + (\text{AU9 or AU10}) + \text{AU43} \tag{1}$$

where AU(4) and AU(43) are always present in pain, one of AU6 and AU7 and one of AU9 and AU10 must be present too. The highest intensity would be selected if both are present.

Each AU is scored on a 6-point scale based on its intensity ranging between "a = 0" for absent intensity and "f = 5" for maximum intensity, except for AU43 which has a binary intensity: 0 (absent) or 1 (present) [16]. For instance, AU6d refers to cheek rising with an intensity of 3.

The Prkachin and Solomon Pain Intensity score (PSPI score) [16] was introduced as the only metric for pain intensity based on facial expressions.

From Formula (1), the PSPI score formula is:

$$\text{PSPI score} = \text{AU4} + \max(\text{AU6}, \text{AU7}) + \max(\text{AU9}, \text{AU10}) + \text{AU43} \tag{2}$$

Consequently, if a frame is coded as:

$$\text{FACS} = \text{AU4a} + \text{AU6d} + \text{AU7d} + \text{AU12d} + \text{AU43} \tag{3}$$

Then, its PSPI score is derived as:

$$\text{PSPI score} = 1 + \max(4, 4) + 1 = 6 \tag{4}$$

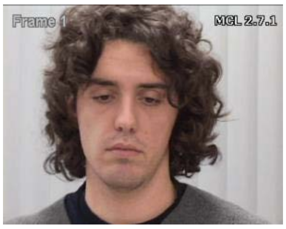
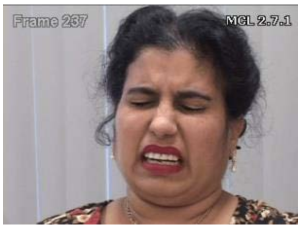
Based on the calculated PSPI score, we define two pain classes as shown in Table 1.

4.2. Exploratory Data Analysis

As mentioned in the previous section, the UNBC-McMaster dataset has 100 subjects (patients) who underwent different kinds of tests in different sequences, from which a sample of frames has been captured. The average size of each frame is 320 × 240 pixels. Figure 5 shows the number of frames captured for each patient (subject).

Based on Figure 5, we can observe that subjects 14, 16 and 19 have the highest number of recorded frames, as opposed to subjects 23, 7 and 12, which have the lowest number of frames. This means that if the training set comprises frames of subject 16 and 23 for example, the model would perform better in the classification of frames of subject 16 than those of subject 23, because it was trained on

Table 1. Pain classes from the UNBC-McMaster dataset [3].

	Pain Classes	
	No Pain	**Pain**
PSPI score	O	1 and above
Facial Expression example		

