Timed-image based deep learning for action recognition in video sequences

Abdourrahmane Atto, Alexandre Benoît, Patrick Lambert

Timed-Image Based Deep Learning for Action Recognition in Video Sequences

Abdourrahmane M. ATTO, Alexandre BENOIT, Patrick LAMBERT

Université Savoie Mont Blanc, LISTIC, France

***Abstract*—The paper addresses two issues relative to machine learning on 2D+X data volumes, where 2D refers to image observation and X denotes a variable that can be associated with time, depth, wavelength, *etc.*. The first issue addressed is conditioning these structured volumes for compatibility with respect to convolutional neural networks operating on 2D image file formats. The second issue is associated with sensitive action detection in the "2D+Time" case (video clips and image time series). For the data conditioning issue, the paper first highlights that referring 2D spatial convolution to its 1D Hilbert based instance is highly accurate for information compressibility upon tight frames of convolutional networks. As a consequence of this compressibility, the paper proposes converting the 2D+X data volume into a single *meta-image* file format, prior to machine learning frameworks. This conversion is such that any 2D frame of the 2D+X data is reshaped as a 1D array indexed by a Hilbert space-filling curve and the third variable X of the initial file format becomes the second variable in the meta-image format. For the sensitive action recognition issue, the paper provides: (i) a 3 category video database involving non-violent, moderate and extreme violence actions; (ii) the conversion of this database into a timed meta-image database from the 2D+Time to 2D conditioning stage described above and (iii) outstanding 2-level and 3-level violence classification results from deep convolutional neural networks operating on meta-image databases.**

***Index Terms*—Data conditioning; Video analysis; Deep learning; Convolution frames; Hilbert space-filling curve; Action recognition; Violence detection.**

I. INTRODUCTION

Convolutional Neural Networks (CNN) have proven outstanding performance in recent image processing engines. Many frameworks, specifically designed and optimized for the image file format, see [1, MatConvNet] for instance, have led to filling the semantic gap between raw images and the high level objects that can be recognized from their contents.

When considering a video 2D+t or a stereoscopic 2D+d file format[1], CNN based feature extraction requires:

- [Option-1] either adapting the network configurations according to dimension extents (by considering dimension extension for network parameters, by separating spatial feature analysis and temporal/depth information processing, *etc.*);
- [Option-2] or relating the 2D+X data to a 2D meta-image format i order to use directly the above frameworks (already optimized for image analysis).

The literature has mainly addressed [Option-1] with a wide range of observable directions. The first direction involves the computational technicality standpoint relatively with 2D+X feature extraction architectures: for instance, CPU[2] and GPU[3] based file architectures have been proposed recently in [2] for 3D convolutions and max-pooling operations that are consistent with MatConvNet. In addition, [3] has proposed some alternative shift and merge modules for spatio-temporal information aggregation, in order to reduce 3D convolution computational complexity, [4] has proposed asymmetric one-directional 3D convolutions whereas [5] has preferred deformable 3D convolutions.

The second direction concerns two-level/stream architectures operating respectively on spatial and temporal (optical flow) features for learning actions in 2D+t video datasets: for instance, [6] has proposed two independent CNNs for learning both still frames and motion between frames, and [7] has considered a refinement of [6] in terms of spatial and temporal information integration at several levels of granularity. Another solution proposed in [8] is as well a two-stream 2D+t architecture where images and their optical flows are processed separately by using convenient convolution operators prior to late fusion. Still in the same direction, [9] has proposed an adaptation of the two-stream framework for long-range video representation by using multiple local features whereas [10] has proposed an extension of the two-stream framework with respect to a multiscale perspective.

Some alternative directions can be found in:

- [11] in terms of learning a projection matrix associated with principal orientations or in [12] from a series of monodimensional temporal convolution operations;
- [13] which adopts a bidirectional Long Short-Term Memory (LSTM) framework for a recurrent feature description strategy, with a constraint being the selection of specific video frames since, otherwise, dimensionality leads to non-tractable algorithms on limited computational resources;
- [14] where graph parsing neural network architectures are developed and [15] where ontology like grammars can be used to disambiguate certain specific situations.

In terms of action recognition benchmarks, most of the above references have highlighted the intricacy in identifying generalizable 2D+X architectures and the most relevant strategy among the directions given above remains an open issue at present time.

* Corresponding author: *Abdourrahmane.Atto@univ-smb.fr*

[1]Convention: in the notation 2D+X, '2D' relates spatial dimensions and the variable X can refer to time ($X \triangleq t$), depth ($X \triangleq d$) or an arbitrary third dimension (abusive notation 3D in some cases).

[2]CPU: Central Processing Unit

[3]GPU: Graphics Processing Unit

On the one hand, the limitation affecting the 2D+X frameworks on large data volumes is the intricacy of nD convolution kernel updating strategies with respect to the capture of tiny objects/events in huge data when n is large. For these huge data and due to the above computational limitation, robust network design is challenging and training is, at present time, subject to assistance: for instance, in [16], only 2D spatial directional convolution operations are used for a first stage training and certain weights obtained are selected to guide the 3D MatConvNet on a patch-by-patch basis. The same holds true for the two-stream spatial and temporal strategy given in [8] and [7]: the approaches are chosen separable (handcrafted extractions of spatial and optical flow features whereas a single 'intelligent' 3D network should have been able to perform this extraction if exploratory of the intrinsic 3D feature space had been straightforward). Another solution to limit computational complexity is the use of compressed domain video representations as in [17]. However, the results obtained by using this approach are slightly less relevant than those obtained by the two-stream fusion stages used in [8] for recognition of homogeneous actions on the same databases. Thus, compression can limit performance depending on its rate.

On the other hand, the major hardware issue when handling huge 2D+X datasets is the limited random-access memory available on standard computer architectures. This leads to limited training capabilities at present time since convergence to a desirable solution cannot be guaranteed when using tiny loads in the optimization batches.

It is worth mentioning that [Option-2] can be achieved by compacting spatial dimensions in 1D format, thus converting a 2D+t video data to a 2D meta-image for instance. But not all 2D-to-1D transforms guaranty nice properties for capturing dependencies that are intrinsic to spatial image features. In order to perform [Option-2] while compacting at best image spatial dependencies, the paper proposes to consider the Hilbert space-filling 1D image description.

The first set of contributions concerns the analysis of Hilbert space-filling curves with respect to a concise and compressible spatial feature representation with respect to convolution operators. This analysis is performed in terms of: (a) maximal spatial shifts loaded in regard to the length of the convolution filter and (b) sparsity degrees of convolution operators when the convolution is performed in the Hilbert 1D domain, see Sections II and III respectively.

The second set of contributions, provided in Section IV as a valuable application of the former contribution, concerns a solution to the challenging issue of *heterogeneous action* recognition in 2D+t data. In contrast with the homogeneous action recognition issue where any category is composed of approximately the same types of motion (for example 'running', 'smiles', *etc.* handled among others in [8] thanks to homogeneous motion databases), the heterogeneous case of violence interpretation (several types of actions having the same consequence: a violence feeling) is very intricate and somewhat subjective. We will present a state of the art on violence detection in Section IV and address violence action recognition on the basis of: (i) violence data benchmarking and (ii) 2D CNN operating heterogeneous action learning from Hilbert based timed meta-image datasets.

To summarize, the major contributions provided through [Option-2] aims at:

- exploiting image spatial compressibility in order to reduce memory load issues,
- learning both spatial and temporal features jointly,
- deriving a framework that makes the use of well-known 2D image based frameworks straightforward.

The main processing steps associated with the paper and developed below are described in the block diagram given by Figure 1.

II. Hilbert space-filling curves: spatial data loads with respect to convolution size

Throughout, $\mathcal{G}(M)$ always refers to a square grid indexing $2^M \times 2^M$ pixels and associated with a left-upper $(0,0)$ corner. This indexing is for convenience: Hilbert space-filling curves can be computed on non-square grids. For the adaptation of Hilbert space-filling constructions to image domain, see [18] (resampling operators) and [19] (lossless compression), among other references.

A. Reshaping 2D grid $\mathcal{G}(M)$ to 1D vector $\mathcal{V}(M)$

Let (m,n) be a point on $\mathcal{G}(M)$. We have $m,n \in \{0,1,\ldots,2^M-1\}$ by definition of $\mathcal{G}(M)$. By decomposing m,n in binary forms, we can write:

$$m=\sum_{\ell=1}^{M}\epsilon_\ell^1 2^{M-\ell}, \quad n=\sum_{\ell=1}^{M}\epsilon_\ell^2 2^{M-\ell}, \tag{1}$$

with $(\epsilon_\ell^1,\epsilon_\ell^2) \in \{0,1\}^2$ for every $\ell \in \{1,2,\ldots,M\}$. Point (m,n) can be univoquely associated with a quaternary 1D array representation $\mathcal{V}(M)=\{0,1,\ldots,4^M-1\}$ as follows:

$$(m,n)\in\mathcal{G}(M) \overset{F}{\longmapsto} k=\sum_{\ell=1}^{M}\xi_\ell 4^{M-\ell}\in\mathcal{V}(M), \tag{2}$$

where $\xi_\ell \in \{0,1,2,3\}$ for every $\ell=0,1,2,\ldots,M$ and F is a bijective application that can be re-written as:

$$\epsilon=\begin{pmatrix}\epsilon_1^1,\epsilon_1^2\\ \epsilon_2^1,\epsilon_2^2\\ \vdots\\ \epsilon_M^1,\epsilon_M^2\end{pmatrix} \overset{F}{\longmapsto} \xi=\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_M\end{pmatrix}$$

with $\epsilon \in \{0,1\}^M \times \{0,1\}^M$ and $\xi \in \{0,1,2,3\}^M$.

