# Temporal Stochastic Softmax for 3D CNNs: An Application in Facial Expression Recognition

Théo Ayral[1], Marco Pedersoli[1], Simon Bacon[2], and Eric Granger[1]
[1] LIVIA, Dept. of Systems Engineering, École de technologie supérieure, Montreal, Canada
[2] Dept. of Health, Kinesiology & Applied Physiology, Concordia University, Montreal, Canada
theo.ayral.1@ens.etsmtl.ca, {marco.pedersoli, eric.granger}@etsmtl.ca
simon.bacon@concordia.ca

## Abstract

*Training deep learning models for accurate spatiotemporal recognition of facial expressions in videos requires significant computational resources. For practical reasons, 3D Convolutional Neural Networks (3D CNNs) are usually trained with relatively short clips randomly extracted from videos. However, such uniform sampling is generally sub-optimal because equal importance is assigned to each temporal clip. In this paper, we present a strategy for efficient video-based training of 3D CNNs. It relies on softmax temporal pooling and a weighted sampling mechanism to select the most relevant training clips. The proposed softmax strategy provides several advantages – a reduced computational complexity due to efficient clip sampling, and an improved accuracy since temporal weighting focuses on more relevant clips during both training and inference. Experimental results obtained with the proposed method on several facial expression recognition benchmarks show the benefits of focusing on more informative clips in training videos. In particular, our approach improves performance and computational cost by reducing the impact of inaccurate trimming and coarse annotation of videos, and heterogeneous distribution of visual information across time.*

## 1. Introduction

Deep Learning (DL) models have been successfully applied in many visual recognition tasks, including detection, classification, tracking, and segmentation, and currently achieve state-of-the-art (SOTA) performance on several image-based benchmarks [4, 20]. Spatiotemporal recognition, in which appearance and motion features play a complementary role, remains a challenging problem in real-world applications. While many DL models are based on spatial feature extraction, specialized mechanisms are needed to manage spatiotemporal data.

In facial expression recognition (FER), the output produced by a 2D Convolutional Neural Network (CNN), *e.g*. VGG or ResNet models, in response to a sequence of frames is typically aggregated or processed by a recurrent neural network which, as seen in AVEC and EmotiW competitions [7], can provide high-level performance [28, 29, 31]. In contrast, 3D CNNs can process a clip as a single input, and jointly analyze appearance and motion to encode spatiotemporal relationships [5, 6, 38]. 3D CNNs have also been integrated as components in FER systems, not always performing well on their own with limited data [11, 31, 51]. However, recent studies have shown the relevance of 3D CNNs for video recognition, highlighting the importance of adopting appropriate training strategies, and integrating extra training data through transfer learning [4, 15, 60]. Along these lines, our work is focused on efficient training strategies for 3D CNNs.

The computational requirements of 3D CNNs represent an important challenge in video recognition applications. Motion adds an extra dimension to model representations (*i.e*. inputs and feature tensors are much larger), and significantly increases the computational and GPU-memory requirements for training a DL model. To address this issue, state-of-the-art 3D models [4, 15] are trained with short, randomly sampled training clips, which is a stochastic approximation of temporal average pooling (see Section 3.1). At inference time, as memory requirements are reduced, temporal average pooling is used. Training with short clips has the advantage of mitigating issues related to GPU memory and video length. The efficiency of this technique in practice suggests that modeling long-range temporal dependencies is not needed in most cases to achieve accurate spatiotemporal recognition. However, a uniform selection of training clips from real-world videos, enforcing equal importance to all frames, raises other issues for training and inference. Clips extracted from a video captured "in the wild" are not all equally relevant, because of inherent characteris-

tics of the tasks or noise in capture conditions. Assigning a global video label to short clips generates noise, and some clips can even be misleading because they do not represent the general aspect of the video. This has important implications for both training and testing phases. For instance, in FER applications, expressions captured in videos vary significantly depending on subjects and capture conditions (*e.g*. illumination and pose). As a result, parts of a video may not contain any relevant information. Also, the expression intensity in FER videos varies through different states (typically onset, apex and offset [19, 63]), and not all these states provide the same discriminative power for spatiotemporal recognition. Moreover, most of the video is typically dominated by neutral state and does not correspond to the sequence-level expression label. To avoid investing computational resources on training with uninformative clips, and to reduce the performance limitation incurred by training on incorrectly labeled clips, it is preferable to learn to sample the most relevant clips.

**Contribution.** In this paper, we present a new temporal stochastic softmax method to efficiently train 3D CNNs for spatiotemporal recognition, with videos of arbitrary length and sequence-level labels. This method leverages a stochastic approximation of softmax temporal pooling for efficient sampling and learning of relevant training clips. Softmax sampling weights are estimated iteratively during training, with lower variance than the REINFORCE method [57], thereby leading to better results. Although uniform clip sampling is often used for its simplicity, empirical results on several FER datasets show that the proposed temporal stochastic softmax provides a better cost-effective training approach for 3D CNNs, achieving a higher level of accuracy, and a shorter training time.

## 2. Related Work

### 2.1. Spatiotemporal models for FER

Given the high level of performance achieved on visual recognition tasks, 2D CNNs have been applied to video classification. These models extract a hierarchy of spatial features from each RGB frame independently [20]. The resulting set of frame-level representations are then aggregated to summarize the entire video. Although this approach can already provide good recognition accuracy [2, 50], it does not leverage temporal information in the data representation. Better suited methods have been proposed for video classification. Recurrent neural networks (RNNs) are commonly used to recognize temporal patterns in the sequence of high-level frame representations produced by a 2D CNN [10, 18, 35]. Alternatively, two-stream convolutional networks [42] explicitly complement the appearance analysis, augmenting still RGB frames with precomputed optical-flow stacks describing low-level motion between frames [39, 49]. 3D CNNs [45, 47] unify the temporal and spatial analysis, treating videos as data volumes. 3D convolutional layers extract a hierarchy of spatiotemporal features from the RGB frame sequence. These powerful models are heavier to operate, because of their high number of parameters, requiring more training data and computational resources. Even so, 3D models are useful in FER. As an example, the small C3D [47] helps for the classification of relatively small video datasets, and can even be used as a deep spatiotemporal feature extractor, combined with RNNs as in [27].