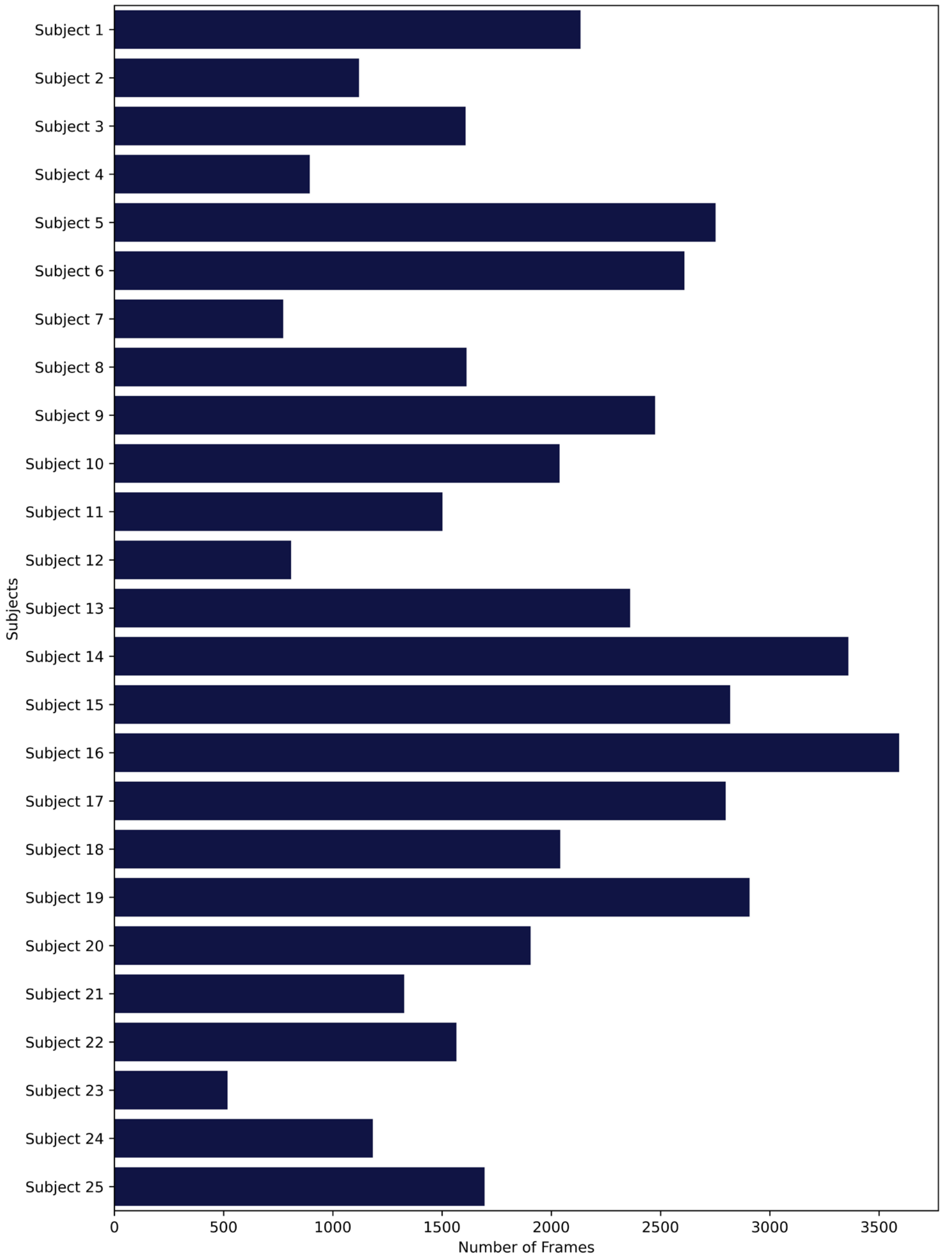


Figure 5. Distribution of frames for each patient (subject).

more examples of subject 16. Hence, subject 16's facial features became familiar to the model. The number of frames belonging to each class (pain/no-pain) is shown in **Figure 6**.

We can note from **Figure 6** that the dataset has imbalanced classes. There is a clear dominance of "No Pain" frames over "Pain" ones. If the model is trained on more "No Pain" examples, it would be biased towards detecting "No Pain" faces, which means that it has more chances to misclassify a "Pain" example as "No Pain", because it got accustomed to "No Pain" examples. In order to have a fair prediction model, to some extent, we will train our model on as many pain examples as no pain ones.