Several choices are possible for the correspondence function F in order to convert the binary representation of $\mathcal{G}(M)$ into a quaternary 1D array enumeration. Among them, one can cite:

- The *lexicographic* ordering $\mathcal{V}^{\mathcal{L}}(M)$, which consists in column-wise concatenations of the 2^M rows of $\mathcal{G}(M)$.
- The *natural recursive quaternary* decomposition of grid $\mathcal{G}(M)$ into adjacent quadrants where we obtain a 1D array $\mathcal{V}^{\mathcal{N}}(M)$ subject to: $\xi_\ell=\epsilon_\ell^1+2\epsilon_\ell^2$.
- The *Hilbert* based representation [20] denoted $\mathcal{V}^H(M)$ hereafter is a variant of the above recursive quaternary decomposition where enumeration involves splits into

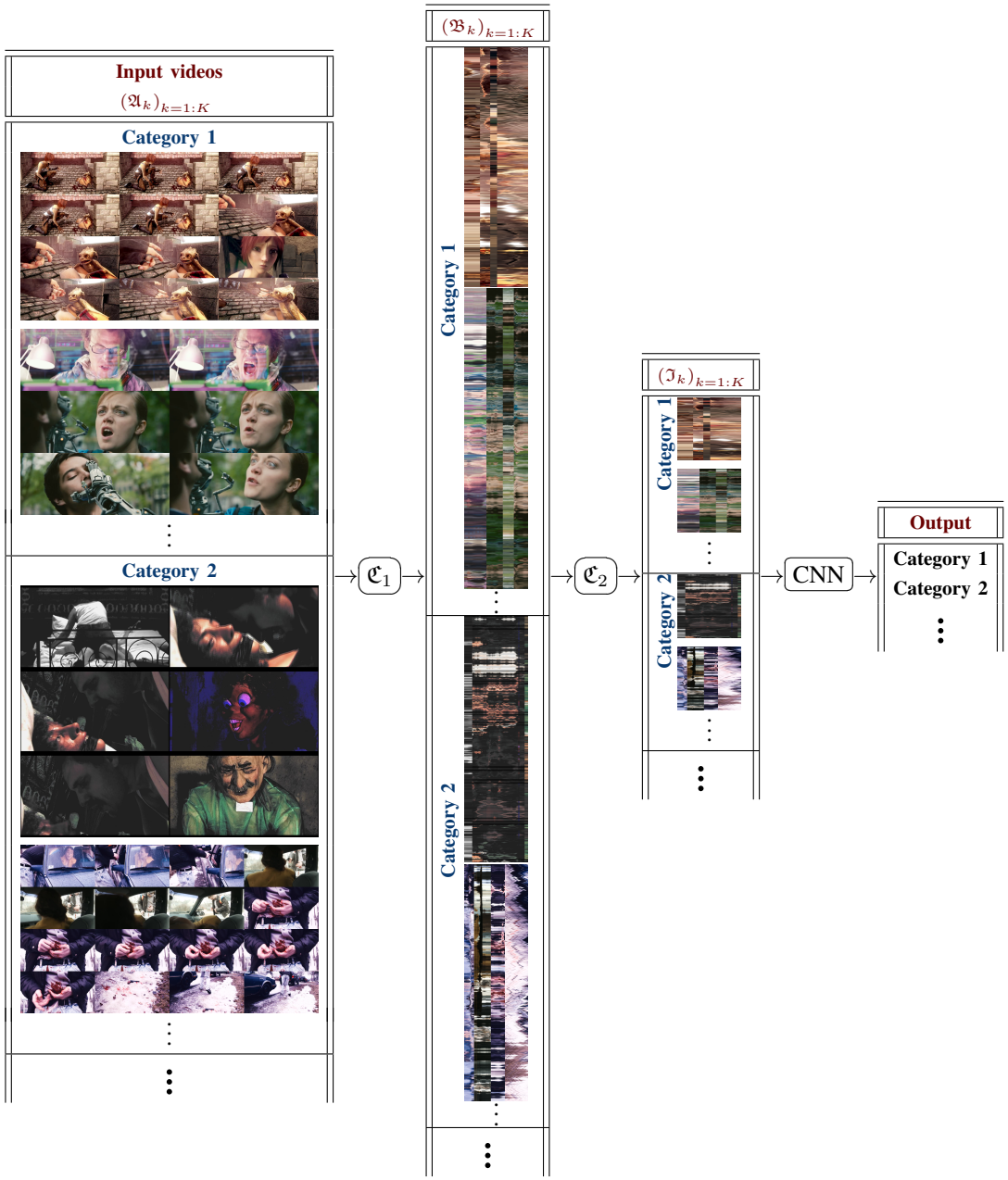

Fig. 1. Block diagram summarizing the learning framework proposed by the paper: the k-th timed-image $\mathfrak{I}_k$ (input of the 2D CNN) is such that $\mathfrak{I}_k = \mathfrak{C}_2 \circ \mathfrak{C}_1 [\mathfrak{A}_k]$ where $\mathfrak{C}_1$ and $\mathfrak{C}_2$ denote respectively the specific Hilbert and compression transforms given in IV-C1, with $\mathfrak{A}_k$ representing the k-th video clip of the dataset.

4 congruent blocs at each recursion. The congruence imposes connections with consecutive k values when moving locally in $\mathcal{G}(M)$ and it can be geometrically explained in terms of several affine transformations on the doubly binary sequence ϵ.

Figure 2 provides lexicographic, natural and Hilbert 1D scan of $\mathcal{G}(M)$ for $M = 2$. As it can be seen in this figure, consecutive indices 7 and 8 are associated with non-close spatial points (m, n) for the lexicographic and natural orderings. This can lead, when using convolution in 1D array domain, to mixing non-homogeneous pixel information. The Hilbert ordering ensure a more relevant description in the sense that the neighbors of index k are associated with a tight spatial neighborhood. More precisely, the following section provides, when M is large, the spatial range loaded by a convolution filter[4] operating in the Hilbert array domain. Before developing this section, we need to select a metric between sample points of $\mathcal{G}(M)$.

Given two points $A = (m, n)$ and $B = (m', n')$ pertaining to $\mathcal{G}(M)$, the Chebyshev distance between these points is associated with the uniform norm of their difference, that is:

$$\mathrm{d}_C(A, B) = \|A - B\|_\infty = \max\{|m - m'|, |n - n'|\}$$

[4]We use the terminology of *filter* with size P to denote the standard *impulse response* of the filter, the latter being in general defined in Fourier domain. This choice is made for sake of abbreviation and because no confusion is possible: Fourier transform is not addressed in the paper.

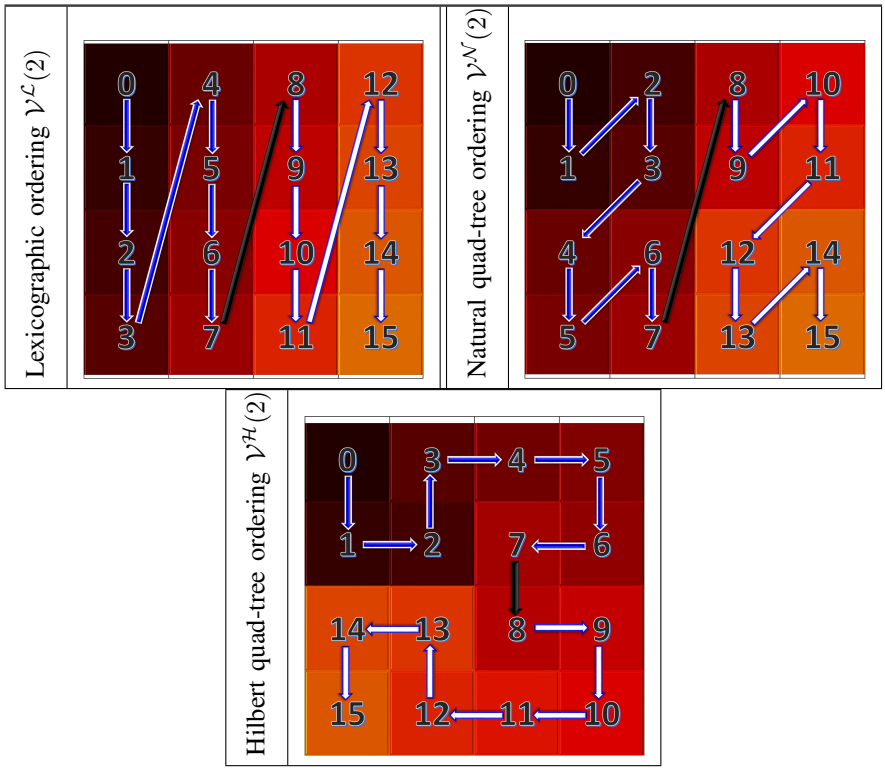

Fig. 2. Correspondance $(m,n) \in \mathcal{G}(2) \mapsto k \in \mathcal{V}^{\bullet}(2)$ for the lexicographic, natural and Hilbert based scan of $\mathcal{G}(2)$. Color ranges from dark-red to bright-orange when k increases from 0 to 15. Hilbert scan properties: [no-jump]: 1D convolution will involve spatially close neighbors and [remoteness]: few pixels, among large-scale neighborhoods, are dislodged to form later, somewhat distant islands but always with spatially close pixels in every island.

