International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Facial Expression-Based Pain Analysis Using Deep Learning Models

[1]Aarti Srivastava [2]Dr. Sanjay Bhadoriya
[1]Research Scholar, Sage University, Indore
[2]Assistant Professor, Sage University, Indore
[1]aartishrivasta@gmail.com [2] sanjaybhadoriya@gmail.com

ABSTRACT
Pain is a key physiological signal of potential harm or dysfunction. While many can describe their pain, others, such as infants, unconscious patients, or those with speech impairments, cannot. In these cases, automated pain assessment is essential. Facial expressions offer a reliable, objective, and noninvasive means for real-time pain analysis. In This paper reviews deep learning algorithms for facial expression-based pain detection, comparing models developed between 2017 and 2025. It examines performance across datasets such as UNBC-McMaster, BioVid, MIntPAIN, and paediatric pain datasets. Advanced architectures, including convolutional neural networks, recurrent neural networks, Vision Transformers, and hybrid ensembles, have achieved up to 98.7% accuracy. The paper concludes with an overview of current challenges and future directions for real-time clinical use.
Keywords: RNN, MIntPAIN, BioVid, UNBC-McMaster, Deep Learning.

1.INTRODUCTION

Pain is defined by the International Association for the Study of Pain (IASP) as “an unpleasant sensory and emotional experience associated with, or resembling that associated with, actual or potential tissue damage.” Pain affects quality of life and is a significant public health issue. During the COVID-19 pandemic, the lack of sufficient healthcare personnel made real-time patient monitoring difficult, highlighting the need for automated pain detection systems.

Pain is typically classified into two categories:

- **Acute Pain:** A short-term, sudden onset of pain, often resulting from injury, surgery, or trauma. It lasts less than three months and is associated with tissue damage.
- **Chronic Pain:** Long-lasting pain that persists beyond the normal healing time, often associated with conditions like arthritis, cancer, or nerve damage.

When patients cannot express pain verbally, automated systems—especially those based on facial expressions—offer a non-invasive and rapid method for assessing pain intensity and ensuring timely clinical interventions.

2. RELATED WORK

Traditional methods such as Local Binary Patterns (LBP), Gabor filters, and the Facial Action Coding System (FACS) were previously used for facial feature extraction in pain detection. However, these methods face limitations such as lighting sensitivity and pose variability [1]. Recent research has shifted toward deep learning methods, particularly:

2.1. RNN-based Model: RNN-based model applied in the pain detection field using facial expression analysis. This model works when we check how the expression changesover time. RNN-basedmodels have been effectively utilized in pain detection, facial expression recognition, and neuroimaging data analysis, among others.

Some RNN-basedmodels work in pain detection.

2.1.1. Hybrid –RNN –ANN for physiological signal pain recognition. Hybrid –RNN-ANN is a bidirectional LSTM (BiLSTM) that is employed to extract temporal features from physiologicalsignals,eg, (EDA, ECG, EMG). This model automatically learns features and feeds them into the ANN to achieve a classification accuracy of 83.3% [1].

2.1.2 FNIRS using BiLSTM:- This architecture describes a bidirectional LSTM network to capture brain activity during pain with the help of functional near-infrared spectroscopy [2].

2.1.3. Pain Net: In this paper, the relational network with RNN Embedding it described on an episode-based training scheme to compare video pain to estimate self-report sequence-level pain [3].

2.1.4. RCNN Regression: This paper processes sequence data from AAM works facial frame using recurrent unit to model to single modularity or sensor data, compares classification performance against random forest and regression model [4].

2.2. CNN-based Model [6]:

This framework combined fine-tuned VGG Face, Principal Component analysis, and a hybrid CNN +RNN model This framework was trained on the Mint PAIN database and evaluated the UNBC-MCMAster data set it show 89% accuracy. Transfer learning with VGG16 for postoperative Pain architecture describe leveraging facial expression analysis hold significant potential to recognize varying degrees of pain. Spatiotemporal CNN 3D-convolutin architecture working on video and find 98.53 accuracy [7]. Mobile Net, GoogleNet, ResNeXt, ResNet18, Dense 161, Combine 2 different models [8].