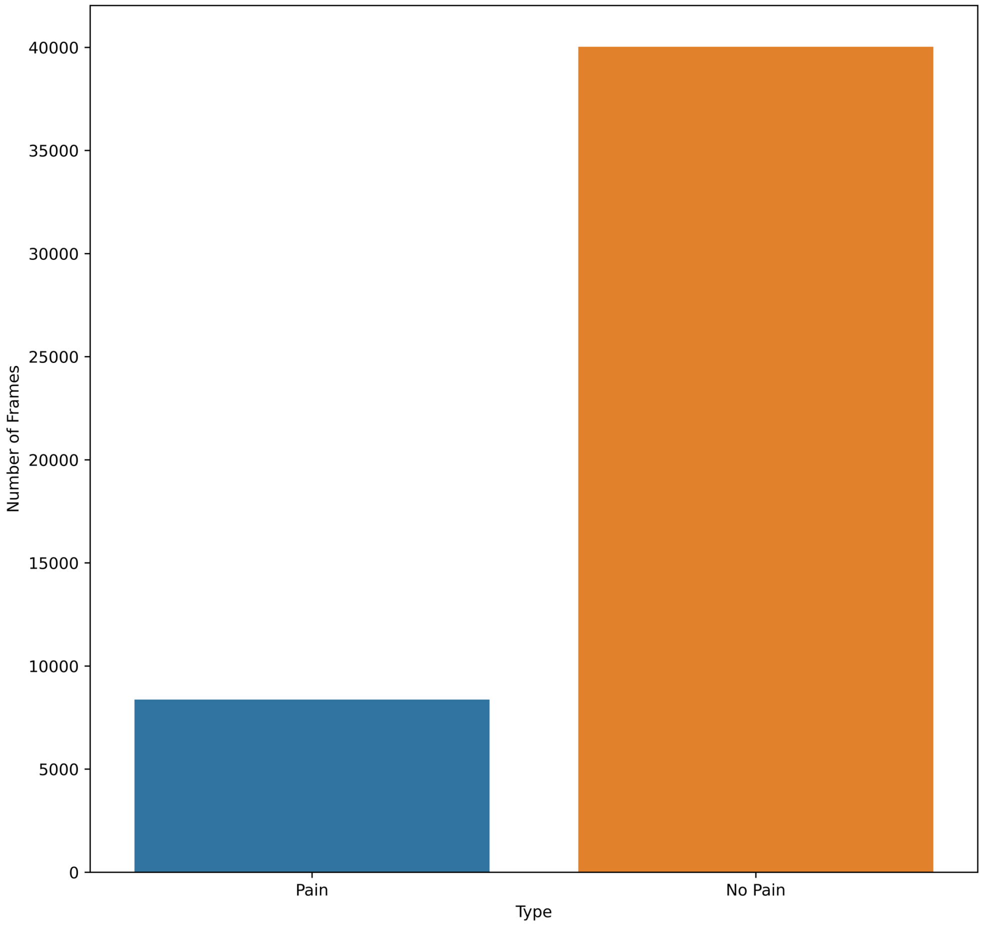


Figure 6. Distribution (classification) of frames between two pain classes.

4.3. Data Processing

Real-world data is often subject to errors, noise and outliers. Before it can be visualized or processed, it needs to be preprocessed and cleaned. Preprocessing is a crucial phase in data analysis. It allows us to do a data quality assessment and resolve all issues which might affect the performance of our model. For numerical and categorical data, the following data dimensions are assessed:

- Completeness, to check if there are missing values.
- Conformity, to verify that all values are stored in a common format.
- Consistency, to check if there are any conflicting values.
- Accuracy, to check if there are incorrect, out of bound or out of date values.
- Duplicates, to check if an entry or a feature is repeated.
- Integrity, to check if there are unreferenced entries.

The process is quite different with images. Images usually have to undergo a couple of transformations such as filtering, cropping, resizing, color grading, rotating and mirroring, mainly to be simplified for faster processing, to be formatted or to be adapted to a certain machine learning algorithm. The following preprocessing steps have been used in this study using the OpenCV library [17] as depicted in **Figure 7**.

