EasyChair Preprint

№ 9929

Toward an Efficient Emotion Recognition from Facial Expressions Using ML

Hmad Zennou, Mohamed Ouhda and Mohamed Baslam

April 6, 2023

Abstract:

The information conveyed by facial expressions can be utilized to identify emotions. When these face expressions are performed, they change over time. Even for people, recognizing certain emotions is a difficult task. Machine learning algorithms are used in this work to recognize emotions in image sequences. It analyzes emotions automatically using cutting-edge deep learning on collected data. This paper compares current state-of-the-art learning algorithms for handling spatiotemporal data and adapting traditional static approaches to deal with image sequences. Expanded versions of CNN, 3D CNN, and Recurrent methods for universal emotion recognition are assessed and contrasted in two public datasets, where the performances are proved, and advantages and disadvantages are addressed. Afterwards, we propose a new end-to-end architecture called Spatio-Temporal Convolutional Features with nested LSTMs (Long short-term memory) that learns multi-level appearance features and temporal dynamics of facial expressions in a common way. More specifically, we use 3D CNN to extract spatio-temporal convolutional features from image sequences representing facial expressions, and the dynamics of facial expressions are actually combined by two sub-LSTMs, Temp-LSTM and Conv-LSTM modeled by a nested LSTM. That is, we use Temp-LSTM to model the temporal dynamics of spatio-temporal features in each convolutional layer, and we use Conv-LSTM to integrate the output of all Temp-LSTMs to obtain multi-level data encoded in hidden layers. Experiments were conducted on two benchmark databases, Oulu-CASIA, and, SASE-FE and the results showed that the proposed method achieved better performance than the expanded versions of CNN, 3D CNN, and Recurrent methods.

1. Introduction and state of the art:

Human faces are full of information, some of this convey the emotion that the individual is expressing. A set of facial expressions are used to recognize emotions in people's faces. The entire range of these expressions must be studied to appropriately recognize emotions. As a result, it's crucial to think of visual sequences that show the whole gamut of emotion. The Figure 1 shows a sequence of images presenting facial expression examples.

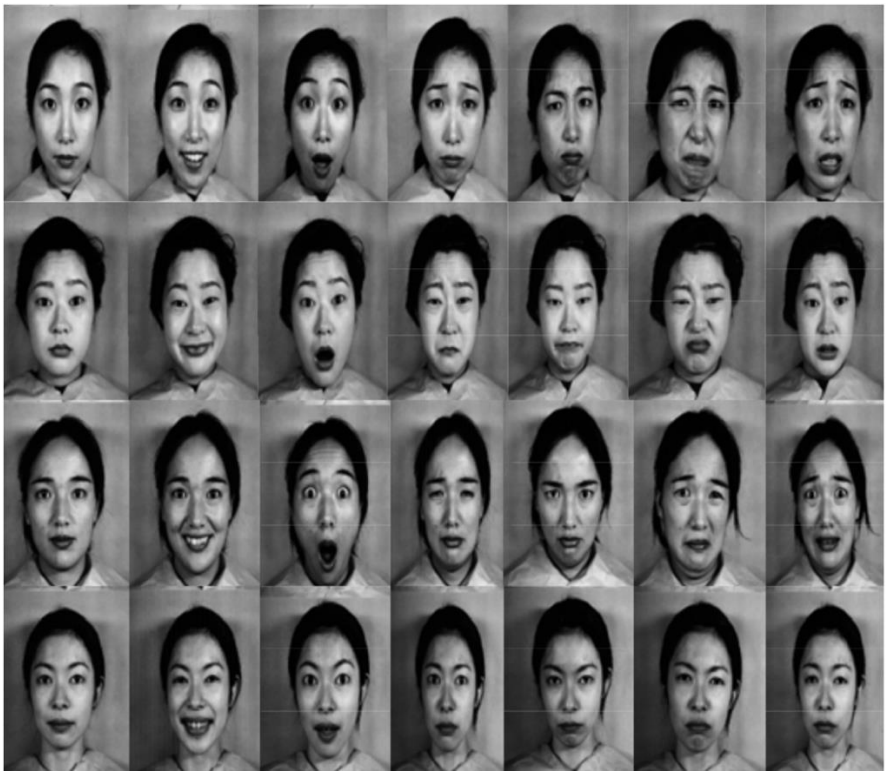


Figure 1: Facial expression samples from JAFFE [13] database. (from [20])

Emotion recognition is such a difficult task that even some humans struggle to identify certain emotions. A trained person can conceal an emotion that is being expressed through facial expressions in the event of blocked and/or micro emotions. Unfortunately, most people are unable to distinguish between genuine and false facial emotions.
In the field of computer vision, deep learning algorithms have achieved excellent success in recognizing human expressions, postures, etc. A sequence of photos that represents the entire face expression of the emotion from start to end has recently been developed.