This metric is particularly relevant for measuring the distance between points that are constrained to belong to a discrete grid in the sense that the circle of radius ρ exactly matches a square containing $(2\rho+1)\times(2\rho+1)$ grid points.

Note the maximal distance measurable on $\mathcal{G}(M)$ is 2^M-1 and it corresponds to the distance between two consecutive corners. This maximal distance corresponds as well to the distance between left-upper corner $(0,0)$ and right-bottom corner $(2^M-1, 2^M-1)$. It is worth emphasizing that point $(2^M-1, 2^M-1)$ is at the same Chebyshev distance from $(0,0)$ as point $(0, 2^M-1)$, which is not true when using the standard Euclidean distance.

B. Spatial range of $\mathcal{G}(M)$ indices loaded by a convolution operator relating the 1D array $\mathcal{V}(M)$ domain

On $\mathcal{G}(M)$ indexing, the convolution of an image I with respect to a 2D filter h is defined by:

$$J(p,q) = \sum_{m,n} h(m,n) I(p-m, q-n)$$

where (m,n) ranges over the support of h in the latter equation.

For $\mathcal{V}(M)$ description of $\mathcal{G}(M)$, pixel $I(F^{-1}(k))$ corresponds to $I(m,n)$ thanks to Eq. (2) and convolution involves a 1D filter denoted h^* from now on. The convolution of $I(F^{-1}(k))$ with respect to h^* is given by:

$$J^*(\ell) = \sum_k h^*(k) I(F^{-1}(\ell-k)) \tag{3}$$

The latter can be rewritten, with notation convention $F^{-1}(k) = (U(k), V(k))$

$$J^*(U(\ell), V(\ell)) = \sum_k h^*(k) I(U(\ell-k), V(\ell-k)) \tag{4}$$

Assuming that:

$$h^* = \{h^*(k), k = 0, 1, \ldots, P-1\}$$

then the indices of image I loaded for computing $J^*(U(\ell), V(\ell))$ corresponds to: $\{F^{-1}(\ell), F^{-1}(\ell-1), \ldots, F^{-1}(\ell-P+1)\}$.

1) Case of a convolution operator with size 2: For $P=2$, the loaded indices are $\{F^{-1}(\ell), F^{-1}(\ell-1)\}$ and one can remark that:

- the Hilbert ordering (notation $F \triangleq F_{\mathcal{H}}$ in Eq. (2)) yields:

$$\mathrm{d}_C(F_{\mathcal{H}}^{-1}(\ell), F_{\mathcal{H}}^{-1}(\ell-1)) = 1 \tag{5}$$

  for any ℓ such that both ℓ and $\ell-1$ pertain to $\mathcal{V}^{\mathcal{H}}(M)$,
- whereas the Lexicographic and natural orderings (notations $F \triangleq F_{\mathcal{L}}$ and $F \triangleq F_{\mathcal{N}}$ respectively in Eq. (2)) lead to:

$$\max_{\ell \in \mathcal{V}^{\mathcal{L}}(M)} \mathrm{d}_C(F_{\mathcal{L}}^{-1}(\ell), F_{\mathcal{L}}^{-1}(\ell-1)) = 2^M - 1 \tag{6}$$

  and

$$\max_{\ell \in \mathcal{V}^{\mathcal{N}}(M)} \mathrm{d}_C(F_{\mathcal{N}}^{-1}(\ell), F_{\mathcal{N}}^{-1}(\ell-1)) = 2^M - 1 \tag{7}$$

Eq. (5) highlights that the 'previous' pixel $I(F_{\mathcal{H}}^{-1}(\ell-1))$ is always spatially very close to the current pixel $I(F_{\mathcal{H}}^{-1}(\ell))$

when considering the $\mathcal{V}^{\mathcal{H}}(M)$ Hilbert ordering. However, this pixel can be far, spatially, from the current pixel position in the lexicographic and natural quaternary orderings thanks to Eqs. (6) and (7). This good property of $F_{\mathcal{H}}$ is a consequence of the neighborhood-based congruence constraint imposed at any step of the Hilbert recursive filling.

Another consequence of Eq. (5) is that: when h^* is chosen so as to perform a discrete order differencing, $h^* = [1, -1]$ for instance, then the corresponding convolution[5] performed in the Hilbert domain involves, spatially, the difference between $I(m, n)$ and only one among $\{I(m-1, n), I(m, n-1), I(m-1, n-1)\}$. Thus, this convolution performs a direct vertical, horizontal or diagonal pixel differencing alternatively with respect to k. In a wavelet based convolution framework associated with Haar differencing filters, this amounts to say that vertical, horizontal and diagonal Haar coefficients are fused into a single detail subband containing all these 3 types of edge information.

From now on, we focus only on Hilbert space-filling image description and we omit symbol $\mathcal{H}$ in $F_{\mathcal{H}}$ and $\mathcal{V}^{\mathcal{H}}$ since no confusion is possible. The following provides for this description, the characterization of Chebyshev distances associated with spatial indices loaded by a convolution operator with size $P > 2$.

2) Case of a convolution operator with size $P > 2$: In contrast with Eq. (5), $\mathrm{d}_C(F^{-1}(\ell), F^{-1}(\ell - q)) > 1$ in general as long as $q > 1$.

For characterizing the spatial load of the set $\{F^{-1}(\ell), F^{-1}(\ell - 1), \ldots, F^{-1}(\ell - P + 1)\}$, we need to determine for any $k \in \mathcal{V}(M)$, the distances:

$$\mathfrak{X}_{Q,M}[k] = \mathrm{d}_C(F^{-1}(k), F^{-1}(k - Q)) \tag{8}$$

when $Q = 1, 2, \ldots, P - 1$.

We recall that $\mathfrak{X}_{Q,M}[k]$ provides the maximal spatial index shift with respect to Chebyshev distance when looking at the Q-st point far behind the current position k in Hilbert $\mathcal{V}(M)$ indexing. Let

$$\mathfrak{M}^{\wedge}{}_{Q,M} = \max_{k \in \mathcal{V}(M)} \mathfrak{X}_{Q,M}[k]$$

Then for any $M \geqslant 1$, we have:

$$\mathfrak{M}^{\wedge}{}_{Q,M} \leqslant 2\lceil \sqrt{Q} \rceil + 1 \tag{9}$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to a given real number.

Eq. (9) shows that whatever the size of grid $\mathcal{G}(M)$, then $(m_Q, n_Q) = F^{-1}(k - Q)$ pertains to a very close spatial neighborhood of $(m, n) = F^{-1}(k)$ since the bound $2\lceil \sqrt{Q} \rceil + 1$ is small and independent with M. Table I provides $\mathfrak{M}^{\wedge}{}_{Q,9}$ when $Q = 1, 2, \ldots, 32$. This table highlights that for a filter size $P = 33$, then the spatial range loaded on $\mathcal{G}(M)$ for a single convolution operation is included in a small 12×12 box. Table I also provides the statistics

$$\mathfrak{M}^{\circ}{}_{Q,M} = \mathrm{mode}\,\{\mathfrak{X}_{Q,M}[k] : k \in \mathcal{V}(M)\}$$

[5]This convolution performed in the Hilbert domain yields sequence: $\left(I(F_{\mathcal{H}}^{-1}(k)) - I(F_{\mathcal{H}}^{-1}(k-1)\right)_{k=0,1,\ldots,4^M-1}$.

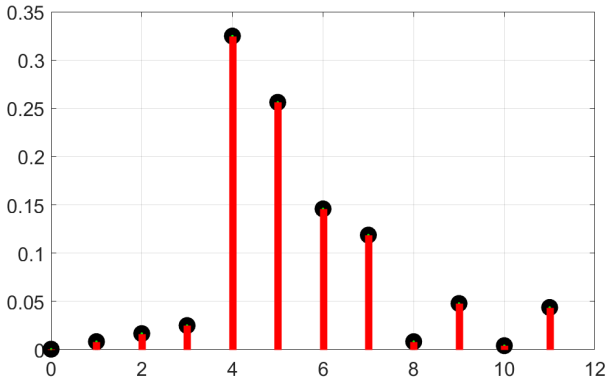

Fig. 3. Distribution of $\mathfrak{X}_{32,9}$ (see Eq. (8)). Set $\mathfrak{X}_{32,9}$ contains 2^{18} index values, but a 32 distance in the Hilbert indexing yields a maximum distance of 11 when relating to the spatial domain. In a pictorial way, this figure highlights that for an image grid with size $2^9 \times 2^9 = 512 \times 512$, the 32 length "convolution snake" runs along the whole grid in almost always extremely curled up position: the distance from his head to his tail does not exceed 11 consecutive horizontal (or vertical) pixels, this distance varies between 4 and 7 pixels most of the time.

for $M = 9$ and $Q = 1, 2, \ldots, 32$, in order to emphasize the location of the most occurrent value in $\mathfrak{X}_{Q,M}$. For information on the whole distributional behavior of $\mathfrak{X}_{Q,M}$, we show an example for fixed $Q = 32, M = 9$ at Figure 3. This figure shows that more than 90% of the selected spatial indices pertain to a square neighborhood with side 8, the latter being very close to $\sqrt{32} \approx 5.66$.