3. DEEP LEARNING MODEL FOR FEATURE EXTRACTION

In the pain intensity classification framework, feature extraction plays a critical role in capturing the subtle facial cues associated with varying levels of pain. Using deep CNNs like ResNet, VGG, DenseNet, ENet, and SqueezeNet, multiple convolutional layers automatically learn hierarchical features from raw input pictures. Along with low-level features (like edges and textures), these networks also pull out high-level semantic features (like brow furrowing, eye squinting, muscle contractions) that show pain expressions in lower layers. Utilizing the Grey Level Co-occurrence Matrix (GLCM), useful texture features are extracted from facial pictures for the purpose of classifying pain. Looking at how often pairs of pixel values happen at a certain distance and orientation in a picture is how GLCM figures out the spatial relationship of pixel intensities. This matrix shows different texture features, like contrast, correlation, energy, and homogeneity, that show how pain emotions change the patterns and textures in different parts of the face [9].

3.1 Deep Learning Architecture

A-InceptionV3: As a member of the Inception family, InceptionV3 is made to be accurate while also using little computer power. 48 or so layers [20]. It has sections called "inception"

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

that use filters of different sizes to pick up features at different sizes. At the same time, 1x1, 3x3, and 5x5 convolutions are used. A mix of these convolutions is used in this method. Uses the functions for group normalization and ReLUactivation [9].

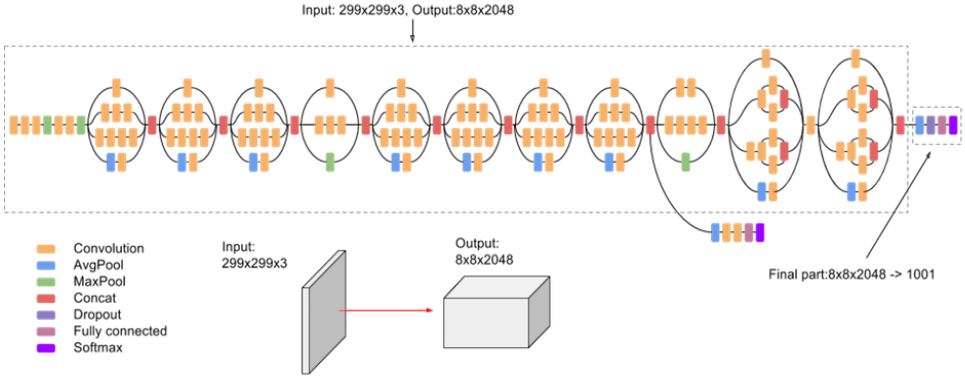

Figure.1 InceptionV3 layer architecture [9].

3.2 Inception Module:

Output=Concatenate (Conv1x1(input), Conv3x3(input), Conv5x5(input), MaxPooling(input))

3.3 Auxiliary Classifier:

Output=Softmax (FC(GlobalAveragePooling(input), units))

B-ResNet50: Residents of ResNet50 (Residual Network) are famous for using leftover blocks. It is made up of 50 layers, and the shortcut links between them help fix the problem of vanishing gradients during training. Residual blocks have skip links that let the input go straight through one or more layers. So, the network can learn how to use leftover functions.

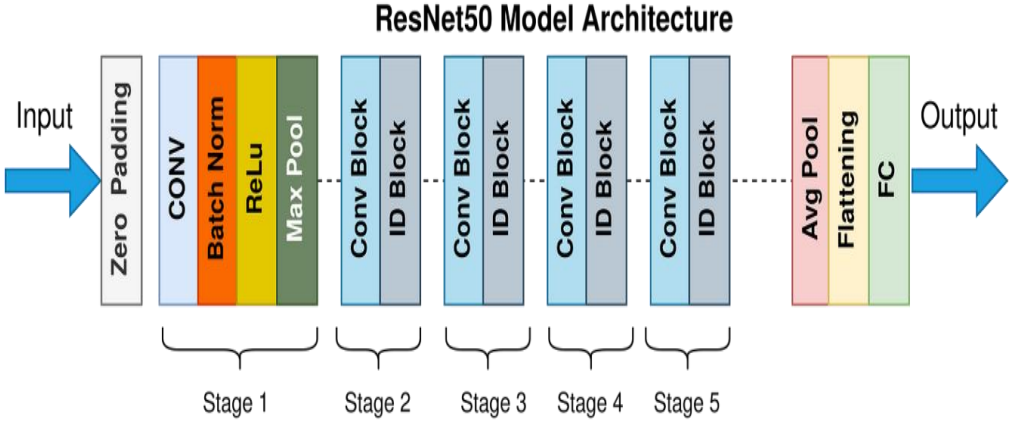

Figure. 2 ResNet50 layer architecture [9]