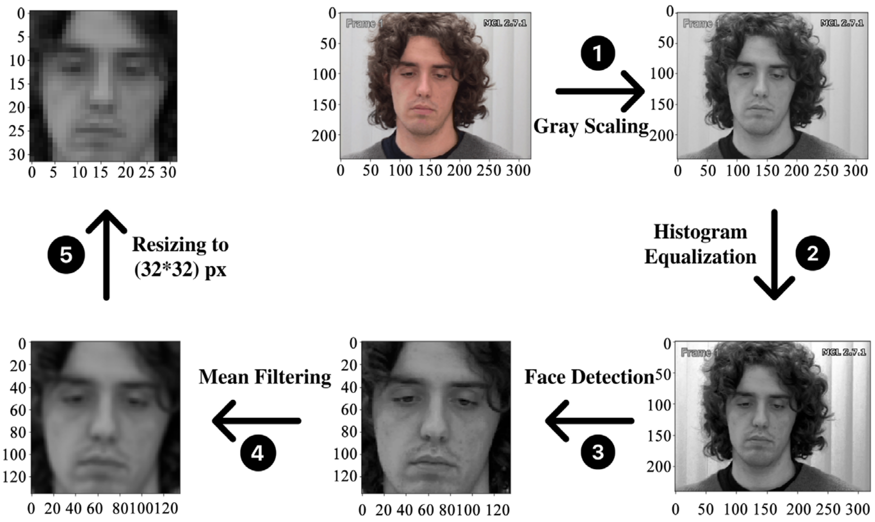


Figure 7. Preprocessing steps applied to one image from the dataset [3].

- Gray scaling, to keep one channel in each image and simplify the convolution operations. Colors are not impactful in this problem. However, they can be of the essence in other problems (e.g., the classification of fruits into different quality classes based on many factors, including their color).
- Histogram equalization, to unify and improve the contrast of every image for better edge detection. Consequently, images would neither be too bright nor too dark.
- Face detection, using the predefined Haarcascade Frontal Face Detector [18].
- Image cropping, to keep the frontal face of the patient only.
- Mean filtering, to eliminate unrepresentative pixels. In this step, every pixel is replaced with the average value of its neighbors, including its own value.
- Normalization/Standardization, to keep pixel intensities within the range [−1, 1] and have small standardized values.
- Resizing, to adapt sizes to the CNN architecture.

4.4. Data Modeling

The CNN model was built based on the architecture described in Section 3, trained on equally-distributed images for each class, using batches of 10 images, and evaluated using the cross entropy loss and the accuracy of each batch in each epoch, as shown in Figure 8 and Figure 9.

From Figure 8, we can notice a random variation in the accuracies, as the model was being trained on each batch for the first time (first epochs). After about 550 iterations, the learning accuracies started ranging between 80% and 100%, with two falls to 70%, which exhibits success in the model's hyperparameter tuning.

Similarly, losses started from around 0.7 in the first epochs, as we can see in Figure 9. After about 700 iterations, they started ranging between 0 and 0.3, with occasional rises between 0.4 and 0.6.