On action recognition, results from Carreira and Zisserman [4] and Hara *et al*. [15] show that 3D models integrating efficient transfer-learning, or simply pretrained on recent large video datasets perform better than 2D CNNs combined with RNNs. Indeed, the progressive augmentation in available training data was followed by huge improvements in the performance of 3D models on action recognition tasks [22]. However, in the context of facial expression recognition (FER), datasets like Acted Facial Expressions in the Wild [8] (AFEW) are still smaller by an order of magnitude, limiting the performance of 3D CNNs. Leveraging appearance and motion analysis jointly is still an issue [9], although spatiotemporal methods have been studied for a long time [17]. In the EmotiW 2017 challenge (based on the AFEW dataset), the second place was achieved without using spatiotemporal features [23]. This direction was followed by the OL_UC team of Vielzeuf *et al*. [50] who made a point of not using any motion features for simplicity and efficiency.

However, experiments in the study of Valstar and Pantic [48] suggest that temporal analysis should play a role for FER, not only on the high-level description of expression dynamics throughout the video but also in the detection of local motion features. In the more general context of action recognition, Sevilla-Lara *et al*. [40] and Huang *et al*. [16] evaluated the importance of motion analysis, and the ability of state-of-the-art models to capture it. They also hypothesized that long-term patterns are not necessary for action recognition, but a wide receptive field helps to capture the most relevant frames [30]. Our work is in line with these considerations, as the proposed stochastic softmax consists of an efficient weighting of short video clips for 3D CNNs.

### 2.2. Efficient weighting of clips

Works related to ours are ones that study techniques to improve training with better sample selection, or to improve inference using a weighted aggregation method. Temporal aggregation of features for 3D CNNs is usually performed with average pooling [4, 15, 9, 49] (more details are provided in Section 3.1). The theoretical analysis of Boureau *et al*. [3], as well as experimental results [33], shows that

max and average pooling have different ranges of expertise, depending on the input size and the sparsity of features. To help find the most adapted type of pooling, softmax is proposed as a parameterizable generalization of these two pooling techniques.

The pooling strategy used for temporal feature aggregation also has a great influence on training, by deciding which part of the input will generate gradients. For 3D CNNs, as training is performed on short clips (usually 8 to 64 frames), the receptive field of the aggregation mechanism is limited, and the pooled features have a different distribution than at test time, with longer videos. Consequently, other weighting methods have to be developed. Studies on importance sampling [21, 52] have made clear that all samples do not have the same relevance to the training process. However, for most video classification datasets (*e.g*. AFEW [8] and Kinetics [22]), labels are not available for each of the frames or clips, but only at the video level. Related issues are discussed by Zhu *et al*. [64], as they consider short-clip training as a weakly supervised learning problem within the action recognition task with video-level labels. In this multiple instance learning (MIL) framework, a bag of several clips is fed to the model and temporal max pooling is used to learn only from the best scored clip. This is a way of providing better training data by selecting the most informative temporal windows *a posteriori*. This approach still requires the use of several clips per sample at each epoch, which is problematic for 3D models. To be able to select clips before evaluating them with the entire model, this method is complemented with a motion metric computed offline. In the context of action recognition, the hypothesis is that relevant clips are the ones with more motion. This does not hold for FER, so our method is purely based on classification scores, and we compute them online with a single training-clip per sample for each epoch, to save computation with the 3D CNN.

Also to cope with capture, trimming and labelling noise, the weighted C3D [51] integrates a softmax layer to give more importance to relevant clips during training. All windows are evaluated in the early epochs of training and their scores are used to weight the training loss, reducing the effect of uninformative or wrongly labeled clips on the model parameter updates. The weighting strategy is amplified throughout training, from average to max. This principle of weighted training constitutes a basis of our work, with the idea of identifying relevant training clips for 3D CNNs. Yet we replace the loss weighting by a stochastic sampling mechanism to avoid computing gradients that would be inhibited by the softmax, and we rethink the distribution estimation method to remove the computational overhead of evaluation. This allows us to train a model with more parameters than the small C3D.

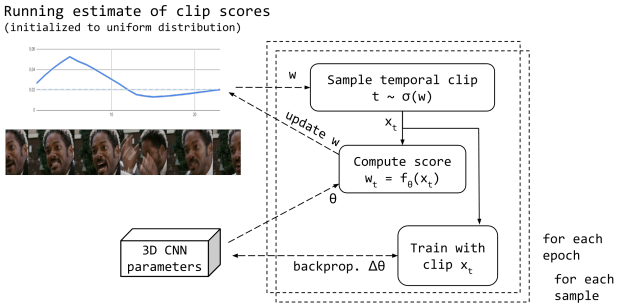


Figure 1. Illustration of weighted clip sampling for temporal stochastic softmax training. Within each video, training clips are sampled based on the softmax of their classification-score distribution. Running estimates of clip scores are updated at every iteration with the classification score of the selected clip.