Facial expressions play a crucial role in human communication, particularly when it comes to conveying emotions. With the rapid advancement of artificial intelligence's computer vision techniques, a great deal of effort has gone into recognizing facial expressions.
Viola and Jones [24], convolutional neural networks (CNN) [6], and support vector machines (SVM) over histogram of gradient (HOG) features [16] are some of the methods used for face detection. Then, for face segmentation, some works proposed complex algorithms such as face correction and background elimination, to extract the face and therefore limit the search space [23].
Many expression recognition researchers discover fiducial points such as face expressions after the face has been recognized. To make it easier to discern expressions, some academics have recommended rotating or frontalizing the face. Depending on whether the challenge is in greyscale, RGB, infrared, or any other modality, different works presented alternative techniques. This procedure is known as face alignment, and it has proven crucial for boosting the accuracy of facial expression identification.
Since some of these initiatives employ multimodal techniques, a fusion of these inputs is required. Depending on the stage at which the data is fused, there are a variety of techniques to link them. One option is to employ a two-stream architecture capable of fusing spatial and temporal data [8]. Middle fuse is another fusion approach that mixes several modalities in the intermediate levels [14].

Let's first browse some RGB Emotion Recognition works. The Cohn-Kanade (CK) database, first reported by Kanade et al. [23], is designed to recognize individual facial emotion-specified expression. Lucey et al. [12] later developed it into the CK+. These two datasets use seven core emotion categories: Anger, Contempt, Disgust, Fear, Happy, Sadness, and Surprise, as well as 30 face action units (AUs) that reflect facial muscle contractions.
To track the face and extract features, Kanade et al. [9] propose an Active Appearance Model (AAM). AAMs match a previously unseen source image containing the object of interest to a pre-defined linear shape model. Similarity-normalized shape (SPTS) and canonical appearance are two of the traits discovered (CAPP). They then utilize a support vector machines (SVMs)-based linear classifier to distinguish facial expressions and emotions [23].
The use of multi-modality is another approach to emotion recognition in literature. In this approach, the dataset employs multiple data inputs to recognize face expressions for emotion detection rather than only one modality, which is used RGB.
Wang et al [25] introduced the Natural Visible and Infrared Facial Expression database (NVIE). The dataset includes visual and thermal videos that were recorded simultaneously, as well as posed and spontaneous reactions. Liu et al. [11] provide a model, employing both thermal and visual footage, based on the USTC-NVIE dataset and the MMSE (BP4D+) database. Their method employs a fisher vector that is combined with local and global trajectory features. Based on the retrieved features, Gaussian mixture models are built.
Deep Architectures have been increasingly popular in recent years, owing to their shown success in outperforming earlier state-of-the-art techniques. By accounting for non-linear feature interactions, these deep architectures have overcome the limitations. The emoFBVP database of multimodal recordings was introduced by Ranganathan et al. [18]. Face, body gestures, voice, and physiological signals make up the multi-modality. Next, they propose a Convolutional Deep Belief Model (CDBN) for emotion recognition using this dataset [18]. Restricted Boltzmann Machines (RBMs) are an extension of Convolutional Restricted Boltzmann Machines (CRBMs). The RBMs are stacked to produce a convolutional deep belief network (CDBN). Layered generative models, or CDBNs, are generative models that are trained layer by layer. Ruiz-Garcia et al. [19] present a pre-trained deep CNN as a Stacked Convolutional AutoEncoder (SCAE). The SCAE is unsupervisedly taught in a greedy layer-wise manner. The model is trained using the Karolinska Directed Emotional Faces (KDEF) dataset [1] for face expression recognition.
The goal of this paper is to compare computer vision techniques for recognizing emotions in face expressions from image sequences. Models for static photos and image sequences are included in the datasets for comparison. It also applies to deep learning models with various inputs. The goal of this comparison is to determine the benefits and drawbacks of the various deep learning models that have been examined.

Paper organization:

The rest of this paper is organized as follows; Section 2 presents the used deep learning models. Section 3 is dedicated to the experiments results and discussions, and finally conclusions are drown in Section 4.

2. Preliminaries :

This section introduces the deep learning models that will be used for testing. The section is split into three parts. The CNN-based models are presented first, followed by the 3D CNN model, and finally the RNN models.

1. Convolutional Neural Networks :

Parkhi, Vedaldi, and Zisserman [17] from the University of Oxford designed and implemented the VGG-Face neural network. The CNN descriptors are calculated using a

VGG16-based CNN and tested on the Labelled Faces in the Wild [7] and YouTube Faces [26] datasets. A 224x224 face image must be used as the CNN's input. The VGG-Face is made up of 18 layers, including convolutional, pooling, and activation layers. The 18 layers are divided into 11 blocks, with Convolutional layers being the first eight, followed by non-linearities like ReLU and max pooling. Fully Connected Layers (FC) [17] are the last three blocks. Table 1 shows the network's architecture in detail.

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Table 1: Fully connected layers are listed as convolutional layers (from [17])

The following table presents the CNN parameters:

Table 2: CNN parameters

Parameters	values
Loss Function	Sparse Categorical Crossentropy
Optimizer	Adam

2. 3D Convolutional Network:

Tran et al. [22] introduced C3D, a 3D CNN model that can learn spatio-temporal properties. Figure 4.3 shows the model architecture, which includes 8 convolution, 5 max-pooling, and 2 fully connected layers, as well as a SoftMax output layer.

