# Pain Detection: A Comparative Study of Different Complexity Architectures

Dallaglio Frignani, Nicolò
261228@studenti.unimore.it

Filieri, Martina
274638@studenti.unimore.it

Fini, Giada
271166@studenti.unimore.it

## Abstract

Numerous scientific studies have highlighted the significant short- and long-term effects of early-life pain exposure. This paper explores the implementation of various machine learning architectures aimed at maximizing accuracy in pain detection, particularly in environments with limited datasets. We present three main architectures, including a geometry-based model, a 3D network, and a retrieval algorithm, each designed to classify pain as real or fake. Our results demonstrate that while less complex models can achieve high accuracy with smaller datasets, more sophisticated architectures like the 3D network require larger data sets to be effective.

## Contents

## 1 Introduction

Early developmental exposure to unexpected and repeated painful stimuli has been shown to alter pain sensitivity and perception, potentially leading to conditions like allodynia and hyperalgesia. Such experiences also affect the stress-response system, leading to elevated basal cortisol levels, and can negatively influence postnatal growth, as evidenced by slower gains in body weight and head circumference. There is substantial evidence associating early-life pain with structural and functional changes in the brain.

One major challenge in detecting pain, especially in neonates, is the subjectivity of medical assessments. In some cases, neonatal distress signals are entirely overlooked. Hence, developing an objective and accurate tool for pain detection is crucial for preventing future harm. This paper compares different architectures — ranging from a complex dual-direction 3D attention model to a simpler geometry-based design — demonstrating that less complex models can perform effectively, even with small datasets. This is especially important given the difficulties in obtaining large amounts of infant video data due to privacy concerns.

# 2 Methods

## 2.1 Geometry-Based Model

This architecture uses majority voting to classify videos based on individual frames. Each video frame is processed using three feature extractors: Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Convolutional Neural Networks (CNN). These extracted features are combined into a single feature vector, which is passed through fully connected layers. The process is repeated for eight frames per video, and the video is ultimately classified as representing either real or fake pain.

## 2.2 3D Model

A 3D adaptation of the 2D DDAMFN model, originally developed for Facial Expression Recognition (FER), was employed. Facial features are extracted using a Mixed Feature Net (MFN) and passed through the Dual Direction Attention Network (DDAN) module. The model is trained on sequential frames to classify pain in videos.

## 2.3 Retrieval Algorithm

This algorithm, built upon the geometry-based model, compares feature vectors extracted from new data with previously classified vectors. The 10 most similar vectors are selected, and a majority vote determines the final classification of the video.

## 2.4 Results

Our results show high accuracy for the geometry-based networks, whereas the 3D network underperformed, likely due to the smaller dataset size. These findings support the notion that 3D models require large datasets for optimal performance, which may not always be feasible.

# 3 Dataset

The dataset comprises 7,504 frames from 938 videos (each containing eight frames). The frames were resized to 112×112 pixels for the geometry-based models and 128×128 for the 3D DDAMFN model. Data augmentation techniques—including rotation, translation, and scaling—were applied to improve network robustness. Additionally, image sharpening was used to enhance frame clarity before feeding them into the architectures.

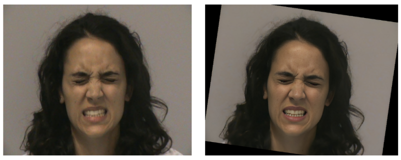


Figure 1: Transformations applied to the Dataset images

# 4 Geometric Network

This hybrid model integrates traditional feature extraction methods, such as HOG and SIFT, with deep learning techniques. The model leverages handcrafted and deep features to improve classification accuracy. A visual representation is displayed below, in Figure 2.

## 4.1 CNN Component

The CNN architecture consists of three convolutional layers, each followed by batch normalization and ReLU activation. Max pooling is applied too after each convolutional block, to reduce the spatial dimensions and to emphasize the strongest activations in the feature maps, which often correspond to the most significant features. Finally, an adaptive average pooling layer reduces the output to a fixed size before flattening it for fully connected layers. Initially, the CNN architecture consisted of six convolutional blocks. However, we identified that the CNN was the primary contributor to overfitting, as its complexity exceeded what was necessary for the available data. To counter this, we reduced the network to three convolutional blocks.