Another interesting mechanism is the SCSampler [24], trained to select the best clips to feed to the classifier at test time. This external and light-weight sampler is trained to predict the relative saliency of clips. By running the classifier on a few sampled clips instead of the entire video, the computational cost at inference time is reduced and classification accuracy is improved. This study shows experimentally that the classification score is a good metric of informativeness of a clip. Our approach is related to their Oracle Sampler, which ranks clips according to their scores for the target class. We adapt this mechanism for training-clip selection, and propose an approach to avoid dense application of the classifier through iterative sampling (Section 3.3).

## 3. Proposed Approach

The main objective of this work is to perform a pooling operation that can select the most important frames of a video sequence. As presented in Section 2.1, softmax pooling seems to be the right candidate because it assigns a specific weight to each short clip of the video. However, standard softmax pooling requires an evaluation of all short clips of a video sequence. This is unfeasible for large 3D models as they require a large amount of memory and computation. In this section we show how to approximate softmax pooling during training such that it requires the evaluation of only one short-clip per video at each training iteration. This makes the training much lighter, while optimizing the same objective function in expectation.

An overview of the stochastic softmax training process is shown in Figure 1. In Section 3.1 we present the motivations and challenges of integrating softmax pooling into the usual short-clip training framework. Then, Section 3.2 discusses solutions for estimating temporal probability distributions iteratively from short clips. Finally, Section 3.3 provides the details of our stochastic softmax pooling, combining both training-clip sampling and softmax pooling.

### 3.1. Softmax pooling with short-clip sampling

Our objective is to minimize the loss $\mathcal{L}$ of a parameterized classifier $f$ on the training dataset. For each video $x$ with label $y$, we minimize $\mathcal{L}(f(x), y)$. For video classification, a typical example of such loss is cross-entropy. If using temporal average pooling, the loss can be written as $\mathcal{L}(\frac{1}{T}\sum_{t=1}^{T} f_t(x), y)$ in which $f_t(x)$ represents the learned features associated to temporal position $t$ in a video of duration $T$. Assuming that the temporal receptive field of the network is limited or that the features extracted for time $t$ only depend on a small temporal neighbourhood (a clip), we can compute the loss as $\mathcal{L}(\frac{1}{T}\sum_{t=1}^{T} f(x_t), y)$ where $x_t$ is a clip associated to time $t$. This assumption is implicitly or explicitly used in most of the recent work on 3D CNNs for video classification [4, 15, 9, 49] because it enables the use of an approximation of the loss:

$$\mathcal{L}(\frac{1}{T}\sum_{j=1}^{T} f(x_j), y) \approx \mathcal{L}(f(x_t), y), \quad t \sim \mathcal{U}(1, T). \quad (1)$$

The loss is computed by sampling different clips throughout training. For each iteration, the loss of a video is approximated using a single clip that is uniformly sampled from each video. This produces significant reduction in computational complexity of each training step and in GPU memory requirements necessary to make the training of 3D CNNs possible. Here we show that this sampling technique with cross-entropy loss is an upper-bound of the real loss. Indeed, cross-entropy loss is convex, and based on Jensen's inequality, the averaged loss computed on all clips $\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(f(x_t), y)$ is an upper-bound of the cross-entropy of the averaged-pooled features $\mathcal{L}(\frac{1}{T}\sum_{t=1}^{T} f(x_t), y)$:

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(f(x_t), y) \geq \mathcal{L}(\frac{1}{T}\sum_{t=1}^{T} f(x_t), y) = \mathcal{L}(f(x), y). \quad (2)$$

The average of clip-level losses is an empirical estimation of the expected loss $\mathbb{E}[\mathcal{L}(f(x_t), y)]$. Instead of computing the entire sum, the expected loss is approximated with a single sample $t$ uniformly sampled as in Eq. (1). Thus, in expectation, the same loss is optimized during training. At test time, memory is less problematic as gradients are not computed, so inference is computed as the average of all video clips, thus a temporal average pooling.

Although this form of training with uniformly sampled clips can be effective, it restricts the temporal pooling to the average pooling strategy. Our work addresses this issue by proposing the more general softmax strategy in the training-clip sampling framework. Temporal average pooling assumes that each sub-region (clip for videos) of the pooled features contains information that is important for the task. It is expected to perform well when videos are short and entail exactly the category that we want to classify [3, 33]. On the other hand, when a video is longer, complex and with possible noise, max pooling is expected to work better. In general, the optimal level of importance for the different parts of a video is unknown, as it depends on the task and the actual data. In this paper, we propose to use weighted pooling in which a weight $p_t$ is associated with each temporal position $t$ of the video. This resembles an attention mechanism [61, 1, 13, 26, 34], but instead of estimating attention weights with a learned layer, $p_t$ is computed as a temporal softmax of the score associated to a given clip. The relative importance attributed to each clip depends on its classification score:

$$f(x) = \sum_{t=1}^{T} \frac{\exp(\gamma f(x_t))}{\sum_{j=1}^{T} \exp(\gamma f(x_j))} f(x_t) = \sum_{t=1}^{T} p_t f(x_t). \quad (3)$$

The softmax operator is parameterized by $\gamma$, the inverse temperature. When $\gamma = 0$, the weight vector $p = (p_1, p_2, ...p_T)$ is equal to the center of the unit simplex, with $p_t = \frac{1}{T}\ \forall t$, so the operator is equivalent to average pooling. In contrast, when $\gamma \to +\infty$ all the weight will be assigned to the highest scoring $t$, as in max pooling. Equation (3) is therefore a generalization of max and average pooling, parameterized by a factor $\gamma$. As with uniform clip sampling, we provide an upper-bound on the training loss obtained with softmax pooling:

$$p_t \sum_{t=1}^{T} \mathcal{L}(f(x_t), y) \geq \mathcal{L}(p_t \sum_{t=1}^{T} f(x_t), y) = \mathcal{L}(f(x), y). \quad (4)$$

The weighting factor $p_t$ of softmax pooling, from Eq. (3), becomes a sampling probability distribution in softmax short-clip training. Indeed, for each video sample $x$, instead of weighting losses obtained from clips sampled uniformly, we directly weight the sampling distributions of clips. In this way we select clips that are more important for training than with uniform sampling.