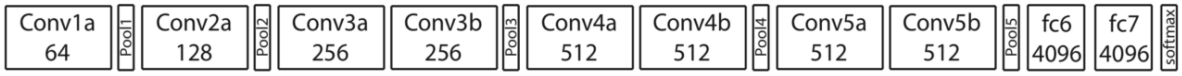


Figure 2:C3D Architecture (from [21])

Alberto Montes' GitHub project (github.com/albertomontesg) [10] provided the C3D model code implementation used in this work. Monte's implementation is a Keras [2] model, which was adapted from Tran et al [21]. Caffe [8] implementation for large-scale video classification.

The following table presents the C3D parameters:

Table 3: 3D CNN parameters

Parameters	values
Loss Function	Sparse Categorical Crossentropy
Optimizer	Adam

3. Recurrent Neural Networks
   - CNN + LSTM

There are two parts in this model. The first component is a CNN that extracts data from a still image. The second component involves layering an LSTM on top of the CNN to gather temporal data between frames. Each image is fed into the CNN, which extracts it at the seventh layer, which is a fully connected layer. The extracted features have a 4096 dimension. These characteristics are then sent into the LSTM.

   - CNN + GRU

This model is similar to the previous model, except it uses a GRU instead of an LSTM. In a two-part network, this model contains both temporal and information networks. The first is a CNN, and the second is a GRU with a CNN on top of it. Each image is fed into the CNN, which extracts features that are then fed into the GRU.
The GRU algorithm is a simpler version of the LSTM algorithm. It's fascinating to compare their performance.
The following table presents the CNN + RNN parameters:

Table 4: CNN + RNN parameters

Parameters	Values
Loss Function	Sparse Categorical Crossentropy
Optimizer	Adam
Number of Layers	3

Datasets:

To test the models, two public datasets were chosen. Both datasets contain videos of different participants making facial expressions that represent a variety of emotions. Both are used for visual sequences, but only one is used for hidden emotion.

1. SASE-FE

The SASE-FE database [15] was used in the first experiments. There are 643 different videos in this collection. A total of 50 people take part in the study. The participants are between the ages of 19 and 36. The dataset uses six universal expressions (figure 3): Happiness, Sadness, Anger, Disgust, Contempt, and Surprise. Each participant in the dataset has two emotional facial expressions, one genuine and one fraudulent.

Figure 3: Example of Six universal emotions in Fake vs Real Expressions dataset.

The following table presents a summary of the dataset content:

Table 5: SASE-FE dataset content details

Number of Subjects	50
Age	19-36
Gender	41% female, 59% male

Race	7.4% african, 77.8% caucasian, 14.8% asian
Number of Videos	643
Frames per second	100
Video Length	3-4 seconds

The dataset has been divided into two parts: training set and testing set. The training set comprises of 80% of the videos, while the validation test is made up of 10% of the videos and the test set is similarly made up of 10%. The training set has 40 participant videos, whereas the validation set contains 8 participant videos.

2. OULU-CASIA

The OULU-CASIA dataset was created by the University of Oulu's Machine Vision Group and the Chinese Academy of Sciences' National Laboratory of Pattern Recognition [27]. Figure 5.3 depicts the six emotions in this dataset (figure 4): happiness, sadness, anger, disgust, fear, and surprise. The participants ranged in age from 23 to 58. Males account for 73.8% of those who took part in the study.

Figure 4:Figure 3: Example of Six universal emotions in the OULU-CASIA dataset.

The following table presents a summary of the dataset elements:

Number of Subjects	80
Age	23-58
Gender	26.2% female, 73.8% male
Race	30 Chinese, 50 Finnish,
Number of Videos	480
Frames per second	25

The dataset has been divided into two parts: training set and testing set. The training videos account for 80% of the total number of videos, while the validation test accounts for 20%. The training set has 65 participant videos, whereas the validation set contains 15 participant videos.

3. Our proposed method:

This section presents the proposed method. Our method consists of three main parts: a 3D CNN module for extracting spatiotemporal convolutional features of facial expressions,

multiple temporal-LSTM modules for capturing the temporal dynamics of facial muscle motions, and a convolutional-LSTM module designed to capture multi-level features encoded in each layer of 3D CNN.
Let's now see in details each component:

a. Spatio-temporal convolutional features :

As facial expression is essentially a dynamic process, we try to directly extract spatio-temporal features of facial expression through simpler methods, namely activity detection, reading detection, gesture detection, etc. [28]. Unlike traditional CNNs that can only handle two-dimensional inputs, 3D CNNs take directly image sequences as input, and thus can literally capture its spatio-temporal features. 3D CNN starts from a sequence of known time-stamped images representing a class of facial emotions, processes the sequence through multiple layers of convolution and pooling, resulting in a collection of spatio-temporal data features which represents the characteristics of an expression.

b. Nested LSTM :

