Large-scale, Fast and Accurate Shot Boundary Detection through Spatio-temporal Convolutional Neural Networks

Ahmed Hassanien[1], Mohamed Elgharib[1], Ahmed Selim[2], Sung-Ho Bae[3], Mohamed Hefeeda[4], and Wojciech Matusik[3]

[1]Qatar Computing Research Institute, HBKU
[2]Trinity College Dublin, CONNECT Center
[3]MIT CSAIL
[4]Simon Fraser University

Abstract—**Shot boundary detection (SBD) is an important pre-processing step for video manipulation. Here, each segment of frames is classified as either sharp, gradual or no transition. Current SBD techniques analyze hand-crafted features and attempt to optimize both detection accuracy and processing speed. However, the heavy computations of optical flow prevents this from happening. To achieve this aim, we present an SBD technique based on spatio-temporal Convolutional Neural Networks (CNN). Since current datasets are not large enough to train an accurate SBD CNN, we are the first to present a very large SBD dataset that allows deep neural networks techniques to be effectively applied. Our dataset contains more than 3.5 million frames of sharp and gradual transitions. The transitions are generated synthetically using image compositing models. Our dataset contain additional 70,000 frames of important hard-negative no transitions. We perform the largest evaluation to date for one SBD algorithm, on real and synthetic data, containing more than 4.85 million frames. In comparison to the state of the art, we outperform dissolve gradual detection, generate competitive performance for sharp detections and produce significant improvement in wipes. In addition, we are up to 11 times faster than the state of the art.**

Index Terms—**Shot Boundary Detection, Convolutional Neural Networks, optical flow, spatio-temporal.**

I. INTRODUCTION

With the wide adoption of digital video, the demand for editing and manipulating video content is in continuous rise. This, however, requires better understanding of videos and their composition. Videos are composed of different camera shots placed after each other. A video shot transitions into another through several forms of visual effect. These visual effects can be classified into two main categories: sharp and gradual [1] as shown in Figure 1. The former is a sudden change of the shot over 1 frame, while gradual transitions occur over multiple frames. Gradual transitions are further classified into dissolve and non-dissolve. The former includes cases such as semi-transparent graduals, fade in and fade out (see Figure 1). Non-dissolve are dominated by wipes (see Figure 1). Wipe graduals have a much wider variety than the dissolve graduals.

Video post-processing techniques are in rising popularity and they cover a wide range of applications. This includes video coding [2], visual quality enhancement [3], [4], graphics rendering [5]–[7], video understanding [8], [9] and many others [10], [11]. Such post-processing techniques, however, are based on assumptions, some of which can be violated during shot transitions. For instance, many techniques assume the presence of one layer at one spatial point, an assumption heavily violated during dissolve transitions. This can lead to unpleasant artifacts as in the case of 2D-to-3D conversion (see Figure 2). Here, the disparity maps can undergo strong artifacts during gradual transitions. Hence, detecting video transitions and assigning a special treatment for them during post-production is an important and desirable step. However, with the high computational demand of many post-production techniques, as well as the real-time requirement of some, shot boundaries detection (SBD) needs to be performed with both high detection accuracy and very fast processing speed.

Current SBD techniques analyze hand-crafted features [1], [12]–[20]. Fast techniques analyze only spatial information such as intensity histogram [13], [19], edges [15], mutual information and others [16]–[18], [20]. Such techniques, while being fast, generate poor detection. To boost detection, motion information is incorporated through optical flow [1], [12], [21], [22]. However, the heavy computations of optical flow [23]–[25] make such techniques slow. As SBD techniques are commonly used as a pre-processing step for video manipulation, optimizing both their detection accuracy and processing speed is important. This, however, remains a challenging problem.

We present DeepSBD, a fast and accurate shot boundary detection through convolutional neural networks (CNN). We exploit big data to achieve high detection performance. In addition, we exploit the parallelizable nature and common GPU implementations of CNNs to achieve fast processing speed. Our technique takes a segment of 16 frames as input, and classifies it as either gradual, sharp or no-transition. It analysis both spatial and temporal information through an effective 3D convolutional network for video processing, inspired by C3D [26].

To train our network, we need a well-annotated very large dataset. Despite datasets already exist from the TRECVID challenge and others [1], [27], experiments show they are not

Fig. 1. Shot transitions are classified into two main categories: sharp and gradual. Gradual transitions are further classified into soft and wipes. Soft include semi-transparent, fade in and fade out. Wipes are the most ill defined form of transitions.

sufficient to train a high accuracy CNN solution. In addition, the vast majority of these datasets are used for testing and evaluating different techniques, and hence should not be used for training. To overcome this problem, we present a very large SBD dataset with clean and accurate annotations capable of training a highly accurate CNN SBD solution. This also allows us to test on all available TRECVID data (3.9 million frames) [1]. The first dataset portion, SBD_Syn, is generated synthetically using image compositing models. It contains 220,339 sharp and gradual segments, each segment contains 16 frames. The second portion, SBD_BT, contains 4,427 no transition segments. They are carefully manually annotated in a way to improve detector's precision; they act as hard-negatives. We optionally use 1 TRECVID release (2005) and another SBD dataset of Baraldi et al. [27] to further improve performance. These datasets have 18,027 total transitions with prior annotations. That is only 7% of all training datasets.

Aspects of novelty of our work include:

1) The first CNN SBD technique. We outperform dissolve gradual detection, generate competitive performance for sharp detections and produce significant improvement in wipes. In addition, we are up to 11 times faster than the state of the art.

2) Introduction of a new very large SBD dataset for training an accurate CNN model. Our dataset contains 3.5 million frames of synthetic transitions and 70,000 frames of hard negative no-transitions.

3) A large wipes dataset containing 1.1 million frames. We will release all our data-sets and code to encourage future research.

4) The largest SBD evaluation to date on 4.85 million frames. 3.9 million frames are from all TRECVID years [1] while most of the rest are synthetically generated.

The next section reviews the state of the art. Here, we discuss the main components of our solution including current SBD techniques, current available SBD datasets and CNN solutions for video spatio-temporal analysis. We then present our SBD solution with emphasize on our detection system and our dataset generation process. Section IV presents detailed

Fig. 2. A figure illustrating the importance of shot boundary detection during 2D-to-3D conversion. Here, we show consecutive frames from two sequences. For each frame we show its original RGB image (left) and the corresponding disparity map (right) as estimated by the 2D-to-3D conversion technique of Bae et al. [7]. The disparity maps is a gray-scale image where white represents objects close to the screen. Note how the disparity captures the humans outlines before and after transitions (first and last frame). During gradual transition, however, the outline is severely destroyed. This is because gradual transitions violate the single layer model used by many video processing techniques. Hence, it is desirable to detect such gradual transitions and to give them a special treatment during post-processing. For this, shot boundary detection is essential.

results and analysis. The results are also supported by a supplementary material (in .pdf format, please examine). Section V is conclusion.