### 3.2. Estimation of the sampling distributions

In Eq. (3), we see that in order to compute $p_t$, $f(x_t)$ must be evaluated on all clips $t$ of a video, which is computationally expensive and consumes considerable memory. We seek to avoid this issue with a stochastic sampling strategy. Thus, instead of computing $p$ as in Eq. (3), we introduce a new variable $q = (q_1, q_1, ..., q_T)$ that estimates $p$ for each video $x$. During training, for each video, $q$ is defined as minimizing the loss. A straightforward way to estimate $q$ is by using REINFORCE [57]. We consider the loss as an expectation over time, sampled with $q$. Thus, its gradient will be:

$$\nabla_q \mathop{\mathbb{E}}_{t\sim q}[\mathcal{L}(f(x_t), y)] = \mathop{\mathbb{E}}_{t\sim q}[\mathcal{L}(f(x_t), y)\nabla_q log(q_t)]. \quad (5)$$

Unfortunately, the gradients estimated with REINFORCE have high variance and, even when using a baseline of the expected cumulative reward, the updates of $q_t$ are too noisy. The potential benefits of sampling are therefore lost by a poor estimation of $q$. Results and limitations of the estimation of sampling distributions with REINFORCE are discussed in Section 4.2. As the distribution parameters are essentially pushed to favour the selection of high scoring (low loss) training clips, it is possible to "shortcut" the REINFORCE optimization by building the sampling distributions directly from the clip scores. Thus, we propose to estimate $q$ in a close form, without the use of gradients. Since $p_t$ is calculated as softmax of $\gamma f(x_t)$, it is possible to store the values of $w_{x,t} = f(x_t)$ directly, and apply softmax when an estimation of $q$ is needed to select the training clip. The probability distribution $q$ is therefore computed from running estimates of scores evaluated at different iterations. This approach is quite simple, does not rely on a noisy estimation of the gradients, and works well in practice. During training, softmax sampling maximizes the classification score for the correct label (providing relevant clips to the model), while keeping diversity in clip sampling to avoid overfitting [12].

### 3.3. Implementation of stochastic softmax

Videos are classified by convolving the 3D CNN in the temporal dimension, over all possible overlapping windows, and implementing a temporal pooling mechanism on the resulting clip-level scores [4]. We combine this evaluation scheme with single-clip extraction during training, as only one clip is used from each video at every epoch. For more details on the implementation, refer to the Supplementary Material.

**Clip Sampling.** The sampler $S$, extracts a clip of $F$ contiguous frames at temporal position $t$ from a video $x$ of arbitrary length $L$. At every epoch of training, we construct batches of training clips. One clip is sampled from each training video. Let $w_x$ be the temporal sequence of $N = L-F+1$ classification-score estimates corresponding to the temporal responses of the classifier convolved over $x$. Then $w_{x,t}$ is the estimated classification score for training clip $x_t$. This score will be the base of our clip weighting. Temporal softmax sampling follows the formula:

$$p(S(x) = x_t) = \frac{\exp(\gamma w_{x,t})}{\sum_{n=1}^{N} \exp(\gamma w_{x,n})} , \quad \forall x_t \subset x. \quad (6)$$

The principle of stochastic softmax training is summarized in Eq. (6). At test time, the relative importance attributed to each clip through temporal pooling is defined as the weighting factor computed from the softmax function in Eq. (3). During training, the temporal weighting is implemented as a sampling probability which translates into a frequency of occurrence in the training iterations. Instead of classifying entire videos and applying temporal weights to the loss afterward, computation is exclusively focused on the selected clip.

**Distribution updates.** The proposed approach stores running estimates $w$ of the score distribution of each training video, and updates them at every epoch, as video clips are sampled. Through iterative clip sampling, we obtain information on the temporal classification score distribution of each video and use it to build its temporal clip selection probability distribution. An overview of the training process is presented in Algorithm 1.

**Algorithm 1:** Stochastic softmax training

```
Initialize w with uniform distributions;
foreach epoch do
    foreach video x in training set do
        compute the clip sampling distribution
          from w_x with Eq. (6) ;
        sample clip x_t, with t = S(w_x) ;
        compute the score for the correct class y :
          w_{x,t} = f(x_t)[y] ;
        update w_x around sampled location t ;
        train with clip x_t by back-propagation ;
    end
end
```

**Training phases.** Efficient training-clip sampling is highly dependent on the accuracy of the temporal distributions. Therefore, we implement several mechanisms to bootstrap the distributions, to make them representative of the informativeness of clips as early as possible during training, without introducing heavy computational overhead. We decompose the training process into three simple steps relative to the sampling mechanism: first, warm-up with uniform sampling and no distribution updates, then, exploration with deterministic sampling and initialization of distributions, and finally exploitation with softmax weighted sampling and distribution updates as described above.

## 4. Results and Discussion

In this section, we perform an ablation study of the method on emotion recognition with AFEW and validate on two datasets of pain detection to evaluate the generalization power in the scope of facial expression recognition. This section only contains the most important results and illustrations to support the paper. For additional experimental results, see the Supplementary Material.