We suggest the so-called Nested LSTM, made up of the MSPP-stand (standardization), Temp-LSTM, and Conv-LSTM, to capture the multi-level features contained in the network's intermediate layers. While Temp-LSTM and Conv-LSTM capture the temporal dynamics and seize the multi-level characteristics or features contained in the separate convolutional layers of the network accordingly, MSSP-stand tries to standardize the spatio-temporal features of different sizes to the same dimension.
The next section presents the Nested LSTM components:

i. MSPP-stand

Since the LSTM inputs need to have the same size, it is impossible to directly input the spatio-temporal information obtained from multiple layers of the 3DCNN into the LSTM unit. To fill this gap, He et al. [29]'s spatial pyramid pooling network (SPP-net) was used as inspiration for our multidimensional spatial pyramid pooling normalization (MSPP-stand) operation. The MSSP-stand is used to standardize spatiotemporal features of various sizes to a single dimension. In order to create a feature vector with a fixed dimension given by the parameter n, we divide a 3D feature map of size $N \times a \times a$ into $N \times n \times n$ subregions (with n is equal to 2, 4, or 8) then summarize the responses inside each subregion using max pooling.

ii. Temp-LSTM and Conv-LSTM

The spatio-temporal features in each layer of the 3D CNN are converted to feature vectors of the same dimension following the MSPP-stand technique. As a result, it is appropriate to continue LSTM's study of the spatiotemporal features, which is an advanced RNN architecture used in facial expression recognition [30, 31] for sequential data analysis. By converting a series of inputs into a series of outputs, the commonly used LSTM can model temporal information, which in most cases can capture the correlations between the spatio-temporal properties retrieved by 3D CNN to a limited extent. Because it is challenging to include all of the appearance features, temporal dynamics, and multi-level features by merely integrating 3D CNN with LSTM, only a small number of conventional approaches based on LSTM fully utilize the information encoded in all the convolutional layers.
We use two LSTM modules, Temp-LSTM and Conv-LSTM, to handle the spatio-temporal data extracted by 3D CNN in order to address the aforementioned problems. A Temp-LSTM is created by stacking LSTM units for each feature vector corresponding to a certain

convolution layer, modeling the temporal dynamics of facial expressions. After that, a Conv-LSTM is built to accept the outputs of Temp-LSTMs as inputs, allowing for the seamless modeling of the needed multi-level features.
Assume that the 3D CNN has *l* total convolutional layers. Consequently, the steps in our proposed approach can be summed up as follows:

$$f_j = 3DCNN(x), j = 1, \dots, l$$
$$f_j^{mssp} = MSSP - stand(f_j), j = 1, \dots, l$$
$$h_j = \mathrm{T} - LSTM_j(f_j^{mssp}), j = 1, \dots, l$$
$$\mathrm{h} = \{h_1, h_2, \dots, h_l\}$$
$$\mathrm{o} = \mathrm{C} - \mathrm{LSTM}\ (\mathrm{h})$$

Where x stands for an image sequence, f_i for the 3D feature map created by the i^{th} convolutional layer of the 3DCNN, h_i for the feature vector from the i^{th} Temp-LSTM module, and o for the classification feature vector.

4. Implementation and discussion:

The steps involved in pre-processing are discussed in this section. The SASE-FE and OULUCASIA datasets both go through the same steps. Videos cannot be used as inputs to non-temporal models; instead, frames from videos must be extracted and used as inputs to the models. A vector containing a succession of these frames is fed into the temporal models as an input.

- Pre-processing :

Each frame of the videos was extracted as an image using a pre-processing technique. There are some frames that are useless because the videos begin with a neutral face expression and then the participant makes the facial expression that corresponds to the emotion. This is considered by the pre-processing, which only keeps frames from half of the video duration to 80% of the video duration.
This guarantees that the frames obtained convey the desired emotion.
Hassner et al [5] proposed a procedure called frontalization, which was used to perform a second change on the datasets. By transforming unconstrained perspectives to constrained, forward facing faces, this method rotates and scales the participant's face, limiting the variability of the location of the faces. Although frontalization can assist reduce variability, it also has significant disadvantages, particularly when the face is partially occluded.
Hassner also proposes soft symmetry, which allows for the estimation of occluded sections of the face when both parts of the face differ. Blending techniques are used to create a symmetrical image on both sides. Figure 5 exhibits a symmetrical image and a symmetrical image with soft symmetry.

Figure 5: Left image presents a Soft Symmetry frontalization process. Right image corresponds to an image with No Symmetry frontalization.

Figure 5 shows the Hassner method [5] for obtaining face landmarks, which consists of 68 fiducial points. These characteristics correspond to sites in the mouth, nose, and eyes, among other places.
These landmarks are then fed into the VGG-Face model as a second input. Because the VGG-Face only accepts photos as input, an intermediate fusion approach is necessary in the first fully connected layers.