II. State of the Art

A. Shot Boundary Detection Techniques

SBD techniques [1], [12], [13], [22] extract features and analyze them temporally. Detection is then performed by finding temporal profiles that fit the examined transition model. Sharp transitions undergo a sudden change in the temporal profile over one frame. Gradual transitions exhibit a more

stretched change in time. Current SBD techniques are classified into two main categories: spatial-only and spatio-temporal analysis based. The former estimates the temporal profile by comparing only spatial features [1], [12]–[20]. A number of spatial features are used such as color histograms [13], [19], edges [15], mutual information and Entropy [16], wavelet representations [12], SURF [28] and many others [17], [18], [20], [29].

Spatial-only SBD methods generate conservative detection accuracy with fast processing speed. Spatio-temporal techniques use optical flow to make detection more robust to scene and camera motions [12], [14], [30], [31], [31]. Such motions can arise due to camera movements and shakiness and often confuse the detection process. Hence, optical flow [23]–[25] between neighboring frames is estimated and removed through frame interpolation. Analysis of the temporal profile is then proceeded as in the spatial-only techniques. Here, motion compensation often reduces false detections of SBD. The main drawback of spatio-temporal techniques, however, is the heavy computations of optical flow.

Among the rich SBD literature, four of the best performing and/or most recent techniques are Liu et al. [22], Yuan et al. [32], Lu et al. [13] and Priya et al. [12]. Lu et al. focuses more on generating fast results and hence they do not incorporate motion information. Their technique is based on assessing temporal discontinuities through HSV histogram. Priya et al. [12] proposed a wavelet based feature vector that measures four main quantities: color, edge, texture, and motion strength. The feature vector is extracted for each frame of a sequence and the temporal profile is estimated through frame differencing. Liu et al. [22] uses a large number of features including color, histogram, edge, motion and related statistical features. Liu et al., Priya et al. and Yuan et al. all focus on high detection accuracy. This, however, comes with the high cost of optical flow. Furthermore, the techniques of Apostolidis et al. [28] and Berladi et al. [29] were recently released. They analyze only spatial information such as SUFR and HSV/color histogram and hence often generate conservative performance with fast processing speed.

To the best of our knowledge, Liu et al. [22] is the latest wipe detector. A candidate transition segment is proposed and the difference between each frame and the start and end frame is calculated. This generates two curves, one for the start and another for the end of the segment. For wipes, the curves should have opposing gradients and somewhat linear. Furthermore, to reduce errors due to camera and object movements, motion compensated frame differencing is used.

B. SBD Datasets

Between the years 2001 to 2007, the National Institute of Standards and Technology (NIST) [33] maintained data for the TRECVID shot boundary detection (SBD) challenge [1]. The dataset contains a wide variety of content including color, gray-scale, indoor, outdoor, outer-space and different levels of noise. The dataset has a total of 4,333,153 frames with 24,423 transitions, 64% of which are sharp. The rest are gradual. Transitions were manually annotated in a way to distinguish between sharp and graduals. Four more releases from a different challenge were maintained by NIST that contain data relevant to SBD. The releases are T2007t, T2007d, T2008 and T2009, containing 34,765,424 frames with 155,902 transitions. The annotations of these data, however, do not distinguish between sharp and graduals. Finally, one more data release related to SBD was generated by Baraldi et al. [27]. Here, the authors addressed the different application of video scene segmentation.

We collected all the SBD related dataset. However, some TRECVID data appear not to exist anymore and/or they can not be tracked. Despite being a large dataset, several factors prevent them to be used for training. First, most of T2001 and T2002 should be removed due to their poor annotations. In addition, at least T2007 and the rest of T2001 should be removed as they are commonly used for evaluation [12], [13], [22]. This leaves at most 15,163 sharp and 7,274 gradual annotations from TRECVID and Baraldi et al. [27]. Experiments show this is not sufficient to train an accurate SBD CNN.

C. Spatio-temporal analysis using CNN

Our solution analyzes both spatial and temporal information through CNN. Hence, our network is related to the literature on video classification. Karpathy et al. [34] proposed multiple approaches for extending the connectivity of CNN to take advantage of the spatio-temporal information. Results show that CNN can generate strong improvement over hand-crafted features. However, the multiple frame models showed a modest improvement compared to the single-frame model. Next, Simonyan et al. [35] proposed a two stream CNN network for video classification. One network analyzes the spatial information while the second analyzes the optical flow field. Their approach generates significant improvement over the single frame model of [34].

Tran et al. [26] presented the first single stream CNN that incorporate both spatial and temporal information at once. Their approach takes multiple frames as input and examines them with 3D spatio-temporal convolutional filters. They handle the problem of activity recognition and performed evaluation on the UCF101 dataset. They outperformed all previous work, including [34], [35]. In addition, their technique is fast as does not require optical flow estimation.

Our solution is a full Shot Boundary Detection (SBD) system consisting of a CNN-based classification step, a merging step and a post-processing step. At the core of our CNN-classification is a spatio-temporal architecture inspired by Tran et al. [26]. Unlike Tran et al. [26], however, our architecture uses batch normalization. Furthermore, our solution contains a component for generating very large well annotated datasets for training our SBD. Results show that all components of our solution, including dataset generation and our full SBD system, play an important role in outperforming the state of the art, both in detection accuracy and processing speed.
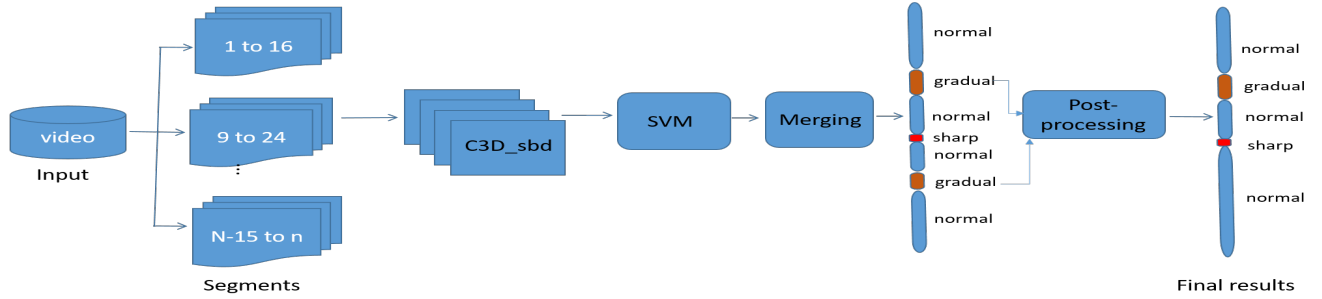
Fig. 3. Overview of our Shot Boundary Detection (SBD) system. A video is divided into segments of 16 frames with an overlap of 8. Each segment is fed to a 3D CNN (see Table I). The output of fc8 is fed to an SVM and labels are assigned. Consecutive segments with the same labeling are merged. Finally, false alarms of gradual transitions are reduced through a histogram-driven temporal differencing.
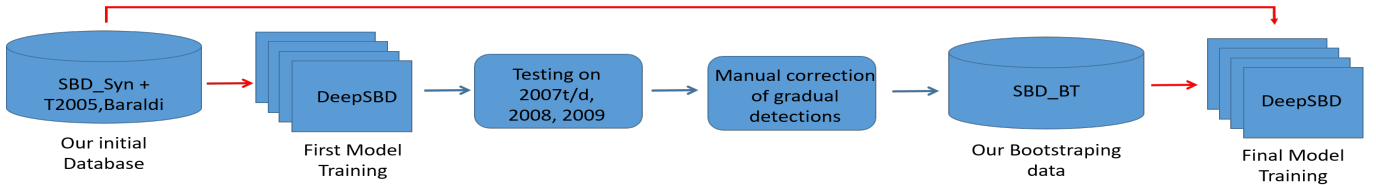

Fig. 4. Our process of data generation. Here, red arrows refer to data fed to CNN training. We first use our synthetic data SBD_Syn and some real data (T2005 and Baraldi et al.) to train our solution. The model runs on 2007t/d, 2008 and 2009. We manually correct gradual detections into sharp, gradual and no transitions. This generates our bootstrapping data SBD_BT. We train the final solution using both SBD_Syn and SBD_BT, and optionally T2005 and Baraldi et al. .