III. Statistical properties of CNN operators with respect to Hilbert space-filling curves

A. Non-stationarity of Hilbert domain convolution outputs

In this section, we are interested on the statistical properties of the convolution output J^* given by Eq. (3). One can first note that if input image I is with constant mean, then the same holds true for J^* since for any ℓ,

$$\begin{aligned}\mathbb{E}\left[J^*(\ell)\right] &= \sum_k h^*(k)\mathbb{E}\left[I(U(\ell - k), V(\ell - k))\right] \\ &= \mathbb{E}[I] \times \sum_k h^*(k)\end{aligned} \tag{10}$$

Thus, we can assume without loss of generality that I is with zero-mean in the following.

The autocorrelation function of J^* is

$$\begin{aligned}\mathbb{E}\left[J^*(\ell)J^*(\ell + \tau)\right] = \sum_{k,k'} h^*(k)h^*(k')\mathbb{E}\big[& \\ & I(U(\ell - k), V(\ell - k)) \times \\ & I\left(U(\ell + \tau - k'), V(\ell + \tau - k')\right)\big]\end{aligned} \tag{11}$$

If I is stationary, then there exists a function Θ such that $\mathbb{E}\big[I(t_1, t_2)I(s_1, s_2)\big] = \Theta(t_1 - t_2, s_1 - s_2)$ and we can write:

$$\begin{aligned}\mathbb{E}[J^*(\ell)J^*(\ell + \tau)] = \sum_{k,k'} h^*(k)h^*(k') \times & \\ \Theta\left(U(\ell - k) - U(\ell + \tau - k'), V(\ell - k)) - V(\ell + \tau - k')\right)\end{aligned} \tag{12}$$

For an arbitrary h^*, the latter quantity cannot be written as a function of τ since Hilbert reshaping functions $\ell \mapsto U(\ell -$

TABLE I
MAXIMUM $\mathfrak{M}^{\wedge}{}_{Q,9}$ AND MODE $\mathfrak{M}^{\circ}{}_{Q,9}$ OF THE DISTRIBUTIONS ASSOCIATED WITH THE DISTANCE SETS $\mathfrak{X}_{Q,9}$ WHEN $Q = 1, 2, \ldots, 32$. WE RECALL THAT $\mathfrak{M}^{\wedge}{}_{Q,9}$ REPRESENTS THE MAXIMAL SPATIAL SHIFT WHEN LOOKING AT THE Q-TH PREVIOUS PIXEL WITH RESPECT TO THE CURRENT POSITION.

Q	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\mathfrak{M}^{\wedge}{}_{Q,9}$	1	2	3	3	3	4	5	5	5	6	7	7	7	6	7	7
$\mathfrak{M}^{\circ}{}_{Q,9}$	1	1	2	2	3	3	2	2	3	3	4	4	5	3	3	4
Q	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
$\mathfrak{M}^{\wedge}{}_{Q,9}$	7	7	7	7	7	8	9	9	9	10	11	11	11	10	11	11
$\mathfrak{M}^{\circ}{}_{Q,9}$	4	5	5	6	4	4	5	6	6	4	4	4	5	5	4	4

$k) - U(\ell + \tau - k')$ and $\ell \mapsto V(\ell - k) - V(\ell + \tau - k')$ are not shift invariants. Thus stationarity of image or image patch, in its standard spatial shift definition, is not in general preserved through the Hilbert 1D image description.

In order to force stationarity from spatial to Hilbert projection, the expectation involved in Eq. (12) must not be applied with respect to shifts of ℓ, but to those of $U(\ell)$ and $V(\ell)$. This amounts to consider filter h^* as defined spatially and being latter transposed to 1D Hilbert description which would add a double indexing process to get pixels in 1D description. Our motivation being to derive an alternative to the full 2D to 1D transpose for both image and filters, we will keep the natural 1D filter indexing and analyze its consequence on several 1D based convolution operators.

B. Compressibility of Hilbert domain convolution outputs

Hilbert indexing is recursive, associated with a non-smooth F function and this makes finding its statistical properties intricate. However, assuming the availability of a large variety of billions of natural images, we can assess the effectiveness and conciseness of Hilbert domain description. This will be performed hereafter in terms of compressibility property with respect to the convolution filters having proven relevance in image processing. We consider the following experimental framework for evaluating this compressibility.

1) Database: The dataset used for compressibility tests is denoted $\mathcal{D}$. It consists of a tree structured library including images extracted from the UCF[6] video database: the tree structure has 101 first level nodes, where every node relates a content *category*. Any category is associated with a set of video examples (category child nodes, second level of the tree). A total of 13 320 video files form the second level of this tree. The third level is composed by images extracted from any video file, the number of images per video being larger than 50 and lower than 500 in general (variable sizes). This database is chosen because it provides a wide range of non-stationarities that can be encountered in indoor (example: category "RockClimbingIndoor") and outdoor (example: category "SoccerPenalty") scenes, including human body features, smooth objects with sharp geometries and natural textures, *etc*. It also has the advantage of providing videos at several spatial resolutions, which makes it relevant with respect to a wide range of modern imaging sensors boarded on a wide range of supports.

[6]UCF-101: see http://crcv.ucf.edu/data/UCF101.php for more information.

2) Compressive operator: We consider a compressive wavelet based convolution operator having the form:

$$\mathcal{T}^{\text{compress}}_{\rho,\eta} = \mathcal{W}^{-1}_{\eta} \circ \mathcal{T}_{\rho} \circ \mathcal{W}_{\eta} \tag{13}$$

where the composition takes into consideration:

- a wavelet transform $\mathcal{W}_{\eta}$ and its inverse $\mathcal{W}^{-1}_{\eta}$ given in a matrix based representation, the parameter η relating the choice of the wavelet name ;
- a thresholding operator $\mathcal{T}_{\rho}$ associated with a compression rate ρ with $0 < \rho < 1$. Operator $\mathcal{T}_{\rho}$ keeps unchanged the $(1-\rho)N$ greatest values of a given vector and force to zero the remaining ρN values of this vector, where N is the vector length (total number of pixels in the context of this paper).

We will keep the same notation for the spatial analog of this operator: the one which applies on 2D image features and for which $\mathcal{W}$ is a standard separable 2D wavelet transform. This consideration is for comparison purpose.

3) Compressibility testing: Given an image I and its Hilbert vectorization $I^*(\ell) = I(U(\ell), V(\ell))$, the compressibility testing consists in:

- selecting ρ and η among their possible values;
- computing $J^*_{\rho,\eta} = \mathcal{T}^{\text{compress}}_{\rho,\eta}(I^*)$
- computing the Peak Signal-to-Noise Ratio (PSNR, in deciBel unit) associated with quality loss due to compression:

$$\text{PSNR}(J^*_{\rho,\eta}) = 10 \log_{10} \frac{255^2 N}{\sum_{\ell=0}^{N-1} (J^*_{\rho,\eta}(\ell) - J^*(\ell))^2} \tag{14}$$

  where N is the total number of pixels, with higher PSNR corresponding to better image quality.

When applying a direct compression of image I by using the standard 2D wavelet based instance of $\mathcal{T}^{\text{compress}}_{\rho,\eta}$, then the output will be simply denoted $J_{\rho,\eta}$.

4) Compressibility results and comparison with a direct 2D approach: Experimental runs are performed and displayed per category for the sake of interpretability. In addition, only Haar wavelet based results are provided (see Figure 4) due to that other wavelets show approximately the same global behavior. From Figure 4, one can remark that the Hilbert based image description yields a more performant compression framework and this, although the fact that 1D image description may intuitively be objectionable.