### 4.1. Experimental methodology

**AFEW.** The Acted Facial Expressions in the Wild dataset of emotion recognition [8] was used to evaluate the proposed and reference training methods. The task is to classify video samples by assigning each of them a single emotion label from the six universal emotions (Anger, Disgust, Fear, Happiness, Sad & Surprise) and Neutral. The performance metric is the classification accuracy (video-level rank-1 accuracy). We do not consider the audio information in our study. The Training set contains 773 samples from 67 movies, with 228 actors. The Validation set contains 383 samples from 33 movies, with 134 actors. The dataset has been constructed in a subject independent manner. AFEW video duration ranges from 0.6s to 5.4s (between 16 and 128 frames), with an average of 2.5s. When using the SeetaFace Engine to crop and align faces [59], the average bounding box of source crop has size 265×265 (from the original 720×576 image). Models are evaluated on the Validation set of AFEW (the Test set being reserved for the EmotiW competition). Model training and hyper-parameter validation were performed on random splits of the Training set. Created from movie samples, AFEW provides close to real-world data, with a wide range of challenges due to the variation in head poses and movements, illumination and backgrounds. Additional noise comes from camera motion, sometimes causing occlusion. Some of these issues are well addressed by the face alignment process. Others can be managed with the proposed temporal weighting mechanism. The dataset was created with a semi-automatic extraction process, producing an imprecise trimming of movie samples. Videos are not always centered on the relevant emotional frames, different scenes can be present in a video, and multiple subjects can be present in the same frame.

**UNBC-McMaster.** The UNBC-McMaster Shoulder Pain database [32] contains 200 image sequences capturing the spontaneous pain expressions of 25 subjects. Sequences vary in length from 48 to more than 500 frames. We follow Wu *et al*. [58] for the evaluation task. The dataset is used in a binary classification setup based on the Observed Pain Intensity (OPI) expert annotations. The 92 sequences with OPI = 0 constitute the negative samples (No Pain), while the Pain class is composed of the 57 sequences with OPI $\geq$ 3. Because of the class imbalance, the evaluation metric used for this task is the classification accuracy at Equal Error Rate on the Receiver Operating Characteristic curve (ROC-EER). We perform leave-one-subject-out cross-validation for the 25 subjects.

**BioVid.** The BioVid Heat Pain Database [55], Part A, is a relatively large heat-pain detection dataset, with 20 frontal video recordings per stimulus level, for each of 87 subjects. Participants received four levels of painful stimuli (PA1 to PA4), adapted to their subject-specific sensitivity. Videos with no stimulus are also present to constitue the BL1 class (No Pain). Biomedical signals are also available but not used in our study. As opposed to UNBC-McMaster, BioVid provides objective labels based on the temperature of the heat-pain inducing device (with subject-specific levels). The task is thus much more difficult, as the objective is not to classify the directly observable expression but its source. All subjects do not react to pain with the same intensity, even though the stimuli are calibrated for each participants. The creators of the database studied this phenomenon and identified participants that did not react visibly to the pain-inducing stimulus [56]. The BioVid videos are even more controlled than UNBC-McMaster and contain less head-pose variations and occlusion [54]. However, the number of irrelevant frames remains an issue, because videos are not trimmed to capture the specific expression but span a fix time window of 5.5 seconds based on the timing of the stimulus. Thus, results on BioVid should specifically evaluate the ability of training-clip selection to favour expressive windows over relatively neutral states usually present at the beginning and end of every video. On this dataset, we use two different evaluation protocols from the literature. The first task is binary classification of Neutral (BL1) and highest level of pain (PA4), with 1740 videos per class. We perform 8-fold subject-independent cross-validation and report classification accuracy. The second setup is similar but the Pain class is extended to PA3 and PA4 videos, as proposed by Yang *et al*. [62]. To work with the class imbalance (1740 No Pain and 3480 Pain videos), Area Under the Receiver Operating Characteristic Curve (ROC-AUC) is used as metric.

For all datasets, our stochastic softmax pooling only requires sequences of RGB frames and the corresponding sequence-level class labels. Facial expression recognition requires a pre-processing step consisting of detection and alignment of faces in each video frame. We employed the SeetaFace Engine from Wu *et al*. [59].

**Architectures.** We evaluate our method with an inflated [4] VGG-16 model [43]. Prior to inflation, the model was pretrained with VGG-Face and FER2013 image datasets [37, 14].

In order to facilitate comparison with the literature on AFEW, we also experiment with a C3D model [47], pretrained with Sports-1M [20]. We use data augmentation consistently across frames[1] for a given video, with horizontal flip, random rotation, crop and color jitter. Models are trained on a single Tesla V100 GPU, with standard SGD with momentum 0.9. We used early stopping on the validation loss. Training clips of 16 frames are selected with our

[1] https://github.com/hassony2/torch_videovision

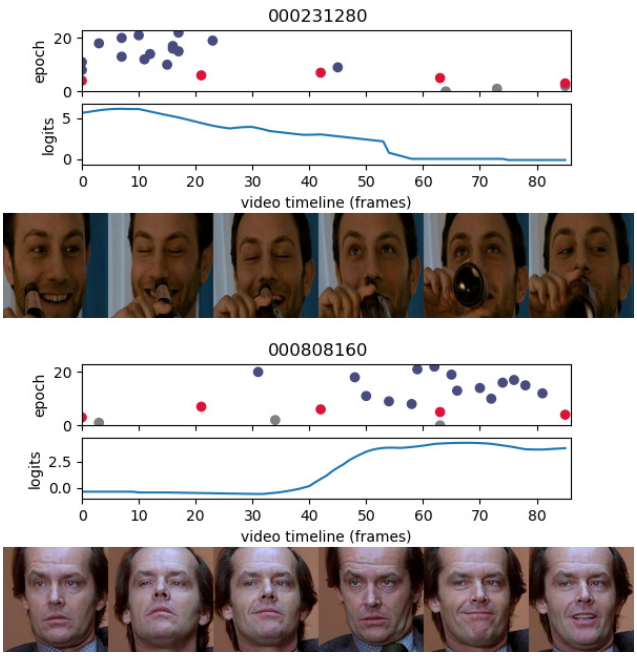


Figure 2. Visualization of sampling distributions (logits) and resulting clip selection during softmax training with $\gamma = 1$, for AFEW videos (Happy). The sampling maps indicate the temporal position of the selected clip at each epoch for a given video example. Colors indicate the training phase: uniform warm-up, deterministic exploration, and weighted sampling. Stochastic softmax avoids selection of occluded and neutral frames.