III. OUR APPROACH

A. Algorithm Design

We present a technique for automatic detection and classification of shot boundaries. We name our technique DeepSBD. A video is divided into segment of frames. Each segment is assigned one of three labels: 1) sharp transition, 2) gradual transition or 3) no transition. We use segments of length 16, with an overlap of 8. Each segment is fed to a deep 3D-CNN that analysis both spatial and temporal information. Our network, C3D_sbd, is inspired by [26] and is trained from scratch for shot boundary detection. The last feature layer is fed to an SVM classifier. This gives the first labeling estimate. Consecutive segments with the same labeling are merged and the result is passed to a post-processing step. The step reduces false positives with little motion. For such segments, we estimate the color histogram of the first and end frame. We measure the Bhattacharyya distance between these histograms. If the distance is small, we declare this segment as no-transition. We use an OpenCV implementation for both color histogram and Bhattacharyya distance, which is very fast.

Figure 3 shows an overview of our detection system. Our network, C3D_sbd, consists of five 3D convolutional layers (see Table I). All convolutional layers are followed by Rectified Linear Unit (ReLU) and pooling layers. The first two convolutional layers are followed by Local Response Normalization (LRN). Two fully connected layers exist, fc6 and fc7, each containing 2049 neurons. The last fully connected layer fc8 contain only 3 neurons, one for each class (sharp, gradual and no transition). In comparison to [26], C3D_sbd uses batch normalization after the first two convolutional layers.

Layer	Kernel (t×y×x×c)×f	Feature map dimension	Followed by
Data	-	20×3×16×112×112	
Conv1	(3×3×3×3)×96	20×96×14×55×55	ReLU LRN
Pool1	-	20×96×12×27×27	
Conv2	(3×3×3×96)×256	20×256×12×29×29	ReLU LRN
Pool2	-	20×256×10×14×14	
Conv3	(3×3×3×256)×384	20×384×10×14×14	ReLU
Conv4	(3×3×3×384)×384	20×384×10×14×14	ReLU
Conv5	(3×3×3×384)×256	20×256×10×14×14	ReLU
Pool5	-	20×256×8×7×7	
Fc6	(8×7×7×256)×2048	20×2048×1×1×1	ReLU Drop 0.5
Fc7	2048×2048	20×2048×1×1×1	ReLU Drop 0.5
Fc8	2048×3	20×3×1×1×1	
Softmax	Label	-	

TABLE I
THE MODEL PARAMETERS OF C3D_sbd. IN COMPARISON TO [26], C3D_sbd USES BATCH NORMALIZATION.

B. Dataset Generation

Training an SBD CNN requires a large and well-annotated dataset. We present two datasets, SBD_Syn (Table II) and SBD_BT (Table III). SBD_Syn is generated synthetically while SBD_BT is generated in a way to improve detector's precision, through bootstrapping. Figure 4 shows the process of generating both datasets. We first use SBD_Syn with T2005 and Baraldi et al. to train from scratch our solution. We run this solution on data from T2007t/d, T2008 and T2009. Due to the massive size of these datasets, however, we only examine segments originally annotated as any form of transition. Note

that original annotations here do not distinguish between sharp or gradual. We closely examine segments detected as graduals. We manually filter them into three classes: gradual, sharp and no transitions. The no-transition represent complicated hard-negative cases such as illumination variation and fast motion (see Figure 5). Finally, we train from scratch a final solution using both SBD_Syn and SBD_BT. We optionally use T2005 and Baraldi et al. to further improve performance. Results show that SBD_BT has a great impact in reducing false detections and improving the overall performance. The supplementary material shows images from the datasets of SBD_Syn and SBD_BT in Figure 1 and Figure 2.

SBD_Syn: Table II shows the content of SBD_Syn. Images from this dataset is shown in the supplementary material (Figure 1). The dataset is generated synthetically through image compositing models [36]. A transition is modeled as a linear combination between the underlying shots

$$I_t(\mathbf{x}) = \alpha_t(\mathbf{x})B_t(\mathbf{x}) + (1 - \alpha_t(\mathbf{x}))F_t(\mathbf{x}) \qquad (1)$$

Here, I_t denotes the observed frame at time t, while B and F are the content from the previous and next shots respectively. α is the mixing parameter between both shots while $\mathbf{x}$ denotes image pixels. The values and distribution of α define the type of shot transition. If no transition exist, then $(\alpha_t, \alpha_{t+1}) = (1, 1)$. A sharp transition, however, have a sudden temporal change with $(\alpha_t, \alpha_{t+1}) = (1, 0)$. For gradual transitions, α changes over time from 1 to 0. This change occurs over a set of frames and hence $(\alpha_t, ...\alpha_t + N) = (1, ., ., 1 - t/N, ., 0)$. N is the transition duration and t is the frame index where $t = 0$ denotes the last frame of the previous shot. Here, the in-between α values are non-binary. This generates the dissolve nature of most gradual transitions (Figure 1). For wipes, α is spatially-varying aswell as temporally-varying.

To generate SBD_Syn we need to define F, B and α in Eq. 1. F and B must not contain any shot transitions. We use the T2007t/d, T2008, T2009 and their annotations to find such frames. We sample F and B in a way to ensure a large offset from the nearest transition. Sharp transitions are generated by applying Eq. 1 with $(\alpha_t, \alpha_{t+1}) = (1, 0)$. Gradual transition generation, however, is more complex. For SBD_Syn we focus on dissolve gradual generation. We randomly select the transition duration N, where $N <= 16$. We also randomly select the transition start and end frames for both B and F. We draw N α samples, where α is modeled with a uniform distribution. We sort all α values in descending order and apply Eq. 1 for each of the N frames.

We train C3D_sbd using balanced data for sharp, gradual and no-transition. We experimented with different data sizes. We found 40,000 segments for each class generate good results. We also train the SVM for sharp and gradual using 110,000 segments for each. For CNN, we use a step learning policy. Learning rate starts with a value of 0.0001 and is reduced gradually by a factor of 10 every two epochs. We use a batch size of 20, and train the model for 6 epochs. That is two epochs for each learning rate of 1e-4, 1e-5, and 1e-6. The momentum value is 0.9. All these values were set empirically to optimize performance. We also found empirically that SVM works better with features from fc8 as opposed to fc6 and fc7.