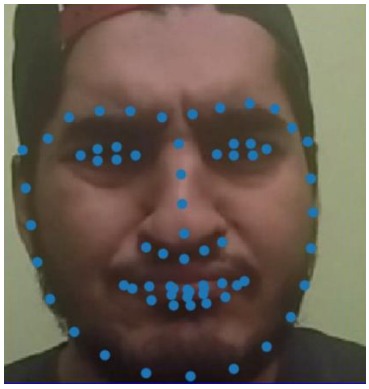


Figure 6: 68 fiducial points superimposed on the detected face

- Experimental Results

This section summarizes the findings from many experiments conducted on various dataset configurations with various model layouts. Only the test accuracy is shown for general purposes.

A. SASE-FE Emotion Results

The first series of experiments focused solely on classifying the six emotions, combining both actual and false feelings. Using the SASE-FE dataset, these experiments include fine-tuning the model. The goal is to evaluate various configurations and select the model with the best test accuracy.

- No Pre-Processing

The experiments are designed to see if adding three fully connected layers and/or a pooling layer after the convolutional layers enhances the CNN's performance. Other tests include freezing all the convolutional layers, freezing only the first few layers, and not freezing any of the layers at all.
The results are shown in the next table.

Table 6: Emotion No Pre-processing configurations Accuracy

Used configuration	Accuracy
No Freezed + No Pooling	0.2842
First Freezed + 3 Fully Connected + No Pooling	0.1970
All Freezed + 3 Fully Connected	0.4281
All Freezed + 3 Fully Connected + Average Pooling	0.4207
All Freezed + 3 Fully Connected + Max Pooling	0.4375
All Freezed + No Pooling	0.4250
All Freezed + Avg Pooling	0.4226
All Freezed + Max Pooling	0.4310

With a test accuracy of 0.4375, the optimal configuration is all Convolutional layers freeze with three Fully Connected layers at the end and Max Pooling layer at the end. Only this configuration will be used in the next experiments.

- Frontalization

The next experiment is to apply the frontalization pre-processed dataset after obtaining the optimal architecture from the previous tests. Both soft and no symmetry are visible in the experiments. The results are shown below.

Table 7: Emotion Frontalization configurations Accuracy

Used configuration	Accuracy
Soft Symmetry	0.454692
No Symmetry	0.594999

The difference between no symmetry and soft symmetry is enormous, as shown in the previous table. The difference amounts to over 15%.

Hassner et al. [6] noted how soft symmetry "may actually be unneeded and potentially even harmful; harming rather than boosting face recognition ability" in some circumstances. Soft Symmetry blends the identified facial features with the surface by modifying it. This mixing, however, is an approximation that can cause noise. Looking at the accuracies, it appears that performing soft symmetry has a significant impact on emotion identification. As a result, the next experiment will solely use no symmetry.

- Two-Stream CNN

According to the previous section's study, no symmetry leads to greater accuracy. Nonetheless, it will be fascinating to see if merging the no symmetry dataset with the extracted face may help boost the accuracy even further. The following experiments include employing a two-stream CNN, which combines a CNN with no symmetry dataset as input and another CNN with no pre-processing dataset as input. Both CNNs are fused before the fully connected layers to achieve this. The architecture is unchanged after the fusing layer. The results are presented below:

Table 8: Emotion Fuse-Stream configuration Accuracy

Used Configuration	Accuracy
Two-Stream	0.583234

The test accuracy for the two-stream CNN was 0.5832, which is lower than the 0.5949 for the No Symmetry CNN. The empirical evaluation with this dataset demonstrates that utilizing Soft-Symmetry is not useful to the emotion recognition task in the situations investigated here. Introducing Soft-Symmetry, as noted by Hassner et al. [12], may "create issues whenever one side of the face is obscured... rendering the final product unrecognizable." Finally, Soft-Symmetry photos will not be used in future models.

B. SASE-FE Hidden Emotion Results

The dataset was divided into fake and real emotions in this set of experiments, yielding a total of 12 classes. Each of the six emotions is divided into two categories: fake and real. Because there are now 12 courses, the test accuracy is projected to be substantially lower.

- Still Images Input

The VGG-Face is used in the first set of trials, with one experiment using photos of the face and the other using the frontalized face. Table 7.4 reveals that frontalization data has a modest advantage over only the face in terms of accuracy.
The second experiment is a two-stream CNN that employs both datasets, with one stream using frontalization and the other using face frontalization. One CNN improves the accuracy of the combined CNN significantly. There are almost four decimal points in the increase.
The third experiment is a middle fuse CNN that uses frontalization as an input and geometry data as output. The test accuracy of 0.2994 improves when the geometry is added to the base accuracy.