stochastic softmax sampling, and temporal softmax pooling is applied to aggregate clip-level scores during inference. For the baseline, clips are selected uniformly and temporal average pooling is applied.

### 4.2. Ablation Study

**Illustrative examples.** Figure 2 displays the sampling map and distribution of temporal stochastic softmax sampling, with two training videos from the AFEW dataset. The proposed method is shown to quickly identify the emotion intensity distribution in the video, thanks to the exploration steps at the beginning of training, and focuses on training clips with the best scores. As observed on the sample images, the video-level label "Happy" does not correspond to the beginning and end of the videos, because of occlusion and neutral state respectively. In both cases, the model avoids these uninformative frames.

**Stochastic softmax sampling strategies.** Table 1 compares results for stochastic softmax training and REINFORCE sampling. Rank-1 accuracy and the number of epochs needed to converge are presented when varying the softmax temperature. Inverse temperature $\gamma = 0$ corresponds to uniform sampling, as generally used in previous approaches. Higher inverse temperatures tend to approximate max pooling. As expected the best results are found in between average and max pooling. This shows that using a softmax pooling is important to achieve optimal performance on this task. A value of $\gamma = 1$ provides the best performance for REINFORCE as well as for our proposed method. However, the proposed softmax training manages to obtain better performance than REINFORCE because it has a lower variance in the estimation of $p$. Also the number of epochs needed to converge for the best temperature is reduced by around 20% compared to uniform sampling. Focusing on the most important parts of the video not only improves the accuracy, it also allows the model to directly focus on important clips, thereby saving training time.

| Inverse Temp. | REINFORCE Acc.(%) | REINFORCE Ep. | Ours Softmax Acc.(%) | Ours Softmax Ep. |
|---|---|---|---|---|
| $\gamma = 0$ | 45.66 ±.21 | 24.6 | 45.66 ±.21 | 24.6 |
| $\gamma = 0.5$ | 46.09 ±.41 | 23.8 | 46.07 ±.27 | 23.6 |
| $\gamma = 1$ | 46.80 ±.63 | 22.5 | **47.35** ±.27 | 20.3 |
| $\gamma = 10$ | 44.52 ±.18 | 17.5 | 46.65 ±.40 | 17.2 |

Table 1. Average performance and duration of REINFORCE and stochastic softmax training on AFEW. Both methods correspond to uniform sampling when $\gamma = 0$.

| Method | Model | Acc. (%) |
|---|---|---|
| Lu *et al.*, 2018 [31] | 3D VGG-16 | 39.36 |
| Fan *et al.*, 2016 [11] | C3D | 39.69 |
| Vielzeuf *et al.*, 2017 [51] | C3D-LSTM | 43.2 |
| | C3D Weighted | 42.1 |
| C3D baseline | C3D (uniform) | 39.95 |
| C3D with Softmax | C3D ($\gamma = 1$) | 42.78 |
| VGG baseline | 3D VGG-16 (unif.) | 45.66 |
| VGG with Softmax | 3D VGG-16 ($\gamma = 1$) | 47.35 |

Table 2. Accuracy of a 3D CNN with stochastic softmax compared to our baseline (same architecture but with uniform training-clip sampling and average pooling) and relevant literature on AFEW.

### 4.3. Results

Table 2 presents our results in comparison with other 3D-CNN approaches on AFEW. Our inflated VGG-16 achieves very good performance compared to other 3D CNNs. Temporal stochastic softmax is able to improve accuracy further. C3D Weighted [51] performs a temporal weighting of the training-clip losses, thus a simplified approximation of our method. We also validate experimentally the effect of stochastic softmax training with a C3D, as this architecture has been extensively studied in the literature. Results show that the training method can be adapted to different architectures of 3D CNNs.

| Method | Model | EER Acc. (%) |
|---|---|---|
| Wu *et al*., 2015 [58] | MIL-HMM | 85.2 |
| Werner *et al*., 2017 [53] | SPTS+CAPP | 91.7* |
| Sikka and Sharma, 2018 [41] | LOMo with SIFT, LBP | 87.0 |
| Kumawat *et al*., 2019 [25] | LBVCNN | 86.55 |
| Our baseline | 3D VGG (unif.) | 86.58 |
| Stochastic Softmax | " ($\gamma = 2$) | 87.21 |

* Accuracy is not computed at ROC-EER.

Table 3. Accuracy of a 3D CNN on UNBC-McMaster, with and without stochastic softmax, compared to related SOTA methods.

| Method | Model | Acc. (%) |
|---|---|---|
| Werner *et al*., 2017 [53] | Standardized FADs | 72.4* |
| Othman *et al*., 2019 [36] | RFc with FADs | 65.8 |
| | Reduced MbNetV2 | 65.5 |
| Thiam *et al*., 2020 [46] | Two-stream VGG | 69.25 |
| Our baseline | 3D VGG (uniform) | 68.12 |
| Stochastic Softmax | 3D VGG ($\gamma = 2$) | 69.60 |

* Using additional information with depth sensor 3D maps.