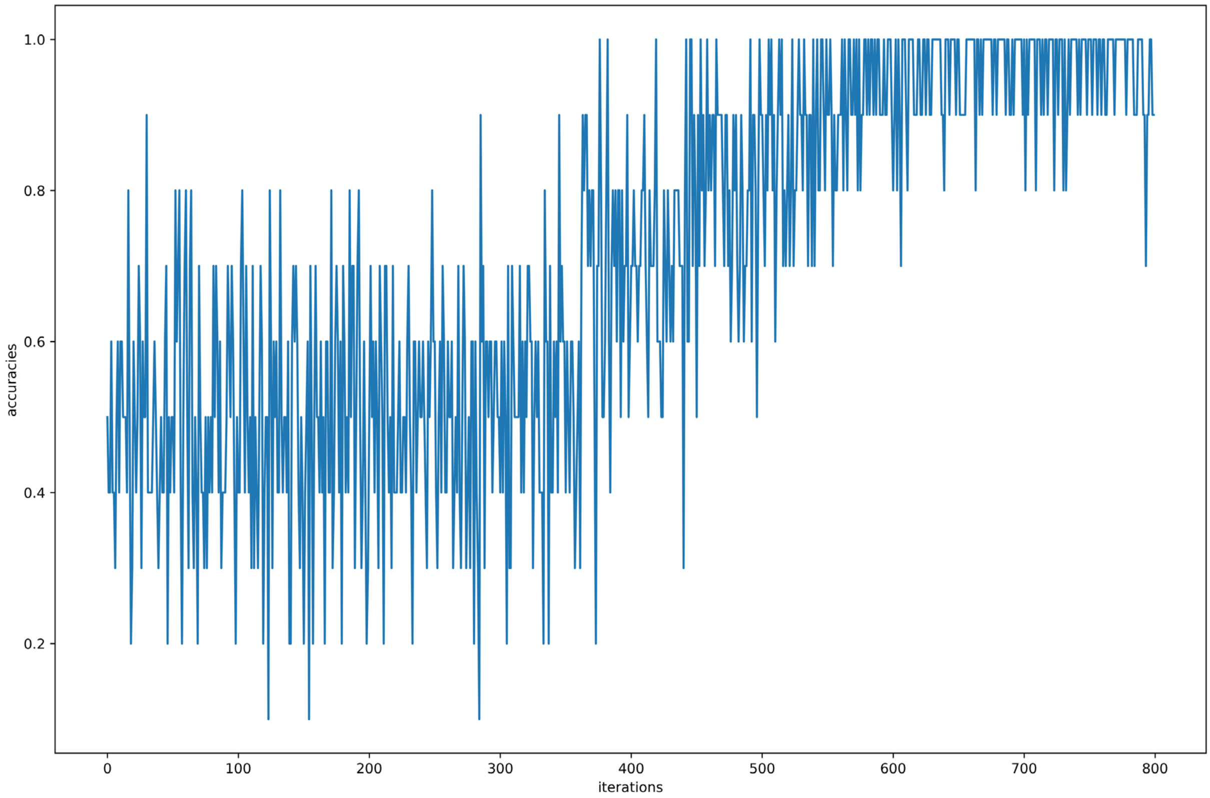


Figure 8. Training accuracies for each batch per epoch.