Table 9: CNNs Hidden configurations Accuracy

Model	Used configuration	Accuracy
CNN	Face Frontalization	0.2806 0.2866
Two-Stream CNN	Frontalization + Face	0.3206
CNN + Geometry	Frontalization + Geometry	0.2994

- Image Sequences Input

The next set of experiments differs from the prior ones in that they now include temporal data. These tests are fed a 5-frame vector containing the range of articulations that each participant uses to express the emotion.
A 3D CNN is used in the first experiment of this type. To get decent performance, most 3D CNNs require a large amount of data. This performance issue is well-known, but the experiment aims to investigate if a 3D CNN can train even with a short dataset. The accuracy of the exam was 0.1281 percent. Despite the decreased accuracy, it is crucial to note that this model incorporates temporal information from image sequences. The model does not employ any fine-tuning and starts the learning process from the beginning.
The fine-tuned model developed with the frontalization preprocess provided in table 7.5 is used in the next tests. Two models were trained, one of which was fine-tuned with a 5-frame input vector. The second step is to extract features from a pre-trained CNN with a vector size of 4096; PCA is then applied to the feature vector to minimize its size, with just the first 100 eigenvectors used.
The second experiment combines a CNN with an LSTM. It's fascinating to note that feature vectors outperform picture vectors in the results. The conclusion is that PCA aids in obtaining the most variables, while the LSTM learns the differences that distinguish each emotion.
The final experiment is the CNN, but this time with a GRU on top. Surprisingly, the model with image vectors as input performs horribly. One thing to keep in mind is that SASE-FE frames begin with a neutral face. The 5 frames picture vector is relatively modest to display the emotion's complete spectrum of expression. The features vector model, on the other hand, has a very high accuracy. Higher even than the LSTM model.

Table 10: Temporal Hidden configurations Accuracy

Model	Used configuration	Accuracy
3D CNN	Frontalization	0.128125

CNN + LSTM	Features Image	0.159200 0.148684
CNN + GRU	Features Image	0.183311 0.084134

C. OULU-CASIA Results

The following experiments were conducted specifically to classify the OULUCASIA dataset's six emotions.

- Still Images Input

The first set of experiments involves fine-tuning the VGG-Face using photos that have not been pre-processed; the second set of tests involves frontalization pre-processed images. Preprocessed photos outperformed non-processed images by nearly 0.03 percent.
The accuracy of the models is improved by this pre-processing. This is because frontalization normalizes the faces and aids the model in learning the differences between emotions; normalizing the images reduces noise, therefore the performance improves in this situation.

The second experiment employs a two-Stream CNN to investigate if combining both the face and frontalization datasets improves the accuracy of the prior two. Frontalization received a 0.2659 percent in earlier studies. With a higher score of 0.2737 percent, the two-stream CNN outperforms the model. This ensures that the model learns all feasible information from the face, even information that may have been lost during frontalization pre-processing.

The third experiment employs an intermediate fusion technique, employing one CNN before concatenating data from the face's geometry after the final convolutional layer. The geometry is made up of 68 fiducial points that are normalized in a new center. It's fascinating to note that this model beats all previous studies by a significant margin, with a test accuracy of 0.4411 percent. The increase is 0.17 percent above the previous highest model.
The following table presents the results.

Table 11: Temporal configurations Accuracy

Model	Configuration	Accuracy
CNN	Face Frontalization	0.2386 0.2659
Two-Stream CNN	Face + Frontalization	0.2737
CNN + Geometry	Frontalization + Geometry	0.4411

- Image Sequence Input

The next experiments differ from the previous ones in that they now use temporal data from image sequences. In this situation, the input is a frame sequence.
Each frame is made up of five consecutive photos taken from the videos.
A 3D CNN is used in the first experiment. Although 3D CNNs are known to require a lot of data to learn well, this model's performance is comparable to the other temporal models shown below.
The fine-tuned model developed with the frontalization preprocess provided in table 11 is used in the following tests. Two types of inputs are tested: image vectors and features vectors. They all follow the same steps.

In the second experiment, a CNN is combined with an LSTM. The picture vector input is noticeably more accurate than the feature vector in this scenario. The difference is less than 0.02%.
A CNN with a GRU built on top is the third and last experiment. The image vector has a greater test accuracy than the features vector, as with the CNN+LSTM. The GRU features vector, on the other hand, has a larger disparity between features and image vectors, at roughly 0.05 percent.

Table 12: Accuracy of Temporal Test

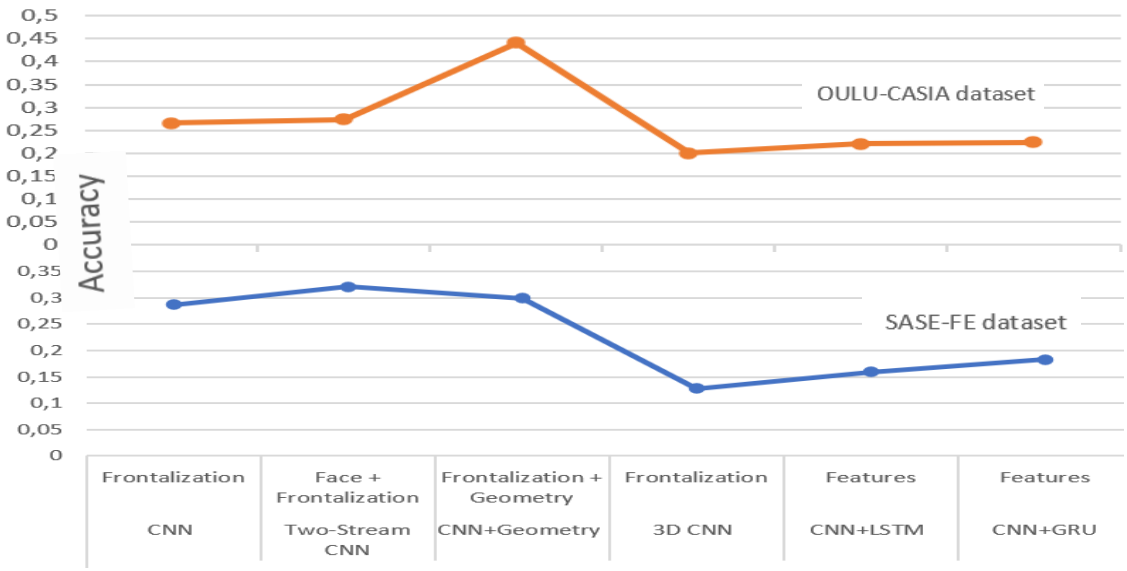
Model	Configuration	Accuracy
3D CNN	Frontalization	0.2000
CNN + LSTM	Features	0.2062
	Image	0.2209
CNN + GRU	Features	0.1797
	Image	0.2241