Residual Block:

Output=ReLU (Add (Conv3x3(input), input))

Global Average Pooling:

Output=Average Pooling (input, pool_size)

Fully Connected Layer (FC):

Output=Softmax (FC(GlobalAveragePooling(input), units))

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

C-SqueezeNet-SqueezeNet is designed to achieve AlexNet-level accuracy on ImageNet with 50x fewer parameters. This makes it efficient for deployment on devices with limited computational resources. The architecture uses a combination of Fire modules and a few traditional convolutional layers [21].

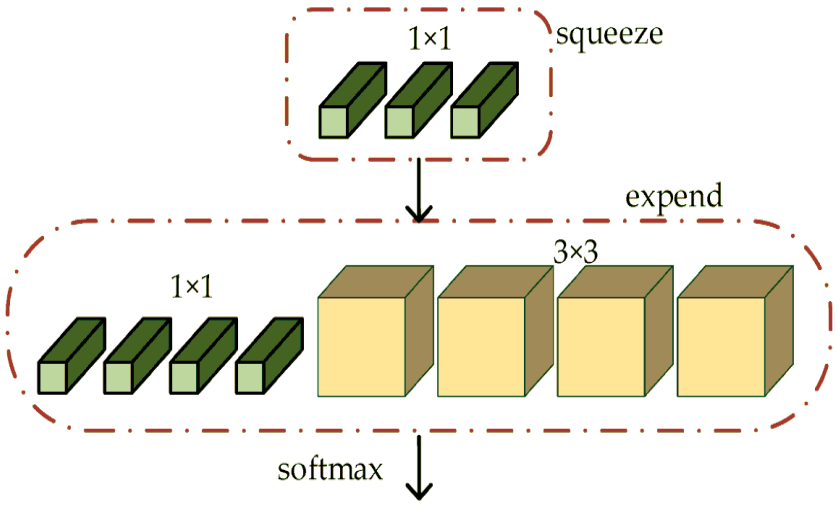

Figure. 3 SqueezeNet-layer architecture [9]

3.4 Layers and Functions:

Squeeze Layer: 1x1 convolutions to reduce the number of input channels. Squeeze=ReLU (Conv1x1(input,filters))

Expand Layer: A mix of 1x1 and 3x3 convolutions to capture spatial features.

Expand1x1=ReLU (Conv1x1(Squeeze,filters))

Expand3x3=ReLU (Conv3x3(Squeeze,filters,padding))

Output:

Output=Concatenate (Expand1x1, Expand3x3)

E-DenseNet-DenseNet (Densely Connected Convolutional Networks) builds a feed-forward network that links every layer to every other layer. It makes gradient flow better and supports feature reuse, which lowers the number of parameters, since each layer gets the feature maps of all the layers that came before it. [22]

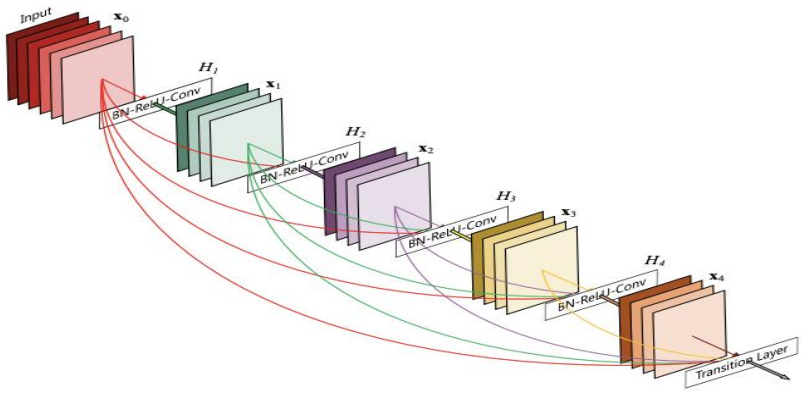

Figure. 4 DenseNet layer architecture [9]

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Dense Block

- Composite Function: Batch normalization followed by ReLU and a 3x3 convolution.
- Composite=ReLU (BatchNorm (Conv3x3(input, filters, padding))
- Concatenation: The output of each layer is concatenated with the input of the following layer.
- Output=Concatenate (input, Composite)