## 4.2 Hybrid Feature Extraction

The model extracts features from three sources: HOG, SIFT, and CNN. HOG and SIFT feature vectors are processed through fully connected layers to reduce dimensionality, while image inputs are passed through the CNN. The concatenated feature vector is then passed through fully connected layers for final classification.

## 4.3 Loss Function

$$Loss = -\sum_{i=1}^{output\_size} y_i \cdot \log(\hat{y}_i)$$

Cross-Entropy Loss was used, which penalizes incorrect classifications by assigning higher loss when the predicted probability of the correct class is low. This encourages the model to increase the predicted probability of the correct class during training.

## 4.4 Final Classification

For the final classification of each video, we employed a majority voting mechanism rather than classifying each individual frame independently. Predictions are made for each frame, and the final classification is determined by the class that receives the majority of votes across all frames.

## 4.5 Conclusion

This hybrid approach, integrating both traditional and deep learning-based feature extraction methods, aims to maximize classification accuracy. The modularity of the code allows for easy experimentation with different architectures and feature extraction methods, making it suitable for a variety of image classification tasks, as a future expansion to perform

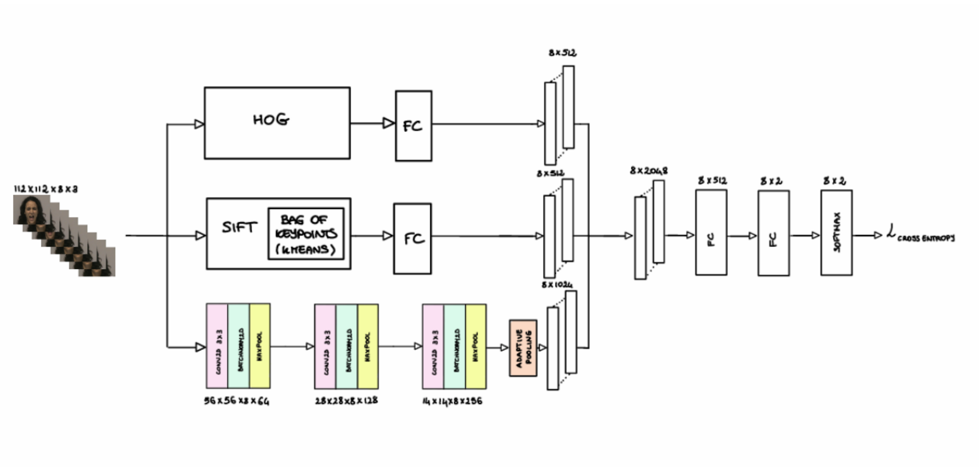


Figure 2: Geometric Network

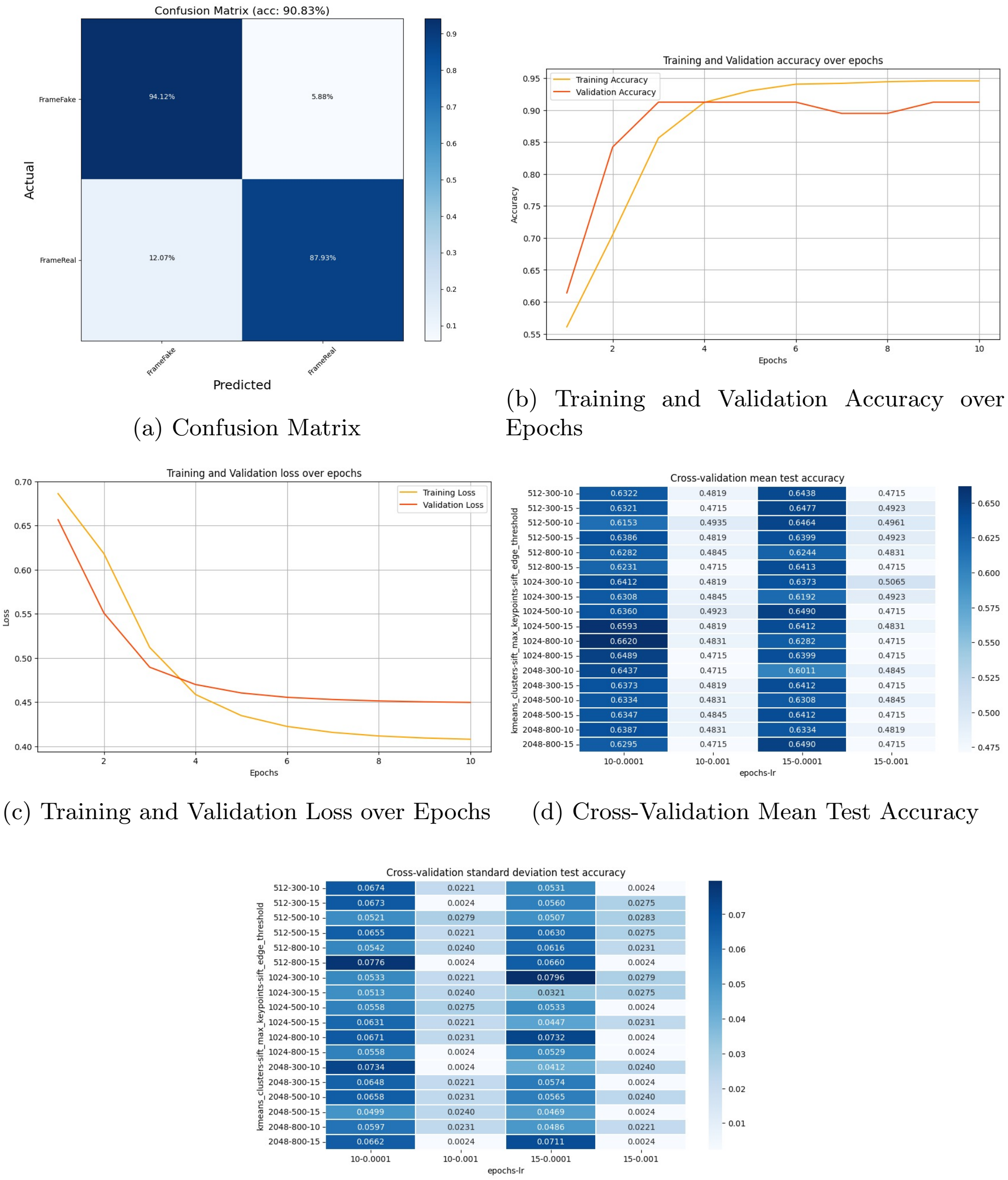