Table 4. Binary Classification accuracy of a 3D CNN on BioVid (BL1 vs. PA4), with and without stochastic softmax, compared to related SOTA methods.

| Method | Model | AUC (%) |
|---|---|---|
| Tavakolian and Hadid, 2019 [44] | 3D ResNet | 82.54 |
| | S3D-G | 83.26 |
| | SCN | 86.02 |
| Our baseline | 3D VGG (uniform) | 82.67 |
| Stochastic Softmax | 3D VGG ($\gamma = 2$) | 84.39 |

Table 5. ROC-AUC results on BioVid (BL1 vs. PA3-4), with and without stochastic softmax, compared to related SOTA methods.

Tables 3, 4 and 5 report the performance of our sampling method on three pain video classification tasks. An aggressive sampling temperature could be expected to generate overfitting on UNBC-McMaster, by reducing the variation in training data which is already very limited. Actually, we found that better results were obtained with a higher temperature, $\gamma = 2$ (Table 3). The classification scores of the model on UNBC-McMaster are generally lower than for AFEW. The temperature parameter allows us to adapt the weighting strategy. Note that with short-clip training, the model cannot adapt its scores to learn a video-level temperature implicitly. Table 4 provides results on the larger BioVid dataset. Temporal max-pooling during inference might not be beneficial, because it is necessary to consider the entirety of the video to classify it. For example, the model could recognize a neutral expression at the beginning of a Pain video, with very high confidence. We considered decoupling the temperatures of the sampling and pooling but do not extensively study this possibility here. We provide preliminary results in Supplementary Material. The good performance improvement obtained with our temporal softmax on BioVid (Table 4) suggests that the method is also relevant to very controlled recordings, with no occlusion and very limited head movement. The benefits of clip sampling does not only consist of limiting the impact of noisy labelling or capture conditions. Temporal weighting addresses a challenge inherent to the task, as facial expressions are events localized in time, and typically preceded and followed by neutral states. This idea is confirmed as the best performance gain of temporal softmax is observed on the second task of BioVid (Table 5). In this scenario, additional samples are gathered in the Pain class (PA3 + PA4). These are videos where temporal weighting is really efficient, as opposed to No Pain videos which do not contain apex or particularly relevant segments. This calls for future work studying the relevance of using different softmax temperatures adapted to each class, as was developed by McFee *et al*. [33] with temporal pooling for audio data.

## 5. Conclusion

We presented a softmax-based training and inference method for 3D CNNs adaptable to the task at hand in terms of computation, regularization and feature aggregation strategy, with no additional trained layer. Our method is designed to enable the use of softmax temporal pooling within the framework of short-clip training, which is the standard way of training 3D models because of computational and GPU-memory limitations. We demonstrated the benefits of softmax temperatures for video classification by considering videos as bags of unequally relevant clips. At test time, a temporal softmax pooling mechanism is able to weight and aggregate information from different clips, with a strategy adapted to the input distribution. Stochastic softmax sampling improves learning by balancing informative and difficult training clips, allowing for faster convergence and limiting the impact of irrelevant clips in the context of weak, sequence-level annotations. These mechanisms provide an improvement in accuracy on all evaluated datasets. Experiments suggested several directions for future work.

## Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2018-04825), Calcul Québec and Compute Canada, and the Canadian Institutes of Health Research (SMC-151518).

## References


[1] M. Aminbeidokhti, M. Pedersoli, P. Cardinal, and E. Granger. Emotion recognition with spatial attention and temporal softmax pooling. In *ICIAR (1)*, volume 11662 of *Lecture Notes in Computer Science*, pages 323–331, 2019.

[2] S. A. Bargal, E. Barsoum, C. Canton-Ferrer, and C. Zhang. Emotion recognition in the wild from videos using images. In *ICMI*, pages 433–436, 2016.

[3] Y-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *ICML*, pages 111–118, 2010.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017.

[5] W. C. de Melo, E. Granger, and A. Hadid. A deep multiscale spatiotemporal network for assessing depression from facial dynamics. *IEEE Trans. Affect. Comput.*, 2020.

[6] W. C. de Melo, E. Granger, and M. B. López. Encoding temporal information for automatic depression recognition from facial analysis. In *ICASSP*, pages 1080–1084, 2020.

[7] A. Dhall. Emotiw 2019: Automatic emotion, engagement and cohesion prediction tasks. In *ICMI*, pages 546–550, 2019.

[8] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE Multim.*, 19(3):34–41, 2012.

[9] A. Diba, M. Fayyaz, V. Sharma, Karami A. H., M. M. Arzani, R. Yousefzadeh, and L. Van Gool. Temporal 3d convnets using temporal transition layer. In *CVPR Workshops*, pages 1117–1121, 2018.

[10] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[11] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *ICMI*, pages 445–450, 2016.

[12] B. Gao and L. Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[13] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019.

[14] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, et al. Challenges in representation learning: A report on three machine learning contests. In *ICONIP (3)*, volume 8228 of *Lecture Notes in Computer Science*, pages 117–124, 2013.

[15] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018.

[16] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, pages 7366–7375, 2018.

[17] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *ICML*, pages 495–502, 2010.

[18] S. E. Kahou, V. Michalski, K. R. Konda, R. Memisevic, and C. J. Pal. Recurrent neural networks for emotion recognition in video. In *ICMI*, pages 467–474, 2015.

[19] S. K. A. Kamarol, M. H. Jaward, H. Kälviäinen, J. Parkkinen, and R. Parthiban. Joint facial expression recognition and intensity estimation based on weighted votes of image sequences. *Pattern Recognit. Lett.*, 92:25–32, 2017.