Transition Layer:

- **Batch Normalization and 1x1 Convolution:** Reduces the number of feature maps.
- **Transition**=ReLU (BatchNorm (Conv1x1(input,filters)))
- **Pooling:** Downsamples the feature maps.
- **Output**=AveragePooling (Transition,pool_size)

F-ResNet34-Figure 5 shows the structure of the ResNet-based networks. Input to the convolutional layer is a colour face picture that has been processed and has 224 × 224 pixels. After making 64 feature maps, they are sent to a max-pooling layer, which finds the maximum value across 3 × 3 spatial neighborhoods with a step of 2 for each channel. Four Residual Layers follow, each with three, four, six, or three DropBlocks

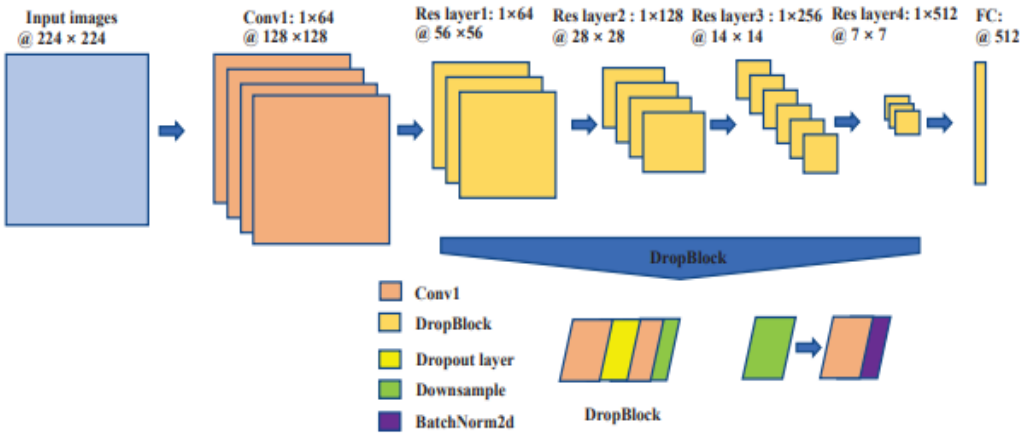

Figure 5. The structure of the ResNet34 CNN Network [9]

The two convolutional layers have different numbers of filters and are 3x3 in size. Each DropBlock also has a dropout layer. Overfitting can be stopped by the dropout layer. On top of that, an adaptable average-pooling layer is used to link the fourth Residual Layers' output for lightweight features. Lastly, the FC layer is used to sort things into groups.

3.5 E-ResNet101

ResNet101, a 101-layer deep convolutional neural network, employs residual blocks to facilitate the training of very deep networks. It is implemented in PyTorch using residual blocks and custom dataset handling. The architecture consists of an initial convolutional layer followed by several residual blocks arranged in stages, and finally an average pooling layer and a fully connected layer for classification.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