pain levels recognition. In Figure 3 are presented some of the results we have obtained.

# 5 3D DDAMFN Model

In our implementation, we adapted the architecture from the paper "A Dual-Direction Attention Mixed Feature Network for Facial Expression Recognition" [1] to handle 3D data instead of 2D data. This model was chosen due to its high accuracy on large datasets like AffectNet, RAF-DB, and FERPlus, all of which are used for Facial Expression Recognition (FER). The FER task is closely related to ours, as recognizing pain expressions is similar to identifying other facial expressions (e.g., neutral, happy, sad, surprised, and fearful). Our modification transformed it into a binary classifier, distinguishing between real and fake pain. A visual representation is given in Figure 4.

## 5.1 MFN Changes

1. Dimensionality shift: The primary update in this version is the shift from 2D to 3D operations. This includes:
   - Replacing `Conv2d` with `Conv3d`.
   - Replacing `BatchNorm2d` with `BatchNorm3d`.
   - Adjusting kernel sizes, strides, and padding from 2D to 3D.
2. Coordinate Attention Module: The 2D `CoordAtt` class has been replaces with a 3D variant, `CoordAtt3D`, which now applies adaptive pooling across three dimensions (time, height, and width), instead of just two (height and width).
3. Depth-wise convolutions: The `Depth_Wise` class and `MDConv` have been updated to support 3D convolutions. The new classes, `Depth_Wise3D` and `MDConv3D` are adapted for 3D input.