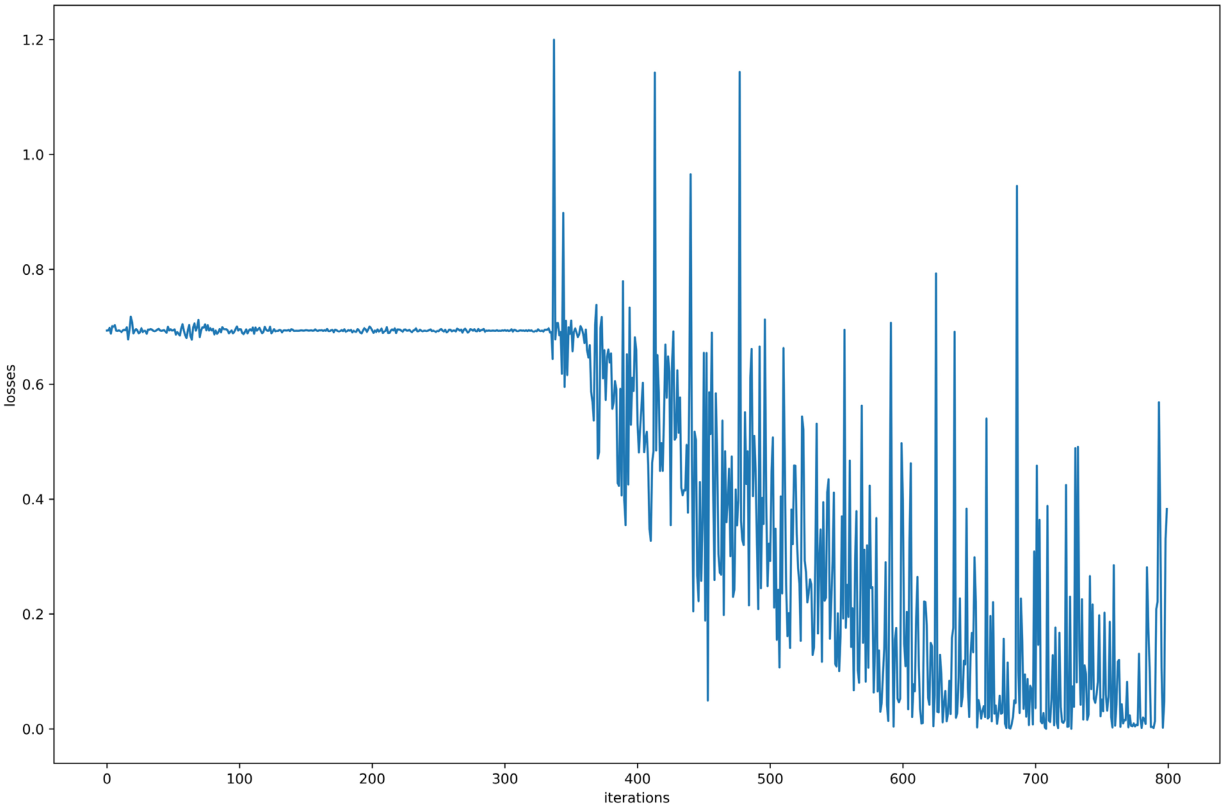


Figure 9. Training losses for each batch per epoch.

5. Performance Analysis

After the training, the model was evaluated on a sample of 400 images based on its accuracy, sensitivity and specificity.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

where:

TP (True Positives) is the number of pain images that were correctly classified.

TN (True Negatives) is the number of no-pain images which were correctly classified.

FP (False Positives) is the number of pain images that were misclassified.

FN (False Negatives) is the number of no-pain images which were misclassified.

Table 2 shows the confusion matrix of the model on the testing set.

Using the above Equations (5) - (7) we calculated the performance scores for our testing dataset.

$$\text{Accuracy} = \frac{200 + 170}{200 + 170 + 0 + 30} = 92.5\%$$

$$\text{Sensitivity} = \frac{200}{200 + 30} = 86.96\%$$

$$\text{Specificity} = \frac{170}{170 + 0} = 100\%$$

In Table 3, we present a comparative study with various accuracy results from different research papers.

Our proposed model achieved an accuracy of 92.5%. In comparison with techniques developed by other researchers, as shown in Table 3, we can say that our pain detection method generates great results on average on the UNBC McMaster Shoulder Pain Expression Archive database, as it overperformed other complex architectures.

Table 2. Confusion matrix.

		Actual Classes	
		Pain	No Pain
Predicted Classes	Pain	TP = 200	FP = 0
	No Pain	FN = 30	TN = 170

Table 3. Comparative study with other researchers' results.

Author	Accuracy
Sourav Dey Roy *et al.* [2]	82.43%
Lijun Yin *et al.* [5]	90.4%
Ghazal Bargshady *et al.* [6]	75.2%
Pau Rodriguez *et al.* [10]	90.2%
Zhanli Chen *et al.* [11]	87%
Reza Kharghanian *et al.* [12]	87.2%
Our Method	92.5%

6. Conclusions and Future Work

In this paper, we have presented a modified CNN model for the automatic pain detection problem. It was built and trained to detect pain through patients' facial expressions, using the UNBC McMaster Shoulder Pain Expression Archive database. Our proposed model can alert medical staffs in a timely manner regarding patients' conditions based on their facial expressions. The model achieved an accuracy of 92.5% which stands in competition with other researchers' results. As we can see, the model is very specific, with an average specificity of 100% based on the testing sample. This means that the model has a perfect performance when no pain is expressed, which in its turn means that the pain detection system is most likely never going to cause false alarms to the medical staff.

As for the sensitivity, the model detects pain in 86.96% of pain situations, which is still excellent, since we know that weak and mild pains do not always infer an emergency. Thus, the error is permitted by the model in some cases, where pain is not expressed with strong intensity. In future improvements, more emphasis will be put on testing different architectures to extract more features from the facial frames. Additionally, we will consider using *Conditional Generative Adversarial Networks* (*CGANs*) to generate synthetic facial images, in order to strengthen the training set.