Datasets	Synthetic Gradual	Synthetic Real
T2007t	19398	19439
T2007d	13656	19607
T2008	39047	32456
T2009	44158	32578
Total	116259	104080

TABLE II

OUR DATASET SBD_SYN IN TERMS OF NUMBER OF SEGMENTS (16 FRAMES EACH). SBD_SYN IS SYNTHETICALLY GENERATED FROM T2007T/D,2008,2009 USING IMAGE COMPOSITING. THE DATA CONTAINS A BALANCED PORTION OF NO-TRANSITIONS.

Transitions	Number of segments
No-transition	4,427
Gradual	11,249
Sharp	359
Total	16,035

TABLE III

OUR SBD_BT DATASET. THE NO-TRANSITION REPRESENT COMPLEX HARD-NEGATIVES (FIGURE 5). WHEN INCLUDED IN TRAINING, PRECISION IS SIGNIFICANTLY IS IMPROVED.

IV. RESULTS

We performed experiments on real data as well as on synthetically generated data. We examined 4,683,552 frames, 81.8% of which are real. Our work is the largest SBD evaluation to date for one algorithm. We asses performance quantitatively using precision (P), recall (R) and F-score (F). Here, we use the standard TRECVID evaluation metric [1] where a transition is detected if it overlaps with the annotations by at least one frame. We report the per-transition performances. During comparison we highlight the best performing technique in **bold**. To account for possible mis-annotations and system error in such large experiment, we claim a technique is superior only if it achieves more then 0.5% P, R, or F improvement over the second best performing technique. Techniques with 0.5% difference are claimed as competitive. We train two models, both using our datasets DSB_Syn and SBD_BT. One of them uses few real data from T2005 and Barladi et al. , denoted by r, at most 7% of the total training data. Both models are competitive to each other. We report results with r in the paper and report the other model in the supplementary material.

We compare against the latest techniques (Lu et al. [13], Priya et al. [12], Apostolidis et al. [28] and Berladi et al. [29]) as well as the best performers in the 7 years of the TRECVID challenge (Yuan et al. [21] and Liu et al. [22]). These tech-
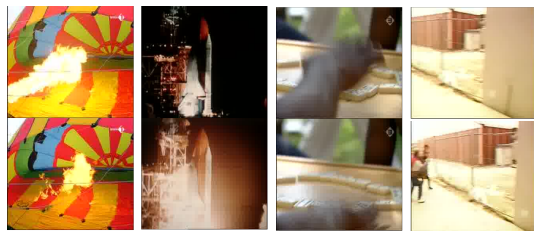

Fig. 5. No-transition samples from our bootstrapping data (SBD_BT). They contain complex cases such as fast motion, occlusion and illumination variation. These cases can be misclassified as graduals. More examples are in the supplementary material, Figure 2.

	videos names
T2001a	BOR10_001, BOR10_002, NAD57, NAD58, anni001, anni005, anni006, anni007, anni00, anni009, anni010
T2001b	BOR03, BOR08, BOR10, BOR12, BOR17

TABLE IV
VIDEOS OF T2001A AND T2001B

niques show the compromise between detection accuracy and processing speed commonly present in SBD. Lu et al. [13] is the fastest of all, but generates conservative performance. Priya et al. [12], Liu et al. [22] and Yuan et al. [21] generate better performance. However, at the cost of heavy optical flow computation. Our results show that DeepSBD optimizes both detection accuracy and processing speed over all current techniques. That is, we outperform gradual detection, generate competitive performance for sharp transitions and produce significant improvement in wipes detection. In addition, we are up to 11 times faster than the state of the art. More detailed results are reported in the supplementary material.

A. Real Sequences

We evaluated our technique on all seven TRECVID releases, from 2001 to 2007. They have a total of 3,831,648 frames, with 8,545 gradual and 14,602 sharp transitions. No test data was included in the training. Table V shows performance evaluation on 6 sequences commonly used in Lu et al. [13] and Priya et al. [12]. The sequences are from T2001a (see Table IV) and present challenging videos from outer-space. The videos include cases such as global illumination variation, smoke, fire and fast non-rigid motion. We outperform Lu et al. in all sequences for both gradual and sharp transition. Furthermore, we outperform Priya et al. in the vast majority of sequences in both transition types.

Table VI compares our technique against Priya et al. [12] on T2007. Note that [12] used a slightly different approach for evaluation than the one recommended by TRECVID [1]. TRECVID recommends estimating the average performance per transition. However, [12] estimated the average performance per sequence. Furthermore, Priya et al. tested on 17 sequences, 7 of which were included in their training set. This biases the results towards Priya et al. [12]. Hence, for fair comparison these 7 sequences should be removed from the 17 test sequences and the comparison should be done on at most 10 sequences. To illustrate this point, we examined our technique with different sizes of the test dataset. Each column of Table VI shows the performance with different size of the test data. With 10 test sequences, our technique outperforms Priya et al. [16] significantly in gradual transitions (0.88 vs. 0.76 f-score) and generates competitive results for sharp transitions. Furthermore, we still outperform Priya et al. even with a test-set of 14 sequences. Here, however, at least 4 sequences are included in Priya et al. training and hence results are biased towards Priya et al. Including these videos in our training is expected to improve performance even further. The spatio-temporal aspect of our solution allow us to generate these high detection accuracy results without

explicitly estimating optical flow. Our experimental results showed that just relying on the spatial information generates very poor performance.

Table VII evaluates DeepSBD on T2004, 2005, 2006 and 2007. To test on 2005, we removed it from our training. We compare against the best TRECVID performers as well as Lu et al. [13]. We significantly outperform Lu et al. in T2007. Furthermore, we outperform the best TRECVID performers, Liu et al. [22] and Yuan et al. [21] on all four datasets. Table VIII evaluates DeepSBD on the remaining TRECVID datasets. T2001b and 2002 annotations contain significant overlap between sharp and gradual transitions. Hence, for them we show the overall combined transitions performance. Furthermore, T2003 is missing 4 videos and hence we could not compare against the reported TRECVID performance. In all sequences we generate good performance. T2001b and 2002 sequences contain strong noise and jitter. Yet, our technique was robust enough to handle such artifacts. Figure 6 (a) shows the precision-recall curves for our DeepSBD on all real TRECVID sequences. Table XVI shows the combined f-score for the RAI dataset [37]. Here, we compare against the techniques of Apostolidis et al. [28] and Berladi et al. [29]. Results show that we significantly outperform both techniques. The supplementary material (Table II-XVI) shows the per sequence results for each of the TRECVID and RAI dataset examined by our technique. This includes much more statistics e.g. true positives (TP), false positives (FP), false negative (FN) and so on.

Table X examines different test configurations for DeepSBD. SVM on fc8 generates better results than on fc6. The post-processing (pp) step improves the performance, especially for T2007. The best performance is obtained with fc8+svm+pp. Figure 7 shows failure cases in gradual transition detection. Too long transitions can get misclassified as False negatives (FN). Here, no enough temporal difference is captured over our 16 frames window. FN can also be generated when both shots have similar texture and color. False positives are largely generated by computer graphics content. Such content have a gradual-like effect. However, they are not semantically classified as a shot transition.