(a) Confusion Matrix

(b) Training and Validation Accuracy over Epochs

(c) Training and Validation Loss over Epochs

(d) Cross-Validation Mean Test Accuracy

(e) Cross-Validation Standard Deviation Test Accuracy

Figure 3: Geometric Network Results

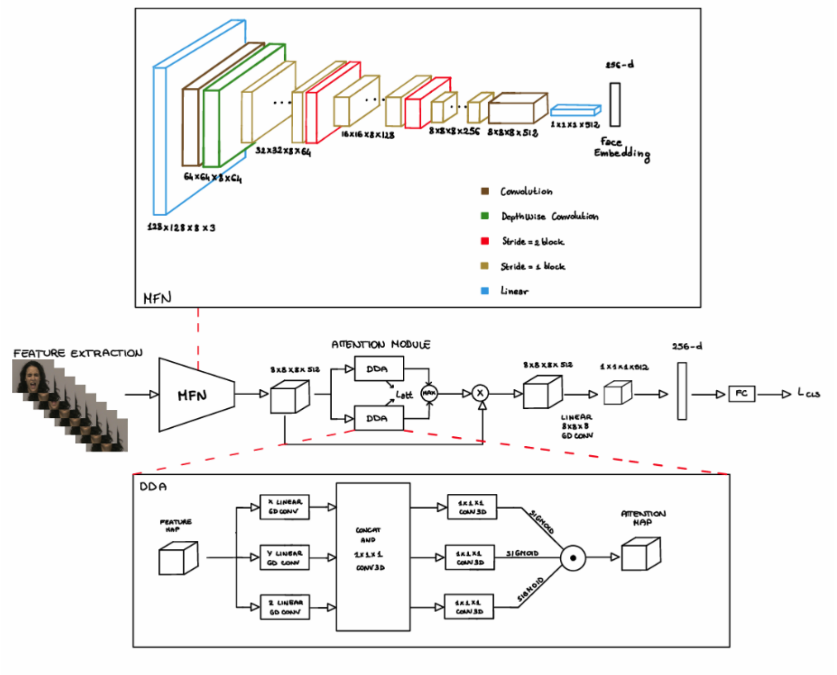


Figure 4: DDAMFN Network

4. Residual Connections: Residual connections are still employed to enable the flow of information across layers, but they have been modified for 3D convolutional operations.

5. Mixed Depth-Wise Convolutions: The `Mix_Depth_Wise` class has been replaced with `Mix_Depth_Wise3D`, which incorporates 3D depth-wise convolutions.

## 5.2 DDAM Changes

For the DDAM architecture, the MFN remains the backbone for feature extraction. The main changes include:

1. Convolutional layers: The original 2D model uses `nn.Conv2d` for 2D convolutions, which has been replaced with `nn.Conv3d` in the 3D version. This allows convolution operations across three dimensions.

2. Batch normalization: Similarly, `nn.BatchNorm2d` used for 2D feature map normalization has been replaced with `nn.BatchNorm3d` to normalize 3D feature maps.

3. Kernel and stride sizes: The kernel and stride sizes in the 3D version have been adapted to accommodate the extra dimension. For example, the kernel size changed from (7,7) in 2D to (8,8,8) in 3D. This adjustment ensures the kernel perfectly overlaps with the feature map, capturing all relevant information.

4. Attention mechanism: The 2D `CoordAtt` module has been replaced with `CoordAtt3D`. In the 3D version, the

attention mechanism now includes the temporal (depth) dimension, and additional operations such as `Linear_t`, `Linear_h`, and `Linear_w` are introduced to handle attention across time, height, and width.

### 5.3 Attention Loss

As in the original architecture, the Mean Squared Error (MSE) loss was employed. The loss is calculated between each pair of attention maps generated by the different dual-direction heads, and the final attention loss is obtained by taking the reciprocal of the sum of these MSE losses. Here, $n$ represents the number of attention heads, while $a_i$ and $a_k$ denote attention maps produced by two different heads.