D. Discussion

For both datasets in table 13, this section discusses the best model for each test. The VGG-Face and C3D models were investigated in this work. With multi-modality, recurrent neural networks, the CNN-based models is improved even further. A voting mechanism is considered when evaluating the models.
The basic CNN is used in the first experiment with pre-processing and no pre-processing data; in both datasets, the frontalization method has a higher accuracy. Frontalization is the process of mapping the face in a confined, forward-facing position. This reduces the variability in face location in the dataset, allowing the models to focus solely on learning the variability of emotion recognition.
In both datasets, the accuracy of the 2-Stream CNN is higher. However, it is the best image model in SASE. Overall, including both inputs aid the models in learning the difference that may have been lost during the frontalization process, which might result in face misalignment.

Table 13: Summary of used models' accuracy

Dataset	Model	Best used Configuration	Accuracy
SASE-FE dataset	CNN	Frontalization	0.2866
	Two-Stream CNN	Face + Frontalization	**0.3206**
	CNN+Geometry	Frontalization + Geometry	0.2994
	3D CNN	Frontalization	0.1281
	CNN+LSTM	Features	0.1592
	CNN+GRU	Features	**0.1833**
OULU-CASIA dataset	CNN	Frontalization	0.2659
	Two-Stream CNN	Face + Frontalization	0.2737
	CNN+Geometry	Frontalization + Geometry	**0.4411**
	3D CNN	Frontalization	0.2000
	CNN+LSTM	Image	0.2209
	CNN+GRU	Image	**0.2241**

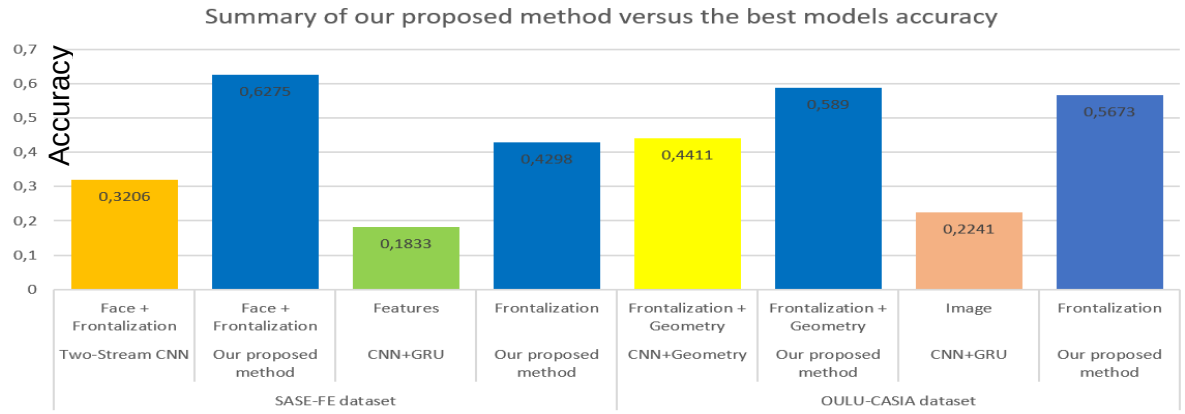


The geometry data improves the performance of middle fusion models in both the SASE-FE and OULU-CASIA datasets when compared to just one input. However, the OULU-CASIA boost is substantially bigger than the SASE-FE boost. OULU earns a 0.17 percent rise compared to SASE's 0.1 percent. All the models evaluated have a higher boost than OULU's Geometry model.