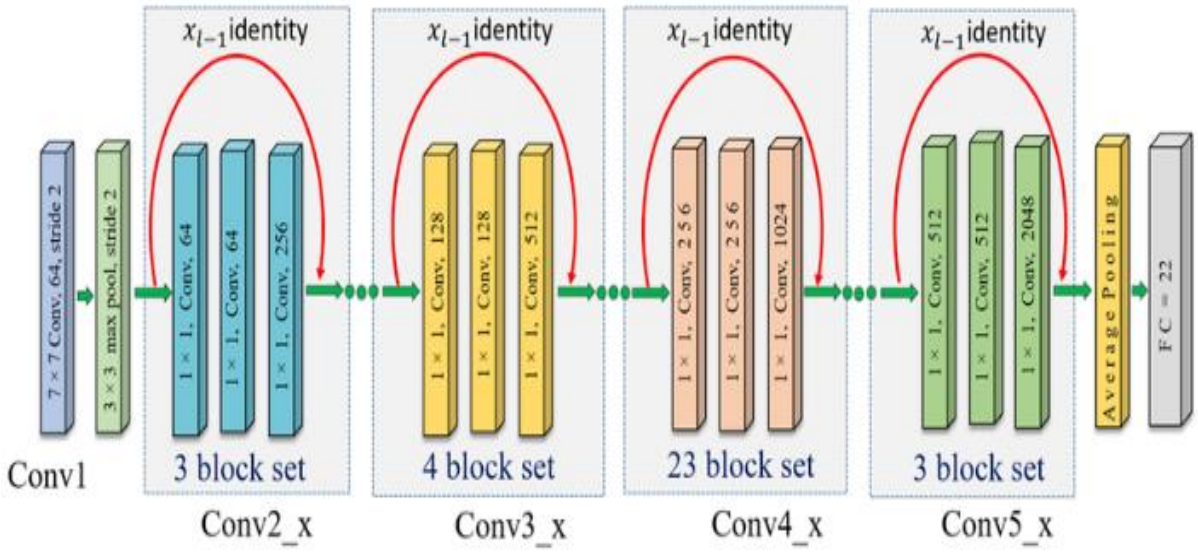

Figure 6. architecture of the ResNet-101 model [9]

3.6 F- ENet-ENet (Efficient Neural Network) is a deep learning model designed for real-time semantic segmentation of images. It is a relatively lightweight network that achieves state-of-the-art performance on several benchmark datasets while requiring less computational resources than other popular models. The architecture of ENet consists of several types of layers, including convolutional layers, max pooling layers, batch normalization layers, and bottleneck layers. The mathematical expression for ENet can be represented as a function f, where f takes as input an image I and outputs a set of segmented pixels M. The function f can be decomposed into a series of layers, which can be represented as follows:

M = f(I) = L_n(L_{n-1}(...(L_2(L_1(I)) Eq.1

Where L_i represents the i-th layer of the network.

The first layer of ENet is a convolutional layer that extracts features from the input image. This is followed by a series of bottleneck layers, which are designed to reduce the computational cost of the network by reducing the number of channels in the feature maps. The bottleneck layer can be represented mathematically as follows:

y = L_{bottleneck}(x) = relu(conv_1x1(x)) * relu(conv_3x3(conv_1x1(x))) Eq.2

where x is the input feature map, conv_1x1 and conv_3x3 are convolutional layers with kernel sizes of 1x1 and 3x3 respectively, relu is the rectified linear unit activation function, and * represents element-wise multiplication.Net also uses a type of layer called a "split-and-merge" layer, which splits the feature maps into smaller parts, applies different operations to each part, and then merges them back together. This helps to reduce the computational cost of the network while maintaining the ability to capture fine-grained details. The output of the final layer of ENet is a set of segmented pixels, which can be represented mathematically as follows:

M = softmax(conv_{final}(L_{drop}(L_{global}(L_{last}(L_{sum}(L_{concat}(x)) ..Eq.3

where conv_{final} is a convolutional layer that produces the final output, L_{drop} is a dropout layer to prevent overfitting, L_{global} is a global average pooling layer, L_{last} is

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

a final convolutional layer, L_{sum} is a layer that sums the feature maps from different stages of the network, L_{concat} is a layer that concatenates feature maps from different layers of the network, and softmax is used to produce a probability distribution over the classes.

4. ANALYSIS OF DIFFERENT MODELS

This table shows year-wise analysis of different models with accuracy and working number of working parameters.

Table 1: Analysis based on accuracy [10]

Model name	Number of parameters [Millions]	ImageNet Top 1 Accuracy	Year
AlexNet	60 M	63.3 %	2012
Inception V1	5 M	69.8 %	2014
VGG 16	138 M	74.4 %	2014
VGG 19	144 M	74.5 %	2014
Inception V2	11.2 M	74.8 %	2015
ResNet-50	26 M	77.15 %	2015
ResNet-152	60 M	78.57 %	2015
Inception V3	27 M	78.8 %	2015
DenseNet-121	8 M	74.98 %	2016
DenseNet-264	22M	77.85 %	2016
BiT-L (ResNet)	928 M	87.54 %	2019
NoisyStudent EfficientNet-L2	480 M	88.4 %	2020
Meta Pseudo Labels	480 M	90.2 %	2021

