ASEAN
Journal of Scientific and Technological Reports
Online ISSN:2773-8752

Research article

A Hybrid Facial Expression Recognition System Based on Machine Learning and Deep Learning Models

Akkapop Prasompon[1*], Prompong Sugunnasi[1], and Atigorn Sanguansri[2]

[1] Faculty of Engineering, Chiang Mai University, Chiang Mai, 50200, Thailand
[2] Faculty of Science and Agricultural Technology, Rajamangala University of Technology Lanna, Nan, 55000, Thailand
* Correspondence: akkapop_praso@cmu.ac.th

Citation:
Prasomon, A.; Sugunnasil, P.; Sanguansri, A. A Hybrid facial expression recognition system basem on machine learning and deep learning models. *ASEAN J. Sci. Tech. Report.* **2026**, *29*(4), e261424. https://doi.org/10.55164/ajstr.v29i4.261424

Article history:
Received: September 21, 2025
Revised: February 14, 2026
Accepted: February 22, 2026
Available online: March 25, 2026

Abstract: Pain assessment through facial expressions is an important area of research because many patients cannot clearly communicate their pain levels. This study presents the first systematic comparison of continuous time-series versus tokenized sequence representations for facial action unit (AU)-based pain classification, introducing a novel application of NLP models (BERT) to discretized AU sequences treated as symbolic text. Two datasets were used: the UNBC-McMaster Shoulder Pain Archive (UNBC-SP) with about 48,000 frames, and the Multimodal Intensity Pain (MIntPain) dataset with about 187,900 frames. Action unit intensities were extracted using the Py-Feat library and then normalized, oversampled, and augmented. A range of models was tested, including random forest, support vector machine, recurrent neural networks, and BERT. Key contributions include: (1) demonstrating that continuous time-series models significantly outperform tokenized approaches (91% vs. 82% accuracy); (2) revealing that classical ensemble methods surpass deep learning on tokenized sequences in data-limited scenarios; and (3) establishing that disruptive augmentations harm performance while conservative methods maintain accuracy. The continuous-time series models achieved the best performance, reaching 91% accuracy on MIntPain and 84% on UNBC-SP, while the tokenized models peaked at 82%. The results suggest that preserving temporal details of facial action units provides an advantage for pain detection, especially with larger datasets, though tokenization may retain value in resource-limited settings. The study highlights the need for larger, more diverse datasets and for validation in real clinical settings to improve the reliability of automatic pain recognition.

Keywords: Facial expression recognition; deep learning; pain assessment; time-series analysis; data tokenization

1. Introduction

For patients with terminal illnesses, medical care involves more than treating the disease; enhancing their quality of life is equally important. Palliative care is a specialized medical service provided alongside primary treatment, focusing on alleviating undesirable symptoms in patients with serious illnesses such as cancer [1]. It has been shown to improve patients' quality of life and reduce the likelihood of depression compared to those not receiving such services [2-3].

According to a 2023 World Health Organization (WHO) report, an estimated 4.4 million people in the WHO European Region, including 140,000

children, require palliative care annually. However, people living in low and middle-income countries are much less likely to receive such care [4]. This disparity limits patients' access to necessary services and resources in certain regions [5]. The limitation and condition of measuring opioid pain relief in certain patients is one of the key reasons that contribute to patients' inability to obtain palliative treatment [4]. Opioids are a group of drugs with the highest analgesic effect that are primarily used to treat severe pain in patients. However, an opioid prescribing policy is rather rigorous because it may lead to addiction and cause side effects, including constipation, respiratory depression, etc. [6] A patient's pain level evaluation by a doctor is one of the criteria to prescribe opioids in a patient [7]. Unfortunately, a patient's limited ability to communicate in some situations could result from severe pain, terminal illness, or a relevant condition like dementia or laryngeal cancer that could cause misdiagnosis [8].

Numerous studies have leveraged computer vision and machine learning to recognize pain from facial images and videos, using frameworks such as the Facial Action Coding System (FACS) [9] and clinical pain scales such as the Numeric Rating Scale (NRS) [10-13]. Deep models, particularly convolutional neural networks (CNNs) [14], have been applied to pain-related facial expression datasets, including the UNBC-McMaster Shoulder Pain Expression Archive and BioVid, demonstrating that automated systems can approximate observer- or self-reported pain ratings under controlled conditions. Despite significant progress in automated facial pain recognition, several important limitations remain. One major issue is that publicly available datasets are relatively small and strongly imbalanced [15-16]. Most samples show no or mild pain, with only a few examples of severe pain, limiting how well deep learning models can learn and apply their knowledge to new cases. A further concern is that the temporal dimension of pain remains underutilized [15] [11-12], as many approaches rely on static image classification,n though facial muscle activity can change within milliseconds [9, 17]. While some recent studies have used LSTM networks or transformers to process continuous AU features over time [9, 12, 18], these methods treat AUs as continuous values and do not leverage the discrete, symbolic nature of facial muscle combinations. After several reviews, to our knowledge, no prior work has applied tokenized sequences of facial action units together with natural language processing models to video-based pain recognition [19]. Discrete facial encoding methods have been developed for general emotion tasks using vector quantization [20], where facial expressions are converted into symbolic tokens "similar to sentences". However, these tokenization frameworks have not been tested for pain intensity classification from video sequences. Chen et al. [19] showed that AU combinations can be represented as low-dimensional features that mimic human coder's decisions, but they did not treat these as discrete symbols in a learnable vocabulary. These gaps motivated our study to test whether tokenized AU sequences can be classified using NLP-based models for pain recognition and to compare this approach with continuous-time series methods on available datasets.

Therefore, we propose a dual-pathway framework that systematically compares two distinct data representation strategies for AU-based pain classification. In the first pathway, frame-level AU intensities are retained as continuous multivariate time series and modeled using neural networks and classical machine learning algorithms. The second pathway discretizes AU sequences through binarization and encodes them as symbolic tokens, creating text-like representations that enable the application of natural language processing (NLP) techniques. Each video frame's AU pattern is converted into a discrete "term" by thresholding AU intensities, and the sequence of frames forms a "passage" representing the entire video. These tokenized sequences are then evaluated using transformer-based models such as BERT [21], neural networks, and classical text classifiers. Both pathways address class imbalance via SMOTE oversampling and explore multiple sequence alignment strategies, as well as time-series augmentation methods. The complete framework is evaluated on two publicly available datasets: the UNBC-McMaster Shoulder Pain Expression Archive (UNBC-SP) [22], which contains approximately 48,000 frames from 25 patients with shoulder injury, and the Multimodal Intensity Pain (MIntPain) database [23] with approximately 187,900 frames from 20 healthy subjects undergoing graded electrical stimulation. By comparing continuous versus tokenized representations across a diverse set of machine learning and deep learning models, this study aims to identify

which data representation strategy and modeling approach best capture pain-related facial dynamics for automated pain assessment.