Afterwards, we compare the best models with our proposed method. The results are shown below:

Table 14 :12 Summary of our proposed method versus the best models accuracy

Dataset	Model	Best used configuration	Accuracy
SASE-FE dataset	Two-Stream CNN	Face + Frontalization	0.3206
	Our proposed method	Face + Frontalization	**0.6275**
	CNN+GRU	Features	0.1833
	Our proposed method	Frontalization	**0.4298**
OULU-CASIA dataset	CNN+Geometry	Frontalization + Geometry	0.4411
	Our proposed method	Frontalization + Geometry	**0.5890**
	CNN+GRU	Image	0.2241
	Our proposed method	Frontalization	**0.5673**



The results above show that our method outperforms the best models in all configuration methods and in both datasets. For SASE-FE dataset, as we can see our method achieves an

averaged accuracy of 62% in the first set of experiments and 42% in the second set. The same for OULU-CASIA dataset, which performs 58% in the first set and 56% in the second set. The results of our method greatly exceeded those of the other methods, 30% compared to Two-Stream CNN, and 24% compared to CNN+GRU for the SASE-FE dataset.

5. Conclusion

In this study, we proposed a new technique method for facial expression recognition. This method seeks to take into account the multi-level characteristics stored in the intermediate layers of the network, as opposed to the majority of existing deep learning-based methods, which derive the classification results based on the outputs of the last fully-connected layer. Deep learning models such as CNNs, 3D CNNs, and RNNs are also examined in this work. On two datasets, SASE-FE and OULU-CASIA, the evaluation focuses on the job of emotion recognition through face expression in image sequences. These datasets contain data for six basic emotions, which are presented in a series of videos and were utilized for emotion classification and actual and concealed emotions using image sequences, respectively.
The evaluation clearly indicated that applying face frontalization for data preparation is superior to doing nothing at all.
Multi-modal models based on the geometry data of the face were used to improve the basic model. The original model was also improved by employing a 2-Stream CNN. Even though both improved the base model, no clear winner emerged.
In the experiment, a 3D CNN model was used, and its use in emotion recognition was demonstrated satisfactorily. Furthermore, the GRU models outperformed the 3D CNN and LSTM spatio-temporal models with image sequence inputs. However, it was impossible to say definitively if utilizing CNN extracted feature vectors as RNN inputs was better than using picture vectors.
Experiments show that our proposed method outperforms the best models that we got from the first experiments.

Bibliographie

[1] M. G. Calvo and D. Lundqvist. Facial expressions of emotion (KDEF): Identification under different display-duration conditions. Behavior Research Methods, 2008.
[2] F. Chollet et al. Keras, 2015.
[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005.
[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. 2016.
[5] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective Face Frontalization in Unconstrained Images.
[6] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.
[7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.
[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding.
[9] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive Database for Facial Expression Analysis.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale
Video Classication with Convolutional Neural Networks. In Large-scale Video Classication with Convolutional Neural Networks, 2014.
[11] P. Liu and L. Yin. Spontaneous Thermal Facial Expression Analysis Based On TrajectoryPooled Fisher Vector Descriptor.
[12] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression.
[13] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding Facial Expressions with Gabor Wavelets.
[14] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. ModDrop: Adaptive multi-modal gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.
[15] I. Ofodile, K. Kulkarni, C. A. Corneanu, S. Escalera, X. Baro, S. Hyniewska, J. Allik, and G. Anbarjafari. Automatic Recognition of Deceptive Facial Expressions of Emotion. 2017.
[16] M. Osadchy, Le Cun. Yann, and M. L. Miller. Synergistic Face Detection and Pose Estimation with Energy-Based Models. The Journal of Machine Learning Research, 2007.
[17] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In Procedings of the British Machine Vision Conference 2015, 2015.
[18] H. Ranganathan, S. Chakraborty, and S. Panchanathan. Multimodal Emotion Recognition using Deep Learning Architectures.
[19] A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade. Stacked Deep Convolutional Auto-Encoders for Emotion Recognition from Facial Expressions.
[20] N. Sarode and S. Bhatia. Facial Expression Recognition. International Journal on Computer Science and Engineering, 02(05):1552–1557, 2010.
[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, 2015.
[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, 2015.
[23] H. C. Vijay Lakshmi and S. PatilKulakarni. Segmentation Algorithm for Multiple Face Detection in Color Images with Skin Tone Regions using Color Spaces and Edge Detection Techniques. IEEE Signal Acquisition and Processing, 2010. ICSAP'10. International Conference on, pages 162–166, 2010.
[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.
[25] S. Wang, Z. Liu, S. Lv, Y. Lv, G. Wu, P. Peng, F. Chen, and X. Wang. A natural visible and infrared facial expression database for expression recognition and emotion inference. IEEE Transactions on Multimedia, 2010.
[26] L. Wolf, T. Hassner, and I. Maoz. Face Recognition in
Unconstrained Videos with Matched Background Similarity.
[27] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietik¨ainen. Facial expression recognition from near-infrared videos. Image and Vision Computing, 2011.
[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489-4497

[29] He, K., Zhang, X., Ren, S., Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8691. Springer, Cham.
[30] Zhao, X. et al. (2016). Peak-Piloted Deep Network for Facial Expression Recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9906. Springer, Cham.
[31] Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, Zhiwu Huang, and Xilin Chen. 2014. Combining Multiple Kernel Methods on Riemannian Manifold for Emotion Recognition in the Wild. In Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14). Association for Computing Machinery, New York, NY, USA, 494–501.