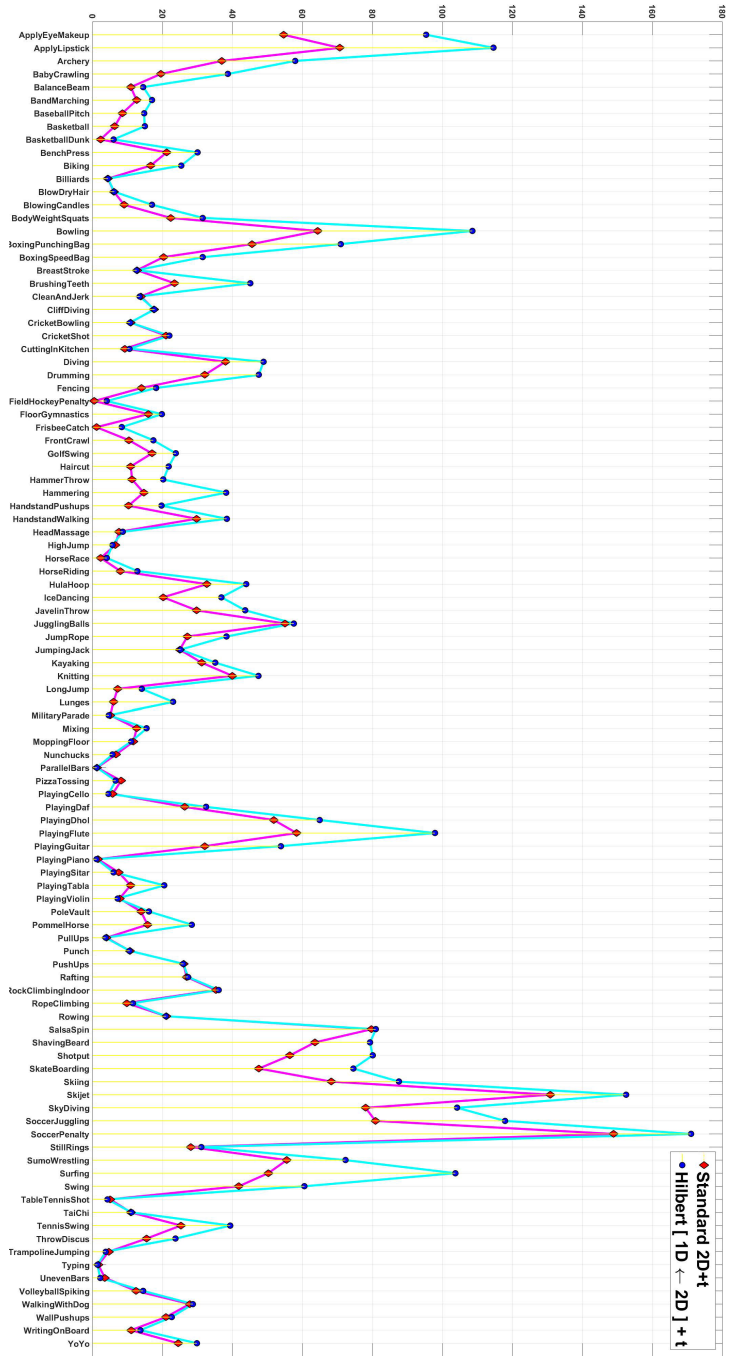

Fig. 4. PSNR($J^*_{\rho,\text{Haar}}$) for compression rate $\rho = 75\%$.

Note also that despite the fact that Haar wavelet transform requires 4 filters in the 2D implementation and 2 filters in the Hilbert based 1D implementation, the convolution outputs have the same nature for both:

- the spatial 2D convolutions with Haar detail filters perform horizontal, vertical and diagonal differencing whatever the input pixel,
- the Hilbert based convolution with 1D detail Haar filter outperforms the above 3 types of differencing in a single run alternatively with respect to pixel location.

One can also remark, from Figure 4, that performance at very high compression rate $\rho = 75\%$ depends on the category content. For categories associated with large scale smooth and uniform objects (bowling, diving, drumming and surfing scenes, as well as zooms of human faces when applying Makeup and Lipstick, among others), performance of both 2D and 1D strategies is higher, and in these cases, the 1D approach is significantly more relevant than the standard 2D framework.

C. Rectifier and max-pooling compositions on Hilbert domain convolution outputs

1) Sparse rectifier: A rectifier operator or Rectified Linear Unit (ReLU) applied to J^* given by Eq. (3) yields outputs:

$$\Gamma^*(\ell) = \mathbb{1}_{J^*(\ell)\geqslant 0} J^*(\ell) \tag{15}$$

If the convolution filter h^* operates as a differencing filter, that is $\sum_k h^*(k) = 0$, then from Eq. (10) and as long as I is piecewise constant, we have $\mathbb{E}\left[J^*(\ell)\right] = 0$ for almost all ℓ (excepted those corresponding to edges of I). In this case where the distribution of J^* is approximately centered, Eq. (15) implies that half content of Γ^* is expected to be zero, causing sparsity for the ReLU neural outputs.

It is well-known that several convolution filters get specialized in edge or transient feature detection during learning process: this means that they operate approximately as differencing filters. Due to the higher compressibility of Hilbert domain coding (see Section III-B), we may expect a higher convolution based differencing effect and thus, a higher sparsity degree for the ReLU neural positioned beyond these convolution outputs.

2) Max-pooling in the Hilbert image domain: In the spatial image domain, standard max-pooling operator involved in CNNs operates on 2D intervals (rectangular image regions) and replaces such a region by a single point associated with the maximal pixel value over this region. When referring to a max-pooling operator applied in the 1D Hilbert domain, any interval with form $[a, b]$ spreads over a specific spatial neighborhood thanks to F^{-1} (see Eq. (2)). The 1D concern provides more flexibility in terms of the selection of non-rectangular windows, for instance, a window size 3 performs in a single image run, max-pooling successively on corners and arcs having several orientations due to Hilbert *zigzag* run (see Figure 2).

In addition, we have: for any natural number r, a 4^r max-pooling operation with stride 4 on $\mathcal{V}(M)$ is equivalent to a $2^r \times 2^r$ max-pooling operation with stride 2×2 on $\mathcal{G}(M)$. The proof of this property relies on the quad-tree structure of the Hilbert split of $\mathcal{G}(M)$ and is let to the reader. When the max-pooling strides involve overlapping windows such as in the framework of [21], then Hilbert domain pooling provides more flexible overlap sizes as these overlaps can be 1, 2, 3, ... pixels, whereas a one-pixel spatial shift on horizontal or vertical axes produces a larger number of overlaps in the standard 2D case (otherwise, one operates row-wise or column-wise operators and thus, loses spatial dependencies).

In the following, we address $2D+t$ data analysis and all the properties highlighted above will be integrated to derive a consistent video-to-image conversion, prior to machine learning for violence detection. This validating application on violence has the following specificity: when one considers heterogeneous actions in the same framework, then performance is weak from a state of the art tour. Thus, a large number of formalisms and frameworks have to be investigated in order to provide relevant solutions.

IV. Visual violence detection in videos: the Hilbert image framework

A. Positioning

1) Homogeneous versus *heterogeneous action recognition:* In terms of action categorization, homogeneity means that the category elements share the same spatio-temporal flow property: for instance, the action consisting in *applying eye makeup* has coarsely the same movements in UCF-101: from the bottle to the eyelashes and eyebrows. There is no other way to put this maskara (at least in these examples) and this makes the action more predictable: eyelashes / eyebrows, a bottle of mascara and an arm movement are enough to determine the action with great certainty.

On the contrary, when we consider a complex concept like visual violence, there are many different properties (multiple causes) that can lead to the same result: a measurable feeling of violence. For example, in a PEGI-16 video category associated with getting rid of game opponents, different types of tools (conventional weapons such as knives, swords, pistol, rifle, bomb, gas, but also standard non-weapon objects like a car, a chair or a bottle that are occasionally transformed into weapons) will lead, in a wide range of movements, to a given level of violence. We are concerned by heterogeneous action recognition insofar as any given category is subject to be composed by a large set of clips reporting very different spatio-temporal properties.

2) Thematic positioning: violence recognition: With the explosion in the number of digital imaging and communication systems, it becomes crucial to develop automatic tools for the supervision of sensitive multimedia contents. Some supervisors for personalizing content delivery are already available in terms of filtering websites and specific web-data, for example *SafeSearch* [22].

However, despite their relevancy when file metadata associated with a content are concise and accurate or when image data contains explicit sensitive symbols, these software solutions are inefficient for violent action recognition because they are not trained for a systematic video analysis on the wide variety of violent, somewhat subjective actions.

Some benchmarking initiatives have thus been proposed recently in order to address emotional impacts of video contents. Among these initiatives, one can mention:

- the construction of Violent Scene Datasets (VSD) from Technicolor [23] and Mediaeval [24], [25] on the basis of evaluation performed by a consortium of experts.
- the proposal of feature extraction and learning for certain specific violence facts, for example the violence:
  - induced by the crowd [26], [27], [28], [29], [30],
  - in cartoon videos [31], [32],
  - associated with the presence of blood [33] or a crime scene [34] or a violent shot [35].

However, the general case remains intricate as even in the limited case of violent shot detection with respect to some Hollywood movies [35], the best strategy involving temporal features and interest points is limited to 72% accuracy. The main issues leading to this limitation are:

- [Issue 1] Generalizability of concepts relating violence with respect to the representativeness and coverability of collected examples: the large number of situations that lead to violent affect feeling and the subjective interpretation of these situations make collecting the representative examples intricate.
- [Issue 2] Generalizability of detection experimentations in terms of compatibility of the video feature extraction stage with respect to existing machine learning frameworks

For handling [Issue 1], the following Section IV-B proposes two experimental benchmarking frameworks: the first considers, in a joint framework, both VSD@Mediaeval and VSD@Technicolor. The second proposes upgrading the database obtained by fusing VSD@Mediaeval and VSD@Technicolor in order to derive a 3-level violence category database.