B. The importance of our datasets

Table I shows the significance and importance of our datasets SBD_Syn and SBD_BT in generating high accuracy detections. We evaluate DeepSBD on T2007 with six different training sets: 1) R_3-5 2) R_3-6 3) R_3-6 + BT, 4) S + r, 5) S + r + BT and 6) and S + BT. S and BT is short for our datasets SBD_Syn and SBD_BT. R_3-6 represent TRECVID real videos and annotations from 2003 to 2006. r is T2005 and Baraldi [27]. Results show that training with R_3-5 generate poor performance. In addition, it limits us to testing on just 3 data-sets (T2001a, T2006 and T2007). Adding T2006 to training improves performance but limits our testing further to 2 data-sets (T2001a and T2007). Adding our bootstrapping data SBD_BT (BT) improves precision and performance significantly. This shows the high quality and importance of our SBD_BT. The best performance, however, is

	D1-anni5	D2-anni6	D3-anni9	D4-anni10	D5-NAD57	D6-NAD58
Lu et al. [13]						
Abrupt	-	0.905	0.754	0.892	-	0.962
Gradual	-	0.817	0.824	0.734	-	0.884
Priya et al. [12]						
Abrupt	**0.85**	0.911	0.842	0.897	0.945	**0.945**
Gradual	0.938	0.885	0.873	0.822	0.809	0.885
DeepSBD (ours)						
Abrupt	0.818	**0.988**	**0.961**	**0.918**	**0.957**	0.904
Gradual	**0.945**	0.885	**0.919**	**0.855**	**0.917**	**0.914**

TABLE V

DEEPSBD EVALUATION ON 6 CHALLENGING SEQUENCES FROM TRECVID 2001 (D1-D6). OUR TECHNIQUE OUTPERFORMS LU ET AL. AND PRIYA ET AL. [12] IN THE VAST MAJORITY OF SEQUENCES. THE IMPROVEMENT IS MORE SIGNIFICANT IN GRADUAL TRANSITIONS.

Size of test-data (in sequences)	9	10	11	12	13	14	15	16	17
Priya et al. [12]									
Abrupt	0.9733	0.974	0.9748	0.9742	0.9737	0.9741	0.9733	0.9737	0.974
Gradual	0.775	0.7578	0.7677	0.7742	0.7811	0.7825	0.7802	**0.7726**	**0.78**
DeepSBD (ours)									
Abrupt	0.9729	0.9749	0.9743	0.974	0.9743	0.9749	0.9733	0.9713	0.9726
Gradual	**0.8962**	**0.8774**	**0.8613**	**0.8395**	**0.8171**	**0.797**	0.7758	0.7507	0.7259

TABLE VI

PER-SEQUENCE F-SCORE ON T2007. WE COMPARE AGAINST PRIYA ET AL. [12] ON TEST-SETS OF DIFFERENT SIZES. SINCE PRIYA ET AL. [12] IS TRAINED ON 7 OUT OF THE TOTAL 17 TEST SEQUENCES, COMPARISON SHOULD BE DONE ON AT MOST 10 SEQUENCES. RESULTS SHOW WE SIGNIFICANTLY OUTPERFORM PRIYA ET AL. WITH A TEST-SET OF 10 SEQUENCES. HERE, OUR GRADUAL TRANSITIONS DETECTOR IS MORE THAN 12% BETTER THAN PRIYA ET AL. IN F-SCORE, A SIGNIFICANT IMPROVEMENT DUE TO OUR CNN SOLUTION. FURTHERMORE, WE STILL OUTPERFORM PRIYA ET AL. WITH A TEST-SET SIZE UP TO 14 SEQUENCES. HERE, HOWEVER, AT LEAST 4 SEQUENCES WERE INCLUDED IN PRIYA ET AL. WHICH BIASES THE RESULTS TOWARDS PRIYA ET AL. INCLUDING THESE VIDEOS IN OUR TRAINING IS EXPECTED TO BOOST OUR PERFORMANCE EVEN FURTHER.

(a) All TRECVID videos   (b) UCF_SBD
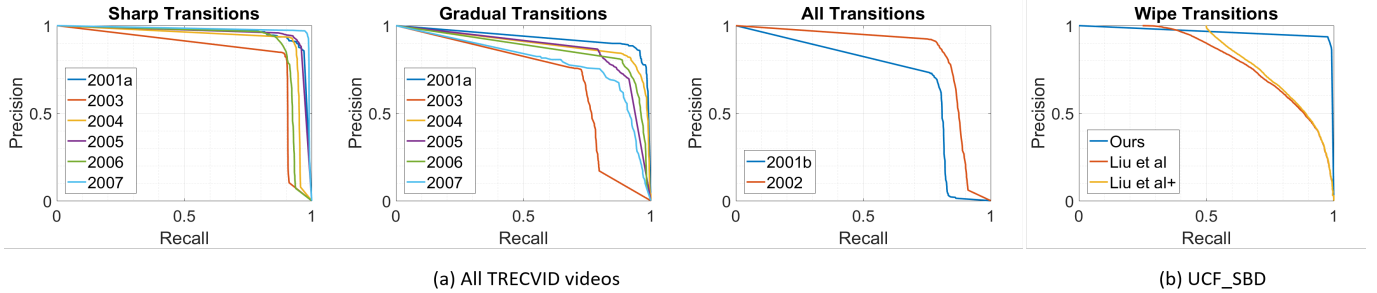
Fig. 6. (a) Precision-Recall of DeepSBD for all TRECVID sequences. (b) Precision-Recall of DeepSBD for wipes. Here, we compare against two implementations of Liu et al. [22] wipe detector. The first implementation examines all frames of UCF101_SBD and '+' examines only frames not classified as sharp nor gradual by DeepSBD. We significantly outperforms both approaches.

	T2004	T2005	T2006	T2007
Best TRECVID performers [1]				
Abrupt	0.929	0.935	0.899	0.972
Gradual	0.806	0.786	0.814	0.753
Lu et al. [13]				
Abrupt	-	-	-	0.761
Gradual	-	-	-	0.618
DeepSBD (ours)				
Abrupt	0.926	0.934	0.895	0.971
Gradual	**0.866**	**0.844**	**0.827**	**0.776**

TABLE VII

COMPARING DEEPSBD AGAINST DIFFERENT TECHNIQUES. WE COMPARE AGAINST TRECVID BEST PERFORMERS, LIU ET AL. [22] FOR 2006/2007 AND AGAINST YUAN ET AL. [32] FOR 2004/2005. WE ALSO COMPARE AGAINST THE LATEST NO OPTICAL FLOW TECHNIQUE OF LU ET AL. [13]. WE OUTPERFORM ALL TECHNIQUES ON ALL DATASETS.

	T2001a	T2001b	T2002	T2003
DeepSBD				
Abrupt	0.931	-	-	0.866
Gradual	0.904	-	-	0.759
Overall	0.918	0.748	0.865	0.8337

TABLE VIII

EVALUATING DEEPSBD ON T2001A, 2001B, 2002 AND 2003.