2. Materials and Methods

2.1 Facial Action Coding System

The Facial Action Coding System (FACS) is an anatomic-based coding system that encodes different facial movements to classify human facial expressions through the combination of multiple action units (AUs) - a facial configuration extracted from changes of visible unique facial muscle, consisting of forty-four different unique actions as shown on the table in Figure 1 where the left column represents single action units, and the right column represents more grossly defined behavioral action unit.

The AU can be calculated by applying the Euclidean distance equation, as stated in equation (1), where d(p, q) is the Euclidean distance from coordinates (p1, p2) to (q1, q2) for calculating the distance between two points from facial landmarks, as shown in Figure 2. For instance, to measure lip corner puller (AU12), we will look at the change of the Euclidean distance d(p36, q48), which is the distance between the right lip corner and right eye corner. As the distance between the two points decreases, the intensity of AU12 will increase.

Single action units		More grossly defined action units	
AU number	Description	AU number	Description
1	Inner brow raiser	8	Lips toward each other
2	Outer brow raiser	19	Tongue show
4	Brow lowerer	21	Neck tightener
5	Upper lid raiser	29	Jaw thrust
6	Cheek raiser	30	Jaw sideways
7	Lid tightener	31	Jaw clencher
9	Nose wrinkler	32	Lip bite
10	Upper lip raiser	33	Cheek blow
11	Nasolabial deepener	34	Cheek puff
12	Lip corner puller	35	Cheek suck
13	Sharp lip puller	36	Tongue bulge
14	Dimpler	37	Lip wipe
15	Lip corner depressor	38	Nostril dilator
16	Lower lip depressor	39	Nostril compressor
17	Chin raiser	43	Eyes closed
18	Lip pucker	45	Blink
20	Lip stretcher	46	Wink
22	Lip funneler		
23	Lip tightener		
24	Lip pressor		
25	Lips part		
26	Jaw drop		
27	Mouth stretch		
28	Lip suck		

Figure 1. List of Action Units in the Facial Action Coding System. (Left: individual action units; Right: combined or higher-level facial actions)

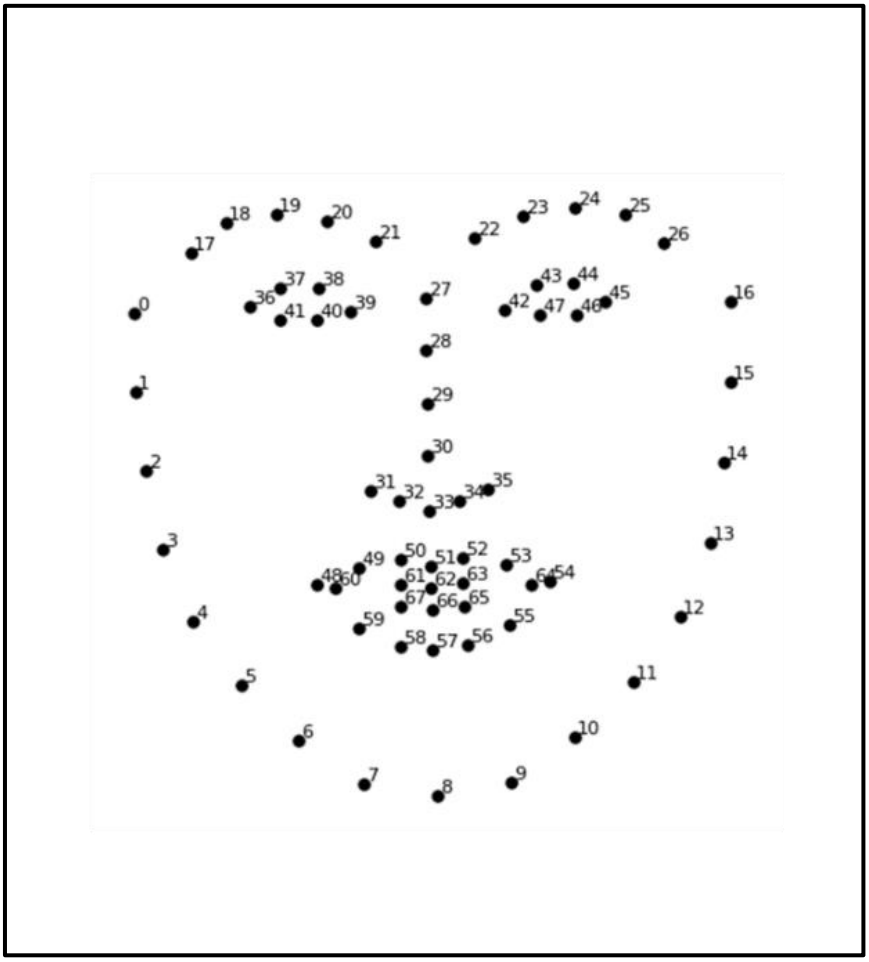

Figure 2. Facial landmarks with 64 key points used for AU intensity calculation

The combination of multiple AUs indicates an individual's expressions. For example, fear expression shows a positive correlation with AU1, AU2, and AU4 [24]. In the clinical context, the use of AU serves as a fundamental indicator for pain assessment through systematic detection of pain-related facial expressions. Prkachin and Solomon established a validated pain measurement framework using four key AUs: AU4 (brow lowering), AU6 or AU7 (narrowing of the eyes), AU9 or AU10 (raising the upper lip), and AU43 (eye closure). This pain measurement method is also known as Prkachin and Solomon Pain Intensity (PSPI) [25]. Facial pain expression can also be assessed using AU25, AU26, and AU27, which collectively represent mouth-opening movements. Research has demonstrated that self-reported pain intensity shows a stronger correlation with these mouth-opening action units than with eye-narrowing and upper-lip-raising movements [26] [27]. Furthermore, studies by Kunz et al. [28] which examined pain-related facial expressions in dementia patients, have identified AU12 (lip corner puller) as a predefined pain-related action unit associated with moderate-to-severe pain levels. Meanwhile, lip movement-related AUs such as AU20 (lip stretcher) and AU24 (lip press) occur more frequently in clinical pain studies than in experimental pain, suggesting that these AUs are more prominent in real-world clinical pain expressions than in brief experimental pain episodes.