$$L_{att} = \frac{1}{\sum_{i=0}^{n} \sum_{k=0}^{n} MSE(a_i, a_k)}, (i \neq j)$$

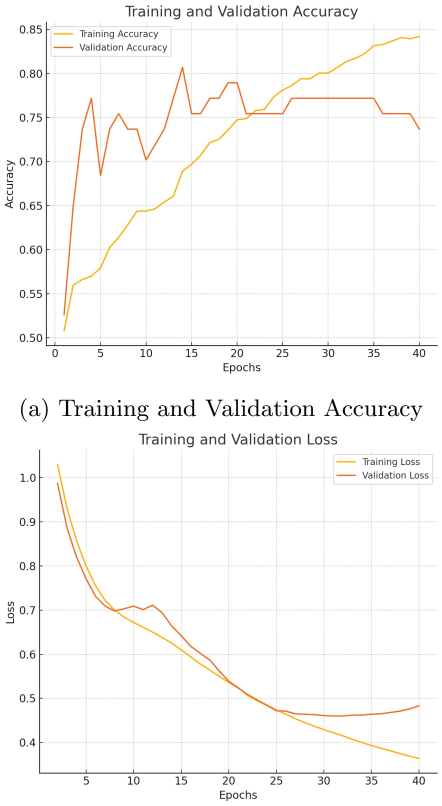
### 5.4 Final Loss

Similarly, the final loss remains largely unchanged, combining the standard cross-entropy loss with the attention loss.

$$L = L_{cls} + \lambda_a L_{att}$$

### 5.5 Results

As anticipated, the 3D architecture did not achieve high accuracy, likely due to the limited dataset — a common challenge in infant pain detection research. The graphs below show that accuracy never exceeds 85%, and when the number of training epochs is too high, the model begins to overfit. This overfitting occurs because the architecture requires more data to generalize effectively.



(a) Training and Validation Accuracy

(b) Training and Validation Loss

Figure 5: 3D DDAMFN Model Results

## 6 Retrieval Algorithm

An additional approach we implemented is a retrieval algorithm. For each frame of the input video (eight in total), the algorithm generates a feature vector and combines them by averaging. It then compares these feature vectors with those generated by the geometric network, using only the correctly classified videos. From this comparison, the algorithm identifies the 10 most similar feature vectors. A majority voting mechanism is then applied to classify the video based on the most frequently predicted class.

### 6.1 Similarity Measure

We used Cosine Similarity as the similarity measure. It was chosen because it is particularly effective for comparing features extracted from images, regardless of the feature vectors' magnitudes. Cosine Similarity is also a highly useful metric in image retrieval tasks. It is calculated by taking the cosine of the angle between two vectors.

$$CosineSimilarity = \frac{A \cdot B}{||A||||B||}$$

### 6.2 Results

The retrieval algorithm, like the geometric model, achieved high accuracy and can be effectively used for classifying new videos.

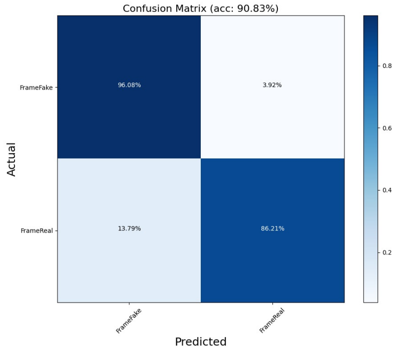


Figure 6: Retrieval Algorithm Confusion Matrix

## 7 Conclusions

The architectures presented here are just a few examples of how this task can be approached. As expected, we demonstrated that overly complex architectures perform poorly when data is limited, despite their potential to extract more precise features. With sufficient data, this model would likely be more robust. In contrast, the simpler geometric network and the retrieval algorithm built upon it performed exceptionally well, achieving over 90% accuracy in classifying real and fake pain.

Due to the academic nature of this project, there are some limitations, particularly the lack of infant data, which required us to train the models using videos of adults. Additionally, several improvements could be made. One promising future direction is the classification of pain intensity, which would be invaluable for medical assessments by not only detecting pain but also measuring its severity, allowing for tempestive intervention when necessary.

Real-world testing is also essential to identify potential weaknesses or incorrect predictions in the model, which would allow for further refinement and improvements.

## References

[1] Saining Zhang, Yuhang Zhang, Ye Zhang, Yufei Wang, and Zhigang Song. A dual-direction attention mixed feature network for facial expression recognition. *Electronics*, 12(17), 2023.