For solving [Issue 2], we propose in IV-C, converting the library of video sequences just obtained to RGB timed-image databases by using Hilbert based 2D→1D projection for any color channel. This will provide us with a more flexible framework that can be spread over all available CNNs platforms having their design constrained to operate on image format.

B. Multi-concept violence detection framework

1) 3-level violence partitioning : VSD@Mediaeval (2015 edition) provides a set of short video clips extracted from 199 movies under Creative Commons license and associated with binary 0 and 1 violent categories. The non-violent category "0" consists of 10398 video clips whereas the violent "1" category is composed by 502 video clips. The binary annotations have been made by a team of Mediaeval international challenge, see [24], [25] for details. The clips cover a wide range of violent actions. However, the binary categorization performed by Mediaeval is very restrictive due to the fact that one can find in the same category, slap and vampirism simulations whereas the difference of affect induced by these actions is clearly huge.

We propose hereafter from an expert-based evaluation system, a split of the violence VSD@Mediaeval category into 2 sub-categories: moderately and extremely violent. In this benchmark, the following experimental setup has been deployed:

- Consider the 'violent' category of VSD@Mediaeval (502 video clips available).
- For any clip of this category, ask examiners and experts to provide a violence scale (emotional impact felt). The scales are chosen to range from 0 to 4, in ascending violence feeling. A total of 19 examiners (researchers and students) have provided a degree for at least one video and 4 experts have meticulously watched and annotated the whole set of clips.
- Fuse, by using a majority vote rule, the scales [0, 2] to form the *moderate violence* category and assign [3, 4] scales to the *extreme violence* category. In case of conflict, the corresponding video file is excluded from the 3-level benchmark. This provides 266 moderate violence examples and 232 extreme violence ones for VSD@Mediaeval (4 violence files have been excluded due to conflict in the majority vote).

From this expert based benchmarking and by taking the non-violent category into account, we obtain a 3-category split.

2) Violent category augmentation from VSD@Technicolor: In order to augment the number of violence examples in the 3-category split of VSD@Mediaeval, we supplemented this database by including VSD@Technicolor dataset.

For VSD@Mediaeval complementation by using VSD@Technicolor, we consider only VSD@Technicolor videos having similar examples in the wide range of situations covered by VSD@Mediaeval. For situations where a VSD@Technicolor scene has no related example in VSD@Mediaeval (case for airplane crashes), we exclude the corresponding files from the 3-level categories. However, we include these examples in the 2-level categories as they are known violent behavior that can balance the large number of 10398 collected non-violent files.

The 2 datasets obtained from this experimental setup described have sample characteristics summarized in Table II. These datasets are referred from now on as VSD-LK where the Level K denotes the number of violence categories, with $K \in \{2, 3\}$ in this paper.

TABLE II
DATABASES DERIVED FROM CATEGORY REFINEMENTS AND COMPLETION ON MEDIAEVAL AND VSD VIOLENCE DATABASES. SPECIFICALLY, FOR THE SAKE OF COMPACTING NEXT TABLES, CATEGORIES *Violence*, *Non-Violence*, *Moderate-Violence*, *Extreme-Violence* ARE ABBREVIATED RESPECTIVELY AS *Is-V*, *No-V*, *Mo-V*, *Ex-V*.

Name	**Categories and number of samples**		
VSD-L2	*Is-V*		*No-V*
	1 137		10398
VSD-L3	*Mo-V*	*Ex-V*	*No-V*
	406	418	10398

Finally, we have derived from an analysis of the different expert annotations, the violence feeling diagram presented in Figure 5. This diagram highlights that if one tries to see violence as a mathematical function, then it has to be increasing when the following variables increase: intensity of impact, duration of violent facts and their frequency (recurrence and ferity), number of protagonists (agent and victims), agent intentionality, victim impactance and violence exhibitance.

3) Representativeness and coverage of VSD-L2 and VSD-L3 : Table III provides 10 major observables relating violence. The observables involve interacting humans, objects and fluids, as well as warning symbols and violence perceptible without explicit interacting causes.

In interacting categories, the direct cause and its implicit/explicit effects are visible in general. For instance, a car accident can imply 2 or more interacting cars, or a car and another object. Implicit human presence in the cars under accident is the direct source of emotional impacts, even if the video does not show car occupants.

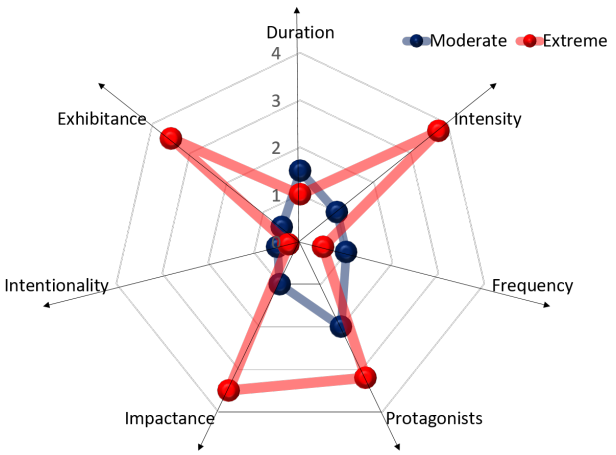

Fig. 5. Balance sheet of main variables governing violence feelings and some annotation examples for moderate and extreme violence occurrence: the blue diagram is representative of a short altercation between few persons and the red one is representative of an airliner crash (involving a large number of victims).

TABLE III
TEN OBSERVABLES AT LARGE REFERRING VIOLENCE. PEGI REFERS TO *Pan European Game Information* (AUDIOVISUAL ANNOTATION SYMBOL FOR PUBLIC AGE RESTRICTIONS).

	Observables	Examples
	Violence from interactions (external induction)	
1	Human/Human	Battle, slap, punch.
2	Human/Object	Surgery, mutilation, projectile throw.
3	Human/Fluid	Drowning, gazing.
4	Human/Animal	Human attack by animals, animal hunting and shoot.
5	Object/Object	Car *versus* car accidents, crash of an airplane (ground or sea).
6	Animal/Animal	Animals fight clubs, predator *versus* prey showdowns.
	Suspicious motions (self-induction)	
7	Living body abnormal motion	Terror or aggressive faces, person falling down
8	Inert structure abnormal motion	Conflagrations, explosion, smoke, flames and ashes, flowing blood.
	Sensitive objects and symbols	
9	Sensitive objects	Guns, weapons, bombs, broken glasses, stripped electrical socket, blood stain frightful masks and veils.
10	Sensitive symbols	PEGI "-18" / "-16" / "-12" / "-10", flammable material signs, swastika injury and hatred posters

In the categories where violence is not due to multiple agent interactions, for example in-air burn of an airplane, the direct cause is not visible in general, but violence feeling is not necessarily diminished.

VSD-L2 and VSD-L3 contain several examples for almost all 10 observables given in Table III, excepted the observables involving animals. These can be collected and processed separately. Note also that some particular sensitive objects and symbols relating violence (example of PEGI symbols) are considered as specific in this paper: their physical and geometrical appearance lead to very limited confusion and disambiguation, when necessary, requires only few additional samples. These observables can be detected very well by using specialized CNNs and as such, the corresponding symbols are excluded from VSD-L2 and VSD-L3. In the sequel, we will thus focus on brute-force violence detection from scratch (without the aid of additional specialized system or expert) on VSD-L2 and VSD-L3.

C. Violence detection: Hilbert based Video to Timed-Image framework

1) Timed-image representation of VSD-L2 and VSD-L3 video clips: Consider a video clip $C = C(x, y, t)$ with length T (number of images) defined on grid $\mathcal{G}(M)$, with $t = 1, 2, \ldots, T$ referring to the time variable. In order to derive a framework directly transposable to standard image classification architectures, we consider a 2D Hilbert based timed meta-image (*timed-image* terminology) representation for this image time series. This representation consists in converting, for any given t, the 2D array $C(\bullet, \bullet, t)$ in a 1D array associated with $\mathcal{V}(M)$ indexing by using Hilbert space-filling curve (see Section II-A). This yields a timed-image with size $4^M \times T$ pixels for clip C. Color components are added as multiple channels when available. For learning purpose and due to limited computing capabilities, we resize the time-images to 1024×256 pixels. A 240p resolution video file (84 480 pixels per image) having 384 image samples will thus be impacted by a high spatial compression rate, but a very low temporal one. The compressibility properties of Hilbert description guarantee less impact on the lowering of the 2D meta-image resolution than when resizing directly the 2D+t video file.

From this representation, video clips in VSD-L2 and VSD-L3 are converted in timed-image databases without loss of information and with the nice compressibility properties with respect to convolution operators illustrated in Section III-B. The 2D timed-image based CNN framework is expected to ensure a good balance in machine learning between spatial (considered as one variable) and temporal variables. This framework also avoids resorting to three directional convolutions where oriented weight updating strategies are intricate, given a direct 2D+t learning framework.