However, these AU combinations would be insufficient if subtle changes in facial muscles remain undetectable by the human eye, as facial expressions can change within 350 milliseconds [9]. Therefore, computer vision techniques are essential in identifying pain in videos of recorded facial expressions. In this study, AUs were extracted using the Py-Feat library [24] in Python. The intensity of the extracted AUs ranges from 0.0 to 1.0, with 1.0 representing the most intense AU detected in the image. Extracted Intensity values were subjected to rolling average smoothing over a specific number of sequential frames to reduce noise from detection failures or periods when subjects displayed minimal facial movement. Since a short period of facial micro expression, approximately 1-6 frames at standard video frame rates of 24–30 fps, to evaluate the impact

of temporal smoothing on pain classification performance without fading pain-triggered frames, we compared a 1-frame window (no smoothing) with a 3-frame rolling average before the binarization step in token sequence modeling. We examine the level of pain following AUs, using multiple studies previously mentioned, including AU04, AU06, AU09, AU10, AU12, AU20, AU24, AU25, AU26, AU27, and AU43. These AUs were implemented for the analysis, selected based on their average correlation with the data labeled with pain values.

2.2 Pain Evaluation

There are numerous methods for evaluating patient pain; however, patient self-report is the most common for determining their pain level. The Numeric Rating Scale (NRS), also sometimes called the Visual Analogue Score (VAS) is the optimal indicator for assessing the pain level in a conscious patient for self-report; it consists of discrete values between zero and ten; 0 indicates "no pain at all," and 10 indicates "pain as severe as you can imagine." In this study, pain labels in the UNBC-SP dataset were classified into four classes as no pain, mild, moderate, and severe based on a 0-10 self-report NRS of zero, between 1 and 3, between 4 and 6, and greater than 6 points, respectively [29], since the MIntPain dataset has already been classified. We did not perform any further label classification on this dataset.

2.3. Methodology

This study proposes a dual-pathway framework for automatic facial pain recognition that integrates continuous-time-series modelling with a symbolic, token-based alternative. In this section, we introduce the corpora, describe preprocessing and feature engineering, outline the modelling strategies, and detail the evaluation protocol. All code was written in Python 3.

2.3.1 Dataset

We surveyed publicly available facial expression databases to identify those suitable for video-based pain level classification. Key criteria for inclusion were: (1) the data consist of face videos (or frame sequences) and (2) each sequence has an associated pain intensity label, preferably on a numerical rating scale (NRS). Several candidate datasets were reviewed [11] , and two were selected for our experiments due to their relevance and the availability of AU annotations. In this work we focus on the (1) UNBC-McMaster Shoulder Pain Expression Archive (UNBC-SP) and (2) Multimodal Intensity Pain (MIntPain) which provide both video and frame-level AU information along with numeric pain intensity annotations (VAS/NRS), making them well-suited for our AU-based approach.

UNBC-SP is a widely used dataset of spontaneous pain expressions in adult patients with shoulder injuries. This archive comprises 200 video sequences totaling 48,398 annotated video frames from 25 patients with rotator-cuff injuries performing prescribed arm movements. Each sequence has a self-reported Visual Analogue Scale (VAS, 0–10) rating and an expert Observer Pain Index. Following Prkachin et al., VAS scores were discretized into four ordinal classes: no pain (0), mild (1–3), moderate (4–6), and severe (7–10) to align with clinical practice while mitigating class imbalance.

The MIntPain database contains 187,939 frames from 1,600 stimulus sweeps across 20 healthy participants who received graded electrical-muscle stimulation. Pain level is annotated on a five-point scale (0–4). Only RGB imagery is provided; consequently, all Action Units (AUs) were re-extracted (Section 3.2) to ensure a homogeneous feature space across datasets.

For each corpus, cases were stratified into a 70 % development set and a 30 % hold-out test set. Within the development fold, we executed a stratified three-fold cross-validation (CV): two folds for training and one for validation, cycling through the folds. Stratified splitting was adopted to preserve the class distribution in each fold, which is particularly important given the class imbalance present in pain datasets [30]. Three folds were chosen as a practical compromise between estimation variance and the limited number of sequences available per class. All model hyperparameter tuning was conducted within these inner CV folds using a nested procedure to prevent information leakage from validation data into hyperparameter selection [31]. The final hold-out set was kept separate and was evaluated exactly once for each trained model configuration, to obtain an unbiased performance estimate.

2.3.2 Preprocessing

2.3.2.1 Action-Unit extraction

Frame-wise AU intensity values were computed for each video frame using the Py-Feat toolbox [24] with a face detector and a ResNet-50 backbone for feature extraction. Detected facial landmarks were aligned using img2pose to normalize head pose. To compensate for varying lighting conditions and camera setups, each sequence's AU time series was standardized to a zero mean and unit variance per AU to preserve temporal dynamics while equalizing scale.

2.3.2.2 Time-series representation

We aggregated frame-level AU vectors into time-series representations for each subject video. Since video lengths varied widely across our dataset, we implemented sequence-length normalization to ensure consistent model input dimensions. Three strategies were evaluated: (1) truncation, which reduced all sequences to the shortest length, discarding excess frames from longer videos; (2) average-length adjustment, which set sequences to the dataset's mean length (rounded to the nearest integer) by truncating longer sequences and padding shorter ones with low-variance Gaussian-noise frames to reach the target length; and (3) padding to the longest length, which extended all sequences to the maximum observed length by adding small, zero-centered Gaussian-noise frames to shorter sequences, approximating near-neutral expressions and avoiding constant-zero padding.

After applying one of these normalization methods, each sequence was further standardized (per sequence) to zero mean and unit variance to address residual differences in overall intensity across subjects or recording conditions. We then addressed class imbalance in the training data by oversampling minority classes using the Synthetic Minority Over-sampling Technique (SMOTE) [32]. SMOTE has demonstrated high effectiveness in oversampling for various sequence-based models with imbalanced datasets [33]. In our implementation, each time-series sequence, originally represented as a tensor $x \in \mathbb{R}^{N\times T\times F}$ were processed, where N is the number of samples after oversampling, T is the number of timesteps, and F is the number of features. Each sequence X_i is flattened into a one-dimensional vector $x_i \in \mathbb{R}^{(T\cdot F)}$ SMOTE then generates synthetic samples by interpolating between a minority sample x_i and one of its nearest neighbors x_j using a random number between 0 and 1, represented as δ, according to equation 1. Finally, the synthetic vectors x_{syn} are reshaped back into the three-dimensional form (T, F) for subsequent analyses.

$$x_{syn} = x_i + \delta(x_j - x_i) \quad (1)$$

In addition to oversampling, we applied data augmentation to the AU time series to expand the effective training set and improve generalization. We generated augmented versions of each sequence using several standard time-series augmentation methods [34]: jittering, scaling, permutation, and time warping. These transformations have been used in prior time-series classification research to simulate variations while preserving class characteristics [34]. We also included a "no augmentation" case for baseline comparison. All augmented sequences inherit the original sequence's label. By training on both original and augmented samples, we aim to increase model robustness to variations in facial movement and potential measurement noise.