	Method in [28]	Method in [29]	Ours
RAI [37]	0.84	0.84	**0.94**

TABLE IX

PROCESSING THE RAI DATASET [37] WITH DIFFERENT TECHNIQUES. HERE WE SHOW THE COMBINED F-SCORE OF THE OVERALL DETECTION. OUR TECHNIQUE SIGNIFICANTLY OUTPERFORMS BOTH APOSTOLIDIS ET AL. [28] AND BERLADI ET AL. [29].

generated when both our datasets SBD_Syn and SBD_BT with r are used for training. In addition to the highest performance,

	fc6+svm	fc8+svm	fc8+svm+pp	fc8+pp
TR2001a				
Gradual	0.882	**0.906**	**0.904**	**0.906**
Sharp	**0.931**	**0.931**	0.931	0.923
TR2006				
Gradual	0.83	**0.841**	0.844	0.843
Sharp	0.887	**0.895**	0.895	0.895
TR2007				
Gradual	0.71	0.732	**0.776**	**0.776**
Sharp	0.955	0.968	**0.971**	**0.973**

TABLE X

COMPARING DEEPSBD WITH DIFFERENT SETTINGS. THE BEST PERFORMANCE ON ALL ALL DATASETS IS OBTAINED WITH fc8 FEATURES + SVM + POST-PROCESSING (PP). SOME OTHER SETTINGS ARE COMPETITIVE (SEE BOLD).

Fig. 7. Failure cases in gradual transition detection.

this option allow us to test on all TRECVID videos, except T2005. Removing r from the training generates the second best performance. This, however, allow us to test on all TRECVID videos, including T2005. The experiment shows the significance and importance of our data-sets. We performed this experiment on several test sets and we found S + r + BT and S + BT are always the top and competitive to each other (see supplementary material, Table. 1). This shows the significance of our datasets and their generation process (Section III-B).

C. Controlled Experiments

We generated a synthetic test-set. Our dataset contain 53,324 segments, divided equally between gradual, sharp, wipes and no-transitions. Each segment is 16 frames long. We generated gradual and sharp transition using image compositing as we

	P	R	F	P	R	F
R_3-5	0.495	0.665	0.568	0.894	0.872	0.883
R_3-6	0.683	0.683	0.683	0.957	0.95	0.953
R_3-6 + BT	0.755	0.705	0.729	0.961	0.961	0.961
S + r	0.722	0.63	0.673	0.979	0.955	**0.967**
S + r + BT	**0.799**	**0.753**	**0.776**	**0.973**	**0.969**	**0.971**
S + BT	0.779	0.714	0.745	0.969	0.966	**0.968**

TABLE XI

TRAINING DEEPSBD WITH DIFFERENT DATASETS. RESULTS SHOW THAT BEST PERFORMANCE IS GENERATED WHEN OUR BOTH SBD_SYN (S) AND SBD_BT (BT) AND r ARE USED. REMOVING ANY REAL SEQUENCES (r) FROM OUR DATASETS GENERATES THE SECOND BEST PERFORMANCE (S+BT). THE ADVANTAGE OF THIS OPTION IS ALLOWING US TO TEST ON ALL TRECVID VIDEOS, INCLUDING T2005 (TABLE VII). THE TABLE ALSO SHOWS SBD_BT CLEARLY IMPROVES THE PRECISION AND OVERALL PERFORMANCE.
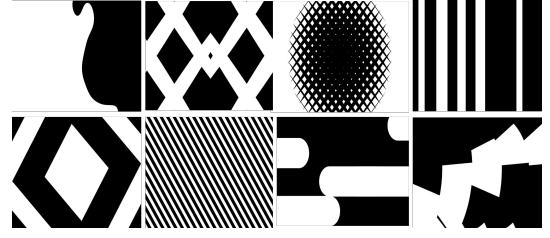
Fig. 8. Samples of the α mattes used for generating the wipes of UCF101_SBD. Examples of the generated wipes are shown in the supplementary material in Figure 3.

Transition Type	Precision	Recall	F-measure
Gradual	0.989	0.995	0.992
Sharp	0.984	0.999	0.992
Wipes	0.976	0.936	0.956

TABLE XII

EVALUATING DEEPSBD ON OUR SYNTHETIC DATASET UCF101_SBD. IN TOTAL UCF101_SBD HAS 53, 253 SEGMENTS, DIVIDED EQUALLY AMONG ALL CLASSES (NORMAL, GRADUAL, SHARP AND WIPES). EACH SEGMENT HAS 16 FRAMES. DEEPSBD GENERATES A VERY HIGH PERFORMANCE IN ALL CLASSES.

did for SDB_Syn (see Eq. 1). Here, we constrain the shots to come from two different UCF101 videos [38]. We present the first large wipes dataset, containing 1.1 million wipe frames (20% test). They are also generated using Eq. 1. Here, however, the opacity values α have more complicated spatio-temporal patterns than sharp and gradual transitions. Figure 8 shows some of the 196 α mattes we used. The supplementary material, Figure 3, show frames from our wipes dataset. We call our synthetic UCF dataset UCF101_SBD.

We train the model using SBD_Syn, SBD_BT and the synthetic wipes. This model generates 4 classes. Table XII evaluates DeepSBD on UCF101_SBD. We generate high performance for all classes, including wipes. Performance is higher than the ones previously reported on the TRECVID sequences. This could be due to the highly accurate annotations of UCF101_SBD. Figure 6 (b) compares our wipe detector against the state of the art of Liu et al. [22]. We evaluate Liu et al. using two strategies. The first examines all frames of UCF101_SBD. The second, '+', examines only frames not detected as gradual nor sharp transitions by DeepSBD. Our technique outperform both approaches significantly.

D. Processing Speed

We examined a TRECVID video of duration 4,096 seconds containing 102,400 frames. We ran the test-phase of DeepSBD with different batch sizes as input. The GPU performs n

	Real-time speed-up factor
DeepSBD	19.3
Liu et al. [1], [22]	3.24
Priya et al. [12]	1.76
Yuan et al. [32]	2.43

TABLE XIII

REAL-TIME SPEED-UP FACTOR FOR DIFFERENT SHOT BOUNDARY DETECTION TECHNIQUES. WE ARE FASTER THAN ALL TECHNIQUES WITH A FACTOR UP TO 11.

iterations until all 102,400 frames are processed. The smaller the batch size, the more iterations required and hence the more time required to process all frames. However, the less memory required. Experiments shows that the processing speed gain from 10 to 100 batch size is not significant. That is between 16-19.3 real-time speed up factor. We use Titan X, a GPU commonly used for deep learning applications. Table XIII compares the processing speed of different SBD techniques. In comparison with the best performing optical-flow based techniques, we are 11 times faster than Priya et al. [12], 6 times faster than Liu et al. [22] and 9.65 times faster than Yuan et al. [32]. The supplementary material shows more analysis of the processing speed in Figure 4-5 (Section II).

E. Deep Analysis on Network Responses

We randomly selected two segments (16 frames) from UCF101 and synthetically generated a sharp and gradual transition using Eq. 1. We treated one of the two sequences as no-transition. We examined all segments using DeepSBD. Figure 9 shows the heat map of some Conv5 filter responses for each transition type. The filters are stacked next to each other, in blocks. The green grid shows filters' borders. Time is the y-axis and space is the x-axis. Vertical space is averaged over the horizontal space. Sharp transitions have abrupt responses in the time axis in form of bright horizontal lines. Gradual transitions have blurred responses in the time axis. No transitions do not show a specific response pattern. The patterns are consistent on several other segments. The supplementary material shows more of such results in Figure 6 (Section III).