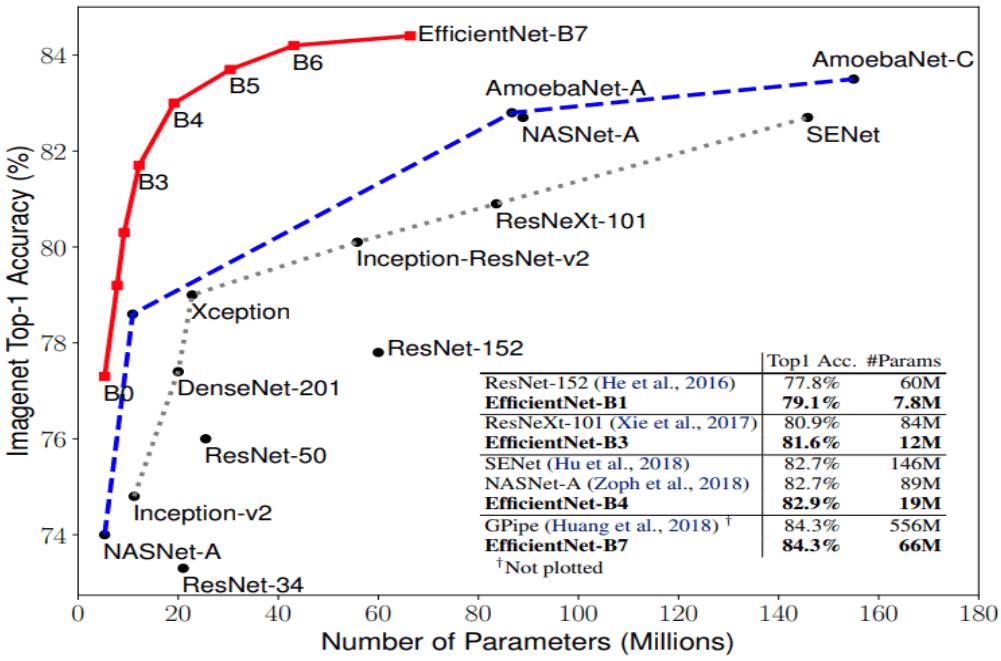

Table 2: Frequently Used data set by Researcher

Ref.	Dataset Name	Year	Dataset Size	Key Features / Modalities
[10]	UNBC-McMaster Shoulder Pain	2011	25	FACS coded facial expressions, pain intensity levels (1–15)
[11]	BioVid Heat Pain	2013	8	Heat-induced pain, FACS, ECG, EDA signals
[12]	BP4D Spontaneous	2014	41	Facial expressions including fear, anger, and pain
[13]	FACS-Based Pain Dataset	2016	140	Facial Action Coding System, facial and physiological signals
[14]	Multimodal Pain Dataset	2018	20	RGB images, thermal images, depth data
[30]	Infant Facial Expression Dataset	–	26	Infant facial expressions related to crying and pain
[31]	iCopeVid	–	49	Video data labeled as Pain / No-Pain
[32]	NPAD-I	–	36	Neonatal Pain, Agitation and Sedation Scale
[33]	APN-db	–	112	Infant pain coding system annota

[10] Best deep CNN architectures and their principles: from AlexNet to EfficientNet.

Deep Learning Approaches (2017–2025)

Timeline of Major Models

Table 3: Major model

S.n	Year	Model	Modality	Highlights
1	2017	DeepFaceLIFT	Facial Expression	Personalized model, ICC improvement
2	2020	Bi-LSTM + fNIRS	Neuroimaging	~90.6% accuracy
3	2021	CNN + LSTM	EDA, ECG	Multi-modal nociceptive pain detection
4	2022	BiLSTM + XGBoost	Cold pain (EDA)	Personalized pain classification
5	2023	PainAttnNet	Transformer, MSCN	Outperforms on BioVid
6	2023	ZNet	Facial Expression	95.4% accuracy, Transfer Learning
7	2024	DoseFormer	GAT + Vitals	AUROC 0.98, F1 0.85
8	2025	PainFormer	Vision Foundation Model	Trained on 10.9M samples, state-of-art

5. PERFORMANCE ANALYSIS ACROSS MODELS

UNBC–McMaster Dataset

- BEiT Transformer: 96.82% accuracy (F1: 94.61%)

- ECCNET: 93.87% → 96% with enhancement
- ShuffleNet V2 + Ensemble: 98.7% accuracy (highest recorded)
- CNN-RNN Ensemble: 86% accuracy

MIntPAIN Dataset

- CNN-RNN Hybrid: 92.3% accuracy
- Temporal Convolutional Network: 94.1% accuracy, AUC 91.3%