2.3.2.3 Tokenized AU Sequence Processing

Nie et al. [35] propose segmenting time series into patch "tokens" for a Vision Transformer-like model, markedly improving performance on long sequences [35]. Thus, we also experimented with a novel representation of the AU time series as tokenized "text" sequences, enabling the use of text-based classification techniques. In this approach, each video's AU sequence is converted into a "sequence" of textual tokens by discretizing the AU intensities and treating each frame as a "word". The conversion process involved several steps. We first applied a rolling average size 1 (no smoothing) and 3 frames to smooth intensity values before binarization. As a result, we produced a "sentence-like" structure, where each frame corresponds to a token describing which AUs are present (equation 2), where $\widetilde{AU}_t^{(i)}$ is the averaged intensity of AU i at frame t, and w is the window size. After smoothing, the class imbalance was again addressed by oversampling minority classes.

$$\widetilde{AU}_t^{(i)} = \frac{1}{w} \sum_{j=t-w+1}^{t} \widetilde{AU}_t^{(i)} \quad (2)$$

Converting graded AU outputs into categorical occurrence labels has been used in facial expression analysis [36]. In our study, we binarized AU intensities using relative thresholds on the normalized 0.00–1.00 Py-Feat scale, similar to how manual FACS coding distinguishes between absent and present AUs using discrete intensity levels. AU intensities were binarized using equation 3 with thresholds $\theta \in \{0.25, 0.50\}$, consistent with common AU practices and corresponding to "low-to-moderate" vs "moderate" activation zones.

$$AU_t^{(i)} = \begin{cases} 1 & \text{if } AU_t^{(i)} > \theta \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

Each binary vector of 11 AUs per frame is then converted into a symbolic representation by formatting each AU value into a binary string and concatenating them into a compact token for that frame. For example, a frame with AU01=1, AU06=0, AU09=1, and so on, is encoded as shown in equation 4, yielding a token such as "10100100011". Tokens from successive frames were joined with whitespace to form a textual sequence for the entire video. Before modeling, a tokenizer was applied, and SMOTE was implemented to address class imbalance.

$$AU\_Str_t = AU04_t + AU06_t + \cdots + AU43_t \quad (4)$$

2.3.3 Modeling

Our objective was to classify each video (the AU sequence) into the correct pain level. We pursued two complementary approaches: (1) sequence-based modeling on continuous AU time series, and (2) token-based modeling on discrete AU text sequences.

2.3.3.1 Continuous-sequence models

Six neural architectures were implemented in TensorFlow 2 to evaluate a range of sequence modeling strategies shown to be effective for time-series classification. Recurrent architectures were selected for their capacity to model sequential dependencies. A three-layer 1D CNN was included to capture local temporal patterns through convolutional filters; vanilla LSTMs and GRUs were included as established gated recurrent units designed to capture long-range temporal patterns; and a bidirectional LSTM was used to exploit both forward and backward context. A simple RNN served as a non-gated baseline, and a hybrid CNN + LSTM combined local feature extraction with sequential modeling. Hidden widths were constrained to 50–250 units, and recurrent depth to 2 layers, to limit overfitting. Networks were optimized with Adam under categorical cross-entropy. In parallel, traditional classifiers were trained on flattened sequences to provide a complementary set of non-sequential baselines. These included random forests (RF), extreme gradient boosting (XGB), support vector machines (SVM), decision trees (DT), Gaussian naïve Bayes (NB), k-nearest neighbors (KNN), and a multivariate classifier (MVC). A 30-trial random search was performed within each CV fold to select hyperparameters.

2.3.3.2 Token-sequence models

In this approach, three model families were explored to assess the suitability of different classification paradigms for the tokenized AU representation. (1) BERT, which is a pre-trained transformer-based language model that has achieved state-of-the-art results in numerous NLP sequence classification tasks, was fine-tuned for multi-class sequence classification; maximum token length equaled the longest training sentence. (2) Classical machine learning classifiers were trained on features extracted from the tokenized sequences to evaluate whether established algorithms can exploit the symbolic co-occurrence patterns. These included Multilayer perceptron (MLP), logistic regression (LR), Extra Trees (ET), DT, RF, NB, KNN, SVM, and XGB. Finally, we also experimented with a lightweight parameter-free method (3) called "Low-Resource," which is a parameter-free 5-NN classifier that employsormalized Compression Distance (NCD) as the similarity metric [37].

2.4 Evaluation

Performance was quantified by overall accuracy defined as the proportion of correctly classified samples: Accuracy is equal to (TP + TN) / (TP + TN + FP + FN), where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative counts, respectively. Accuracy was selected as the primary metric because SMOTE oversampling was applied to balance the class distribution in the training data, thereby reducing bias toward the majority classes that would otherwise make accuracy misleading in imbalanced data. Reported values correspond to the mean ± standard deviation across the three outer CV splits, where the standard deviation provides insight into the stability of model performance across different data partitions. Distributional assumptions were assessed using the Shapiro–Wilk test for normality of the model residuals and Levene's test for homogeneity of variances across groups. Where both assumptions were satisfied, a two-factor mixed ANOVA (alignment strategy × augmentation) was performed to analyze main effects and their interaction. Otherwise, the Kruskal–Wallis test followed by Dunn–Bonferroni post-hoc comparisons was applied. Significance was accepted at p-value < 0.05. Following significant omnibus tests, pairwise comparisons were conducted using Tukey's HSD test for parametric analyses and the Dunn–Bonferroni procedure for non-parametric analyses, both controlling the family-wise error rate. To counteract hyperparameter optimism, the best configuration from each inner CV loop was retrained on the entire development fold before a single evaluation on the holdout data.

3. Results and Discussion

In this section, we present the results of continuous-time series modeling and tokenized sequence modeling, and then discuss their implications. Examination was performed across the UNBC-SP and MIntPain facial pain datasets. The goal of this study was to identify which strategy, and which specific configuration within each strategy, produced the highest classification accuracy for pain detection. Statistical analysis involved mean accuracies from 3-fold cross-validation and, when relevant, subsequent ANOVA or post hoc comparisons. All reported p-values, effect sizes, and confidence intervals stem from these tests.