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[21] A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2530–2539, 2018.

[22] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[23] B. Knyazev, R. Shvetsov, N. Efremova, and A. Kuharenko. Leveraging large face recognition data for emotion classification. In *FG*, pages 692–696, 2018.

[24] B. Korbar, D. Tran, and L. Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, pages 6231–6241, 2019.

[25] S Kumawat, M Verma, and S Raman. LBVCNN: local binary volume convolutional neural network for facial expression recognition from image sequences. In *CVPR Workshops*, pages 207–216, 2019.

[26] J. Li, X. Liu, M. Zhang, and D. Wang. Spatio-temporal deformable 3d convnets with attention for action recognition. *Pattern Recognit.*, 98, 2020.

[27] S. Li and W. Deng. Deep facial expression recognition: A survey. *IEEE Trans. Affect. Comput.*, 2020.

[28] S. Li, W. Zheng, Y. Zong, C. Lu, C. Tang, X. Jiang, J. Liu, and W. Xia. Bi-modality fusion for emotion recognition in the wild. In *ICMI*, pages 589–594, 2019.

[29] C. Liu, T. Tang, K. Lv, and M. Wang. Multi-feature based emotion recognition for video clips. In *ICMI*, pages 630–634, 2018.

[30] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma. T-C3D: temporal convolutional 3d network for real-time action recognition. In *AAAI*, pages 7138–7145, 2018.

[31] C. Lu, W. Zheng, C. Li, C. Tang, S. Liu, S. Yan, and Y. Zong. Multiple spatio-temporal feature learning for video-based emotion recognition in the wild. In *ICMI*, pages 646–652, 2018.

[32] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. A. Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *FG*, pages 57–64, 2011.

[33] B. McFee, J. Salamon, and J. P. Bello. Adaptive pooling operators for weakly labeled sound event detection. *IEEE ACM Trans. Audio Speech Lang. Process.*, 26(11):2180–2193, 2018.

[34] D. Meng, X. Peng, K. Wang, and Y. Qiao. Frame attention networks for facial expression recognition in videos. In *ICIP*, pages 3866–3870, 2019.

[35] J. Y.-H. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.

[36] E. Othman, P. Werner, F. Saxen, A. Al-Hamadi, and S. Walter. Cross-database evaluation of pain recognition from facial video. In *ISPA*, pages 181–186, 2019.

[37] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, pages 41.1–41.12, 2015.

[38] G. Praveen R, E. Granger, and P. Cardinal. Deep weakly-supervised domain adaptation for pain localization in videos. *FG*, 2020.

[39] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black. On the integration of optical flow and action recognition. In *GCPR*, volume 11269 of *Lecture Notes in Computer Science*, pages 281–297, 2018.

[40] L. Sevilla-Lara, S. Zha, Z. Yan, V. Goswami, M. Feiszli, and L. Torresani. Only time can tell: Discovering temporal data for temporal modeling. *arXiv preprint arXiv:1907.08340*, 2019.

[41] K. Sikka and G. Sharma. Discriminatively trained latent ordinal model for video classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(8):1829–1844, 2018.

[42] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[44] M. Tavakolian and A. Hadid. A spatiotemporal convolutional neural network for automatic pain intensity estimation from facial dynamics. *Int. J. Comput. Vis.*, 127(10):1413–1425, 2019.

[45] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV (6)*, volume 6316 of *Lecture Notes in Computer Science*, pages 140–153, 2010.

[46] P. Thiam, H. A. Kestler, and F. Schwenker. Two-stream attention network for pain recognition from video sequences. *Sensors*, 20(3):839, 2020.

[47] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[48] M. F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 42(1):28–43, 2012.

[49] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1510–1517, 2018.

[50] V. Vielzeuf, C. Kervadec, S. Pateux, A. Lechervy, and F. Jurie. An occam's razor view on learning audiovisual emotion recognition with small training sets. In *ICMI*, pages 589–593, 2018.

[51] V. Vielzeuf, S. Pateux, and F. Jurie. Temporal multimodal fusion for video emotion classification in the wild. In *ICMI*, pages 569–576, 2017.

[52] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014.

[53] P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, S. Gruss, and H. C. Traue. Automatic pain assessment with facial activity descriptors. *IEEE Trans. Affect. Comput.*, 8(3):286–299, 2017.

[54] P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, and H. C. Traue. Head movements and postures as pain behavior. *PLOS ONE*, 13(2):1–17, 02 2018.

[55] 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. In *BMVC*, 2013.

[56] P. Werner, A. Al-Hamadi, and S. Walter. Analysis of facial expressiveness during experimentally induced heat pain. In *ACII Workshops*, pages 176–180, 2017.

[57] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992.

[58] C. Wu, S. Wang, and Q. Ji. Multi-instance hidden markov model for facial expression recognition. In *FG*, pages 1–6, 2015.

[59] S. Wu, M. Kan, Z. He, S. Shan, and X. Chen. Funnel-structured cascade for multi-view face detection with alignment-awareness. *Neurocomputing*, 221:138–145, 2017.

[60] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV (15)*, volume 11219 of *Lecture Notes in Computer Science*, pages 318–335, 2018.

[61] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057, 2015.

[62] R. Yang, S. Tong, M. B. López, E. Boutellaa, J. Peng, X. Feng, and A. Hadid. On pain assessment from facial videos using spatio-temporal local descriptors. In *IPTA*, pages 1–6, 2016.

[63] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan. Peak-piloted deep network for facial expression recognition. In *ECCV (2)*, volume 9906 of *Lecture Notes in Computer Science*, pages 425–442, 2016.

[64] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A key volume mining deep framework for action recognition. In *CVPR*, pages 1991–1999, 2016.