2) State-of-the-art performance on VSD-L2 and VSD-L3: In this section, we consider performance evaluation for the top state-of-the-art CNNs with respect to time-image features. This will provide some insights on the framework that leads to more generalization capability with respect to the new VSD-L2 and VSD-L3 time-image representation. The CNNs considered are: Alexnet (BVLC version) [36], VGG-19 [37] and GoogleNet [38], considered in a transfer learning strategy. Alexnet-BVLC has 8 layers with about 60 million parameters, VGG-19 is composed of 19 layers with more than 130 million parameters and GoogleNet involves 22 layers with about 4 million parameters (the parameter count principle must be understood from a training-from-scratch here). Classification features are extracted from the fully-connected layers of these CNNs. No data augmentation procedures have been considered in the experimental setup.

TABLE IV
PERFORMANCE EVALUATION (CONFUSION MATRICES IN %) FROM ALEXNET, VGG-19 AND GOOGLENET TRANSFER LEARNING VSD-L2 AND VSD-L3.

Performance on VSD-L2									
BVLC AlexNet			VGG-19			GoogleNet		
	No-V	*Is-V*		*No-V*	*Is-V*		*No-V*	*Is-V*
No-V	96.81	03.19	*No-V*	54.55	45.45	*No-V*	99.34	00.66
Is-V	68.34	31.66	*Is-V*	40.95	59.05	*Is-V*	84.67	15.33
Accuracy:	64.24		Accuracy:	56.80		Accuracy:	57.33	

Performance on VSD-L3									
	BVLC AlexNet			VGG-19			GoogleNet		
	No-V	*Mo-V*	*Ex-V*	*No-V*	*Mo-V*	*Ex-V*	*No-V*	*Mo-V*	*Ex-V*
No-V	99.45	00.44	00.11	99.15	00.77	00.08	98.46	01.51	00.03
Mo-V	92.96	06.34	00.70	90.14	09.86	00.00	81.69	18.31	00.00
Ex-V	89.04	02.05	08.90	89.73	07.53	02.74	80.14	16.44	03.42
	Accuracy:		38.23	Accuracy:		37.25	Accuracy:		40.07

Experimental results are given respectively for VSD-L2 and VSD-L3 in Table IV in terms of classification confusion matrices and mean classification accuracy. Table IV shows that for VSD-L2, the lightweight Alexnet framework guaranties more generalization properties[7] with respect to this transfer learning setup. All tested frameworks however fail in categorizing the VSD-L3 which involve fine grained violence categories.

In the following, we will thus keep Alexnet framework and a learning from scratch strategy, so as to force CNN weights to be more convenient to the meta-image representation.

3) Learning timed-image violence features from scratch : Table V presents visual violence detection results obtained on VSD-L2 from the timed-image CNN framework, when the CNN is trained from scratch[8]. The CNN model used is a lightweight variant of [36], both pertain to the framework given in [21]. In addition, we have used a weighted entropy function to handle imbalance in VSD categories, the weights used corresponding to the proportions of elements in VSD-L2. This CNN has characteristics summarized in Table VI. The results obtained in Table V show that one can reach a good level of accuracy with a timed-image CNN in heterogeneous[9] action recognition.

We have also provided in Table V and for the sake of comparison, experimental results obtained on a 3D CNN trained from scratch, where convolution kernels and pooling are both 3D [2] for any layer associated with such operators. For this 3D CNN framework, several simulations have shown limited performance on VSD-L2, despite their satisfactory training performance in terms of decay of the loss function and increase of the training batch accuracies. In contrast, it is worth noticing that several simulations have led to more than 75% accuracy for timed-image version of VSD-L2 (variable accuracies on the 2 categories). It is worth noticing that timed-image computation and compression being done prior to the learning stage, computation complexity amounts, when focusing on a single convolution filter, to comparing $\mathcal{O}\left(MNT[\log(MN/4)+\log(T)]/4\right)$ for the 2D timed-image framework and $\mathcal{O}\left(MNT[\log(MN)+\log(T)]\right)$ for the 3D CNN (which is approximately 4.37 times faster for the timed-image framework when assuming a standard 240p[10] video encoding over 160 image frames per video clip and a compression rate of 1/2 both in frame rows and columns).

In addition, for VSD-L3, the 3D CNN framework fails in categorization over a huge number of tests as it assigns almost all tested images to a single category (thus 33% accuracy that are comparable to a deterministic assignment to one class). In contrast, for the timed-image framework, Table VII provides an illustrative example showing more than 20% accuracy increment when compared to a deterministic assignment over one class. These results motivate us in the pursuit of violence data collection and benchmarking as a natural augmentation of the number of moderate and extreme violence examples will lead to a more relevant discrimination for the multi-level violence categorization.

From the intricacy of finding good performance on a direct 3D CNN when no privileged dimension is imposed, we may conjecture that standard weight updating strategies should integrate in the 3D framework, penalty terms with respect to space and time dimensions (so as to penalize highly spatial dimensions and less temporal dimension for instance). Solving this conjecture is out of the scope of the present paper. However, this conjecture is strengthened by the fact that in

[7]The number of output categories seems to play a significant role in terms of generalization: when the number of output categories is very small (VSD-L2 for instance), then the fusion of activations from higher dimensional spaces (VGG-19 and GoogleNet) for deriving a binary output has several consequences: among these consequences, the most important one is blurring activation information by aggregation. However, this assertion needs to be checked carefully over a large class of problems.

[8]Simulations have been performed by using an Nvidia Tesla V100-PCIE having GPU compute capability 7.0 and 16GB of dedicated memory. The mini-batch size used is 128 timed-images (approximately 12 seconds of video clip per timed-image).

[9]Heterogeneous action case study is such that a wide range of different, non-necessarily correlated scenarios, pertain to the same category. Example: car accidents, battle, bloody scenes, terror faces, all are in the same category.

[10]In this case, $M = 240$, $N = 352$ and we recall that in practice, the number of frames T varies depending on the video sequence considered.

TABLE V
CONFUSION MATRIX (AVERAGE RETRIEVAL PER CATEGORY AND AVERAGE INTER-CLASS CONFUSION) FOR VSD-L2 TIMED-IMAGE VERSUS 3D CNN CATEGORIZATION. VSD-L2 IS DESCRIBED IN TABLE II.

Experimental results on VSD-L2									
3D CNN			(2D) Dynamic-Image CNN			(2D) Timed-Image CNN		
Category	*No-V*	*Is-V*	Category	*No-V*	*Is-V*	Category	*No-V*	*Is-V*
No-V	94 %	06 %	*No-V*	71 %	29 %	*No-V*	94 %	6 %
Is-V	78 %	22 %	*Is-V*	43 %	57 %	*Is-V*	36 %	64 %
Accuracy	58 %		Accuracy	64 %		Accuracy	79 %	

TABLE VI
CONVOLUTIONAL NEURAL NETWORK FOR VIOLENCE DETECTION ON VSD-L2 AND VSD-L3 CHARACTERISTICS.

Layer	Content	$\#N$ of Elements	Element size	# of Channels
1	VSD-L2 VSD-L3	11535 11222	$4^M \times T$ = 1024×256	3
2	'Convolution'	96	11×11	3
3	'ReLU'[11]	Element-wise (one to one)		
4	'Normalization'	Cross channel with 5 channels/element		
5	'Max Pooling'	Sub-sampling: maximum over a 3×3 neighborhood		
6	'Convolution'	128	9×9	96
7	'ReLU'	Element-wise (one to one)		
8	'Normalization'	Cross channel with 5 channels/element		
9	'Max Pooling'	Sub-sampling: maximum over a 3×3 neighborhood		
10	'Convolution'	384	7×7	128
11	'ReLU'	Element-wise (one to one)		
12	'Convolution'	192	5×5	384
13	'ReLU'	Element-wise (one to one)		
14	'Convolution'	128	3×3	192
15	'ReLU'	Element-wise (one to one)		
16	'Max Pooling'	Sub-sampling: maximum over a 3×3 neighborhood		
17	'Fully Connected'	Neuron matrix / Sizes [Input 8192 - Output 16]		
18	'ReLU'	Element-wise (one to one)		
19	'Fully Connected'	Neuron matrix / Sizes [Input 16 - Output VSD-L2: 2 / VSD-L3: 3]		
20	'Softmax'	Probability distributions with respect to 4 outputs		
21	'Classification'	Weighted cross-entropy (Output: VSD category)		
Training/testing samples are respectively 65% / 35% of VSD-L2 data.				
Training/testing samples are respectively 75% / 25% of VSD-L3 data.				

other action recognition frameworks involving homogeneous[12] categories, authors have separated spatial and temporal flow features instead of a direct 3D learning on 3D data. We believe that this separation is a somewhat counterintuitive procedure as we may expect the CNN to extract itself the necessary features for a discriminant analysis when dominant dimensions are integrated with respect to training objectives.

To conclude this section, one may question about the best 3D-to-2D strategy among timed-images and the so-called *dynamic images* (consisting in integrating the video motion flow into a 2D map [39], [40]). We believe that the latter sounds relevancy when the camera is fixed, and a single object/person is doing a single elementary action (case of the so-called "homogeneous" action recognition). However, in the context addressed by this paper, a short video clip can involve several angles from several cameras and, in addition, actions are composite (hands, guns, bloods and laugh can be integrated in a single clip): a motion map will then fail to be understandable because of superimposition of too many different sources of displacements. Moreover, inverting camera angles leads to abrupt transition that has consequence, huge displacement fields. This is the reason why the results given in Table V for dynamic image based learning[13] are not very competitive: the "heterogeneous" violence context is not favorable to such an integral flow based violence categorization.