V. CONCLUSION

We presented the first CNN technique for shot boundary detection. Current techniques compromise between detection accuracy and processing speed and use hand-crafted features. We exploit big data to optimize both accuracy and speed. This is important as SBD is a common pre-processing step for video manipulation. We present two large datasets containing 3.57 million frames. One set is generated synthetically while the other is carefully annotated through bootstrapping. We outperform state of the art gradual transition detections, generate competitive performance in sharp transitions and produce significant improvement in wipes detections. Our approach is up to 11 times faster than the state of the art. Future work can examine computer graphics content more closely. We will release our datasets and code to encourage future research.

REFERENCES

[1] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Computer Vision and Image Understanding (CVIU)*, vol. 114, no. 4, pp. 411–418, 2010.

[2] J. Fan, D. K. Y. Yau, W. G. Aref, and A. Rezgui, "Adaptive motion-compensated video coding scheme towards content-based bit rate allocation," *Journal of Electronic Imaging*, vol. 9, no. 4, pp. 521–533, 2000.

[3] Z. Wang, D. Liu, S. Chang, Q. Ling, and Y. Yang, "D3: Deep dual-domain based fast restoration of jpeg-compressed images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2764–2772.

[4] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016, pp. 1646–1654.
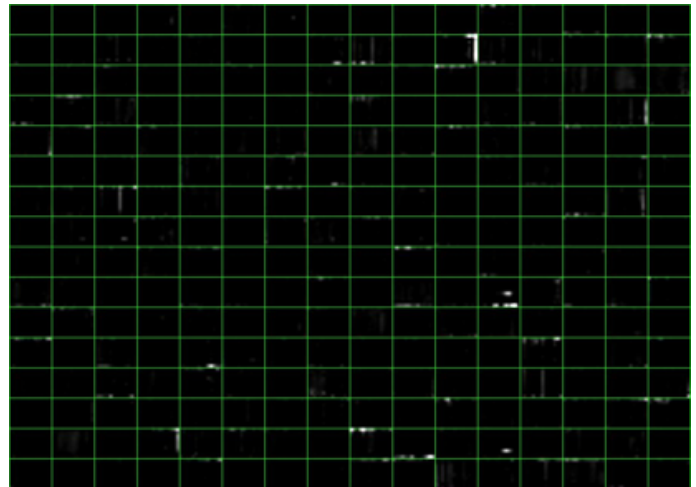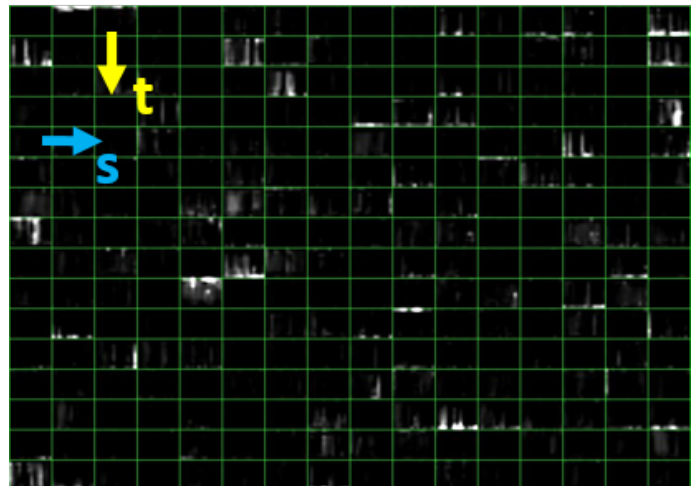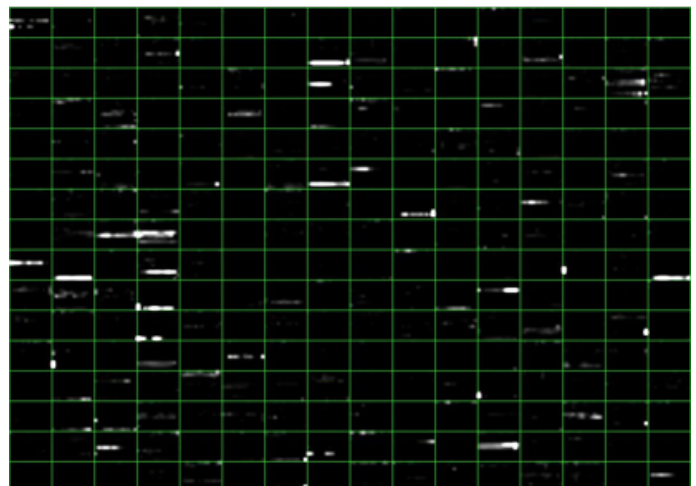
Fig. 9. Filter responses of DeepSBD stacked next to each other. The green grid shows filters' borders. Here, y-axis is time (see blue arrow) and x-axis is space (see yellow arrow). Sharp transitions have an abrupt response in time (bright horizontal lines). Gradual transitions have blurred responses in time. No transition do not show specific patterns. More examples are shown in the supplementary material in Figure 6.

[5] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in *IEEE Internataional Conference on Computer Vision (ICCV)*, 2016, pp. 842–857.

[6] K. Calagari, M. Elgharib, P. Didyk, A. Kaspar, W. Matusik, and M. Hefeeda, "Gradient-based 2d-to-3d conversion for soccer videos," in *ACM Multimedia*, 2015, pp. 331–340.

[7] S. Bae, M. A. Elgharib, M. Hefeeda, and W. Matusik, "Efficient and scalable view generation from a single image using fully convolutional networks," *CoRR*, vol. abs/1705.03737, 2017. [Online]. Available: http://arxiv.org/abs/1705.03737

[8] K. Zahng, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *CVPR*, 2016, pp. 766–782.

[9] ——, "Summary transfer: Exemplar-based subset selection for video summarization." in *CVPR*, 2016, pp. 1059–1067.

[10] Y. Song, M. Redi, J. Vallmitjana, and A. Jaimes, "To click or not to click: Automatic selection of beautiful thumbnails from videos," in *ACM International Conference on Information and Knowledge Management (CIKM)*, 2016, pp. 659–668.

[11] K. Templin, P. Didyk, K. Myszkowski, M. M. Hefeeda, H.-P. Seidel, and W. Matusik, "Modeling and optimizing eye vergence response to stereoscopic cuts," *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, vol. 33, no. 4, 2014.

[12] L. Priya and D. S., "Walsh hadamard transform kernel-based feature vector for shot boundary detection," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 12, pp. 5187–5197, 2014.

[13] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on svd and pattern matching," *TIP*, vol. 22, no. 12, pp. 5136–5145, 2013.

[14] P. P. Mohanta, S. K. Saha, and B. Chanda, "A model-based shot boundary detection technique using frame transition parameters," *IEEE Transactions on Multimedia (TMM)*, vol. 14, no. 1, pp. 223–233, 2012.

[15] D. Adjeroh, M. C. Lee, N. Banda, and U. Kandaswamy, "Adaptive edge-oriented shot boundary detection," *EURASIP Journal on Image and Video Processing*, vol. 2009, no. 1, 2009.