7. Strengths & Limitations

Despite the fact that our dataset had very limited examples (frames from 25 patients only), it could still generalize well and gave excellent and reliable results on newly seen faces, using the least complex architecture possible for such a complex classification problem. However, we would get a much better performing model if we could combine our dataset with other more diversified datasets and give a try to different architectures, which are more adapted and tailored to the pain detection problem than a simple general architecture, which has proven its success with many computer vision problems.

Acknowledgements

The work presented here used the UNBC McMaster Shoulder Pain Expression

Archive database provided by the University of Pittsburgh.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] IASP (2021) International Association for the Study of Pain. https://www.iasp-pain.org/about/

[2] Roy, S.D., Bhowmik, M.K., Saha, P. and Ghosh, A.K. (2016) An Approach for Automatic Pain Detection through Facial Expression. *Procedia Computer Science*, **84**, 99-106.

[3] University of Pittsburgh (2011) UNBC-McMaster Shoulder Pain Expression Archive Database. https://www.pitt.edu/~emotion/um-spread.htm

[4] Xu, X.J., Huang, S.R., Children's Hospital and De Sa, V.R. (2019) Pain Evaluation in Video Using Extended Multitask Learning from Multidimensional Measurements. *Proceedings of Machine Learning Research*, **116**, 141-154.

[5] Yin, L., Chen, X., Sun, Y., Worm, T. and Reale, M. (2008) A High-Resolution 3D Dynamic Facial Expression Database. 2008 8*th IEEE International Conference on Automatic Face and Gesture Recognition*, Amsterdam, 17-19 September 2008.

[6] Bargshady, G., Soar, J., Zhou, X., Deo, R.C., Whittaker, F. and Wang, H. (2019) A Joint Deep Neural Network Model for Pain Recognition from Face. 2019 *IEEE* 4*th International Conference on Computer and Communication Systems*, Singapore, 23-25 February 2019, 52-56.

[7] Dai, L., Broekens, J. and Truong, K.P. (2019) Real-Time Pain Detection in Facial Expressions for Health Robotics. 2019 8*th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos*, Cambridge, 3-6 September 2019.

[8] Tavakolian, M. and Hadid, A. (2018) Deep Binary Representation of Facial Expressions: A Novel Framework for Automatic Pain Intensity Recognition. *International Conference on Image Processing*, Athens, 7-10 October 2018.

[9] Hammal, Z. and Cohn, J.F. (2012) Automatic Detection of Pain Intensity. *Proceedings of the ACM International Conference on Multimodal Interaction (ICMI'*12*)*, **2012**, 47-52.

[10] Rodriguez, P., *et al.* (2017) Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification. *IEEE Transactions on Cybernetics*, 1-11. https://doi.org/10.1109/TCYB.2017.2662199

[11] Chen, Z., Ansari, R. and Wilkie, D. (2019) Learning Pain from Action Unit Combinations: A Weakly Supervised Approach via Multiple Instance Learning. *IEEE Transactions on Affective Computing*, 1 p. https://doi.org/10.1109/TAFFC.2019.2949314

[12] Kharghanian, R., Peiravi, A. and Moradi, F. (2016) Pain Detection from Facial Images Using Unsupervised Feature Learning Approach. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Orlando, 16-20 August 2016.

[13] (2016) What is the VG Gneural Network? https://www.quora.com/What-is-the-VGG-neural-network

[14] Lucey, P., Cohn, J.F., Prkachin, K.M., Solomon, P.E. and Matthews, I. (2011) Painful Data: The UNBC-McMaster Shoulder Pain Expression Archive Database. 2011 *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, Santa Barbara, 21-25 March 2011.

[15] What-When-How (2020) Facial Expression Recognition Techniques Part 1. https://what-when-how.com/face-recognition/facial-expression-recognition-face-recognition-techniques-part-1/

[16] Prkachin, K. and Solomon, P. (2008) The Structure, Reliability and Validity of Pain Expression: Evidence from Patients with Shoulder Pain. *Pain*, **139**, 267-274.

[17] OpenCV (2021). https://opencv.org

[18] Haarcascade Frontal Face Detector (2021). https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml