

UTRECHT UNIVERSITY

THESIS FOR MSc GAME AND MEDIA TECHNOLOGY
INFOMGMT

Realistic Painful Expression Synthesis Using Generative Models

Authors:

Yuyu Chen

Student Number: 9752129

y.chen24@students.uu.nl

Supervisor:

Assist. Prof. Dr. I.Önal Ertuğrul

i.onalertugrul@uu.nl

Second examiner:

Assist. Prof. Dr. Z.Yumak

z.yumak@uu.nl

July 21, 2023



Utrecht
University

Abstract

Accurate pain assessment is crucial to understand the severity of a patient’s condition and for developing appropriate treatment. In recent years, a new research trend has focused on Automatic Pain Assessment (APA) through facial expressions. Compared to human observation, it provides continuous monitoring and a more objective assessment. Moreover, it also holds potential benefits for patients with severe cognitive or communication impairments. So far, many learning-based approaches have been proposed through facial expressions. However, these methods often did not generalize well to real-world conditions, primarily due to the lack of diverse pain-related facial expression data.

Traditional data augmentation methods have been used to expand the facial painful expression dataset, but they only provide limited diversity. Generative Adversarial Networks (GANs) is a promising technique. This project mainly focuses on the application of GANs to generate synthetic but realistic facial pain expressions, aimed at effectively augmenting the existing dataset. We have selected the UNBC-McMaster dataset for augmentation, a database widely recognized in the APA field. Our analysis of this dataset identified a number of issues, including insufficient data volume, imbalanced distribution of data labels and the range of head pose.

To overcome these challenges, we propose a novel method. First, we implement GANimation, a network that enables us control over the activation magnitude of each Action Units (AUs), allowing for their combination into desired expressions. We then fine-tune this network with the UNBC-McMaster dataset, promoting within-domain generation to reduce domain variance. Additionally, a 3D registration technique is applied throughout both the training and testing phases to counter the issue of head pose range problems in the original dataset. Finally, We propose an innovative scoring mechanism that allows us to generate high-quality and diverse pain expressions through the fine-tuned GANimation network. With the proposed method, we can effectively create balanced and sufficiently diverse pain datasets.

In the experiment evaluation, we first assess the quality of the generated images using both quantitative and qualitative evaluation methods. The results demonstrate a significant improvement in the quality of the fine-tuned GANimation network compared to the images generated without fine-tuning. Then, we employ the Fréchet Inception Distance (FID) metric to evaluate the overall quality of the augmented dataset. The effectiveness of the proposed 3D registration technique is validated here. Outcomes show that the synthetic dataset generated based on the scoring mechanism is closely to the original, especially when using 3D-registered images as inputs. Lastly, we train the synthetic dataset for classification using existing Convolutional Neural Networks (CNN). The results indicate that the synthetic dataset has the potential ability to improve current APA classification algorithms.

Contents

1	Introduction	4
2	Related work	6
2.1	Overview of Automatic Pain Assessment	6
2.2	Painful Data of Facial Expressions	12
2.3	Data Augmentation	14
2.4	Generative Adversarial Network (GANs)	17
3	Selected Dataset Overview and Analysis	29
3.1	Dataset selection: UNBC-McMaster dataset	29
3.2	Dataset analysis	31
4	Methodology	33
4.1	Manipulating AU through GANimation	33
4.1.1	Training Phase	33
4.1.2	Testing Phase and Result Analysis	35
4.1.3	Finetuning	37
4.1.4	Generated Result after Finetuning	39
4.2	AU Combinations for Pain Expression Generation	40
4.2.1	Components of Pain Formula	40
4.2.2	Generated Result Analysis	42
4.2.3	Scoring Mechanism	43
4.3	Painful Dataset Augmentation	47
4.4	Painful Dataset Evaluation	49
5	Experiment Results	51
5.1	Image Quality Assessment	51
5.2	GAN-based Network Measurement	52
5.3	Performance Evaluation of the Trained Classifier	53
6	Discussion and Conclusion	57
6.1	Review of Research Questions	57
6.2	Limitations	59
6.2.1	Label Discrepancy	59
6.2.2	Computational Limitations	60
6.2.3	Test Dataset Volume	60
6.3	Future work	61
6.4	Conclusion	62

1 Introduction

How much does it hurt? In the medical field, pain assessment is not only critical for characterizing pain and identifying underlying mechanisms but can also guide the decision-making process regarding pharmacological treatment [1]. Accordingly, it is believed that a valid and reliable assessment of pain is essential for both clinical trials and effective pain management [2]. Unfortunately, pain is not always adequately assessed and managed, due to its complex and subjective nature [3] and the lack of accurate physiological or clinical signs for objective pain measurement [4].

Currently, the gold standard of pain assessment for both the presence and intensity is self-report of patients, in line with the clinical definition of pain, which states, "Pain is whatever the experiencing person says it is, existing whenever he/she says it does" [5]. However, this method isn't universally applicable. Infants or patients with severe cognitive or communication limitations may not be able to give a self-report of their pain verbally, in writing, or by other means such as a finger span [6] or eye blinking to indicate yes or no responses [7]. Several studies have already shown that these patients often receive less pain medication compared to those who can verbally express their pain [8]. In this context, it becomes crucial to assess pain through behavioural indicators such as facial expressions, vocalizations, and body movements [9]–[11]. These behavioural indicators can provide critical insights into pain, which forms the basis of **Automated Pain Assessment (APA)**.

Among all behavioural indicators, facial expression is considered to be the most prominent and salient nonverbal pain behaviour [12]. Since researchers have already demonstrated that facial responses to pain have very promising diagnostic validity [13], there have been significant efforts towards identifying reliable and valid facial indicators of pain. As a result, increasing research attention is being focused on the development of **APA based on facial expressions**. In particular, advanced information technologies such as machine learning, computer vision has been used to this field recently [14]. For instance, the latest research presents a novel enhanced deep neural network framework specifically designed for the effective detection of pain intensity [15]. These approaches offer the ability of continuous pain monitoring, compared to the traditional assessments performed by human observers, which has the potential to prevent delayed treatment due to missing severe pain events. Additionally, it may provide a more objective assessment compared to the human observer, whose judgment could be biased by personal factors such as their relationship with the patient [16] or even the patient's physical appearance [17].

Nevertheless, a significant limitation of these approaches is a lack of painful data. Acquiring appropriate data is challenging, especially for populations like infants, critically ill patients, elderly or cognitively impaired patients due to ethical concerns [18]. Yet, Facebook [19] and Google [20] have already demon-

strated the importance of large-scale datasets in developing high-quality models. Learning-based approaches rely heavily on large and complex training sets to generalize well in unconstrained environments.

To overcome this challenge, we aim to use **data augmentation** schemes [21], effectively increasing the amount and diversity of data. Typically, data augmentation consists of simple modifications to the dataset images, such as panning, rotating, flipping, and scaling. But the diversity obtained from those typical methods is limited [22], leading to the need for synthetic data examples [23]. In this project, we aim to generate synthetic facial expression data of pain to enrich the dataset effectively.

Generative adversarial networks (GAN) are one of the promising techniques for the synthesis of images [24], including the generation of high-quality realistic natural facial expressions [25]–[27]. Here, we focus on using the GAN framework to synthesise high quality painful expressions for data augmentation. Therefore, the following research questions and corresponding sub-questions are defined for this research project:

- **Primary Research Question**

How can the application of GANs be utilized to generate synthetic but realistic facial pain expressions, with the goal of effectively augmenting the existing dataset?

Based on this research question and the previous research analysis, the three sub-questions are as follows.

- **Sub-research Question I** Can the dataset augmentation scheme generate realistic or high-quality synthetic pain expressions?
- **Sub-research Question II**
How does applying 2D and 3D face registration before pain expression generation impact the performance of the model?
- **Sub-research Question III**
What is the potential impact of the synthetic pain expression dataset on the performance of existing APA approaches?

In the following sections, we will further develop these problems and explore possible solutions. The remaining sections of this paper are organized as follows: Section 2 gives an overview of related work. Next, section 3 describe the selected painful dataset in this project. The details of our unique approach are then articulated in Section 4, where we elaborate on the specifics of our proposed methodology. Finally, Section 5 and 6 provides an evaluation of the results and a discussion of the proposed solution, respectively.

2 Related work

In this section, we provide some basic knowledge of this project. First, we give an overview of the APA method, which focus on the current learning-based approaches. This is followed by an in-depth exploration and analysis of pain-induced facial expressions. Besides, we describe data augmentation, the existing solutions to current challenges. Finally, we emphasise the potential of Generative Adversarial Networks (GANs) as promising tools to address these challenges, laying the groundwork for the methodological advances introduced by this project.

2.1 Overview of Automatic Pain Assessment

Pain Assessment differs from pain recognition or detection, which focuses on identifying the presence or absence of pain. It is primarily quantifies the intensity of pain, answering the question [28], how much does it hurt? Clinically speaking, pain intensity is defined as the magnitude of experienced pain. There are usually two methods to measure pain intensity:

- **Self-report.**

Self-report is commonly considered as the standard for pain assessment due to the subjective nature of pain. It is considered patient-centred and offers retrospective accounts of events, experiences, and behaviour. It is also a convenient and cost-effective method of assessment [16]. The most common strategies are verbal rating scales (VRSs), numerical rating scales (NRSs), visual analog scales (VASs), and graphical scales, see Fig.1.

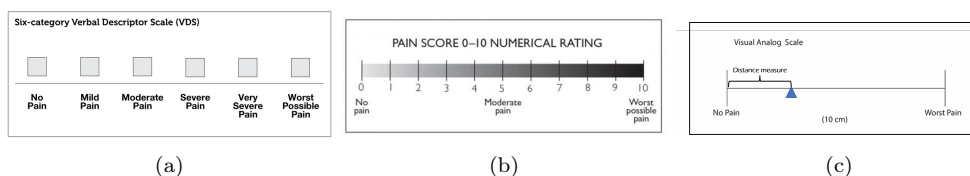


Figure 1: Common strategies of self-report [29]: (a) An example of verbal rating scales. (b) An example of numerical rating scales. (c) An example of visual analog scales.

Several clinical studies have verified the accuracy and reliability of self-reported assessment, advocating that they should be used whenever possible [30]. However, it is important to note that these methods require cognitive abilities and functioning from the individual, and this may not always be the case. Additionally, self-report may be influenced by factors such as reporting bias, memory discrepancies, and variations in verbal aptitude [31] as they are a controlled and goal-oriented response.

- **Observational Scales.**

The observation scale is typically employed when the patient displays severe cognitive impairment or has difficulties with communication. The American Geriatrics Society Panel on Persistent Pain in Older Persons published six common pain indicators: facial expressions, verbalizations or vocalizations, body movements, changes in interpersonal interactions, changes in activity patterns or routines, and mental status changes [32].

However, this method presents significant challenges and is susceptible to subjective bias and observer errors in judgment [33]. Furthermore, it is not possible for human caregivers to provide constant monitoring of patients, which may result in missed instances of pain or delayed detection of changes by the human observer.

Indeed, both self-reported and observational scales have limitations that may lead to inadequate pain management, particularly for critically ill patients where such shortcomings can be life-threatening [34].

To address these limitations, **Automated Pain Assessment (APA)** has been proposed as an attractive alternative to traditional pain assessment methods with the goal of reducing both reporting bias from patients and observing bias from physicians. Besides, this approach enables continuous, automatic, and real-time pain assessments, enabling prompt responses by physicians to improve the overall patient experience.

A typical process for APA, as illustrated in Fig.2, typically involves the following steps:

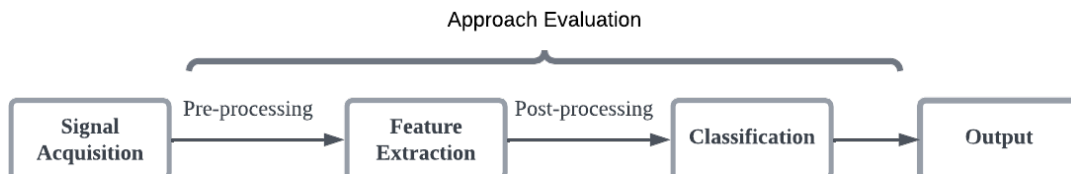


Figure 2: General steps involved in automatic pain assessment

Specifically, the process involves the collection of signals related to pain, followed by pre-processing such as data cleaning, and feature extraction. Afterwards, the acquired signals are evaluated by different recognition approaches, such as pattern matching or machine learning algorithms, to produce the final pain assessment results. To date, numerous studies have explored the use of various modalities, either singularly or in combination, for signal acquisition in automated pain assessment.

Generally, it can be broadly divided into signals based on physiological responses and behavioural responses. In specific, physiological responses to pain stem from neural interactions [35] that lead to changes in various physiological signals [36], such as heart rate variability [37], skin conductance or electro-dermal activity [38], electromyography [39], electroencephalography [40], and functional

magnetic resonance imaging (fMRI) [41]. On the other hand, behavioural responses involve protective and communicative actions [42], such as vocalisations [9], body movements [11] and facial expressions [43].

Compared to signals based on physiological responses, behaviour-based signals can be recorded in a contactless and non-intrusive manner. Additionally, some physiological response-based signals, such as those based on brain activation, are often limited to experimental pain conditions and are both costly to obtain and require extensive preparation, particularly for EEG recordings. Given these considerations, behavioral responses are often the preferred choice for signal acquisition in many studies. In the context of this research project, we specifically focus on facial expressions as the acquired signal.

So far, only a few studies have focused on APA compared with automatic pain recognition or detection. Here are some representative works in recent years.

- **AAM/SVM system**

This approach [44] was proposed in 2012 by the same authors who published the database. It is called AAM/SVM system. The Active Appearance Model (AAM) was used to track the face and extract visual features. Support vector machines (SVMs) were then used to classify individual action units as well as pain, see Fig.3.

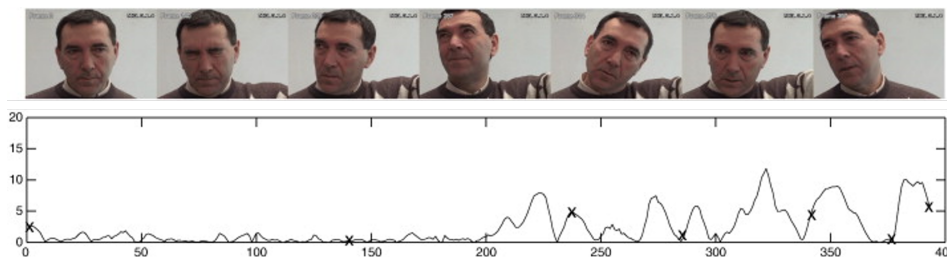


Figure 3: Examples output of AAM/SVM system [44]

In the experiment part, it used the area under the receiver-operator characteristic (ROC) curve (AUC) as the performance measure. The best overall performance of 81.8 is achieved by fusing several features.

- **Hidden Markov Model (HMM) learning**

Wu et al. [45] formulate expression recognition as a multi-instance learning problem. A discriminative multi-instance Hidden Markov Model (HMM) learning algorithm is proposed, where the HMM is used to capture the dynamics within instances.

In the experiment part, the leave-one-subject-out cross-validation method is implemented. Furthermore, the ground truth here is the observed pain intensity (OPI), which ranges from 0 (no pain observed) to 5 (extreme pain observed). The experimental results show that the algorithm achieves an

accuracy of 85.23% and an F_1 -score of 0.78, which also proves the effectiveness of the algorithm.

- **Recurrent Convolutional Neural Network Regression**

Zhou et al. [46], a Recurrent Convolutional Neural Network (RCNN) based real-time regression framework is proposed. See Fig.4 as the resulting output.

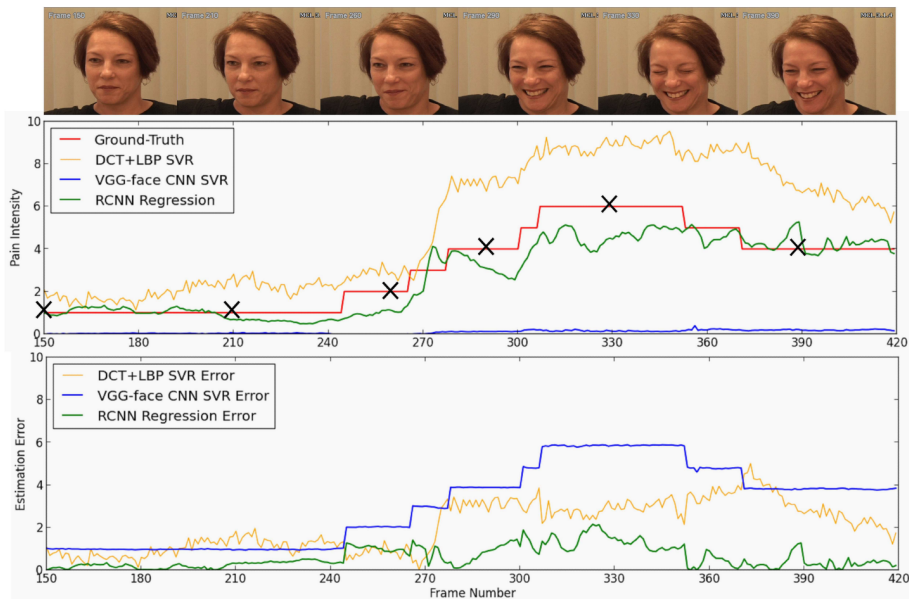


Figure 4: Examples output of RCNN regression framework [46]

The green line is the result of the proposed method. As we can see, the output is stable and smooth, and can also avoid unstable jumps or peaks among frames.

In the quantitative analysis phase, **25-fold cross-validation** was used to assess the method. The average Mean Squared Error (MSE) and Pearson Product-moment Correlation Coefficient (PCC) were applied as the performance metrics. And promising results of the average MSE and PCC of 1.54 and 0.65 were got respectively. Furthermore, the approach can be performed in real time and has a high computational efficiency.

- **A Joint Deep Neural Network Model**

Bargshady et al. [47] proposed another method called joint deep network. It uses a hybrid method to solve the classification problem of the pain data in four classes (no pain, weak pain, mid pain, and strong pain). Specifically, the method uses two different recurrent neural networks (RNNs). Both of them were pre-trained in a Visual Geometric Group Face Convolutional Neural Network (VGGFace CNN) and connected together as a network to estimate pain intensity levels. Here, the VGGFace model is designed for

face recognition and uses millions of face images for training. The whole process is given in Fig.5

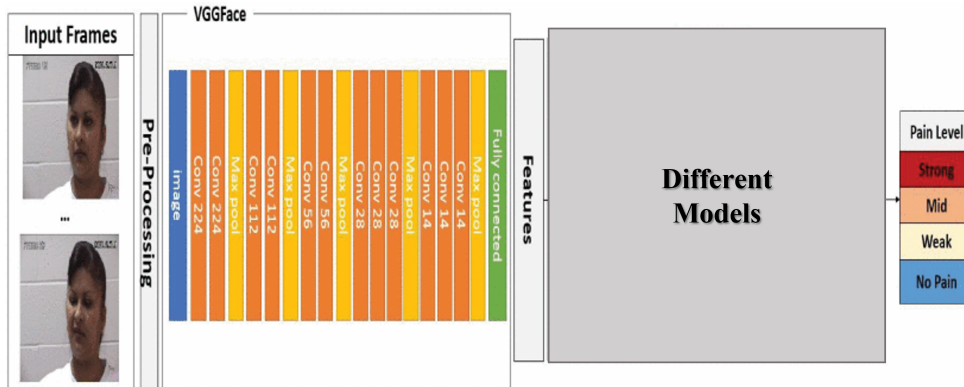


Figure 5: The architecture network model including VGGFace [47]

As we can see, firstly VGGFace is finetuned and used for feature extraction. Then, in the experiment part, the performance of the various models (Convolutional Neural Network (CNN), Deep Convolutional Networks (DCNs)) in the pre-training scenario are compared and evaluated against each other.

The experiment metrics are the accuracy and the area under the Receiver-Operator Characteristic curve (AUC). Besides, leave-one-out cross-validation was applied. The results are shown in the table.

Model	Accuracy	AUC
CNN	54.45%	44.8%
VGGFace+CNN	57.3%	52%
VGGFace +DNN1 + DNN2	73%	58.5%
Bargshady et al [47]	75.2%	82.7%

Table 1: Accuracy and AUC obtained in [47]

The final result of a joint deep neural network model is 75.2% accuracy and 82.7% AUC.

- **Ensemble Deep Learning Model**

In 2020, Bargshady et al. [48] proposed an ensemble deep learning approach. The resulting Ensemble Deep Learning Model (EDLM) integrates three streams of independent CNN-RNN-based networks. In the meantime, features were extracted from the facial images using a fine-tuned VGGFace algorithm combined with a Principal Component Analysis (PCA) method.

To enable a rigorous evaluation of the proposed EDLM model, various performance measures are used, including the Classification Mean Absolute Error (MAE), Mean Squared Error (MSE), Accuracy, Area under the ROC Curve (AUC) and F-score were utilized. And **10-fold cross-validation** was used for training and evaluation. The average performance of the proposed model is in table 2.

MSE	MAE	Accuracy	AUC
0.081	0.103	86%	90.5%

Table 2: The average performance of EDLM model [48]

The EDLM model shows good performance, including the feature extraction part.

The above representation is arranged in the order of publication time. We can see that recent research has demonstrated a significant interest in using various learning-based approaches, particularly deep learning, for APA. However, even the latest methods [48] only achieve an accuracy rate of 86%, which is not exactly an ideal result and there is still room for improvement.

2.2 Painful Data of Facial Expressions

We first provide a specific description as well as a detailed analysis of the acquired signal - the painful expressions.

Facial expressions have been widely recognized as a reliable indicator of pain intensity in behaviour-based signals. A change in facial expression can be seen as a change in pain intensity [49]. Several studies have already encoded pain as a series of action units (AU) based on the Facial Action Coding System (FACS). The Facial Action Coding System (FACS) [50] was originally developed by Ekman and Friesen. It is an anatomically based classification system for facial movements that examines the changes in shape and appearance produced by the facial muscles. Each muscle movement is considered to be an action unit (AU), see Fig.6 as an example.

Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46

Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28

Figure 6: Action Units in the Facial Action Coding System (FACS) [51]

FACS has 44 AUs in total which can be used to measure emotional stimuli. In other words, using FACS, human coders can manually encode almost any anatomically possible facial expression, deconstructing it into specific action units (AU) and the time periods during which they produce it.

As for the pain, it was relatively consistent across a range of clinical pain conditions and experimental pain patterns [52]. So, pain expression is widely characterised by the activation of a small group of facial muscles and can be encoded by a set of corresponding action units (AUs): brow lowering (AU 4), orbital tightening (AU 6 and AU 7), levator labii raise (AU 9 and AU 10) and eye closure (AU 43) [53], see Fig. 7.



Figure 7: Examples of facial expressions associated with pain [54]

Then, for the purpose of pain assessment, the Prkachin and Solomon Pain Intensity (PSPI) metric [55] has been widely used in current research. This is a well-validated method of measuring pain based on the Facial Action Coding System (FACS). In PSPI scale, each of the action units are measured on a six-point ordinal scale (0 = absent, 5 = maximum). The **pain formula** is defined as:

$$\text{Pain} = \text{AU4} + \max(\text{AU6}||\text{AU7}) + \max(\text{AU9}||\text{AU10}) + \text{AU43} \quad (1)$$

Here, AU43 can be either 0 or 1, while the other Action Units (AU4, AU6, AU7, AU9, and AU10) can have a maximum of 5 intensity levels. The resulting pain scale is composed of $3 * 5 + 1 = 16$ levels.

Until now, many studies of APA based on facial expressions have been proposed on predicting Prkachin and Solomon's Pain Intensity Index (PSPI), which measures pain as a linear combination of facial action units.

2.3 Data Augmentation

As we mentioned in 2.1, the state-of-the-art method of APA still faces significant challenges. One of the main problems is that these learning-based models, particularly deep learning models, are typically driven by the large size of the networks reaching millions of parameters [56]. Since these models are already inherently complex, several efforts to improve the generalization performance of these models have resulted in the development of increasingly intricate architectures [21], such as AlexNet [22] to VGG-16 [57], ResNet [58], Inception-V3 [59], and DenseNet [60]. For instance, the well-known Convolutional Neural Network (CNN) architecture, VGG16, consists of 16 layers of neurons and contains a total of 138 million parameters. These kinds of large networks are heavily reliant on large amounts of high-quality data for training to avoid overfitting in order to ensure good generalization. Therefore, it is clear that **data serve as the foundation for the development of a learning-based approach for APA.**

As we mentioned in 2.2, a major challenge in moving forward with APA is the difficulty of collecting and annotating data. The usual data acquisition process is to record videos of cognitively healthy individuals by experimentally inducing acute pain and other distressing states in a controlled laboratory setting.

So far, several datasets are available for automated pain assessment. Here, some publicly available datasets are listed for details in Table 3.

Database	Subjects	Sample Size
Infant COPE [61]	26 neonates	204 facial images
UNBC-McMaster [62]	25 shoulder pain patients	200 image sequences
BioVid [63]	90 healthy adults	8700 videos
Hi4D-ADSIP [64]	80 healthy adults	3360 3D sequences
BP4D-Spontaneous [65]	41 healthy adults	328 3D videos

Table 3: Public painful dataset

Different pain induction methods were used to create the different painful datasets. For instance, the BioVid dataset employs heat stimuli under lab conditions to induce acute pain, while the BP4D-Spontaneous dataset utilizes cold stimuli. Besides, there are different annotation methods to describe facial expressions and the pain or emotion experienced. Common methods include self-report, observer scale, stimulus type, stimulus level, and AU-based scores. Some datasets, such as **UNBC-McMaster** offer multiple forms of annotation. All these databases have significantly contributed to advancements in APA.

However, compared to large datasets in other fields such as ImageNet [66] with over 14 million images, or Facebook’s DeepFace system [67] trained on a dataset of 4 million facial images, the above-mentioned datasets are much smaller, often consisting of thousands rather than millions of samples. This can

be due to a number of reasons. First, collecting and annotating data related to pain expression is challenging. Pain, particularly chronic pain, is a complex and subjective phenomenon that is difficult to accurately capture and label. Privacy and ethical considerations also limit the amount of data that can be collected. In the context of APA, Despite its potential for measuring pain intensity, FACS is time-consuming and subjective. First, it requires manual coding by trained experts who need to take over a hundred hours of training to become proficient [68]. In fact, well-trained FACS specialists need about two hours to annotate a one-minute video. It is also a subjective way and therefore prone to bias. As a result, providing manually FACS annotated frames in the databases is difficult in the APA field. Therefore, the dataset currently available for APA can be considered small.

Although functional solutions such as dropout regularisation, batch normalisation, migration learning and pre-training have been developed to try to extend deep learning to applications on smaller datasets, **data augmentation** still addresses overfitting from the root of the problem [21].

Data augmentation is a frequent technique used in deep learning to increase the amount of data by generating new data from existing data, making the dataset more diverse. It can be seen as a regularisation technique used to reduce the generalisation error of the model [69]. It is usually classified into two main categories, **basic image manipulations** (such as flipping, transposing, and colour space manipulations) and **deep learning approaches** (for example, on GANs) [70].

As for the first categories, it typically consists of simple modifications to the dataset images such as translation, rotation, flip and scale. See Fig.8 as an example.

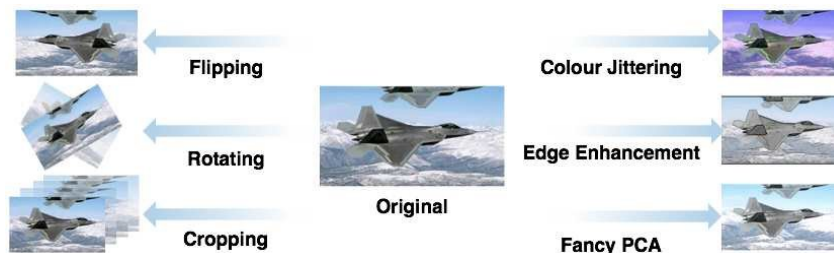


Figure 8: Example of data augmentation approach [71]

This has been already used in the painful expression dataset. For instance, Huang et al. [72] use such traditional methods including noise addition, contrast adjustment, brightness adjustment and image inversion in the **UNBC-McMaster** dataset, see Fig.9. After these approaches, the database is expanded more than 10 times.

While the above-mentioned techniques provide some relief, they do not get to the core of the problem. Techniques like typical data augmentation methods only provide limited diversity of data. This has inspired the use of deep learning approaches to generate **synthetic data** which can introduce more variability.

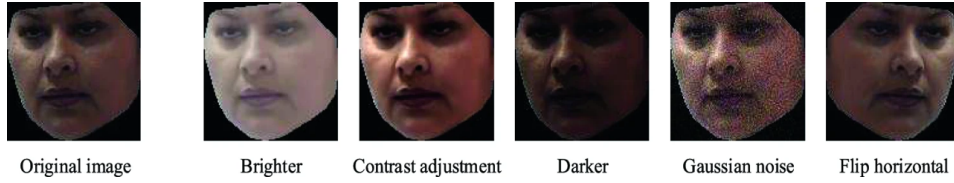


Figure 9: Data visualization after data augmentation in [72]

Synthetic data is artificially generated information as an alternative to real-world data [73]. Besides, it provides detailed ground truth annotation and is a low-cost and scalable alternative to manually annotating images [74]. As a result, the generation of synthetic data is increasingly used to overcome the burden of creating large datasets for training learning models, especially deep neural networks [75]. There are many methods of data synthesis, such as realistic image rendering and learning-based image synthesis [75]. Among the various methods, Generative Adversarial Networks (GANs) [76] are considered as a promising method.

So far, GAN has already been applied to many domains and data augmentation in one of them. Recently, researchers have shown great enthusiasm for GAN-based image synthesis for data augmentation. For instance, a recent study [77] using synthetic data generated by GAN to train cancer detection algorithms has achieved striking results. The results show that compared with training on an original dataset, the algorithm performs better on synthetic data. Besides, Niinuma et al. [78] has already demonstrated that networks trained on synthetic facial expressions outperform networks trained on actual facial expressions too.

2.4 Generative Adversarial Network (GANs)

Generative Adversarial Network (GAN), as its name implies, is a generative model that learns to make realistic data adversarially [79]. It is inspired by game theory, where generators and discriminators compete with each other in the training process [80]. The general architecture of GAN is illustrated in Fig.10

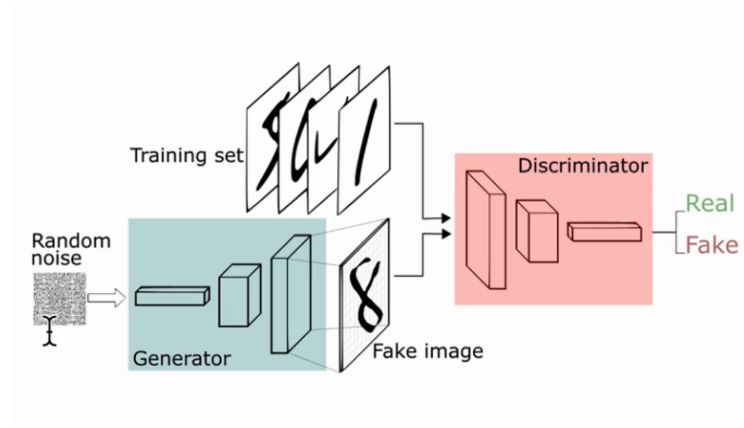


Figure 10: The architecture of generative adversarial networks (GAN) [81]

As we can see, a GAN has two parts:

- **Generator G**

The principle of generator G is to generate fake data to fit the potential distribution of the real data as much as possible. Specifically, the input of G is random noise z often with a uniform or normal distribution. Then the generated instances $G(z)$ become negative training examples for the discriminator D .

- **Discriminator D**

Discriminator D is a binary classifier to determine whether the given is "real" or not. It takes both the real samples from training set and $G(z)$ as input X . The output is the probability $D(X)$ that the sample is real.

Then, the training process proceeds in alternating period [82]:

1. **The D trains for one or more epochs.**
2. **The G trains for one or more epochs.**
3. **Repeat steps 1 and 2 until the model convergence.**

Notice, the parameters of G are kept constant during the training phase of D and vice versa.

In the training process, the goal of G is to generate images $G(z)$ as realistic as possible to deceive D , while D 's goal is to try to separate the images generated

by G from the real ones. In this way, D and G constitute a dynamic adversarial process. The final result is that, in the optimal state, G can generate a picture $G(z)$ that is sufficiently "fake" to be true. In other words, for D , it is difficult to determine whether the image generated by G is real or not, i.e. a Nash equilibrium is reached, so that $D_{(G(z))} = 0.5$. At this point, the convergence goal of the model is for G to be able to generate real data from random noise.

The above are the core ideas of GAN. In mathematical terms, as described by Goodfellow and his co-authors in the original paper [76], in GAN, G and D use a joint loss function, where G tries to minimise the function and D is trying to maximise it. The specific formula in the paper is as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2)$$

Here, $D(x)$ corresponds to a loss of two items which are $\mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log(D(\mathbf{x}))]$ and $\mathbb{E}_{\mathbf{z} \sim P_z} [\log(1 - D(G(\mathbf{z})))]$ respectively. The $\log(x)$ monotonically increasing over $[0, 1]$ because $D(x) \in [0, 1]$. Thus, we expect the real data $D(x)$ to converge to 1, so the expectation of the first term increases. Then, in the second term, $D(G(z))$ is the expectation of the generated data. It tends to approach 0 and is incremental. So, the overall expectation corresponding to $D(x)$ is larger which means $\max_{(D)}$. As for G , the loss is $\mathbb{E}_{\mathbf{z} \sim P_z} [\log(1 - D(G(\mathbf{z})))]$. We want the generated data to be closer to the real data, so $D(G(z))$ converges to 1 and the overall expectation is smaller, i.e. $\min_{(G)}$. In summary, the discriminator D aims to maximize loss and the generator G aims to minimize loss. Therefore, this process is also known as a two-player min-max game.

Essentially, the above-introduced is the concept of GAN which provide a broad framework. It has motivated the development of various GAN-based variants, which have proven effective in addressing practical challenges in a variety of applications and scenarios. Here, we will introduce some popular variants of GAN:

- **Deep Convolutional GAN**

Deep Convolutional Generative Adversarial Networks (DCGAN) [83] was the first structure that adopted convolutional networks into GAN. Before this work, CNN-based GANs have been unsuccessful. In this paper, a series of effective constraints on the network structure is proposed to make the training of CNN-based GANs more stable. The main contributions are:

- Replacing all pooling layers with convolution in both generator and discriminator.
- Removal of the fully connected layer, such as global average pooling.
- Use Batch Normalization in both the generator and discriminator.

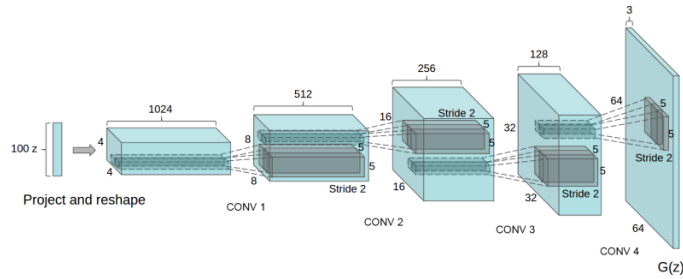


Figure 11: The Generator of DCGAN [80]

See Fig.11 as a visualization of an example model architecture

DCGAN has greatly improved the stability of GAN training and the quality of the generated results.

- **Conditional GAN**

Conditional Generative Adversarial Networks (CGANs) [84] is another extension to the original GAN. As the name implies, it allows the GAN to produce results that are conditional, that is, the final output can be controlled by artificially changing the vector of inputs (See Fig.12). Specifically, both the generator and the discriminator can add additional information y as a condition. Here y can be any information, such as category information, or other modal data. The optimization process of CGAN is

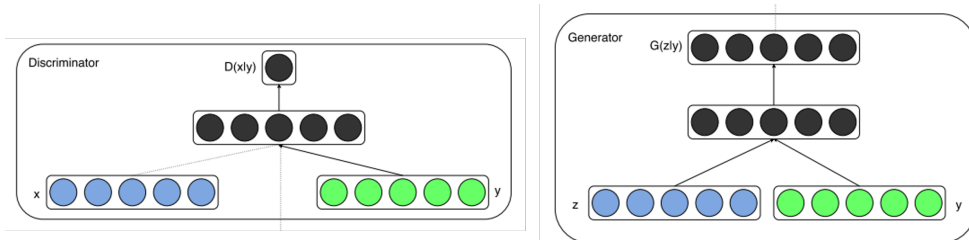


Figure 12: The CGAN [84]

a binary minimal maximal game problem with conditions:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x | c)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z | c)))] \quad (3)$$

Compared with GAN, one of the main advantage of CGAN is that it can generate data with user-defined properties.

- **Cycle GAN**

Cycle-Consistent Adversarial Networks (CycleGAN) [85] is widely used in domain adaptation. Sample data can be converted without pairing, for example zebra to horse, see Fig.13.



Figure 13: Example of image transformation [85]

Unlike the previously mentioned GANs, CycleGAN has two discriminators and two generators. As it is shown in Fig.14, the horse is passed through generator G to produce a picture of a zebra with its original shape. Immediately afterwards, another generator F is used to restore the freshly generated zebra picture to the previous horse's shape. Finally, two discriminators are used to determine the authenticity of the generated zebra and the real horse respectively. And vice versa.

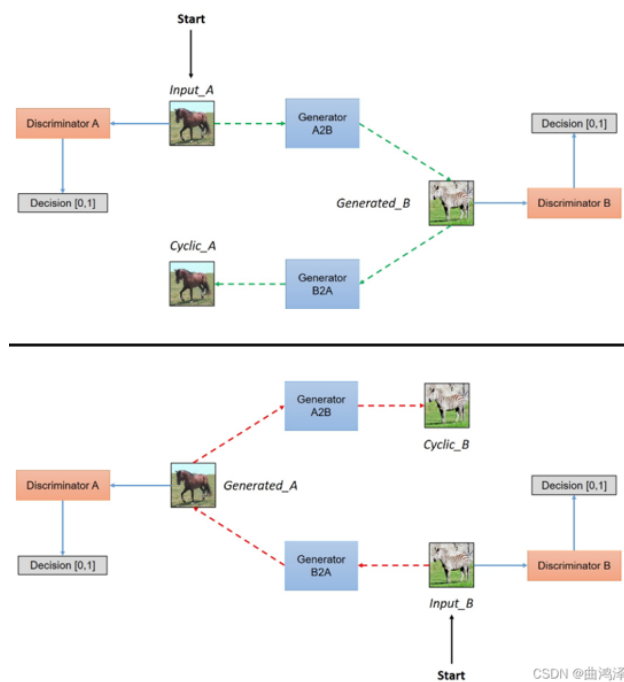


Figure 14: Network Architecture of CycleGAN [86]

During training, the discriminator and the generator are trained separately. As for the loss function, the Loss of CycleGAN consists of two parts:

$$Loss = Loss_{GAN} + Loss_{cycle} \quad (4)$$

$Loss_{cycle}$ refers to the Cycle Consistency Loss. It is a guarantee that the output image of the generator is only different in style from the input image, while the content is the same. For instance, in the training of horse to zebra, the generator generates a picture of a zebra. At this time, if the shape of the generated zebra differs significantly from the original horse, but the generated zebra is particularly realistic, then this is not what we expect. The Cycle Consistency Loss is to prevent this kind of situations. When the horse image is passed through the two generators and reconstructed back to the original horse, the two images are subtracted to calculate the difference between them. If the distance between the two is approximately smaller, that means that the two images are more similar. The specific formula is as follows.

$$\begin{aligned} Loss_{cycle} &= \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1] \end{aligned} \quad (5)$$

On the other hand, $Loss_{GAN}$ is to ensure that the generator produces a more realistic picture, with the following formula.

$$\begin{aligned} Loss_{GAN} &= \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) \\ &= \mathbb{E}_{y \sim p_{data}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log (1 - D_Y(G(x)))] \\ &+ \mathbb{E}_{x \sim p_{data}(x)} [\log D_X(x)] + \mathbb{E}_{y \sim p_{data}(y)} [\log (1 - D_X(F(y)))] \end{aligned} \quad (6)$$

Overall, CycleGAN is a ring-shaped structure consisting of two generators and two discriminators. Compared to other Domain Adaptation models, such as Pix2Pix [87], it requires data from only two domains, without the need for a strict matching relationship.

- **Wasserstein Gans**

Wasserstein Generative Adversarial Networks (WGAN) [88] have been proposed by optimizing the objective function while the previous-mentioned methods optimize the architecture of GANs.

It first theoretically analyses the problems of the original GAN and thus gives targeted improvement points in literature [89]. Then, compared to the original GAN implementation process, WGAN has changed four things:

1. The last layer of the discriminator removes the sigmoid.
2. Generators and discriminators do not take \log for the loss.
3. After each update of the discriminator parameters cut off their absolute value to no more than a fixed constant c .
4. No need for momentum-based optimization algorithms, including Momentum and Adam, RMSProp, SGD recommended

After these changes, WGAN achieves the following:

1. A thorough solution to the problem of unstable GAN training. This removes the need to carefully balance the training levels of generators and discriminators.
2. It helps in avoiding the issue of the generator producing a limited variety of samples, known as collapse mode [90], and ensuring the diversity of the generated samples.
3. The training process has a numerical value to indicate the progress of the training. A smaller value means that the GAN is better trained, in other words, the better quality of the images produced by the generator.
4. All of the above benefits can be achieved with the simplest of multi-layer fully connected networks.

Overall, WGAN solves the problem of training instability and provides a reliable metric of the training process that is highly correlated with the quality of the generated samples.

- **WGAN-GP**

Although WGAN has made progress in stable training, it can sometimes generate poor samples and still be challenging to converge. The weight clipping strategy used in WGAN to enforce the Lipschitz constraint on the discriminator restricts the weights to a fixed range, which can cause several issues during training. For instance, in some cases, it may limit the learning capacity of the network and prevent the discriminator from adequately discriminating between real and generated samples. To address this issue, Gulrajani et al. [91] proposed WGAN-GP as an improvement to WGAN. Here GP refers to Gradient Penalty.

In WGAN-GP, the weight clipping method is replaced with a gradient penalty term, where the norm of the gradient of the discriminator with respect to its input (i.e., the generated image) is constrained to a fixed value.

$$L = \underbrace{\mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} [D(\tilde{\mathbf{x}})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(\mathbf{x})]}_{\text{Original discriminator loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]}_{\text{The gradient penalty}}. \quad (7)$$

As we can see, the differences between WGAN and WGAN-GP may appear subtle, as both use the same regular term, called GP (gradient penalty), to improve the Lipschitz constraint on the discriminator. However, in WGAN-GP, the GP term is added acting as a regularization penalty that constrains the norm of the gradient of the discriminator with respect to its input. By imposing this additional constraint, the WGAN-GP approach

avoids the problems of weight clipping in WGAN and ensures that the discriminator learns a smoother decision boundary. This simple change leads to more stable training and higher-quality generated samples.

In this project, our aim is to use synthetic facial expressions to do the data augmentation for the painful expression dataset in order to achieve better performance for the APA. While there is no previous work on generating synthetic pain expressions, the **facial expression synthesis** for inspiration can give some inspiration.

Facial expression synthesis refers to generating realistic synthetic images of facial expressions using GANs. In this context, the goal is to generate synthetic images of faces exhibiting different emotions or expressions, such as happiness, sadness, anger, and surprise. Pain is not usually considered an emotion as it is defined as a physical sensation rather than a mental or emotional state. However, from the FACS perspective, they share similarities. For example, sadness can be coded by AU1 (Inner Brow Raiser), AU4 (Brow Lowerer) and AU15 (Lip Corner Depressor). While as we mentioned in Section .2.3, pain is encoded by another set of corresponding AUs.

Numerous productive studies have been conducted in the area of Facial Expression Synthesis, yielding significant advancements in the field. Recently, the utilization of GAN-based methods has emerged as a widely sought-after research trend within the field of Facial Expression Synthesis:

- **StarGAN**

The StarGAN [26] was developed to address the challenge of multi-domain image conversion. Prior to StarGAN, other GAN models such as CycleGAN were limited to converting between only two domains. This meant that to perform conversions across C domains, $C \times (C - 1)$ separate models would need to be trained. In contrast, StarGAN requires only a single model to be trained, while still delivering exceptional results, see Fig.15.

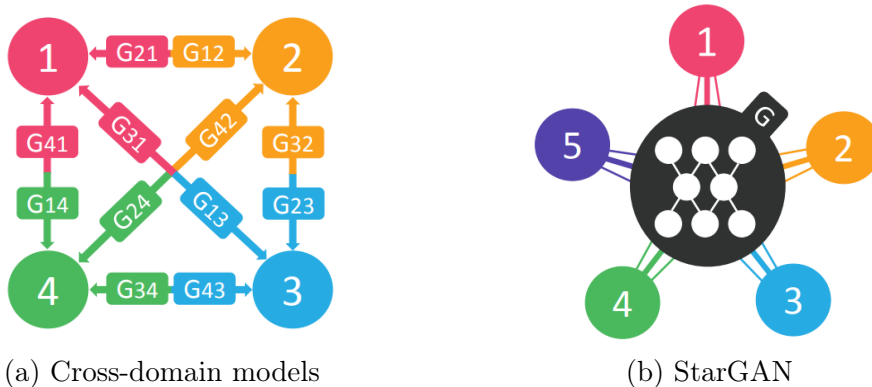


Figure 15: Comparison between cross-domain models and StarGAN [26]

The defining characteristic of StarGAN is its integration of domain control

information, similar to the architecture of a CGAN, making this its main contribution to the field, see Fig.16.

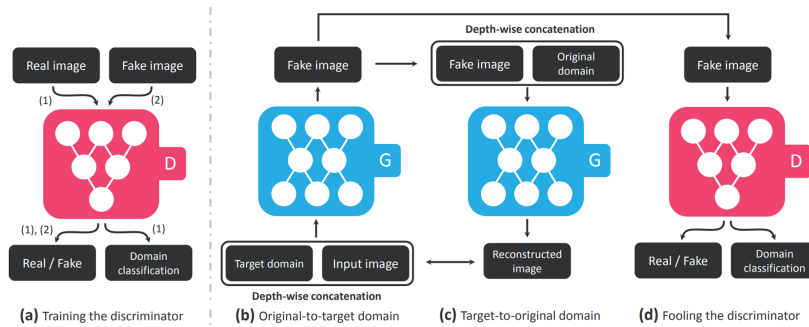


Figure 16: Overview architecture of StarGAN [26]

One of its key applications is the transformation of facial attributes, see Fig.17.



Figure 17: Overview architecture of StarGAN [92]

It uses the training dataset of CelebFaces Attributes (CelebA) and Radboud Faces Database (RaFD).

- **GANimation**

While StartGAN has been successful in generating facial expressions, it is limited to the creation of discontinuous expressions. GANimation [93] addresses this limitation by introducing a novel GAN conditioning scheme that's based on AU annotation. By describing the anatomical facial movements that define human expressions, GANimation allows for controlling the activation amplitude of each AU and combining several of them. In addition, the model can be trained, using an unsupervised strategy with only images annotated with activated AUs. The network is also robust to changing background and illumination conditions using an attention mechanism.

The GANimation network consists of two generators (G) and a discriminator (D), combining the concepts of conditional GAN and CycleGAN. It uses FACS to enable the generators to produce a sequence of smoothly transitioning expressions. Besides, unlike traditional CycleGANs that have two generators, the network employs a single generator, which generates expressions based on the input expression codes. The cycle is achieved by swapping the expression codes between the generator's output and input. See Fig.18 as a simplified version of the GANimation network.

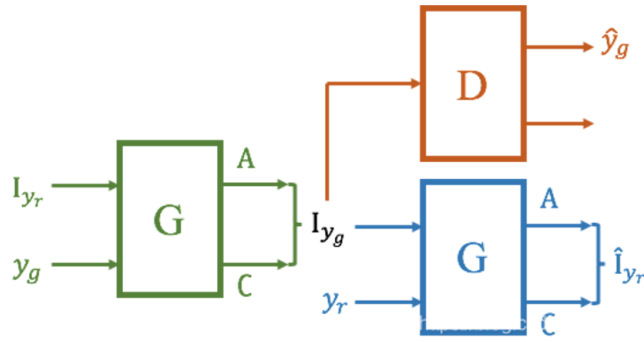


Figure 18: Simplified version of GANimation network [93]

The method consists of two modules, as shown in the figure. First, a generator $G(I_{y_r}, Y_g)$ is trained to produce a facial expression image that closely resembles the target expression specified by the input image I_y and the expression code Y_g . The generator is executed twice: first to produce the generated image I_{y_g} from the reference image I_{y_r} (the mapping relationship: $I_{y_r} \rightarrow I_{y_g}$), and then to regenerate I_{y_r} from I_{y_g} . Second, a discriminator $D(I_{y_g})$ is trained using WGAN-GP to evaluate the authenticity of the generated images.

In addition, another important contribution of the proposed model is the attention mechanism in the generator, see Fig.19. This mechanism enables the model to emphasize the areas that require deformation and to keep the remaining facial features and the background as unchanged as possible. By focusing on specific regions, the model can produce more accurate and robust facial expressions.

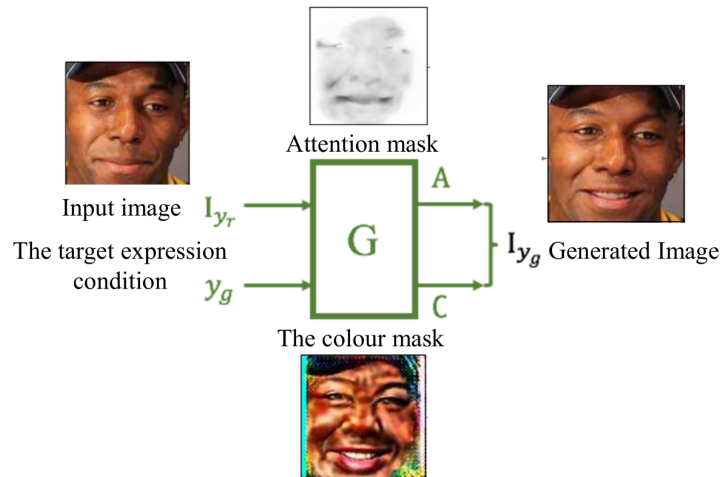


Figure 19: The attention mechanism: Attention-based generator [93]

As for the loss function, it consists of four items:

– **Image Adversarial Loss**

Image adversarial loss is a GAN loss calculated using WGAN-GP. It is included to obtain realistic synthesized images and to ensure that the distribution of the generated images is similar to the distribution of the training images.

– **Attention Loss**

Since the data does not contain a ground truth for the Attention Mask, the generator trained to produce the mask can often reach a saturation value of 1, rendering it ineffective. To address this issue and ensure a smooth transition between generated images, the paper introduces a Total Variation Regularization of the Attention Mask A . The equation is as follows:

$$\lambda_{\text{TV}} \mathbb{E}_{\mathbf{I}_{y_o} \sim \mathbb{P}_o} \left[\sum_{i,j}^{H,W} [(\mathbf{A}_{i+1,j} - \mathbf{A}_{i,j})^2 + (\mathbf{A}_{i,j+1} - \mathbf{A}_{i,j})^2] \right] + \mathbb{E}_{\mathbf{I}_{y_o} \sim \mathbb{P}_o} [\|\mathbf{A}\|_2] \quad (8)$$

– **Conditional Expression Loss**

The purpose of Conditional Expression Loss is to allow the generator to learn to generate target expressions based on the expression code, see the equation below.

$$\mathbb{E}_{\mathbf{I}_{y_o} \sim \mathbb{P}_o} [\|D_y(G(\mathbf{I}_{y_o} | \mathbf{y}_f)) - \mathbf{y}_f\|_2^2] + \mathbb{E}_{\mathbf{I}_{y_o} \sim \mathbb{P}_o} [\|D_y(\mathbf{I}_{y_o}) - \mathbf{y}_o\|_2^2] \quad (9)$$

– **Identity Loss**

Identity Loss is actually the cycle-consistency-loss in CycleGAN, which is designed to guarantee that faces in both input and output images belong to the same person.

These four components add up to the total loss function.

The resulting image is obtained by manipulating the input image I_{y_r} , which is indicated by the green square in Fig.20, using the parameter α .

Here, the parameter α controls the degree of activation of the target action unit (AU) involved in the smile expression. CelebA is used for the training database.

In summary, one of the most important aspects of GANimation is its ability to adapt facial expressions based on action units (AU). By allowing the generator to learn how to manipulate specific AUs, the network can generate a wide variety of facial expressions in a completely unsupervised manner. It provides a more flexible and scalable framework for facial expression synthesis.



Figure 20: The Examples of expression animation in a continuous domain [93]

After generating data using GANs, it is often necessary to evaluate the quality and diversity of the results. In this context, Inception Score (IS) [94] and the Fréchet Inception Distance (FID) [95] are commonly employed as metrics to assess and compare the efficacy of diverse GAN models. Below, we will provide the description for each of these metrics respectively.

- **Inception Score (IS)**

The basic idea of the Inception Score (IS) is to use an image classifier to evaluate the quality of generated images. Specifically, the classifier used is the Inception Net-V3, which has been trained on the ImageNet dataset. This metric is commonly used to measure both the quality and diversity of the generated images from GAN networks. IS is calculated using a specific formula, which is as follows.

$$IS(G) = \exp(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x) || p(y))) \quad (10)$$

Here, D_{KL} is the formula for KL-divergence which is a measure of the difference between two probability distributions.

The objective of GANs is to generate clear and diverse images, hence a higher IS score indicates better image quality. Additionally, since Inception Net-V3 is trained on a 1000-class classification task, the maximum attainable IS score is limited by the number of classes, with $IS(G) \leq 1000$.

However, the IS score has limitations. The IS score heavily relies on the chosen classifier and is an indirect means of evaluating the quality of the generated images. Furthermore, it does not account for the specific dissimilarities between real and generated data. Since the Inception Score is

based on ImageNet, any data that does not resemble ImageNet may be judged as not real (fake) according to the IS metric.

- **Fréchet Inception Distance (FID)**

Unlike IS, Fréchet Inception Distance (FID) directly measures the distance between the generated data and the real data at the feature level, without the use of a classifier. As a result, it is often used to directly evaluate the similarity between the generated image and the real image.

The top layer of a pre-trained neural network is known to extract high-level information about an image and reflect the essence of the image to some extent. FID takes advantage of this by using the 2048-dimensional vector extracted before the fully connected layer of a pre-trained Inception V3 as a feature of the image. It directly considers the distance between the generated data and the real data at the feature level, without relying on a classifier. The formula for calculating FID is as follows.

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2} \right) \quad (11)$$

The formula involves the trace of a matrix, which is the sum of its diagonal elements, commonly known as "trace" in matrix theory. It also uses the mean (μ) and the covariance (Σ) of the distribution, where r represents the real image and g represents the generated image.

A smaller value of FID indicates that the distributions of the real and generated images are closer to each other, which in means that the quality and diversity of the generated images are better.

Compared with IS, FID exhibits greater robustness to noise. Furthermore, unlike IS which assesses the authenticity of generated data by comparing it with ImageNet data, FID evaluates by comparing the generated data directly with the training data.

Overall, due to its comparative advantages, FID is currently a widely used evaluation metric in the evaluation of GAN networks.

3 Selected Dataset Overview and Analysis

In this project, the selected dataset is **The Unbc-Mcmaster Shoulder Pain Expression Archive Database (UNBC-McMaster dataset)**. It is one of the most common databases to be used in the APA field. In this section, a full description and analysis of the selected dataset will be provided.

3.1 Dataset selection: UNBC-McMaster dataset

UNBC-McMaster dataset is a collection of video sequences aimed at automatically detecting facial pain in patients [62]. It was compiled by researchers at McMaster University and the University of Northern British Columbia and includes spontaneous facial expressions, Facial Action Coding System (FACS) coded frames, frame-by-frame and sequence-level pain levels, and Active Appearance Model (AAM) landmarks [96]. To facilitate pain-related research and supplement existing datasets, a portion of the data, collected from participants undergoing range-of-motion tests while experiencing shoulder pain, was first made available in March 2011.

The available dataset consisted of 25 individuals. During the data-collecting phase, all participants were asked to perform movements in a laboratory room in both active and passive conditions. The active test asks the participants to stand and is instructed to move the limb as far as possible. In contrast, the passive test was performed by a physical therapist who moved the limb until it reached maximum range or was asked to stop by the participants.

In both active and passive tests, two digital cameras recorded the participants' facial expressions. Then, in the assessment part, participants rated the pain produced by each test verbally, where a card listing verbal pain descriptors was given in advance. Furthermore, independent observers rated pain intensity from the offline recorded videos at the sequence level. Notice, these observers had considerable training in the identification of pain expression.

Fig.21 gives an example of the above process.



Figure 21: The sequences example from **UNBC-McMaster** dataset [62]

Then, the frame-by-frame level FACS codes and pain rating (PSPI) are implemented.

First, the collected data were coded using the Frame-by-Frame Facial Action Coding System (FACS). Here, the FACS coding only focused on movements potentially related to pain, see Fig.22.

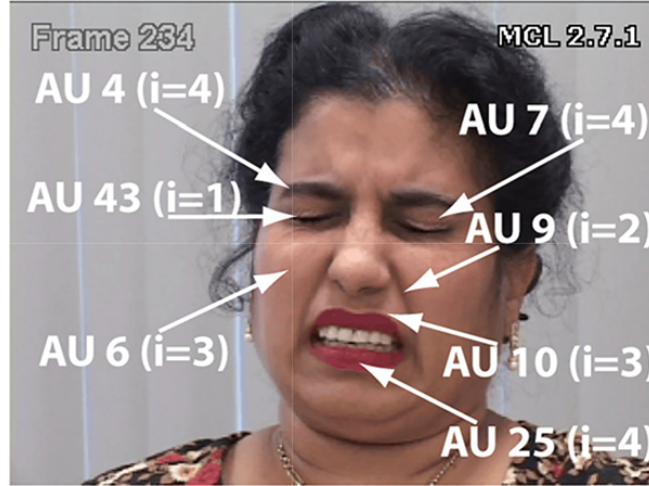


Figure 22: An example with the corresponding AU and their intensity [53]

Then, we calculate the PSPI score based on the AU intensities. For instance, the PSPI of Fig.21 is $\text{Pain} = 4 + \text{Max}(3, 4) + \text{Max}(2, 3) + 1 = 4 + 4 + 3 + 1 = 12$.

Briefly, the entire available **UNBC-McMaster** dataset has 200 sequences from 25 different subjects. From the 200 sequences, a total of 48398 frames have been FACS coded and PSPI scaled. Here the PSPI score was computed to quantify pain intensity in 16 discrete levels (0-15).

Notice, not all the images of the **UNBC-McMaster dataset** can be shown. We have been granted permission to use a select set of images (TV095, JL047, IB109, DM039, AK064) for electronic media, with the condition that we acknowledge the copyright holder (@jeffery Cohn). Therefore, all test cases presented in this thesis are from this subset of licences.

3.2 Dataset analysis

Next, we give a detailed analysis of the selected dataset. The specific analysis is as follows:

- **Imbalanced label distribution**

The UNBC-McMaster database has an uneven distribution of PSPI scores between the different intensity levels, see table.4.

PSPI	0	1-2	3-4	5-6	7-8	9-10	11-12	13-14	15
% total	82.71	10.87	4.57	1.06	0.27	0.20	0.26	0.05	0.01

Table 4: Data distribution of **UNBC-McMaster** dataset

The percentage of each pain intensity relative to the total number of frames is given there ($N = 48398$).

As we can see, 82.71% of frames had a PSPI score of 0 and 17.02% of frames had a PSPI score ≥ 1 .

- **Insufficient volume of data**

There is an insufficient amount of data for most pain levels in the database. For example, there are only 895 images for pain intensity greater than 5. Besides, the **UNBC-McMaster** database contains data on only 25 people. Both factors make it far from enough to train machine learning models that can generalize to the unseen subject.

- **Range of head pose**

Even though faces are nearly frontal, out-of-plane head rotation is present in several frames, see Fig.23.



Figure 23: An example of head movement in the database [43]

The video sequences have various durations, with sequences lasting from 90 to 700 frames. As a result, the faces in the images are not aligned and no semantic correspondence is established across subjects and frames.

Overall, the **UNBC-McMaster** database presents two major challenges in terms of being imbalanced and insufficient for a multi-class classification task.

This makes it challenging to train machine learning models that can generalize well. Furthermore, the range of head pose may cause misalignment of facial expressions and affect the accuracy of feature extraction and analysis [97].

4 Methodology

In this section, we provide a detailed methodology of GANimation as a tool for augmenting pain databases, see Fig .24 as the overall pipeline.

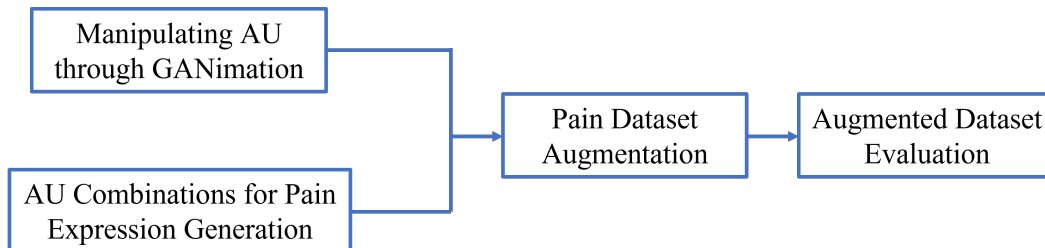


Figure 24: The Overall Pipeline

We first give a detailed implementation of GANimation, which enables us to control the magnitude of activation of each AU. This will give us the ability to generate facial expressions effectively. Then, a novel mechanism is proposed for generating sufficiently diverse pain expressions through varied AU combinations. Based on these two methods, a comprehensive solution for generating a balanced and sufficiently diverse dataset is proposed. Finally, we give the evaluation procedure for the augmented datasets, which will assess the effectiveness of our augmentation strategy.

4.1 Manipulating AU through GANimation

After a comprehensive investigation and comparative analysis of various GAN networks in section 2.4, we choose **GANimation** [93] as the foundational architecture for our project. This decision was based on the unique features of GANimation, the ability to precisely control the activation amplitude of each action unit (AU) and the ability to combine multiple AUs. In this section, we give the detailed implementation process, result analysis and finetuning improvements.

4.1.1 Training Phase

We use a reimplementation of GANimation [98] based on Pytorch. Compared with the original GANimation, its codes are cleaner and well structured as well as providing a more powerful test function. As for the training part, the specific process is as follows:

- **Data Pre-processing**

We use the dataset called EmotionNet [99] which contains more than 400k in-the-wild face images to train the GANimation network. See Fig.25 is

the example of the query obtained when retrieving all images identified as happy and fearful by the algorithm.



Query by emotion	Number of images	Retrieved images
Happiness	35,498	
Fear	2,462	
Query by Action Units	Number of images	Retrieved images

Figure 25: The Examples from EmotionNet dataset [99]

The dataset was collected using several web search engines to specifically select images with faces as features and associated with sentiment keywords in WordNet [100].

For every image in the dataset, we first extract the face bounding box and crop the face from the images. Here we use **face_recognition**, a simple face recognition library to recognise and manipulate faces from Python or the command line [101]. The final image output size is $128 * 128$.

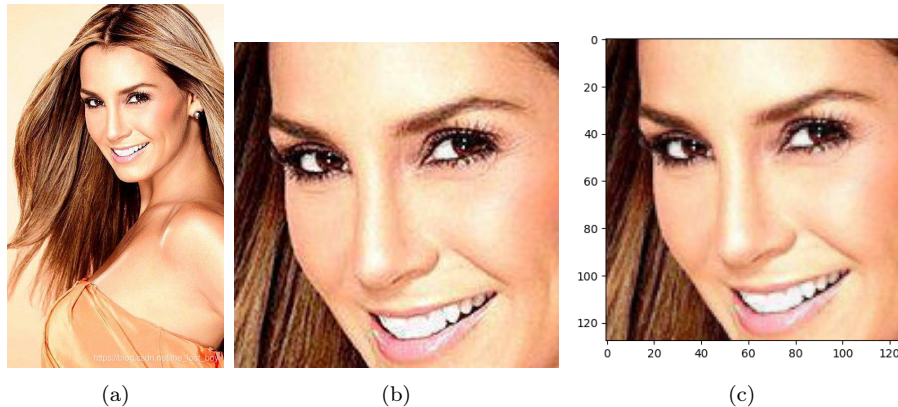


Figure 26: Image [102] pre-processing example: (a) the original image (b) Face bounding box is extracted (c) Face is cropped from the images and resized to $128 * 128$.

Fig. 26 illustrates an example of face detection by face_recognition, with the detected face location's pixel coordinates as follows: Top: 118, Left: 118, Bottom: 304, Right: 304.

- **Obtaining AU Annotations**

As we mentioned in the section 2.3, the creation of annotated databases based on the FACS system, including Action Units (AUs) and their respective intensities, requires a lot of expert coders and their time [103]. This has led to a notable lacking of large-scale, annotated databases and video sequences of facial expressions.

To address this challenge, here, we use OpenFace [104] to extract a vector of AU intensities from available facial expression datasets. The Action Units used in this project is: [AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU12, AU14, AU15, AU17, AU20, AU23, AU25, AU26, AU45].

AU intensity assigned by OpenFace is a number between 0 and 5 with two decimal places. Here, we divide all values by 5 and normalize them to within the interval 0-1.

- **Training Setting**

We trained the model with 410k images from the EmotionNet dataset to reduce training time. Then, we choose **Adam** [105] with a **learning rate** of 0.0001, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and **batch size** 10. The model is trained for 30 epochs with a linear decay rate of 0 over the last 10 epochs. For every 5 optimization steps, the critic network we perform a single optimization step of the generator.

4.1.2 Testing Phase and Result Analysis

After completing the training phase of the model, we do the testing part.

For this, we first selected images from the **UNBC-McMaster dataset** with a pain intensity score of 0 as the input test images. Since every subject in the dataset contains an extensive subset of data marked with zero pain intensity, we gave priority to the frontal images with eyes open as input data. Then, we follow the same image pre-processing procedures as in the training phase. Besides, to improve the clarity of the generation result, we perform the **alignment** by using the alignment function in Openface to position the head pose directly in front. See Fig.27 as an example Input image for the following experiments.

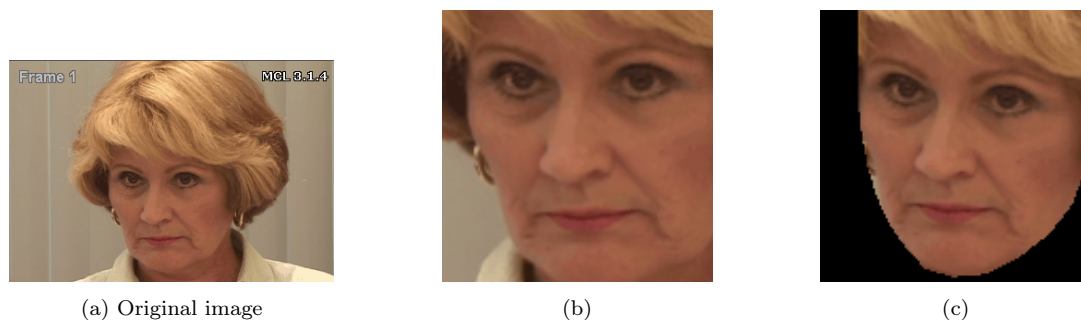


Figure 27: Image Pre-processing in the testing phase: (a) Original image selected from the dataset (b) The same pre-processing procedure as the training phase (c) Final input test image.

We carried out the experiment for all pain-related AUs, initiating at an intensity of 0.2. The intensity was subsequently increased incrementally by 0.2 in each step. Fig.28 is the generation result. As we can see, the generated results

are very noisy, especially when the intensity increases.

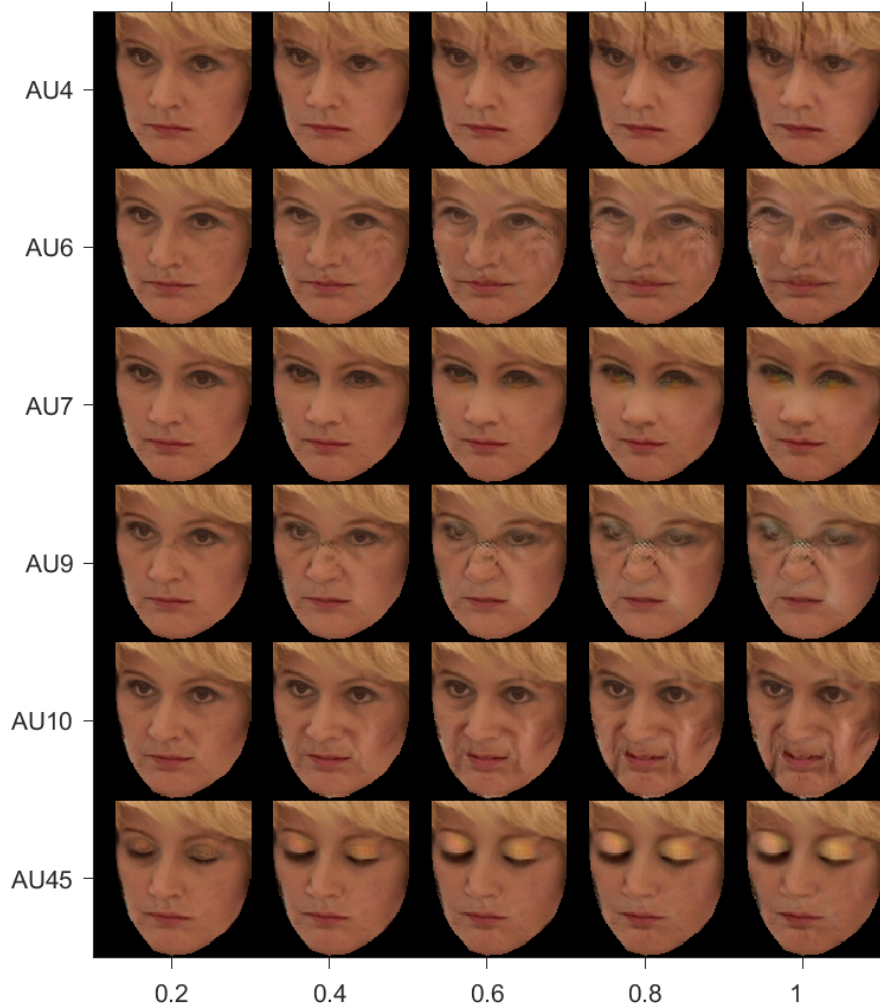


Figure 28: Generation result obtained from GANimation

We attribute these to two main factors through detailed analysis:

- **Domain Variations:**

The GANimation network was trained on the EmotionNet dataset and then tested with images from the UNBC-McMaster dataset. Unfortunately, the results have proven to be somewhat noisy. We attribute this primarily to the different environments of the two datasets.

The EmotionNet dataset contains **in-the-wild** face images. Here, in-the-wild means images downloaded from the Internet such as social media. In

other words, it was captured in a variety of uncontrolled settings with varying light conditions and camera quality. In contrast, the UNBC-McMaster dataset was collected under more controlled laboratory conditions. Such differences in the data collection environment (**external factors**) may introduce unpredictable variables, such as differences in lighting and image quality, which can affect the performance of our models.

Overall, we believe it is these domain variations that affect the performance of our model when transforming from one dataset to another.

- **Residual features after pre-processing:**

Although we pre-process the data in both training and testing phases, we still found that certain elements such as hair were still present. This can affect the training results. For example, the Brow Lowerer, represented by AU4, may be obscured by hair. We think this is another reason which would affect the quality of the generated images.

4.1.3 Finetuning

Based on the above-mentioned analysis results, we propose two strategies for fine-tuning the GANimation network in order to improve the generation results.

- **Within domain generation:**

To reduce the effects of Domain Variations, we propose **fine-tuning GANimation using the UNBC-McMaster dataset**, thereby solving the issues we previously encountered from the EmotionNet.

By fine-tuning the model in this way, we can create coherence between the training and testing environments. Given that the UNBC dataset has uniform settings in terms of resolution, camera, view angles, and age groups, it can provide a more consistent benchmark for testing the model. In other words, this solution essentially provides so-called 'within domain generation', in which both the training and testing data come from the same source or domain. We believe by reducing these domain variations between the domains of training and testing sets, we will get less noisy results, leading to more reliable and precise outcomes.

- **3D registration:**

We perform **3D registration** to solve the residual features of both the training and testing phases, aiming to the reduction of residual features.

As we analyze in 3.2, out-of-plane head rotation is present in several frames. Preliminary pre-processing such as 2D alignment using OpenFace proved insufficient, as demonstrated in Fig.29b. Therefore, we apply 3D registration to every image in the UNBC-McMaster dataset. Our approach is informed by the 3D registration method presented by Giacomo et al. [106]. In specific, it uses PRNet [107] which gets a 2D face image as

input, performs 3D registration without requiring person-specific training, and outputs a dense 3D mesh of the face. The result is shown in Fig.29c.

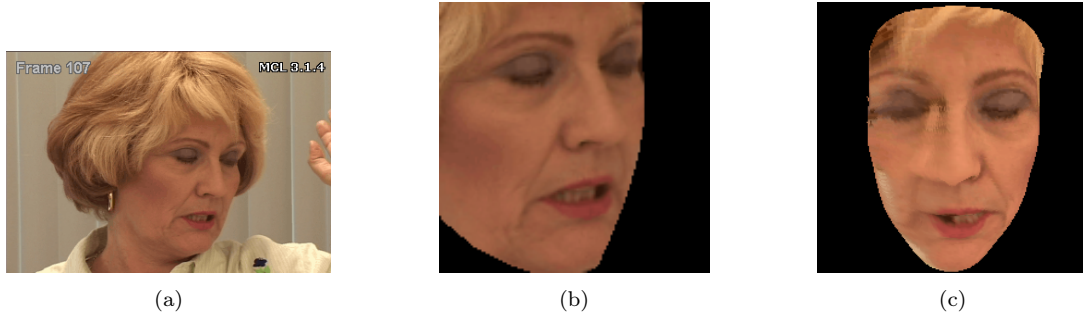


Figure 29: Out-of-plane head rotation and alignment: (a) Example of out-of-plane head rotation (b) Alignment using Openface. (c) 3D registration.

Then, we begin to finetune the GANimation based on the two strategies. The detailed process is as follows:

1. Step 1: Dataset splitting

We first split the training dataset and the test dataset. The whole UNBC-McMaster dataset has a total of 25 subjects, and we choose 20 subjects for the training set and reserve the remaining 5 for the test set.

2. Step 2: 3D registration

We perform 3D registration of the data in the training set. Notice, the out result also shows that five images (see Fig.30 as a shown example), failed to be extracted and we removed them from the training set.

3. Step 3: Obtaining AUs labels

We used OpenFace to extract AU from the 3D-registered images in the training set. The extracted results are saved in the CSV files with the same name as the corresponding image.



Figure 30: Part of failed examples

After completing the above pre-processing steps, we finetune the GANimation that is pre-trained on EmotionNet with UNBC-McMaster dataset. Here, we keep the training setting the same as those used for training EmotionNet, see 4.1.1.

4.1.4 Generated Result after Finetuning

In the testing part of the fine-tuned GANimation, we carried out the same experiment setting and input image as in section 4.1.2.

A visual comparison of results, illustrated in Fig.31, indicates that images have a higher quality compared to those produced without fine-tuning.

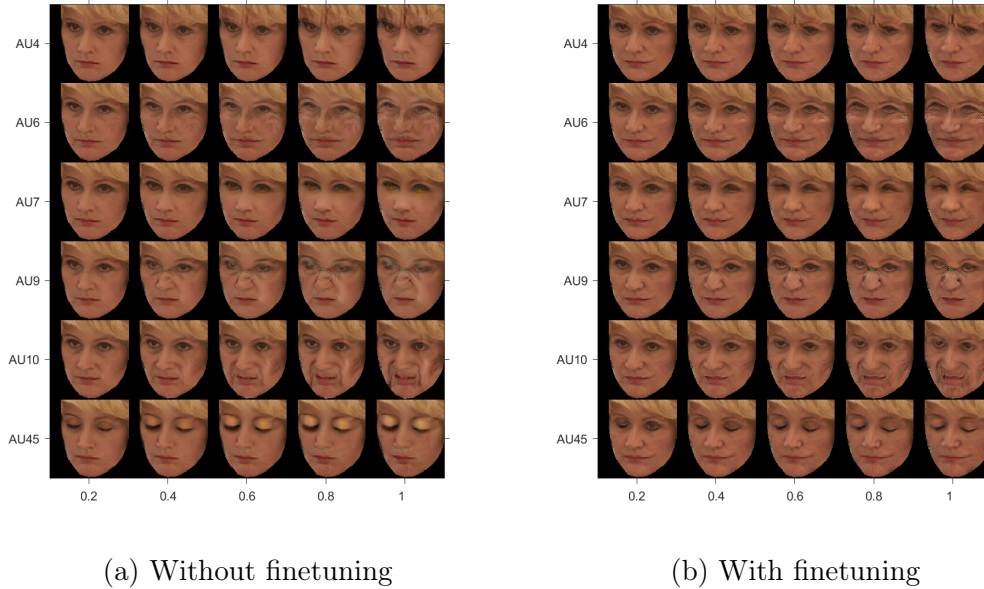


Figure 31: Comparison of generated results (2D aligned).

More generation results of the permitted-shown cases can be found in appendix A. We think these supplemental examples will provide a wider perspective. Besides, a comprehensive analysis of the comparison between the generated images with fine-tuning and without fine-tuning GANimation network is provided in Section 5.1.

In conclusion, the fine-tuned GANimation model can be used as an effective tool for generating painful expressions in our project.

4.2 AU Combinations for Pain Expression Generation

So far, we can already control the magnitude of activation of each AU and combine several of them through GANimation. Our goal now is to manipulate these AUs in order to generate pain expressions of various intensities. The overall solution for pain expression generation is:

By manipulating the Action Units (AU) through GANimation to generate the images, we calculated the different pain levels of the generated images according to the pain formula 1.

In this part, we will first describe the components of the pain formulation. We will then analyse and identify specific AUs that contribute to noisy output. Lastly, a scoring mechanism is proposed which can help us select the most effective combination of AUs to generate clear, realistic expressions of pain.

4.2.1 Components of Pain Formula

As it shows in the pain formula 1, only the action units [AU4, AU6, AU7, AU9, AU10, AU43] contribute to the pain level calculation. In our GANimation implementation, we use OpenFace to generate [AU4, AU6, AU7, AU9, AU10]. However, OpenFace does not generate AU43. Since it is not possible for us to introduce a new label like AU43 into the GANimation system. Therefore, we propose an alternative strategy to **approximate the AU43, which was included in OpenFace-generated labels, to AU45.**

In specific, a review of the relevant literature shows that AU43 denotes Eyes Closed, which is scientifically defined as ‘the relaxation of levator palpebrae superiors. AU45 refers to blinking, defined as relaxation of levator palpebrae superioris; contraction of orbicularis oculi (pars palpebralis). In the pain formula, AU43 can be either 0 (eyes closed) or 1 (eyes opened). Based on these definitions, we make the following assumption: **when a blink (AU45) reaches a certain intensity, it can be considered as an eye closed (AU43).**

We then conduct the following experiments to validate this assumption and find the threshold. Here, we keep all other AU equal to 0 and only change the intensity of AU43. Fig.32 presents images generated by GANimation with different AU45 values.

As we can see, when $AU45 = 0.4$, the eye is observed to be fully closed. Therefore, We have come to the following conclusion:

$$AU43 = \begin{cases} 0 & AU45 < 0.4 \\ 1 & AU45 \geq 0.4 \end{cases} \quad (12)$$

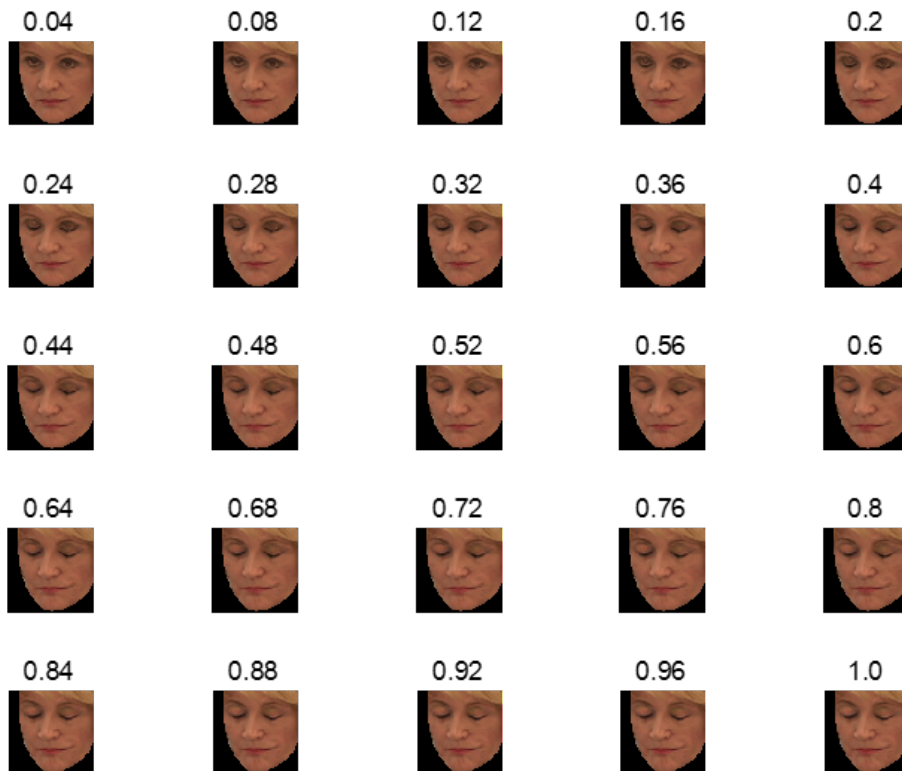


Figure 32: The generated images if different AU45 values

With the components of the pain formula defined, we can now generate different levels of painful expressions by manipulating various action units.

To start, We define a **Pre-determined List**. In every pre-determined list, we assign varying values to pain-related Action Units ([AU4, AU6, AU7, AU9, AU10, AU43]), while setting all other AUs not associated with pain to 0. Here, since the AUs are initially normalised in GANimation from 0-5 to 0-1, we present the normalised values in this context. Then we calculate the corresponding PSPI level. See table 5, each row is examples of predetermined lists.

AU4	AU6	AU7	AU9	AU10	AU45	PSPI
0	0	0	0.2	0	0	1
0.2	0	0	0	0.4	0	3
0	0.6	0	0	0	0.4	4
0.6	0	0.8	0.8	0	0.4	12

Table 5: Examples of predetermined lists

By giving the input image and the Pre-determined List, GANimation can generate pain expression. Fig.33 is the generation results derived from the Pre-determined List, corresponding to each row in table 5.



Figure 33: Generation results according to table 5

4.2.2 Generated Result Analysis

By conducting a series of experiments aimed at generating various levels of pain expressions using different pre-determined lists, we made several notable observations.

- **Observation 1:**

The noise is more noticeable in AU9, see Fig.34 below.



Figure 34: The generated images with different AU9 values

- **Observation 2:**

The noise level increases as the AU intensity value increases. See Fig.35 as examples.

- **Observation 3:**

The overlapping effect of multiple action units (AU) occurs when multiple action units (AU) exhibit intensity at the same time, which may have some negative effects on the generated results.

As we can see from Fig.36, we recognized that Action Units 4 (AU4), 6 (AU6), 7 (AU7), and 45 (AU45) are all associated with the upper facial region.

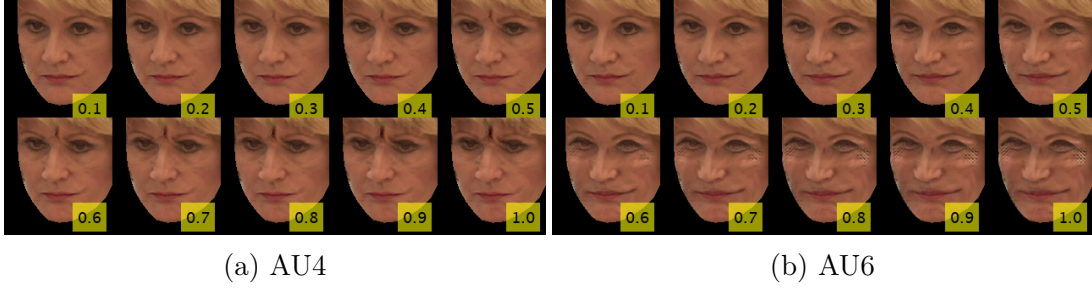


Figure 35: The generated images with AU values from 0.1 to 1.0



Figure 36: The FACS description of pain-related AU. (a) Brow Lowerer (b)Cheek Raiser (c)Lid Tightener (d)Nose Wrinkler (e)Upper Lip Raiser (f)Eyes Closed

Having each of these four AUs simultaneously exhibiting intensity could potentially result in overlapping effects, which may distort the accuracy of the results or render them more complex to interpret. Fig.37 clearly shows the generated result of multiple AUs.

Each row corresponds to a different AU intensity, while each column represents the set of AUs represented by the corresponding number.

In conclusion, our observations demonstrate that different pre-determined lists indeed generate pain expressions of varying quality.

4.2.3 Scoring Mechanism

In this part, we introduce a scoring mechanism to find the optimal Pre-determined Lists.

First, we defined a **Pre-determined Matrix** M which consists of all possible **Pre-determined Lists** l_i . Specifically, we consider values ranging from 0 to 5 for each of the following AUs: [AU04, AU06, AU07, AU09, AU10] and either 0 or 1 for [AU45]. The corresponding pain intensity PI was calculated based on the pain formula. Then, we normalize [AU04, AU06, AU07, AU09, AU10] to a 0-1 range and set the value of AU45 according to formula 12.

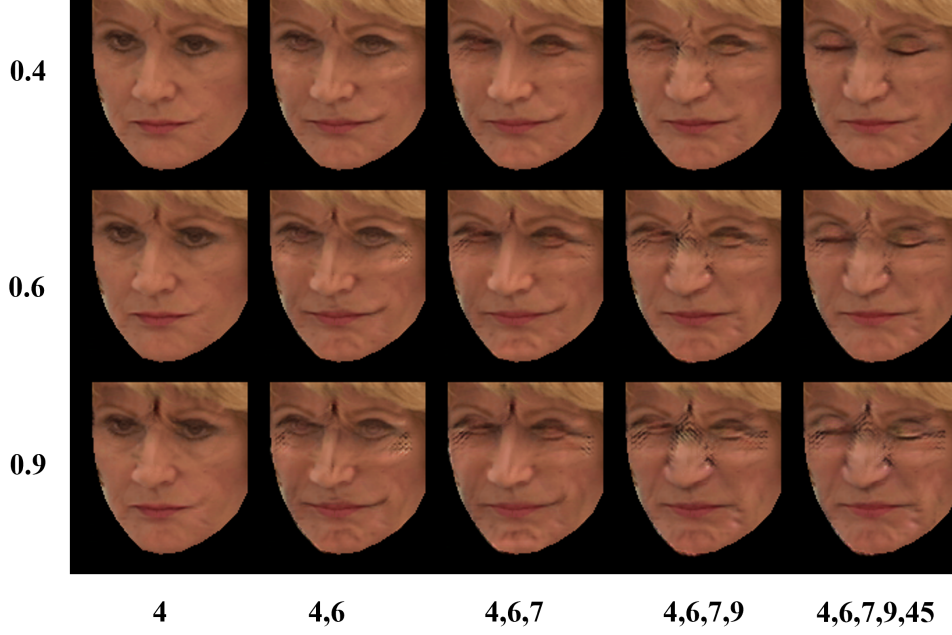


Figure 37: The generated images of multiple AUs

PSPI	0	1	2	3	4	5	6	7	8
Types	1	8	33	96	225	456	806	1240	1686
PSPI	9	10	11	12	13	14	15	16	
Types	2056	2246	2136	1785	1344	873	440	121	

Table 6: Statistical data of Pre-determined Matrix

Table 6 gives the count of all possible combinations with the same pain levels in the Pre-determined Matrix.

Then, the scoring mechanism based on the observations for every l_i in M is as follows:

- **Initialization.**

For each Pre-determined List in the Pre-determined Matrix, we set the initial score $S = \{S_{AU_n} | n = 4, 6, 7, 9, 10, 45\}$ of each Action Unit (AU04, AU06, AU07, AU09, AU10, AU45) to 10. Here, we set $V = \{v_{AU_n} | n = 4, 6, 7, 9, 10, 45\}$ to represent the value of each AU.

- **Adjustments based on Observation 1**

If the value of AU09 is greater than 0.3, the initial score of AU09 is: $S_{AU_9} = S_{AU_9} - 2 \times 10 \times v_{AU_9}$

- **Adjustments based on Observation 2**

If the value of any AU among AU04, AU06, AU07, or AU10 is greater than 0.8, the initial score for that particular AU is: $S_{AU_n} = S_{AU_n} - 10 \times v_{AU_n}$.

- **Adjustments based on Observation 3**

The scores $Score$ for each of AU04, AU06, AU07, AU09, AU10, and AU45 are summed to obtain a total score, $Score = Sum\{S_{AU_n} | n = 4, 6, 7, 9, 10, 45\}$. Then, we count the number n of AUs among AU04, AU06, AU07, AU09, and AU45 which is bigger than 0.8. The total score $Score = Score - n * 2$.

- **Shuffling the images**

For rows with the same pain level PI , the rows are sorted based on the total score $Score$, from highest to lowest. In cases where the total score is identical, we Shuffled the rows.

Fig.38 gives examples of pain level 10. We randomly choose the Pre-determined List with different score values.

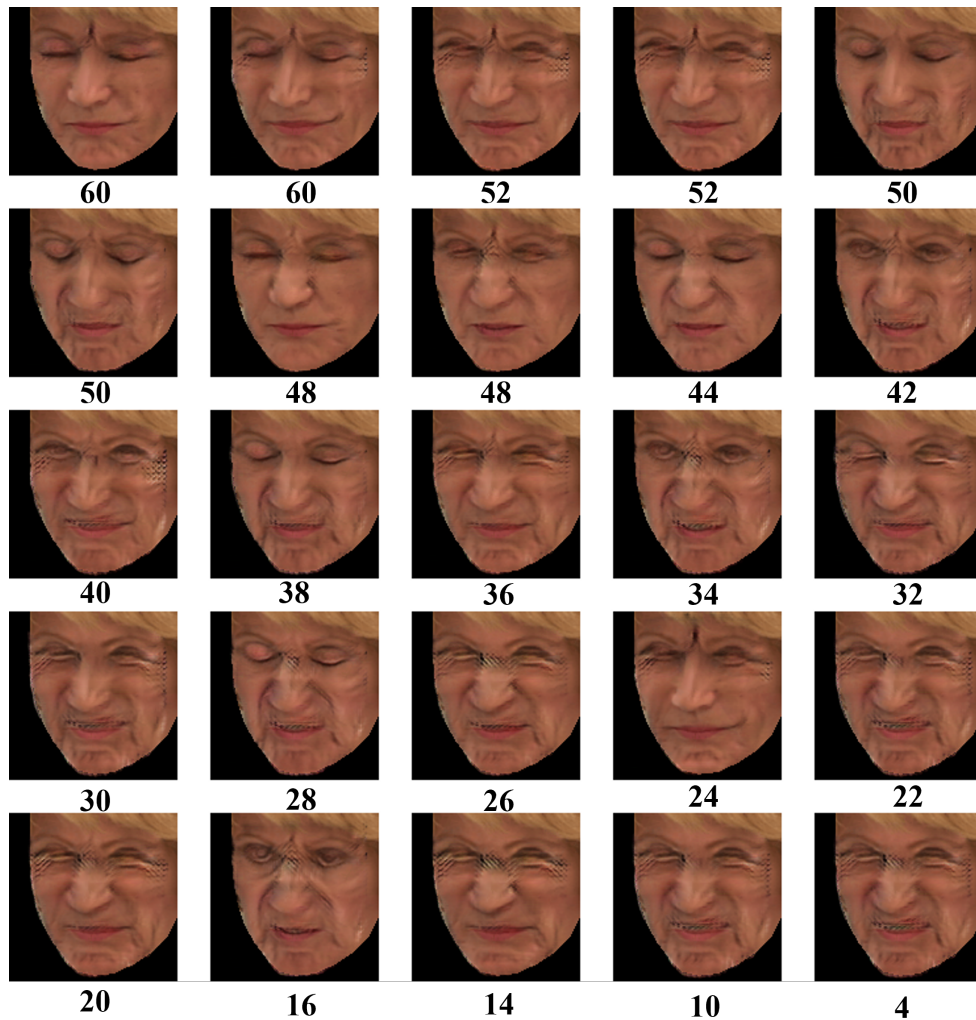


Figure 38: The generated images with a different score.

From a visual perspective, it can be seen that as the score decreases, there

are simultaneously decreases in the quality of the generated image. We select scores above 35 as the threshold for viable Pre-determined Lists. Table 7 is the calculated number of Pre-determined Lists where the score (S) exceeds 35 within each pain level.

Pain Level	1	2	3	4	5	6	7	8
Number	8	33	96	225	454	796	1211	1625
Pain Level	9	10	11	12	13	14	15	16
Number	1896	1887	1634	1212	693	271	58	0

Table 7: Selected Pre-determined Lists

In conclusion, we consider the pain expressions generated from these pre-determined lists to meet the required quality standards for our study. Therefore, the outputs from these lists were chosen for the next phase of database generation.

4.3 Painful Dataset Augmentation

Having presented the strategy for generating diverse pain expressions, we first give the detailed implementation of generate a sufficient and balanced painful dataset by using the fine-tuned GANimation network.

In this project, we aim to augment the UNBC-McMaster dataset. Correspondingly, we trained our fine-tuned GANimation network on the same dataset to achieve the within-domain generation. In this case, we need to make sure that the fine-tuned GANimation network only generated images of the five individuals included in the test set in order to prevent any bias. To perform the data augmentation on the entire dataset, images from all 25 individuals need to be generated. Therefore, we **fine-tune the GANimation five times**.

For each run, we randomly selected five individuals for the test set, and the remaining 20 were used as the training set. Table 8 gives 5 different test sets and here we assigned numerical identifiers from 1 to 25 to each individual. The GANimation network was then fine-tuned using the different versions of training sets. As a result, we obtain five versions of the fine-tuned network.

Version 1	2	4	12	16	24
Version 2	3	5	8	9	20
Version 3	7	10	18	21	23
Version 4	1	6	13	19	22
Version 5	11	14	15	17	25

Table 8: Test data of each version of fine-tuned network

To start with, we extract the data for each person at a pain level of 0, as defined by the database labels. Then, with the above-mentioned five versions fine-tuned GANimation network, we can generate pain expression for every individual based on the scoring mechanism.

With the scoring mechanism, we can have a sufficiently diverse dataset. However, the project also requires for balanced across various pain levels. As such, we choose to generate the same number of images n per individual, for each pain level, ensuring a uniform distribution of data across the entire pain dataset. If the number of Pre-determined lists for each pain level smaller than n , we just randomly selected 100 Pre-determined lists for each pain level to generate the corresponding images. Otherwise, we can choose more than one image as input, and use all available Pre-determined lists to generate the images. After that we randomly select n images for each pain level. Through this method, we can successfully create a sufficiently diverse and balanced dataset, effectively addressing the previously existing problems.

Notice, as we mentioned in section 4.1.2, the residual features remaining after pre-processing may also have some negative impact the test set. To further discussed, we have two types of input images format in this context. One type

is the images with 2D alignment, and the other comprises images with 3D registration. A detailed discussion on the performance of these two types of images with the GANimation network will be provided in the section [5.2](#).

4.4 Painful Dataset Evaluation

Having completed the data augmentation process, this part gives a comprehensive evaluation strategy for the augmented dataset. This evaluation strategy is significant as it confirms the robustness, applicability, and potential value of the generated data in the APA field.

Fig.39 provides the overall evaluation pipeline.

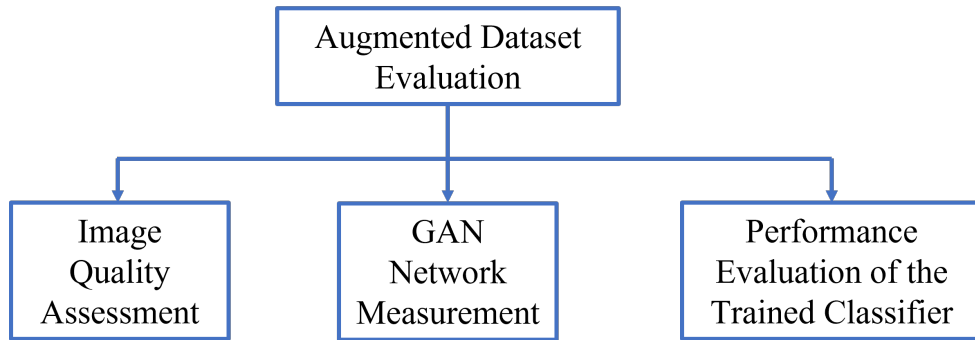


Figure 39: The overall evaluation pipeline

It can be divided into three key stages:

- **Image Quality Assessment:**

As the first step in the evaluation process, our goal is to measure the quality of the images generated during the data augmentation process. This evaluation is to ensure that the images are of sufficiently high quality. Our approach combines both qualitative and quantitative research methods.

In the qualitative evaluation, we use **visual assessments** which provide us with an immediate, intuitive grasp of the quality of generated images. Factors like clarity and realism can be considered here. Then, to complement and balance the subjective nature of the qualitative evaluation (visual assessment), we introduce a quantitative metric, namely the **Cumulative Probability of Blur Detection (CPBD)**. It is a no-reference image blur metric, which is derived from research into human perception of blur at different contrast values. It is an image quality metric that matches the characteristics of human vision. Larger values on the CPBD scale reflect clearer detail and less blur, allowing us to quantify the quality of sharpening in filtered images. By using CPBD, we ensure a comprehensive, multifaceted evaluation of the quality of generated images.

- **GAN-based Network Measurement:**

After the image quality assessment, we can focus on the evaluation of the GAN network itself. Here, we use **the Fréchet Inception Distance (FID)**.

As we mentioned in section 2.4, it is a popular metric used in the evaluation of generative models, particularly Generative Adversarial Networks (GANs). It compares the distribution of generated images with the distribution of a set of real images (**Ground Truth**) [95]. A low FID score means that the distribution of the generated image is more similar to the real image, indicating that the generated image is very similar to the real image. An FID score of zero means that the two distributions are the same, though this is rarely achieved in practice due to the inherent complexity and variability of image data.

In sum, the FID score can provide a measure of the similarity between the distribution of the generated images and the distribution of the original dataset, offering a precise insight into the ability of the GAN network to reproduce the original data feature.

- **Performance Evaluation of the Trained Classifier:**

The final stage of our evaluation process assesses the potential of the enhanced dataset for practical applications. For this purpose, we trained a simple classifier for the APA task using an augmented dataset. We then analyze the performance of this classifier to understand how the augmented dataset potentially contributes to improving existing APA approaches. This allows us to evaluate whether our data augmentation work produces datasets that are effective in improving model performance and generalisation in real-world environments.

In sum, our evaluation strategy is designed not only to validate the quality and effectiveness of the generated dataset but also to highlight areas of potential improvement and further research of current APA approaches.

5 Experiment Results

This section provides the corresponding experiment results following the instruction of Section 4.4.

5.1 Image Quality Assessment

For the first experiment, a critical goal was to assess the quality of image generation and then to validate the generation quality of the fine-tuned GANimation network.

We start by using a sample of test images with 2D alignment. Specifically, we randomly select one individual from each of the five different versions of the fine-tuned GANimation network as the test subject. For each of these subjects, we generate 100 pain images corresponding to each pain level (ranging from 1-15) by using the scoring mechanism for each pain level(1-15). This process will be repeated twice: once with the GANimation network without finetuning and once with the finetuning GANimation network.

In section 4.1.4, we have already presented a qualitative measure of image quality through a comparative visual inspection. It shows that the images generated by the fine-tuned network appear sharper and more realistic than those generated before the fine-tuned. Then, we use CPBD to provide a quantitative measure of image quality. In specific, We calculate the CPBD values for each generated image for each test subject and then compute the average. This effectively captures the average generated image quality for each object.

The table 9 summarises our experimental results, providing a clear comparison between the GANimation network with and without finetuning.

Test subjects	16	8	18	33	22
Without fine-tuning	0.273	0.386	0.325	0.230	0.340
With fine-tuning	0.318	0.419	0.347	0.256	0.378

Table 9: The average CPBD value

Across all test subjects, the images generated by the finetuned network consistently exhibit higher average CPBD values than those generated without finetuning. This increase in CPBD values can be directly translated to an enhancement in image quality: higher CPBD values indicate clearer detail and less blur in the produced images. This observation is also consistent with our qualitative (visual) assessment.

In conclusion, our findings provide compelling evidence that fine-tuning the GANimation network significantly enhances the quality of the generated images.

5.2 GAN-based Network Measurement

The second part of our experiment is dedicated to the implementation of the GAN-based Network Measurement, which is a necessary step to assess the quality and diversity of the generated datasets.

FID is a Full Reference (FR) metric, which measures the similarity between the real dataset and generated dataset. Therefore, for each individual, we define two groups: Reference group F_r , representing the ground truth, and Generated group F_g .

- F_r : This group includes all images of pain level m_i (except for pain level 0) from the original dataset.
- F_g : For each level m_i , generated 100 painful images based on the scoring mechanism .

In this experiment, we first randomly select a fine-tuned version of the GAN-imation model for testing purposes. Then, for each test subject, we choose the same image after 2D alignment and 3D registration, respectively. Here, in order to control the variables, all images in the FR group are processed in the form corresponding to their input (either 2D or 3D).

The table presented below provides FID values for different types of input.

Test subjects	2D aligned	3D aligned
2	67.676	40.461
4	49.099	42.499
12	72.905	59.733
16	66.376	65.797
24	72.905	41.308

Table 10: The FID value of finetuned GANimation version 1

By checking Table 10, we observe that for a given GAN network, the distribution of the generated data of the 3D aligned input image is more closely aligned with the original distribution compared to the 2D aligned ones. This result could indicate that the use of 3D registration could produce a more realistic dataset which can better reflect the distribution of images in the original dataset. Thus, we can say 3D registration is more effective than 2D alignment in generating a more authentic distribution of images.

5.3 Performance Evaluation of the Trained Classifier

For the final experiment, we focus on the performance analysis of the trained classifier. This analysis aims to provide an understanding of how our generated dataset can contribute to enhancing the existing APA approaches.

In this task, we pool PSPI scores into a simplified 6-point scale instead of 16 levels of classification. The scale ranges from 0 (indicating no pain) to 5 (signifying strong pain), making the task more practical and manageable.

In terms of data partitioning for training and testing purposes, we adopt the widely-used 80:20 ratio. This ratio corresponds to using 80% of the data for training the model and the remaining 20% for testing its performance. Given a total of 25 individuals in the dataset, this division results in 20 individuals' data used for training and 5 for testing. The selection of these 5 testing datasets is done randomly to ensure unbiased results. Table 11 and table 12 provide a detailed look at the actual distribution of the training and testing sets separately.

The Class	Label	PSPI Score	Number of Images
0	No Pain	0	33112
1	Mild Pain	1-3	5470
2	Moderate Pain	4-6	854
3	Severe Pain	7-9	87
4	Very severe pain	10-12	148
5	Strong pain	>13	18

Table 11: The distribution of the training set in the original dataset

The Class	Label	PSPI Score	Number of Images
0	No Pain	0	6862
1	Mild Pain	1-3	1200
2	Moderate Pain	4-6	459
3	Severe Pain	7-9	77
4	Very severe pain	10-12	43
5	Strong pain	>13	10

Table 12: The distribution of the testing set in the original dataset

In this experiment, the generated dataset consists of 100 images for each individual, per pain level.

As for the classification model, we choose the network from the Off-the-Shelf CNN. In particular, we use ConvNeXtSmall [108], a model from Keras Applications with pre-trained weights [109]. This decision is motivated by the robust performance demonstrated by these architectures across various vision tasks, as evidenced by the results in the ImageNet Large Scale Visual Recognition Challenge [110]. Such CNNs, trained on millions of images for object

category classification, are remarkably suitable for direct pain assessment applications. Specifically, we used the pre-trained ConvNeXtSmall model as the base and added custom layers: a Global Average Pooling2D layer to transform the feature maps into individual vectors, a dense layer with 1024 neurons for deeper abstraction, and a final dense layer with 6 neurons for 6-category (0-5 pain levels) classification.

As for the training setting, we used the Adam optimizer with a 0.0001 learning rate, a categorical cross-entropy loss function for compilation, and tracked accuracy as a metric during training and validation. The model was trained for 12 epochs using training data with a batch size of 128.

We defined two groups for comparison: the control group used the original 3D-registered dataset (original dataset), and the experiment group used the generated dataset for training (generated dataset). Both groups used the same split for datasets. Notice, here the test set of the experiment group is the same as the control group instead of the generated date.

The plots of model accuracy and loss for both training sessions are as follows:

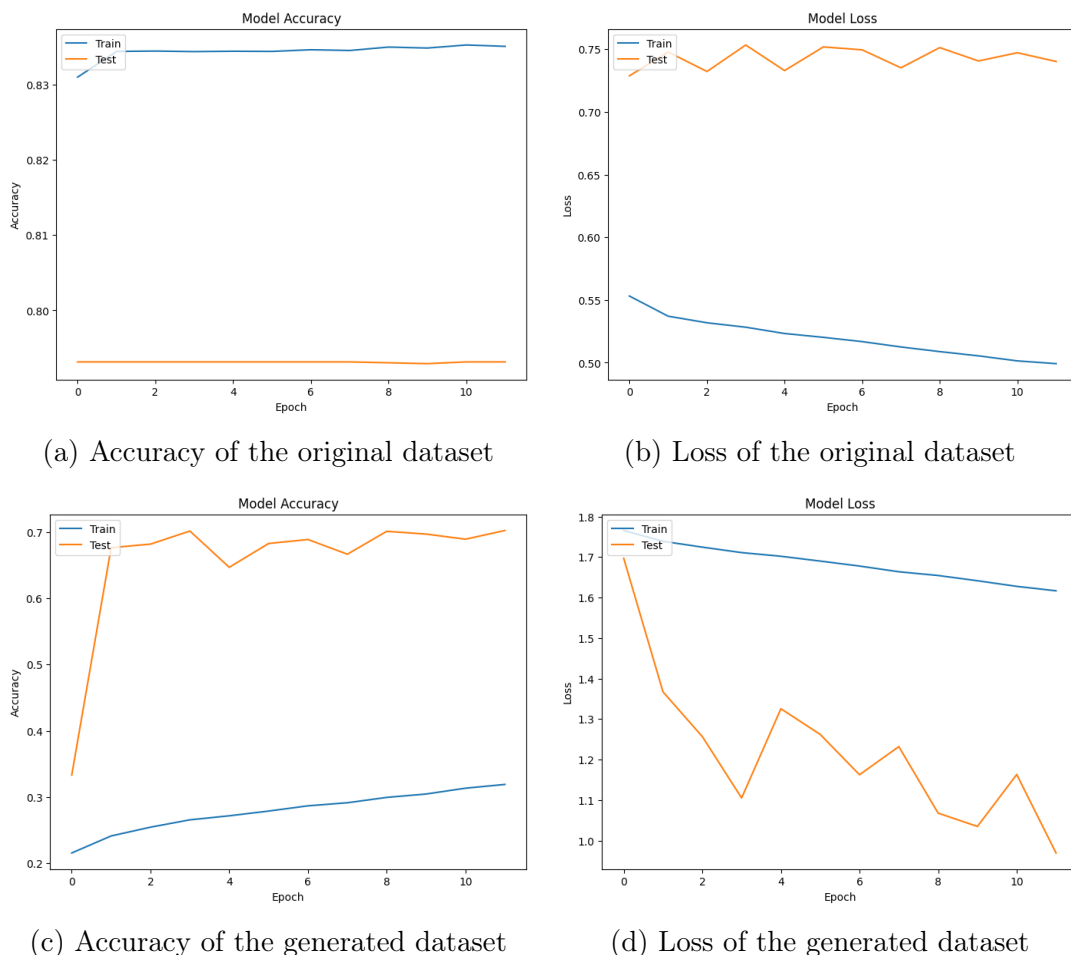


Figure 40: Model accuracy and loss during training

The results for both the generated dataset and original dataset are presented in the table below.

Class	Precision	Recall	F1 score	Support
0	0.79	1	0.88	6867
1	0	0	0	1202
2	0	0	0	459
3	0	0	0	77
4	0	0	0	43
5	0	0	0	10
Accuracy			0.79	8658
Macro avg	0.13	0.17	0.15	8658
Weighted avg	0.63	0.79	0.7	8658

Table 13: The result of the original dataset

Class	Precision	Recall	F1 score	Support
0	0.79	0.87	0.83	6867
1	0.11	0.1	0.11	1202
2	0	0	0	459
3	0	0	0	77
4	0	0	0	43
5	0	0	0	10
Accuracy			0.7	8658
Macro avg	0.15	0.16	0.16	8658
Weighted avg	0.64	0.7	0.67	8658

Table 14: The result of the generated dataset

A comparison of these results reveals that the accuracy for class 0 is roughly the same between the two groups. However, for class 1, the experimental group outperformed the original dataset in terms of both accuracy and recall, suggesting an enhanced ability to identify class 1 labels.

Specifically, despite the original dataset showing a higher overall accuracy (0.79) compared to the generated dataset (0.70), this could potentially be misleading. Since the original dataset appears to classify most instances into class 0, this may lead to high accuracy but weak performance for class 1. In contrast, the generated dataset attempted to recognize instances from class 1 and possibly others, which might have resulted in a slightly reduced overall accuracy but improved performance on other metrics. Furthermore, while the original dataset achieved a recall of 1 for class 0, the generated dataset provided a better balance between accuracy and recall, producing a higher F1 score (0.83 instead of 0.88).

In summary, the generated dataset showed better performance in identifying class 1 instances and maintained a balanced performance in class 0. However, both groups showed room for improvement in recognising the rest of the classes. A more comprehensive analysis is provided in the section [6.2](#).

6 Discussion and Conclusion

In this section, we re-examine the research questions, summarise the findings and describe the experimental results in detail. We then discuss the analysis of potential enhancement pathways based on the experimental results.

6.1 Review of Research Questions

In our project, the primary research question is: **How can the application of GANs be utilized to generate synthetic but realistic facial pain expressions, with the goal of effectively augmenting the existing dataset?**

We explore this research question by conducting three distinct but related sub-research questions, which helped design and evaluate the effectiveness of the dataset augmentation scheme and assess its potential to bolster the performance of existing APA approaches.

Our first research question focuses on the quality of GAN-based generated images: **Can the dataset augmentation scheme generate realistic synthetic painful expressions?** This question is essential since the ability to generate such images of painful expressions is the foundation of its practical application.

In the initial implementation of GANimation based on the 'in-the-wild' EmotionNet dataset, we get unsatisfactory results. Further investigation shows it is caused by two main factors: Domain Variations and Residual features post-processing. To overcome these two challenges and to produce more realistic and higher quality images of painful expressions, we proposed two strategies: 1) fine-tuning the GANimation network using the UNBC-McMaster dataset, and 2) processing the images from the UNBC database with 3D registration, in order to realize the 'within-domain generation'.

Compared the performance of those two versions of GANimation - one with fine-tuning and the other without. Both qualitative visual assessments and quantitative CPBD measurements validate the quality of the generated images. The experimental results corroborate the effectiveness of the proposed solution in generating satisfactory-quality images, thereby validating that the GAN-based augmentation scheme can generate realistic synthetic painful expressions.

The second research question is **How do 2D and 3D face registration before pain expression generation impact the performance of the model?** Compared with the first research questions which focused on the image perspective, this one is more concerned with the dataset perspective. Essentially, as a dataset augmentation method, we need to produce a dataset of synthetic images that are not only high quality but also sufficiently diverse.

To achieve this, we designed an innovative scoring mechanism to generate a balanced and sufficiently diverse dataset with high-quality images. Besides, we also make an assumption that the 3D registration before pain expression

generation can have a positive impact on the performance of the GANimation model. In the experiment part, we employed the Fréchet Inception Distance (FID) to measure the similarity between the distribution of the generated and original datasets. Our experiments confirm this hypothesis by demonstrating that the images generated after 3D alignment not only maintain a high quality and diversity but also show a similar distribution to the original database, thus validating our scoring mechanism.

The last research question highlights the application part: **How does the potential impact of the synthetic pain expression dataset on the performance of existing APA approaches?** Through this question, we aim to show the application of the generated dataset in real-world scenarios, particularly in training a machine learning classifier for the APA field.

To explore this, we trained a convolutional neural network (CNN) classifier using the synthetic dataset. In the experimental section, we evaluate this trained classifier using Precision, Recall and F1 scores. The results show an improvement in the ability to identify certain pain categories compared to the original model, especially in under-represented categories. Although the performance of the CNN classifier did not reach the level we had expected, it provides a promising baseline of the potential for improving the learning-based APA approach.

6.2 Limitations

Our research has produced promising results, but the performance of the trained classifier did not reach what we initially expected. In this section, we identify several factors that may have contributed to this variation.

6.2.1 Label Discrepancy

We think one of the primary factors that could account for this variation is the mismatch between manual labels provided in the original dataset and the automated annotations generated by OpenFace.

As we mentioned in 2.3, the creation of annotated databases based on the Facial Action Coding System (FACS) requires significant time and effort. This leads to a lack of large databases of facial expressions with FACS annotations. To address this problem, we use the annotation provided by the OpenFace in the training of the GANimation network, which provides a fully automated solution for AU annotations. However, this approach posed specific challenges in the following part of the research. See Fig 41 as an example of these discrepancy.



Figure 41: Example image

In the manual annotation, Table 15 gives the AUs intensities. And a pain level of 10 (PSPI score) is calculated based on them. However, the AU intensities generated by OpenFace are quite apparent.

	AU04	AU06	AU07	AU09	AU10	AU43	Level
Manual annotation	5	4	0	0	0	1	10
OpenFace annotation	2	1	1	0	1	1	5

Table 15: The annotation AU labels

This discrepancy causes a dilemma of 'ground truth'. Since our aim is to generate painful facial expressions by manipulating various AU intensities, which

means that the annotations (AU intensities and pain levels) are based on OpenFace. However, manual annotations are typically considered the most reliable ground truth because they were carefully annotated by experts. Therefore, in the testing part of the classifier, we use the manual label as the ground truth. Ideally, both manually and automatically generated AU labels should align. However, we found a considerable discrepancy between them. This discrepancy resulted in a considerably different in the generated training data distribution compared to the actual distribution observed during the testing phase.

In short, the classifier is trained using OpenFace labels as the ground truth, but these labels do not accurately reflect the actual ground truth, i.e., manual labels, during the testing phase. This can lead to discrepancies between the training and testing scenarios, which in turn can have a negative impact on the performance of the classifier.

6.2.2 Computational Limitations

Another potential limitation lies in the training duration of the classifier. In specific, our classifier was trained for only 12 epochs, considerably fewer than many similar studies. For instance, E.Morabit et al. [111] use 200 epochs to train an Off-the-Shell CNN classifier. The reason for our shorter training duration is our research was conducted on Google Colab, which has certain computational limitations. We think these kinds of limitations may limit the training of the classifier, thereby influencing the quality of results.

6.2.3 Test Dataset Volume

Our test dataset may not be sufficient to accurately assess the effectiveness of the classifier.

Even though we performed data augmentation on the training dataset, the test dataset we use in the classifier was directly from the original dataset without any augmentation. In other words, the test dataset still suffers from the issues of insufficient data volume and imbalanced label problems. This implies that our test dataset might not have been sufficiently comprehensive or diverse to provide an accurate evaluation of the classifier’s effectiveness.

6.3 Future work

Looking forward, we believe that there is still room for improvement in the proposed method. As we mentioned in the previous section, one significant challenge that emerged during our research was the Label Discrepancy between OpenFace-generated and manual labels. To overcome this challenge, we think the following strategies could be the future exploration.

- **Mapping Between OpenFace and Manual Labels**

One approach could be trying to find the correlation or mapping between OpenFace-generated labels and manual labels. This mapping could help to bridge the gap between these two sets of labels, potentially improving the model's performance by ensuring a more consistent ground truth across both training and testing scenarios.

- **Use of Manual Labels in Fine-tuning**

Another approach is to use manual labels during fine-tuning, rather than OpenFace-generated labels. We think this may align the training and testing scenarios better.

Besides, the integration of more diverse and comprehensive pain datasets will always be a possible solution. Such datasets will not only help to train the GAN networks but also provides a more sufficient set of test cases for the classifier.

In sum, we think that future work could be more focused on mitigating the problem of discrepancy between labels as well as introducing more painful datasets.

6.4 Conclusion

In conclusion, this research has presented an innovative method that uses GANs for creating a balanced, and sufficiently diverse synthetic dataset of painful facial expressions.

The study introduced a novel method, which uses the GANimation network to control the magnitude of activation of each Action Unit (AUs) so that the desired expressions can be synthesised. A unique approach was then employed for fine-tuning GANimation network based on the UNBC-McMaster dataset, which facilitated the within-domain generation and reduced domain variance. Moreover, we provide a 3D registration technique in both the training and testing phases, which can effectively manage the head-range problems. Finally, a scoring mechanism was introduced to facilitate the generation of balanced, and sufficiently diverse pain expressions through the fine-tuning GANimation networks.

The experiment result shows the effectiveness of the proposed method, highlighting the quality and potential values of improving the current APA approaches.

Appendix A Appendix A: Generation Results Comparison

- Test subject: AK064 Input test image:

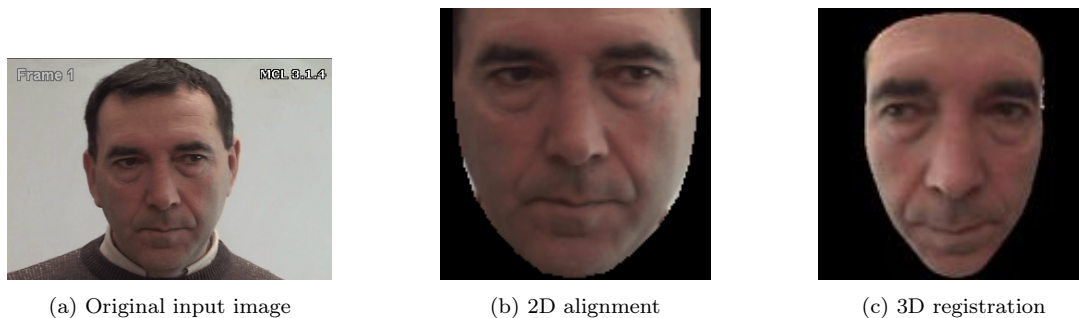


Figure 42: Test Input of AK064.

Fig.43 is the result with a 2D alignment image as input:

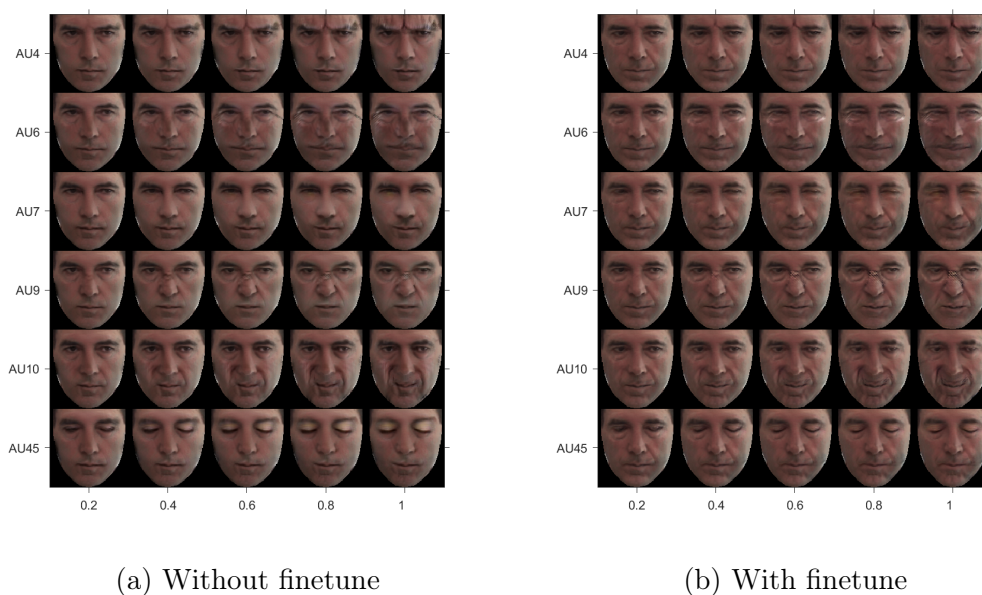
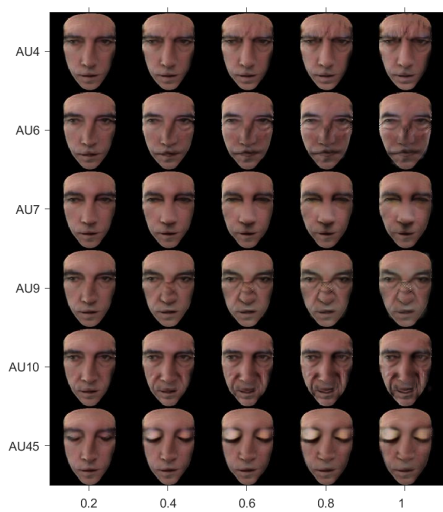
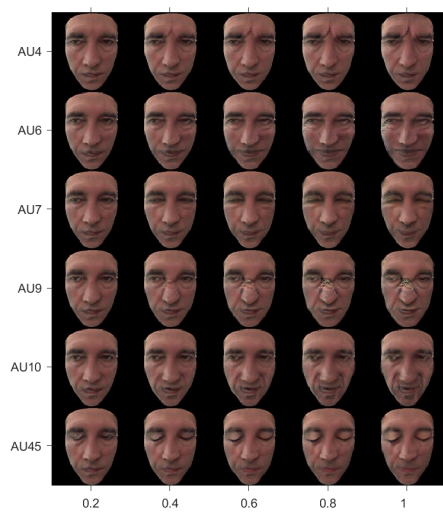


Figure 43: Comparison of generated results (2D alignment)

Fig.44 is the result with a 3D registration image as input.



(a) Without finetune



(b) With finetune

Figure 44: Comparison of generated results (3D registration)

- **Test subject: IB109** Input test image:

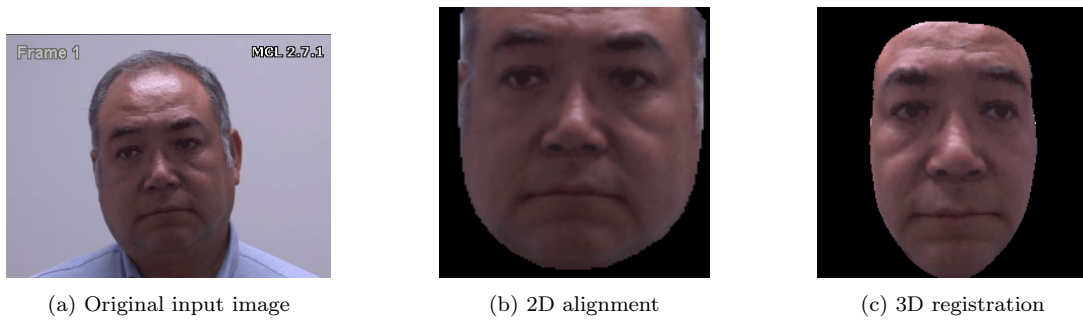
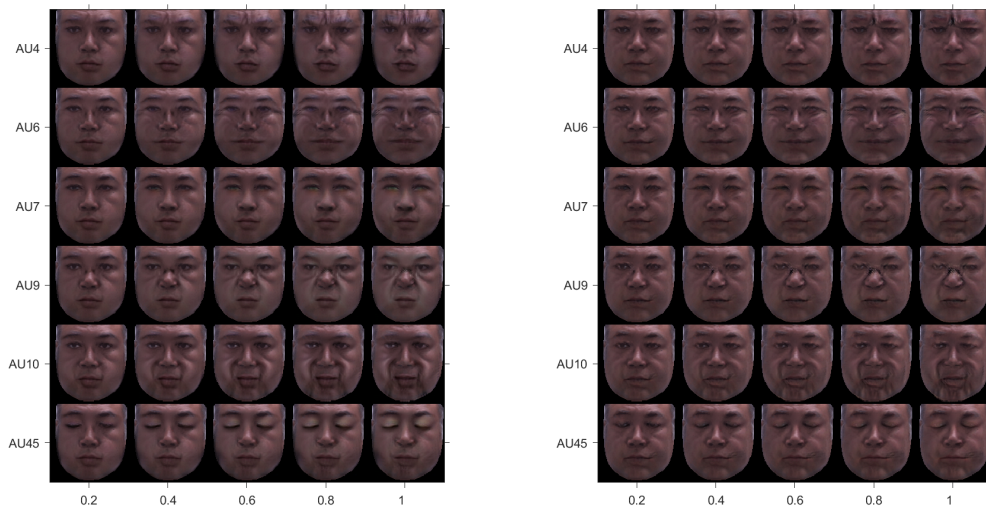


Figure 45: Test Input of IB109.

Fig.46 is the result with a 2D alignment image as input:

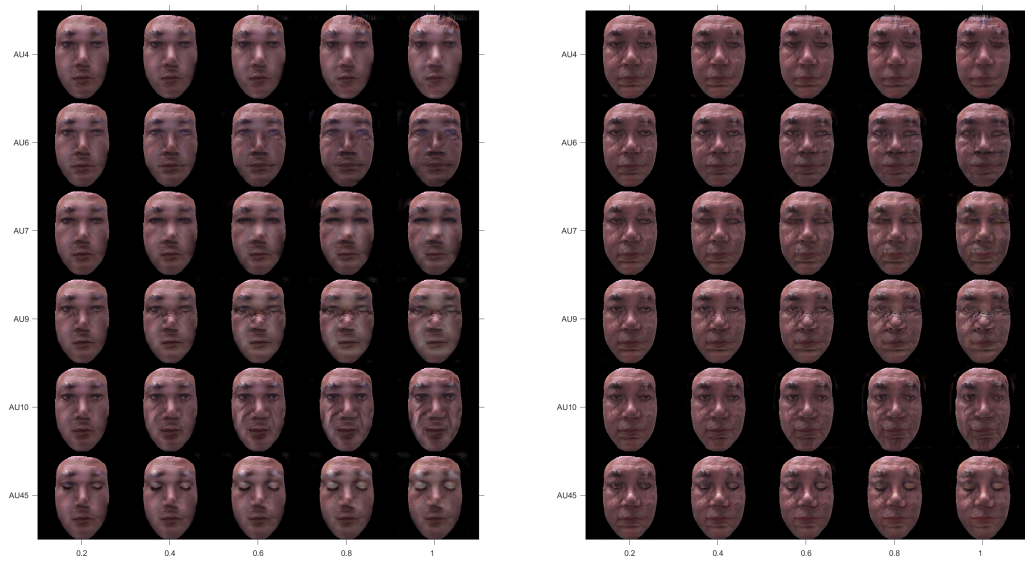


(a) Without finetune

(b) With finetune

Figure 46: Comparison of generated results (2D alignment)

Fig.47 is the result with a 3D registration image as input.