Pediatric & ICU Datasets

- CPANN (CPEC): 82.1% accuracy
- PFECIC (ICU Children): 88.3% accuracy, F1: 88.5%

Neuroimaging-Based

- CNN-LSTM (fNIRS): 91.2% accuracy
- EEG-based CNN/RNN: 87–88% accuracy (F1: ~93.2%)

Comparative Algorithm based on performance

Table 4: Performance analysis

S.no	Algorithm/model Type	Data Set	Performance
1.	ShuffleNet V2 + XGBoost Ensemble	UNBC–McMaster	98.7% Accuracy
2	ECCNET (VGG + MobileNet + GoogleNet)	UNBC–McMaster	~96% Accuracy
3.	CNN–LSTM Hybrid	BioVid / UNBC	83.1% Accuracy, AUC 93.3%
4.	ZNet (Transfer Learning)	UNBC–McMaster	95.4% Accuracy
5.	PSRS (Pain Sentiment CNN)	UNBC–McMaster	83.7% Accuracy, F1 0.8253
6.	Vision Transformers (ViT, ViViT)	UNBC–McMaster	F1 ~0.49–0.55

6. CHALLENGES AND FUTURE DIRECTIONS

Using deep learning model there are so many challenges like Dataset Limitations: if our data set is too small then we do not find an accurate result. lack of other like demographic diversity, unbalanced classes. Secondly, Generalization: Many models perform well on one dataset but poorly on others third one is Real-Time Deployment: Limited hospital-ready systems fourth is Multimodal Fusion: Combining facial, physiological, EEG, and audio features shows promise another is Explainability: Need for interpretable AI in healthcare other is Low-resource Deployment: Lightweight models needed for mobile & rural healthcare settings at last but not least Federated Learning: Enables privacy-preserving pain detection across multiple hospitals

7. CONCLUSION

Facial expression-based pain detection using deep learning has evolved rapidly in the past decade. State-of-the-art models like Painin Former and ShuffleNet ensembles have reached near-human-level accuracy in specific datasets. However, generalization, real-time application, and interpretability remain key challenges. The future lies in developing robust,

lightweight, and multimodal systems that can be deployed in hospitals, home care, and remote settings—ultimately leading to improved patient care and pain management.

References

1. Runwag,KexuFeng,Weigh,"Hybrid RNN-ANN based deep Physiological Network for pain Recognition",AnnyIntconf ,IEEE med Bio soc,2020, page no. 5584-5587.
2. Raul Fernqdez,RaisRomero,"Pain assessment based on FNIRS Using Bidirectional LSTM ",arxivcs,24 dec 2020.
3. Ming Bishay,GrahamPage,MohamaadMavadati," PainNet: statistical relation network with Episode–based training for pain estimation", Arxiv,8 April 2025.
4. Jingzhou,Xiaopena,Feish," Recurrent Conventional Neural network regression for continue pain intensity estimation in video",3 May 2010.
5. Ehsan Othman, Phillppwerner, Frerksaxena," An automatic system for continuous pain intensity monitoring based on analyzing data from uni and bidirectional and multimodal", Sensor2022, 15 July 2022, pg 210-222.
6. Ghazal Bargshady ,Xujuan Zhou, RavineshCdeo," Ensemble neural network approach detectionpain intensity from facial Exprisson ",Pubmed,7 september 2020,DOI :10.1016 ,artmed 2020.
7. Ji-Tuozhang,Xiao-Yihu,WenduanPubmed ," application of deeplearning based facial pain recognition model for post operative pain assment",13 june2025,DIO:10.1016/Jjeline.2025.111898.
8. MohamaadTavakoilan,AbdenoulHodid," Spatiotemporal convolutional Neural network for automatic pain intensity Estimation from facial Dynamics ",Innternational journal of Computer vision ,2019.
9. ShrivastavaAarti,BhadouraiyaSanajy,"Efficient Pain Recognition via Deep Learning: A Comparative Analysis of Lightweight and Deep CNN Models". (2025). Ianna Journal of Interdisciplinary Studies,ISSN(O):2735-9891,ISSN(P):2735-9883, 7(2), 42-61.
10. AI Summer, "Best deep CNN architectures and their principles: from AlexNet to EfficientNet", [Online]. Available: https://theaisummer.com/cnn-architectures/. [Accessed: Sep. 9, 2025].