3.1 Continuous time-series modeling – UNBC-SP dataset

For the continuous-AU strategy, we evaluated three sequence alignment methods (Truncation, Averaging, and Padding) combined with five augmentation schemes (None, Jitter, Scaling, Time Warping, and Permutation). On the UNBC-SP dataset, results indicated that model performance varied significantly by augmentation ($F(4,42)=3.02$, $p=0.027$), although multiple pairwise comparisons did not detect consistently large differences, except for Permutation, which significantly reduced accuracy in several classifiers (e.g., SVM dropped from 79% to 51%). The alignment strategy also approached significance ($p=0.0585$), with truncation and averaging generally outperforming padding. Across all continuous models on the UNBC-SP dataset, the highest accuracy reached about 84% (RF under Averaging with no augmentation). Some deep learning architectures, such as Bi-directional LSTM, achieved around 69% but did not surpass the best classical models. Table 1 summarizes the mean accuracy results for the UNBC-SP dataset on continuous-model experiments under each alignment and augmentation condition.

3.2 Continuous time-series modeling - MIntPain dataset

In contrast, the continuous-model approach on the MIntPain dataset yielded higher peak performance (see Table 2). The top accuracy was approximately 91%, achieved by an SVM classifier with a truncation alignment and three augmentation approaches, including no augmentation, jittering, and scaling method, followed by an RF with a truncation alignment and two augmentation approaches, including no augmentation and jittering, and a GRU-based model with a truncation alignment and scaling augmentation method, that both approached 90%. Statistical tests indicated that alignment ($F(2,42)=6.88$, $p<0.005$) and augmentation ($F(4,42)=4.11$, $p=0.006$) significantly influenced performance on MIntPain. Post-hoc comparisons showed that disruptive augmentations, such as permutation, undermined results on datasets like the UNBC-SP, while less intrusive operations, such as jittering or minor Scaling, introduced no significant detriment compared to no augmentation.

Table 1. Mean accuracy % for continuous time-series modeling using the UNBC-SP dataset with various alignments and augmentation methods

	Truncate					Padding					Average				
	N	**JT**	**SC**	**PM**	**TW**	**N**	**JT**	**SC**	**PM**	**TW**	**N**	**JT**	**SC**	**PM**	**TW**
1DCNN	0.60	0.60	0.64	0.44	0.59	0.42	0.39	0.44	0.29	0.26	0.66	0.63	0.65	0.51	0.43
LSTM	0.44	0.42	0.45	0.30	0.29	0.25	0.25	0.25	0.34	0.25	0.21	0.23	0.28	0.23	0.25
RNN	0.61	0.60	0.61	0.40	0.62	0.31	0.60	0.35	0.25	0.27	0.54	0.51	0.51	0.40	0.52
GRU	0.51	0.51	0.51	0.40	0.54	0.33	0.27	0.36	0.38	0.28	0.38	0.43	0.46	0.29	0.49
Bi LSTM	0.69	0.69	0.65	0.45	0.63	0.59	0.58	0.52	0.31	0.38	0.65	0.67	0.68	0.50	0.60
CNN+LSTM	0.49	0.47	0.54	0.34	0.36	0.28	0.25	0.25	0.25	0.29	0.19	0.35	0.27	0.27	0.26
DT	0.56	0.50	0.47	0.36	0.47	0.63	0.64	0.56	0.47	0.58	0.61	0.57	0.59	0.36	0.59
RF	0.69	0.74	0.71	0.49	0.63	0.72	0.72	0.66	0.56	0.66	**0.84**	0.74	0.72	0.50	0.64
NB	0.60	0.60	0.60	0.40	0.29	0.49	0.49	0.50	0.35	0.48	0.58	0.58	0.57	0.40	0.46
KNN	0.57	0.57	0.56	0.39	0.58	0.70	0.70	0.69	0.52	0.62	0.66	0.66	0.67	0.47	0.55
SVM	**0.79**	**0.79**	0.78	0.51	0.68	0.80	**0.81**	0.79	0.52	0.58	0.83	0.83	0.83	0.52	0.59
XGB	0.66	0.67	0.68	0.44	0.58	0.75	0.74	0.73	0.53	0.69	0.74	0.74	0.75	0.51	0.68
Arima	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20

Table 2. Mean accuracy % for continuous time-series modeling using the MIntPain dataset with various alignments and augmentation methods

	Truncate					Padding					Average				
	N	**JT**	**SC**	**PM**	**TW**	**N**	**JT**	**SC**	**PM**	**TW**	**N**	**JT**	**SC**	**PM**	**TW**
1DCNN	0.84	0.85	0.84	0.51	0.73	0.86	0.85	0.85	0.48	0.33	0.85	0.84	0.84	0.51	0.54
LSTM	0.81	0.82	0.80	0.67	0.77	0.51	0.37	0.34	0.27	0.24	0.68	0.69	0.69	0.55	0.50
RNN	0.84	0.84	0.82	0.54	0.73	0.44	0.48	0.41	0.25	0.25	0.70	0.77	0.78	0.40	0.38
GRU	0.82	0.81	0.90	0.66	0.77	0.69	0.66	0.59	0.49	0.39	0.73	0.76	0.74	0.59	0.58
Bi LSTM	0.85	0.85	0.84	0.59	0.78	0.73	0.72	0.61	0.45	0.54	0.74	0.79	0.77	0.50	0.65
CNN+LSTM	0.84	0.83	0.84	0.66	0.71	0.22	0.23	0.23	0.23	0.22	0.70	0.68	0.68	0.51	0.46
MVC	0.33	0.31	0.29	0.28	0.28	0.33	0.32	0.27	0.25	0.24	0.31	0.31	0.29	0.25	0.23
DT	0.57	0.55	0.52	0.31	0.51	0.56	0.53	0.48	0.32	0.52	0.27	0.55	0.48	0.32	0.53
RF	0.90	0.90	0.89	0.52	0.81	0.89	0.88	0.82	0.53	0.78	0.83	0.89	0.86	0.50	0.79
NB	0.39	0.39	0.39	0.29	0.31	0.27	0.27	0.27	0.21	0.21	0.32	0.32	0.32	0.26	0.23
KNN	0.77	0.77	0.77	0.45	0.76	0.77	0.76	0.76	0.43	0.48	0.77	0.76	0.76	0.43	0.61
SVM	**0.91**	**0.91**	**0.91**	0.56	0.85	0.88	0.88	0.88	0.52	0.45	0.83	0.87	0.87	0.51	0.61
XGB	0.87	0.86	0.85	0.51	0.77	**0.88**	**0.88**	0.85	0.53	0.78	**0.86**	0.87	0.85	0.49	0.77
Arima	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20

3.3 Tokenized sequence modeling – UNBC-SP dataset

For the token-based approach, we focused on two key preprocessing parameters: the smoothing window (1-frame vs. 3-frame averaging) and the binarization threshold (e.g., 0.25 vs. 0.5). A two-way ANOVA revealed a statistically significant effect of smoothing ($F(1,12)=7.92$, $p<0.05$) such that a 3-frame average lowered accuracy for most classifiers. By contrast, the binarized threshold manipulations had little effect on overall performance ($F(1,12)=0.44$, $p=0.52$). Several machine learning models, especially ensemble-based methods such as KNN, ET, and RF, achieved high accuracy (up to 82%). In comparison, the recurrent deep networks and the BERT-based classifiers struggled with tokenized sequences, often performing at chance levels of 20 to 30 percent accuracy for the four-class task. Statistical tests supported a significant performance

gap between classical and deep models on tokenized UNBC-SP data ($p<0.001$), with classical methods outperforming the neural architectures by large margins.

3.4 Tokenized sequence modeling – MIntPain dataset

Tokenized results on the MIntPain dataset were qualitatively similar to those of the UNBC-SP. Sequence data with a 1-frame window yielded better accuracy than 3-frame smoothing, although the smoothing effect did not reach statistical significance (ANOVA, $F(1,12) = 0.36$, $p = 0.56$). The binarization threshold again had no notable impact ($p = 0.56$). The best-performing models on token data were classical ensemble methods, achieving accuracies around 82%, comparable to the UNBC-SP token results. Neural network models (LSTMs, BERT, etc.) generally did not surpass random guessing on tokenized sequences for MIntPain, suggesting difficulty learning from the discrete token representation with limited data. For example, ensemble methods such as KNN, ET, and RF significantly outperformed deep models on token data, such as the UNBC-SP dataset. Table 3 summarizes the token-based modeling accuracies across various settings. It shows the limited influence of threshold choices and the consistently stronger performance of classical algorithms over deep learning in this context.

Table 3. Mean accuracy % for tokenized sequence modeling with UNBC-SP and MIntPain dataset, modeling with each different average window (1-frame and 3-frame) and threshold (0.25, 0.5)

	UNBC-SP				**MIntPain**			
Rolling Average	**1-frame window**		**3-frame window**		**1-frame window**		**3-frame window**	
Threshold	**0.25**	**0.5**	**0.25**	**0.5**	**0.25**	**0.5**	**0.25**	**0.5**
RNN	0.43	0.43	0.43	0.2	0.5	0.5	0.5	0.5
LSTM	0.18	0.18	0.43	0.18	0.5	0.5	0.5	0.5
GRU	0.43	0.18	0.2	0.43	0.5	0.5	0.5	0.5
Bi-LSTM	0.4	0.33	0.3	0.43	0.5	0.5	0.5	0.5
BERT	0.43	0.43	0.43	0.43	0.5	0.5	0.5	0.5
MLP	0.74	0.74	0.68	0.67	0.75	0.72	0.69	0.65
LR	0.3	0.29	0.24	0.31	0.25	0.25	0.24	0.25
ExtraTree	**0.82**	0.81	0.71	0.71	**0.82**	**0.82**	0.73	0.69
DT	0.58	0.58	0.56	0.56	0.58	0.59	0.56	0.55
RF	0.79	0.77	0.69	0.69	0.79	0.8	0.71	0.67
NB	0.23	0.23	0.21	0.22	0.22	0.22	0.21	0.22
KNN	0.78	0.75	0.67	0.66	0.78	0.78	0.69	0.64
SVM	0.68	0.74	0.61	0.68	0.68	0.72	0.62	0.64
XGB	0.67	0.63	0.57	0.59	0.58	0.58	0.55	0.51
Low Resource	0.23	0.23	0.25	0.24	0.2	0.21	0.23	0.21

3.5 Comparison of Continuous vs. Token approaches

A direct comparison between the continuous and tokenized strategies highlights a few important points. On the MIntPain dataset, continuous-time series modeling achieved higher overall accuracy (peak ~91%) than the token-based approach (peak ~82%), and this difference was statistically significant ($p < 0.01$). In contrast, on the smaller UNBC-SP dataset, the top continuous and tokenized accuracies were very similar (~84% vs. 82%) and not significantly different ($p = 0.34$). Classical machine-learning algorithms emerged as strong contenders across both continuous and tokenized pipelines, whereas deep learning architectures were more effective on continuous data and generally less competitive in the tokenized condition. Post hoc Tukey tests confirmed that the permutation augmentation consistently reduced performance across models (in both strategies and datasets), and that smoothing frames before tokenization often lowered accuracy without providing benefits.

3.6 Discussion

The study results show that continuous-time series analysis combined with classical machine learning, and, in some settings, with deep learning, yields higher or at least comparable performance to the sentence tokenization approach on both datasets. Tokenization reduced accuracy across most models, especially in complex deep learning architectures. This pattern likely reflects the small sample size and data imbalance, which favor lower-complexity classifiers. Considering the results together, retaining continuous action-unit dynamics confers a clear advantage over the tokenized-sequence strategy. Notably, classical models such as random forests and ET often matched or surpassed deep learning, which, in some configurations, approached chance. This gap can be interpreted through several factors related to data representation and model capacity. One issue came from the data preprocessing step: the binarization step in the tokenization process removes the graded intensity information from each AU, causing models to lose access to the continuous intensity gradients they rely on for capturing temporal dynamics. The dataset sizes of approximately 200 sequences for UNBC-SP and approximately 1,600 sequences for MIntPain dataset are also the big issue that leading to insufficient for data classification using neural network architecture, especially, BERT, which was designed for large scale language modeling, showed accuracy results close to random chance when trained on these limited tokenized sequences, indicating that the model failed to converge on meaningful representations of the tokenized input. While binarization reduces information content for complex models, this simplification may flatten data patterns, helping tree-based and margin-based classifiers find useful decision boundaries. Models such as RF, ET, and SVM can effectively exploit co-occurrence patterns of binary AU activations without requiring full intensity resolution, which explains their relatively stronger performance on tokenized data. However, the result 82% accuracy with the sentence tokenization method and 91% with the continuous time series method supports the feasibility of using temporal dynamics data, such as facial action units, to assist in classifying facial expression characteristics in human pain, which is consistent with previous research [38] that emphasizes motion cues in pain recognition.

The statistical analysis shows that specific preprocessing choices materially affect outcomes. The truncation and average length alignment generally outperformed padding. The permutation augmentation technique reduced accuracy by disrupting the temporal order. Permutation augmentation strategies result in reduced accuracy because they disrupt temporal order. Simultaneously, although data tested with jittering and scaling augmentation methods may demonstrate higher accuracy in some datasets, statistical tests indicate that outcomes may show minimal or no differences compared to not implementing these methods. For the tokenization process, it was found that 3-frame window smoothing reduces accuracy, and the different thresholds used to binarize AU intensity data in this study do not significantly affect results. This supports the micro-expression principle that AU changes over short time intervals are crucial for pain differentiation, as smoothing these values removes important information from the dataset. Therefore, the results of this study indicate that preserving fine-scale temporal data details and avoiding disturbance or temporal-sequence transformation of data represent an appropriate method for analyzing facial pain from AU data in sequence format.

Finally, this study reflects both the support and limitations of pain classification from facial AU sequences. First, dataset size and quality remain primary factors determining model performance, as this study demonstrates that the MIntPain dataset, with its larger data volume, produces clearly improved results compared to the UNBC-SP dataset. Second, preserving the data sequence, avoiding unnecessary data reduction, and managing data balance are crucial for developing pain-detection systems from facial expressions. Finally, although CV evaluation includes controlled hyperparameter selection and statistical testing, external validation covering diverse data sources, equipment, and populations, along with calibration, will enhance readiness for clinical applications.

4. Conclusion

This study developed a novel dual-pathway system for facial pain recognition by combining continuous time-series analysis of action units with a tokenized sequence approach. Instead of focusing primarily on continuous time-series representations of AU data, this study applied a tokenized AU sequence

approach, treating frame-level binarized AUs as text-like tokens, for facial pain classification using NLP-based models. A systematic comparison between continuous and tokenized representation strategies across multiple classifier families, including traditional machine learning, deep learning, and low-resource methods, was introduced in this study. The classification architecture was tested on two datasets, UNBC-SP and MIntPain, and showed that continuous models yielded the best results, with accuracies of up to 91% on the SVM classifier using the MIntPain dataset and 84% on the RF classifier using the UNBC-SP dataset. Tokenization was less effective overall but still reached about 82% accuracy with classical models on the UNBC-SP dataset, suggesting that it may be useful in certain conditions. Augmentation methods, such as permutation, reduced accuracy, while smoothing did not improve performance for tokenized models. The results also indicate that preserving the temporal sequence of action units is important for accurate pain classification. The findings of this study contribute to the development of non-invasive pain assessment tools for patients in palliative care settings who have limited ability to self-report pain, such as those with dementia, laryngeal cancer, or end-stage illness. Continuous AU monitoring from video can capture micro-expressions that indicate pain at time scales difficult for human observers to detect, which could serve as a decision-support tool for healthcare providers in evaluating pain levels. Importantly, this study clarifies which data representations and model architectures can extract discriminative features from AU data and which cannot. These empirical insights can serve as a reference for future research on facial pain recognition, enabling researchers to select appropriate tools and methods more efficiently and accelerating progress toward practical clinical applications.

The main limitation of this work is the small size and restricted scope of available datasets. The models were tested only on controlled data, and their performance in real-world settings is still uncertain. There was also no independent verification of the pain labels, and only a limited set of preprocessing parameters was examined. Future work should focus on expanding datasets to include a wider range of patients and conditions, as well as testing in natural, uncontrolled environments. It will also be important to include expert review of labels, explore more advanced sequence models such as Transformers, and test how different frame rates affect accuracy. For practical use in clinics, further steps will be needed to address video quality, privacy, and usability. Finally, the most important challenge in pain analysis research for palliative care patients is developing systems capable of identifying background pain or non-stimulus-evoked pain, which would help overcome the communication barriers that currently limit pain assessment and improve quality of life in this patient group. With these improvements, automated facial pain recognition could become a reliable tool to support healthcare providers in monitoring and managing patient pain.

5. Acknowledgement

The author would like to express gratitude to Asst. Prof. Dr. Prompong Sugunnasil for his continued support and guidance throughout this study. The author is also grateful to the dataset owners who provided access to their resources and made this research possible. Finally, the author would like to acknowledge the earlier work of Atigorn Sanguansri, which provided an important inspiration for the direction of this study.

Author Contributions:

Conceptualization:	AP (Akkapop Prasompon), PS (Prompong Sugunnasil), AS (Atigorn Sanguansri)
Methodology:	AP and PS
Software:	AP
Validation:	AP and PS
Formal analysis:	AP
Investigation:	AP
Resources:	AP and PS
Data Curation:	AP
Writing - original draft:	AP
Writing - review & editing:	AP and PS
Visualization:	AP
Supervision:	PS
Project administration:	PS

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Miller, K. D.; Leticia, N.; Theresa, D. Cancer Treatment and Survivorship Statistics, 2022. *CA Cancer J. Clin.* **2022**, *5*, 409–436. https://doi.org/10.3322/caac.21731

[2] Parikh, R. B.; Kirch, R. A.; Thomas, J. S. Early Specialty Palliative Care—Translating Data in Oncology into Practice. *N. Engl. J. Med.* **2013**, *24*, 2347–2351. https://doi.org/10.1056/NEJMsb1305469

[3] Laura, A. P. M. Why and How to Integrate Early Palliative Care into Cutting-Edge Personalized Cancer Care. *J. Clin. Oncol. Educ. Book* **2024**.

[4] World Health Organization. *Palliative Care*; World Health Organization: Geneva, Switzerland, 2023; https://www.who.int/europe/news-room/fact-sheets/item/palliative-care (accessed June 1, 2023).

[5] Clark, D.; Nicole, B.; Clelland, E. G. Mapping Levels of Palliative Care Development in 198 Countries: The Situation in 2017. *J. Pain Symptom Manage.* **2020**, 794–807. https://doi.org/10.1016/j.jpainsymman.2019.11.009

[6] Dowell, D.; Ragan, K.; Jones, C. CDC Clinical Practice Guideline for Prescribing Opioids for Pain—United States, 2022. *MMWR Recomm. Rep.* **2022**. https://doi.org/10.15585/mmwr.rr7103a1

[7] Cohen, B.; Leigh, J. R.; Charles, V. P. *Opioid Analgesics*; StatPearls Publishing: Treasure Island, FL, USA, 2017.

[8] De, S.; Gioacchino, D. Using AI to Detect Pain through Facial Expressions: A Review. *Bioengineering* **2023**, 548. https://doi.org/10.3390/bioengineering10050548

[9] Ekman, P.; Friesen, W. V. Facial Action Coding System. *Environ. Psychol. Nonverbal Behav.* **1978**. https://doi.org/10.1037/t27734-000

[10] Safikhani, S.; Gries, K. S.; Trudeau, J. J. Response Scale Selection in Adult Pain Measures: Results from a Literature Review. *J. Patient-Rep. Outcomes* **2018**, *2*, 1–9. https://doi.org/10.1186/s41687-018-0053-6

[11] Fang, R.; Hosseini, E.; Zhang, R. Survey on Pain Detection Using Machine Learning Models: Narrative Review. *JMIR AI* **2025**, *4*, e53026. https://doi.org/10.2196/53026

[12] Wen, C. T.; Du, T.; Teo, J. C. Automated Pain Detection Using Facial Expression in Adult Patients with a Customized Spatial Temporal Attention Long Short-Term Memory (STA-LSTM) Network. *Sci. Rep.* **2025**, *15*, 13429. https://doi.org/10.1038/s41598-025-97885-5

[13] Chen, Z.; Ansari, R.; Wilkie, D. J. Automated Detection of Pain from Facial Expressions: A Rule-Based Approach Using AAM. In *Proceedings of SPIE*; 2012; Vol. 8314, 83143O. https://doi.org/10.1117/12.912537

[14] Takalkar, M. A.; Min, X. Image-Based Facial Micro-Expression Recognition Using Deep Learning on Small Datasets. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*; 2017; pp 1–7. https://doi.org/10.1109/DICTA.2017.8227443

[15] Hassan, T.; Seus, D.; Wollenberg, J.; Weitz, K.; Kunz, M.; Lautenbacher, S.; Garbas, J.-U.; Schmid, U. Automatic Detection of Pain from Facial Expression: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1815–1831. https://doi.org/10.1109/TPAMI.2019.2958341

[16] Lautenbacher, S.; Hassan, T.; Seuss, D. Automatic Coding of Facial Expressions of Pain: Are We There Yet? *Pain Res. Manag.* **2022**, *2022*. https://doi.org/10.1155/2022/6635496

[17] Chongwen, W.; Wang, Z. Progressive Multi-Scale Vision Transformer for Facial Action Unit Detection. *Front. Neurorobot.* **2022**, *15*. https://doi.org/10.3389/fnbot.2021.824592

[18] Pouromran, F.; Lin, Y.; Kamarthi, S. Personalized Deep Bi-LSTM RNN Based Model for Pain Intensity Classification Using EDA Signal. *Sensors* **2022**, *21*, 8087. https://doi.org/10.3390/s22218087

[19] Chen, Z.; Ansari, R.; Wilkie, D. J. Learning Pain from Action Unit Combinations: A Weakly Supervised Approach via Multiple Instance Learning. *IEEE Trans. Affect. Comput.* **2022**, *31*, 135–146. https://doi.org/10.1109/TAFFC.2019.2949314

[20] Tran, M.; Siniukov, M.; Jin, Z.; Soleymani, M. Discrete Facial Encoding: A Framework for Data-Driven Facial Display Discovery. *arXiv* **2025**, arXiv:2510.01662

[21] Devlin, J.; Chang, M.-W.; Lee, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*; 2019; pp 4171–4186.

[22] Lucey, P.; Cohn, J. F.; Prkachin, K. M. Painful Data: The UNBC-McMaster Shoulder Pain Expression Archive Database. In *Proceedings of the IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*; 2011; pp 57–64. https://doi.org/10.1109/FG.2011.5771462

[23] Haque, M. A.; Bautista, R. B.; Noroozi, F. Deep Multimodal Pain Recognition: A Database and Comparison of Spatio-Temporal Visual Modalities. In *Proceedings of the IEEE Int. Conf. Autom. Face Gesture Recognit. (FG 2018)*; 2018; pp 250–257. https://doi.org/10.1109/FG.2018.00044

[24] Cheong, J. H.; Jolly, E.; Xie, T. Py-Feat: Python Facial Expression Analysis Toolbox. *Affect. Sci.* **2023**, 781–796. https://doi.org/10.1007/s42761-023-00191-4

[25] Prkachin, K. M. The Consistency of Facial Expressions of Pain: A Comparison across Modalities. *Pain* **1992**, *51*, 297–306. https://doi.org/10.1016/0304-3959(92)90213-U

[26] Kunz, M.; Meixner, D.; Lautenbacher, S. Facial Muscle Movements Encoding Pain—A Systematic Review. *Pain* **2019**, *160*, 535–549. https://doi.org/10.1097/j.pain.0000000000001424

[27] Dildine, T.; Atlas, L. The Need for Diversity in Research on Facial Expressions of Pain. *Pain* **2019**, *160*, 1901–1902. https://doi.org/10.1097/j.pain.0000000000001593

[28] Atee, M.; Hoti, K.; Chivers, P.; Hughes, J. D. Faces of Pain in Dementia: Learnings from a Real-World Study Using a Technology-Enabled Pain Assessment Tool. *Front. Pain Res.* **2022**, *3*. https://doi.org/10.3389/fpain.2022.827551

[29] Boonstra, A. M.; Stewart, R. E.; Köke, A. J. Cut-Off Points for Mild, Moderate, and Severe Pain on the Numeric Rating Scale for Pain in Patients with Chronic Musculoskeletal Pain: Variability and Influence of Sex and Catastrophizing. *Front. Psychol.* **2016**, *7*, 1466. https://doi.org/10.3389/fpsyg.2016.01466

[30] Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the Int. Joint Conf. Artif. Intell. (IJCAI)*; 1995; pp 1137–1143.

[31] Cawley, G. C. On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.

[32] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*. https://doi.org/10.1613/jair.953

[33] Mujahid, M.; Kına, E.; Rustam, F.; et al. Data Oversampling and Imbalanced Datasets: An Investigation of Performance for Machine Learning and Feature Engineering. *J. Big Data* **2024**, *11*, 87. https://doi.org/10.1186/s40537-024-00943-4

[34] Brian, K. I.; Seiichi, U. Time Series Data Augmentation. In *Proceedings of the Int. Conf. Pattern Recognit. (ICPR)*; 2020.

[35] Nie, Y.; Nguyen, N. H.; Sinthong, P. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. *arXiv* **2023**, arXiv:2211.14730.

[36] Girard, J. M.; Cohn, J. F.; Torre, F. D. L. Estimating Smile Intensity: A Better Way. *Pattern Recognit. Lett.* **2015**, *16*, 12–21. https://doi.org/10.1016/j.patrec.2014.10.004

[37] Jiang, Z.; Yang, M.; Tsirlin, M.; Tang, R.; Dai, Y.; Lin, J. "Low-Resource" Text Classification: A Parameter-Free Classification Method with Compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*; 2023; pp 6810–6828. https://doi.org/10.18653/v1/2023.findings-acl.426

[38] Broomé, S.; Gleerup, K. B.; Andersen, P. H. Dynamics Are Important for the Recognition of Equine Pain in Video. In *Proceedings of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*; 2019; pp 12667–12676. https://doi.org/10.1109/CVPR.2019.01295