(a) Without finetune

(b) With finetune

Figure 47: Comparison of generated results (3D registration)

- **Test subject: TV095** Input test image:



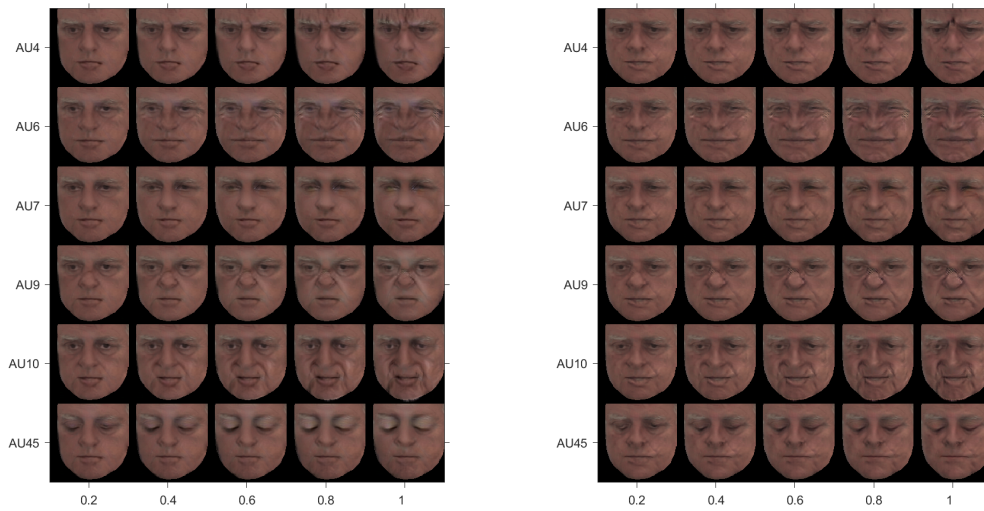
(a) Original input image



(b) 2D alignment

Figure 48: Input test image of TV095

Fig.49 is the result with a 2D alignment image as input:

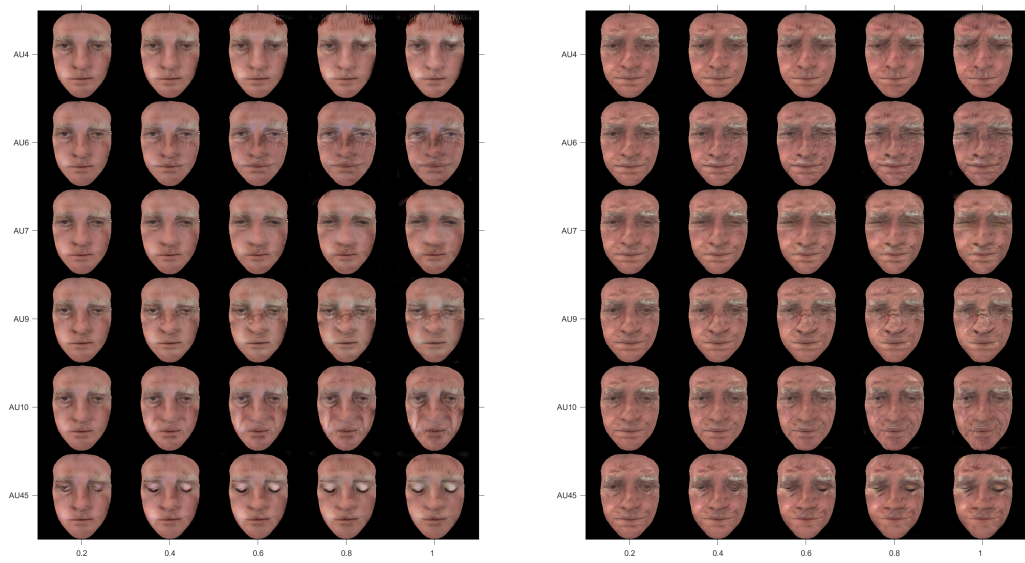


(a) Without finetune

(b) With finetune

Figure 49: Comparison of generated results (2D alignment)

Fig.50 is the result with a 3D registration image as input.



(a) Without finetune

(b) With finetune

Figure 50: Comparison of generated results (3D registration)

References

- [1] A. Caraceni and M. Shkodra, “Cancer pain assessment and classification,” *Cancers*, vol. 11, no. 4, p. 510, 2019.
- [2] H. Breivik, P.-C. Borchgrevink, S.-M. Allen, *et al.*, “Assessment of pain,” *BJA: British Journal of Anaesthesia*, vol. 101, no. 1, pp. 17–24, 2008.
- [3] H. E. Merskey, “Classification of chronic pain: Descriptions of chronic pain syndromes and definitions of pain terms,” *Pain*, 1986.
- [4] K. Ho, J. Spence, and M. F. Murphy, “Review of pain-measurement tools,” *Annals of emergency medicine*, vol. 27, no. 4, pp. 427–432, 1996.
- [5] M. McCaffery, *Nursing practice theories related to cognition, bodily pain, and man-environment interactions*. University of California Print. Office, 1968.
- [6] S. Merkel, T. Voepel-Lewis, and S. Malviya, “Pain control: Pain assessment in infants and young children: The flacc scale,” *The American journal of nursing*, vol. 102, no. 10, pp. 55–58, 2002.
- [7] C. Pasero and M. McCaffery, *Pain Assessment and Pharmacologic Management-E-Book*. Elsevier Health Sciences, 2010.
- [8] R. S. Morrison and A. L. Siu, “A comparison of pain and its treatment in advanced dementia and cognitively intact patients with hip fracture,” *Journal of pain and symptom management*, vol. 19, no. 4, pp. 240–248, 2000.
- [9] K. A. Puntillo, A. B. Morris, C. L. Thompson, J. Stanik-Hutt, C. A. White, and L. R. Wild, “Pain behaviors observed during six common procedures: Results from thunder project ii,” *Critical care medicine*, vol. 32, no. 2, pp. 421–427, 2004.
- [10] C. Gélinas, L. Fillion, K. A. Puntillo, C. Viens, and M. Fortier, “Validation of the critical-care pain observation tool in adult patients,” *American Journal of Critical Care*, vol. 15, no. 4, pp. 420–427, 2006.
- [11] J. Young, J. Siffleet, S. Nikoletti, and T. Shaw, “Use of a behavioural pain scale to assess pain in ventilated, unconscious and/or sedated patients,” *Intensive and Critical Care Nursing*, vol. 22, no. 1, pp. 32–39, 2006.
- [12] K. Herr, K. Bjoro, and S. Decker, “Tools for assessment of pain in nonverbal older adults with dementia: A state-of-the-science review,” *Journal of pain and symptom management*, vol. 31, no. 2, pp. 170–192, 2006.
- [13] K. D. Craig, K. M. Prkachin, and R. E. Grunau, “The facial expression of pain.” 2011.

- [14] P. Werner, D. Lopez-Martinez, S. Walter, A. Al-Hamadi, S. Gruss, and R. W. Picard, “Automatic recognition methods supporting pain assessment: A survey,” *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 530–552, 2022. DOI: [10.1109/TAFFC.2019.2946774](https://doi.org/10.1109/TAFFC.2019.2946774).
- [15] G. Bargshady, X. Zhou, R. C. Deo, J. Soar, F. Whittaker, and H. Wang, “Enhanced deep learning algorithm development to detect pain intensity from facial expression images,” *Expert Systems with Applications*, vol. 149, p. 113305, 2020.
- [16] K. D. Craig, “The social communication model of pain.,” *Canadian Psychology/Psychologie canadienne*, vol. 50, no. 1, p. 22, 2009.
- [17] K. D. Craig, “The facial expression of pain better than a thousand words?” *APS Journal*, vol. 1, no. 3, pp. 153–162, 1992.
- [18] T. Hassan, D. Seuß, J. Wollenberg, *et al.*, “Automatic detection of pain from facial expressions: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1815–1831, 2021. DOI: [10.1109/TPAMI.2019.2958341](https://doi.org/10.1109/TPAMI.2019.2958341).
- [19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
- [20] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [21] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [23] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Synthetic data augmentation using gan for improved liver lesion classification,” in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, IEEE, 2018, pp. 289–293.
- [24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [25] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan, “Geometry guided adversarial facial expression synthesis,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 627–635.

- [26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.
- [27] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, “Invertible conditional gans for image editing,” *arXiv preprint arXiv:1611.06355*, 2016.
- [28] S. Gibson, B. Katz, T. Corran, M. Farrell, and R. Helme, “Pain in older persons,” *Disability and Rehabilitation*, vol. 16, no. 3, pp. 127–139, 1994.
- [29] T. Donvito. “How to use a pain scale to assess your pain.” (2021), [Online]. Available: <https://anesthesiaexperts.com/uncategorized/pain-scale-assess-pain/> (visited on 03/22/2021).
- [30] S. J. Closs, B. Barr, M. Briggs, K. Cash, and K. Seers, “A comparison of five pain assessment scales for nursing home residents with varying degrees of cognitive impairment,” *Journal of pain and symptom management*, vol. 27, no. 3, pp. 196–205, 2004.
- [31] H. Wang, J. Yang, and P. Li, “How and when goal-oriented self-regulation improves college students’ well-being: A weekly diary study,” *Current psychology*, vol. 41, no. 11, pp. 7532–7543, 2022.
- [32] R. Van Herk, M. Van Dijk, F. P. Baar, D. Tibboel, and R. De Wit, “Observation scales for pain assessment in older adults with cognitive impairments or communication difficulties,” *Nursing Research*, vol. 56, no. 1, pp. 34–43, 2007.
- [33] R. H. Gracely, P. McGrath, and R. Dubner, “Ratio scales of sensory and affective verbal pain descriptors,” *Pain*, vol. 5, no. 1, pp. 5–18, 1978.
- [34] K. Shannon and T. Bucknall, “Pain assessment in critical care: What have we learnt from research,” *Intensive and critical care nursing*, vol. 19, no. 3, pp. 154–162, 2003.
- [35] E. E. Benarroch, “Pain-autonomic interactions: A selective review,” *Clinical Autonomic Research*, vol. 11, no. 6, pp. 343–349, 2001.
- [36] B. Kidd, S. Cruwys, P. Mapp, and D. Blake, “Role of the sympathetic nervous system in chronic joint pain and inflammation,” *Annals of the rheumatic diseases*, vol. 51, no. 11, p. 1188, 1992.
- [37] R. Logier, R. Jounwaz, R. Vidal, M. Jeanne, *et al.*, “From pain to stress evaluation using heart rate variability analysis: Development of an evaluation platform,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, IEEE, 2010, pp. 3852–3855.
- [38] D. Harrison, S. Boyce, P. Loughnan, P. Dargaville, H. Storm, and L. Johnston, “Skin conductance as a measure of pain and stress in hospitalised infants,” *Early human development*, vol. 82, no. 9, pp. 603–608, 2006.

- [39] M. Oliveira, A. R. Machado, V. Chagas, T. C. Granado, A. A. Pereira, and A. O. Andrade, "On the use of evoked potentials for quantification of pain," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2012, pp. 1578–1581.
- [40] R.-R. Nir, A. Sinai, E. Raz, E. Sprecher, and D. Yarnitsky, "Pain assessment by continuous eeg: Association between subjective perception of tonic pain and peak frequency of alpha oscillations during stimulation and at rest," *Brain research*, vol. 1344, pp. 77–86, 2010.
- [41] A. Marquand, M. Howard, M. Brammer, C. Chu, S. Coen, and J. Mourão-Miranda, "Quantitative prediction of subjective pain intensity from whole-brain fmri data using gaussian processes," *Neuroimage*, vol. 49, no. 3, pp. 2178–2189, 2010.
- [42] A. C. d. C. Williams, "Facial expression of pain: An evolutionary account," *Behavioral and brain sciences*, vol. 25, no. 4, pp. 439–455, 2002.
- [43] P. Lucey, J. F. Cohn, I. Matthews, *et al.*, "Automatically detecting pain in video through facial action units," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 664–674, 2010.
- [44] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, S. Chew, and I. Matthews, "Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database," *Image and Vision Computing*, vol. 30, no. 3, pp. 197–205, 2012.
- [45] C. Wu, S. Wang, and Q. Ji, "Multi-instance hidden markov model for facial expression recognition," in *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, IEEE, vol. 1, 2015, pp. 1–6.
- [46] J. Zhou, X. Hong, F. Su, and G. Zhao, "Recurrent convolutional neural network regression for continuous pain intensity estimation in video," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 84–92.
- [47] G. Bargshady, J. Soar, X. Zhou, R. C. Deo, F. Whittaker, and H. Wang, "A joint deep neural network model for pain recognition from face," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, 2019, pp. 52–56.
- [48] G. Bargshady, X. Zhou, R. C. Deo, J. Soar, F. Whittaker, and H. Wang, "Ensemble neural network approach detecting pain intensity from facial expressions," *Artificial Intelligence in Medicine*, vol. 109, p. 101954, 2020.
- [49] K. M. Prkachin, "Assessing pain by facial expression: Facial expression as nexus," *Pain Research and Management*, vol. 14, no. 1, pp. 53–58, 2009.
- [50] P. Ekman and W. V. Friesen, "Facial action coding system," *Environmental Psychology & Nonverbal Behavior*, 1978.

- [51] R. Zhi, M. Liu, and D. Zhang, “A comprehensive survey on automatic facial action unit analysis,” *The Visual Computer*, vol. 36, pp. 1067–1093, 2020.
- [52] M. Kunz, S. Scharmann, U. Hemmeter, K. Schepelmann, and S. Lautenbacher, “The facial expression of pain in patients with dementia,” *PAIN®*, vol. 133, no. 1-3, pp. 221–228, 2007.
- [53] Z. Hammal and J. F. Cohn, “Automatic detection of pain intensity,” in *Proceedings of the 14th ACM international conference on Multimodal interaction*, 2012, pp. 47–52.
- [54] P. Werner, D. Lopez-Martinez, S. Walter, A. Al-Hamadi, S. Gruss, and R. W. Picard, “Automatic recognition methods supporting pain assessment: A survey,” *IEEE Transactions on Affective Computing*, vol. 13, no. 1, pp. 530–552, 2019.
- [55] K. M. Prkachin and P. E. Solomon, “The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain,” *Pain*, vol. 139, no. 2, pp. 267–274, 2008.
- [56] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 international interdisciplinary PhD workshop (IIPhDW)*, IEEE, 2018, pp. 117–122.
- [57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [58] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [60] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [61] S. Brahnham, C.-F. Chuang, R. S. Sexton, and F. Y. Shih, “Machine assessment of neonatal facial expressions of acute pain,” *Decision Support Systems*, vol. 43, no. 4, pp. 1242–1254, 2007.
- [62] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, “Painful data: The unbc-mcmaster shoulder pain expression archive database,” in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, IEEE, 2011, pp. 57–64.

- [63] P. Werner, A. Al-Hamadi, R. Niese, S. Walter, S. Gruss, and H. C. Traue, “Towards pain monitoring: Facial expression, head pose, a new database, an automatic system and remaining challenges,” in *Proceedings of the British Machine Vision Conference*, 2013, pp. 1–13.
- [64] B. J. Matuszewski, W. Quan, and L.-K. Shark, “High-resolution comprehensive 3-d dynamic database for facial articulation analysis,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2011, pp. 2128–2135.
- [65] X. Zhang, L. Yin, J. F. Cohn, *et al.*, “Bp4d-spontaneous: A high-resolution spontaneous 3d dynamic facial expression database,” *Image and Vision Computing*, vol. 32, no. 10, pp. 692–706, 2014.
- [66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.
- [67] S. I. Serengil and A. Ozpinar, “Lightface: A hybrid deep face recognition framework,” in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2020, pp. 23–27. DOI: [10.1109/ASYU50717.2020.9259802](https://doi.org/10.1109/ASYU50717.2020.9259802). [Online]. Available: <https://doi.org/10.1109/ASYU50717.2020.9259802>.
- [68] G. C. Littlewort, M. S. Bartlett, and K. Lee, “Automatic coding of facial expressions displayed during posed and genuine pain,” *Image and Vision Computing*, vol. 27, no. 12, pp. 1797–1803, 2009.
- [69] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [70] L. Nanni, M. Paci, S. Brahnam, and A. Lumini, “Comparison of different image data augmentation approaches,” *Journal of imaging*, vol. 7, no. 12, p. 254, 2021.
- [71] L. Taylor and G. Nitschke, “Improving deep learning with generic data augmentation,” in *2018 IEEE symposium series on computational intelligence (SSCI)*, IEEE, 2018, pp. 1542–1547.
- [72] Y. Huang, L. Qing, S. Xu, L. Wang, and Y. Peng, “Hybnet: A hybrid network structure for pain intensity estimation,” *The Visual Computer*, pp. 1–12, 2022.
- [73] S. I. Nikolenko, *Synthetic data for deep learning*. Springer, 2021, vol. 174.
- [74] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic data for text localisation in natural images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2315–2324.
- [75] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, and V. Chari, “Learning to generate synthetic data via compositing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 461–470.

- [76] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [77] F. H. K. d. S. Tanaka and C. Aranha, “Data augmentation using gans,” *arXiv preprint arXiv:1904.09135*, 2019.
- [78] K. Niinuma, I. O. Ertugrul, J. F. Cohn, and L. A. Jeni, “Synthetic expressions are better than real for learning to detect facial actions,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1248–1257.
- [79] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” *arXiv preprint arXiv:1701.00160*, 2016.
- [80] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, “Recent progress on generative adversarial networks (gans): A survey,” *IEEE access*, vol. 7, pp. 36 322–36 333, 2019.
- [81] Thalles. “A short introduction to generative adversarial networks.” (), [Online]. Available: <https://sthalles.github.io/intro-to-gans/>. (accessed: 01.09.2016).
- [82] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, *et al.*, “Recent advances of generative adversarial networks in computer vision,” *IEEE Access*, vol. 7, pp. 14 985–15 006, 2018.
- [83] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [84] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [85] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [86] H. Bansal and A. Rathore. “Understanding and implementing cyclegan.” (2017), [Online]. Available: <https://junyanz.github.io/CycleGAN/>.
- [87] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [88] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*, PMLR, 2017, pp. 214–223.
- [89] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv:1701.04862*, 2017.

- [90] V. Kushwaha, G. Nandi, *et al.*, “Study of prevention of mode collapse in generative adversarial network (gan),” in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, IEEE, 2020, pp. 1–6.
- [91] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” *Advances in neural information processing systems*, vol. 30, 2017.
- [92] Y. Choi. “StarGAN - Official PyTorch Implementation.” (2018), [Online]. Available: <https://github.com/yunjey/stargan>.
- [93] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, “Ganimation: Anatomically-aware facial animation from a single image,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 818–833.
- [94] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *Advances in neural information processing systems*, vol. 29, 2016.
- [95] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.
- [96] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [97] N. Rathee and D. Ganotra, “An efficient approach for facial action unit intensity detection using distance metric learning based on cosine similarity,” *Signal, Image and Video Processing*, vol. 12, pp. 1141–1148, 2018.
- [98] Donydchen. “Ganimation – an out-of-the-box replicate.” (2018), [Online]. Available: https://github.com/donydchen/ganimation_replicate.
- [99] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, “Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5562–5570.
- [100] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [101] A. Geitgey. “Face_recognition.” (), [Online]. Available: https://github.com/ageitgey/face_recognition.
- [102] Z. Liu, P. Luo, X. Wang, and X. Tang, *Celebfaces attributes (celeba) dataset*, Online; accessed 8-July-2023, 2023. [Online]. Available: <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>.

- [103] S. Du, Y. Tao, and A. M. Martinez, “Compound facial expressions of emotion,” *Proceedings of the national academy of sciences*, vol. 111, no. 15, E1454–E1462, 2014.
- [104] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, “Openface 2.0: Facial behavior analysis toolkit,” in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, IEEE, 2018, pp. 59–66.
- [105] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [106] G. Fiorentini, I. O. Ertugrul, and A. A. Salah, “Fully-attentive and interpretable: Vision and video vision transformers for pain detection,” *arXiv preprint arXiv:2210.15769*, 2022.
- [107] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, “Joint 3d face reconstruction and dense alignment with position map regression network,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 534–551.
- [108] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.
- [109] Keras, *Keras applications*, Online, 2023. [Online]. Available: <https://keras.io/api/applications/>.
- [110] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [111] S. El Morabit, A. Rivenq, M.-E.-n. Zighem, A. Hadid, A. Ouahabi, and A. Taleb-Ahmed, “Automatic pain estimation from facial expressions: A comparative analysis using off-the-shelf cnn architectures,” *Electronics*, vol. 10, no. 16, p. 1926, 2021.