V. CONCLUSION AND PROSPECTS

This work has addressed intrinsic 3D feature learning from Hilbert based meta-image description of 3D data. The 3D=2D+X description has been designed on the basis of duality between spatial 2D observations and the additional dimension X that can relate time, wavelength or depth information. The aim of this description was obtaining a good balance between spatial information (compacted in one dimension) and the additional information provided by variations in X.

[12]Homogeneous action case study is such that any given category is composed by visually correlated features. Example: in a 'soccer penalty kick' category, all video samples are visually correlated in the sense that the scene contains a ball and the optical flow shows interaction between this ball and a foot.

[13]We have used the same CNN and same training parameters as for the timed-image framework presented in Table V, the dynamic images being computed as in [39].

TABLE VII
CONFUSION MATRIX (AVERAGE RETRIEVAL PER CATEGORY AND AVERAGE INTER-CLASS CONFUSION) FOR VSD-L3 TIMED-IMAGE CATEGORIZATION. VSD-L3 IS DESCRIBED IN TABLE II.

Experimental results on VSD-L3			
2D-Timed-Image CNN			
Category	*No-V*	*Mo-V*	*Ex-V*
No-V	84 %	8 %	8 %
Mo-V	43 %	30 %	27 %
Ex-V	29 %	14 %	57 %
Mean accuracy	57 %		

More specifically, we have shown that Hilbert based scan of a 2D grid yields a highly compressible 1D data description when the grid relates image content. Thus, by concatenating the third X dimension observations to this Hilbert 1D data description, we have derived an image description (termed meta-image) of the 2D+X data. Different characterizations of the meta-image description have also been derived.

From experimental simulations, it appears that this description makes obtaining good performance possible in heterogeneous visual violence action recognition from 2D+t video data. More precisely, in this case of sensitive action recognition, the issue addressed has concerned providing an alternative to handcrafted supervision and learning of 2D+t discriminant features. Indeed, it is more trivial to deploy existing image based learning frameworks on the meta-image description (called timed-image when $X = t$) than expending energy in the search of stealth 2D+t information in 3 directions.

From a theoretical point of view, the meta-image framework is equivalent in some sense to a direct 3D framework: the reverse implication is trivial because a 3D convolution filter can be easily reshaped in the Hilbert meta-image domain prior to analysis. However, the direct implication is computationally intricate: for instance, given a 1D convolution in the Hilbert domain, its 2D concern involves imposing some zeros at specific locations on the 2D grid and further concatenate the third variable coefficients by respecting the same rule, which is not thinkable for any iteration of the several hundreds of convolution filters associated with a deep CNN.

A problem of both theoretical and practical interests is the analysis of this framework in several problems where the additional dimension X can play several types of roles in information retrieval. This will allow, depending on performance decays with respect to information compression in this dimension, understanding which feature plays the main discriminant role in practice, where it is worth noting that information compression can be due to large convolution strides or pooling sizes, to data re-sampling prior to learning, *etc*. Such analysis will require several years of simulations on a wide range of heterogeneous visual actions and is reported here as an open issue.

We end by providing an experimental fact observed for bidirectional recurrent LSTM: despite the fact that Hilbert timed-image provides a natural way of converting video data into vectorized and dependent sequences of time series, training failed (no decaying trend in the error function) for a wide range of LSTM setups tested for VSD classification. This can be due to the heterogeneous issue for which examples provided in VSD show very few, almost no explicit temporal similarities. We may expect performance for LSTM on timed-images involved in homogeneous action recognition frameworks.

In future work, we plan to investigate fusion approaches between timed-image features and their corresponding spatial *versus* optical flow information: the corresponding setup may involve specifying several learning parameters in a pre-training stage. This approach can be seen as a 3 stream extension, in a framework similar to [7], [6], but involving all 2D spatial, 2D timed-image and 1D optical flow features.

ACKNOWLEDGEMENT

The work was supported by grant SAINS of the Université Savoie Mont Blanc, France. Numerical simulations have been performed thanks to the facilities offered by MUST computing center of Université Savoie Mont Blanc and French *National Agency of Research* PHOENIX ANR-15-CE23-0012 workstations. The violence datasets post-processed and analyzed were supplied by MEDIAEVAL and TECHNICOLOR initiatives. The authors are grateful to the USMB/LISTIC members who have participated in video clip annotations. The authors are also very grateful to Cécile BARBIER, Tao LAURENT, Rim TRABELSI and Nicolas MEGER for their participation in terms of computing advice, insightful comments, design of the annotation website, as well as the evaluation protocol.

REFERENCES

[1] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 689–692. [Online]. Available: http://doi.acm.org/10.1145/2733373.2807412

[2] P. Sun, "Mexconv3d: Matlab mex implementation of the basic operations for 3d (volume) convolutional neural network," *Framework, Avaliable online*, vol. 54, 2016. [Online]. Available: https://github.com/pengsun/MexConv3D

[3] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank, "Asymmetric 3d convolutional neural networks for action recognition," *Pattern Recognition*, vol. 85, pp. 1 – 12, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318302632

[5] J. Li, X. Liu, M. Zhang, and D. Wang, "Spatio-temporal deformable 3d convnets with attention for action recognition," *Pattern Recognition*, vol. 98, p. 107037, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320319303383

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 568–576. [Online]. Available: http://dl.acm.org/citation.cfm?id=2968826.2968890

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, July 2017, pp. 4724–4733.

[9] J. Zhang and H. Hu, "Domain learning joint with semantic adaptation for human action recognition," *Pattern Recognition*, vol. 90, pp. 196 – 209, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320319300470

[10] D. Zhang, L. He, Z. Tu, S. Zhang, F. Han, and B. Yang, "Learning motion representation for real-time spatio-temporal action localization," *Pattern Recognition*, vol. 103, p. 107312, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320320301163

[11] L. Chen, Z. Song, J. Lu, and J. Zhou, "Learning principal orientations and residual descriptor for action recognition," *Pattern Recognition*, vol. 86, pp. 14 – 26, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318303157

[12] P. Wang, L. Liu, C. Shen, and H. T. Shen, "Order-aware convolutional pooling for video based action recognition," *Pattern Recognition*, vol. 91, pp. 357 – 365, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320319301013

[13] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[14] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," *AAAI Conference on Artificial Intelligence*, Jan 2018. [Online]. Available: http://par.nsf.gov/biblio/10056961

[15] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[16] Y.-G. Yoon, P. Dai, J. Wohlwend, J.-B. Chang, A. H. Marblestone, and E. S. Boyden, "Feasibility of 3d reconstruction of neural morphology using expansion microscopy and barcode-guided agglomeration," *Frontiers in Computational Neuroscience*, vol. 11, no. 97, 2017.

[17] A. Chadha, A. Abbas, and Y. Andreopoulos, "Compressed-domain video classification with deep neural networks: "there's way too much information to decode the matrix"," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 1832–1836.

[18] J. Valantinas, "On the use of space-filling curves in changing image dimensionality," *Information Technology and Control*, vol. 34.4, 2005.

[19] J.-Y. Liang, C.-S. Chen, C.-H. Huang, and L. Liu, "Lossless compression of medical images using hilbert space-filling curves," *Computerized Medical Imaging and Graphics*, vol. 32, no. 3, pp. 174 – 182, 2008.

[20] D. Hilbert, *Über die stetige Abbildung einer Linie auf ein Flächenstück*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1935, pp. 1–2.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[22] Google Web Search, "Safesearch for google file system," ©Google, 2003-2018.

[23] M. Schedi, M. Sjöberg, I. Mironică, B. Ionescu, V. L. Quang, Y. Jiang, and C. Demarty, "Vsd2014: A dataset for violent scenes detection in hollywood movies and web videos," in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2015, pp. 1–6.

[24] C. Demarty, B. Ionescu, Y. Jiang, V. L. Quang, M. Schedl, and C. Penet, "Benchmarking violent scenes detection in movies," in *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2014, pp. 1–6.

[25] M. Sjöberg, Y. Baveye, H. Wang, V. L. Quang, B. Ionescu, E. Dellandréa, M. Schedl, C.-H. Demarty, and L. Chen, "The MediaEval 2015 Affective Impact of Movies Task," in *Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany, September 14-15, CEUR-WS.org, ISSN 1613-0073*, 2015. [Online]. Available: http://ceur-ws.org/Vol-1436/Paper_1.pdf

[26] W. Song, D. Zhang, X. Zhao, J. Yu, R. Zheng, and A. Wang, "A novel violent video detection scheme based on modified 3d convolutional neural networks." *IEEE Access*, pp. 1–1, 2019.

[27] T. Senst, V. Eiselein, A. Kuhn, and T. Sikora, "Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2945–2956, Dec 2017.

[28] S. Choudhary, N. Ojha, and V. Singh, "Real-time crowd behavior detection using sift feature extraction technique in video sequences," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, June 2017, pp. 936–940.

[29] N. Zhuang, J. Ye, and K. A. Hua, "Convolutional dlstm for crowd scene