[16] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 16, no. 1, pp. 82–91, 2006.

[17] J. Lankinen and J.-K. Kämäräinen, "Video shot boundary detection using visual bag-of-words," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2013.

[18] D. Lelescu and D. Schonfeld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 106–117, 2003.

[19] C. Zhang and W. Wang, "A robust and efficient shot boundary detection approach based on fisher criterion," in *ACM Multimedia*, 2012, pp. 701–704.

[20] D. M. Thounaojam, T. Khelchandra, K. M. Singh, and S. Roy, "A genetic algorithm and fuzzy logic approach for video shot boundary detection," *Computaional intelligence and neuroscience*, vol. 2016, 2016.

[21] J. Yuan, W. Zheng, L. Ding, D. Wang, Z. Tong, H. Wang, J. L. J. Wu, F. Lin, and B. Zhang, "Tsinghua university at trecvid 2004: Shot boundary detection and high-level feature extraction," in *TRECVID Workshop*, 2004.

[22] Z. Liu, E. Zavesky, D. Gibbson, B. Shahraray, and P. Haffner, "At&t research at trecvid 2007," in *TRECVID Workshop*, 2007.

[23] M. W. Tao, J. Bai, P. Kohli, and S. Paris, "Simpleflow: A non-iterative, sublinear optical flow algorithm," *Computer Graphics Forum (Eurographics)*, vol. 31, no. 2, 2012.

[24] A. Dosovitskiy, P. Fischery, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766.

[25] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision (IJCV)*, vol. 92, no. 1, pp. 1–31, 2011.

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

[27] L. Baraldi, C. Grana, and R. Cucchiara, "A deep siamese network for scene detection in broadcast videos," in *ACM Multimedia*, 2015, pp. 1199–1202.

[28] E. Apostolidis and V. Mezaris, "Fast shot segmentation combining global and local visual descriptors," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6583–6587.

[29] L. Baraldi, C. Grana, and R. Cucchiara, "Shot and scene detection via hierarchical clustering for re-using broadcast video," in *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2015, pp. 1–11.

[30] S. Lian, "Automatic video temporal segmentation based on multiple features," *Soft Computing*, vol. 15, no. 3, pp. 469–482, 2011.

[31] Y. Kawai, H. Sumiyoshi, and N. Yagi, "Shot boundary detection at trecvid 2007," in *TRECVID Workshop*, 2007.

[32] J. Yuan, H. Wang, L. Xiao, D. Wang, D. Ding, Y. Zuo, Z. Tong, X. Liu, S. Xu, W. Zheng, X. Li, Z. Si, J. Li, F. Lin, and B. Zhang, "Tsinghua university at trecvid 2005," in *TRECVID Workshop*, 2005.

[33] N. I. of Standards and Technology, "http://trecvid.nist.gov/trecvid.data.html," https://www.nist.gov/, 2017.

[34] 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.

[35] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[36] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, pp. 1647–1654, 2007.

[37] R. T. Network, "The rai scuola video archives," http://www.scuola.rai.it/, 2015.

[38] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1212.html#abs-1212-0402

Supplementary Material: Large-scale, Fast and Accurate Shot Boundary Detection through Spatio-temporal Convolutional Neural Networks

Ahmed Hassanien[1], Mohamed Elgharib[1], Ahmed Selim[2], Sung-Ho Bae[3], Mohamed Hefeeda[4], and Wojciech Matusik[3]

[1]Qatar Computing Research Institute, HBKU
[2]Trinity College Dublin, CONNECT Center
[3]MIT CSAIL
[4]Simon Fraser University

I. Our data-sets

Figure 1 shows samples from the gradual transitions class of our dataset (SBD_Syn). Our data is synthetically generated through image compositing. It is diverse, containing a wide variety of colors, texture, objects, motion and so on. Figure 2 shows hard negative samples from our bootstrapping data (SBD_BT). The samples contain challenging cases that commonly confuse gradual transition detectors e.g. fast motion, fast zoom in, illumination changes, object occlusion, strong lighting, and so on. Figure 3 shows 10 sequences from our synthetically generated wipes dataset. The sequences show some variety of the alpha mattes used to generate wipes.

Tab. I shows the significance and importance of our synthetic SBD_Syn and bootstrapping SBD_BT datasets in generating high accuracy detections. We evaluate our technique, DeepSBD, on different datasets with six different training sets: 1) R_3-5 2) R_3-6 3) R_3-6 + BT, 4) S + r, 5) S + r + BT and 6) and S + BT. S and BT is short for our datasets SBD_Syn and SBD_BT. R_3-6 represent TRECVID real videos and annotations from 2003 to 2006. r is T2005 and Baraldi. Results show that training with R_3-5 generate poor performance. In addition, it limits us to testing on just 3 data-sets (T2001a, T2006 and T2007). Adding T2006 to training improves performance but limits our testing further to 2 data-sets (T2001a and T2007). Adding our bootstrapping data SBD_BT (BT) improves precision and performance significantly. This shows the high quality and importance of our SBD_BT. The best performance, however, is generated when both our datasets SBD_Syn and SBD_BT with r are used for training. In addition to the highest performance, this option allow us to test on all TRECVID videos, except T2005. Removing r from the training generates a competitive performance. This, however, allow us to test on all TRECVID videos, including T2005. The experiment shows the significance and importance of our data-sets. We performed this experiment on several test sets and we found S + r + BT and S + BT are always the top and competitive to each other. This shows the significance of our datasets.

Tab. II-XVI shows detailed per video results for different testing sets. For each testing dataset, we report the results using two different training-sets (S+r+BT and S+BT). We show: the number of transitions (#T), true positives (TP), false positives (FP), false negatives (FN), precision (P), recall (R) and F-measure (F).

II. Processing speed

Figure 4-5 examines the processing speed (test-phase) of our technique with different batch sizes as input. We ran our model on 6,394 segments. Each segment is 16 frames long, and hence our test-set contains 102,304 frames. Figure 4 reports the total processing speed in seconds while Figure 5 reports the real-time speed up factor. Tab. XVII shows detailed analysis of this experiment. For each batch size we ran our technique twice to ensure consistency. Results show that the processing speed gain from 10 to 100 batch size is not significant. Thats between 16-19.3 real-time speed up factor.

III. Deep Analysis on Network Responses

Figure 6 visualizes the feature response of our technique. We show the visualization of two different image sequences. For each sequence, we randomly selected two segments (16 frames) from UCF101 and synthetically generated a sharp and gradual transition using image compositing models. We treated one of the two sequences as no-transition. We examined all segments using our technique, DeepSBD. Figure 6 shows the heat map of some Conv5 filter responses for each transition type. The filters are stacked next to each other, in blocks. The green grid shows some filters' borders. Time is the y-axis and space is the x-axis. Vertical space is averaged over the horizontal space. Sharp transitions have abrupt responses in the time axis in form of bright horizontal lines. Gradual transitions have blurred responses in the time axis. No transitions do not show a specific response pattern. The learned patterns of the three classes capture meaningful and discriminative information for the different types of shot transitions. Such information generate high detection accuracy as shown through out our results.
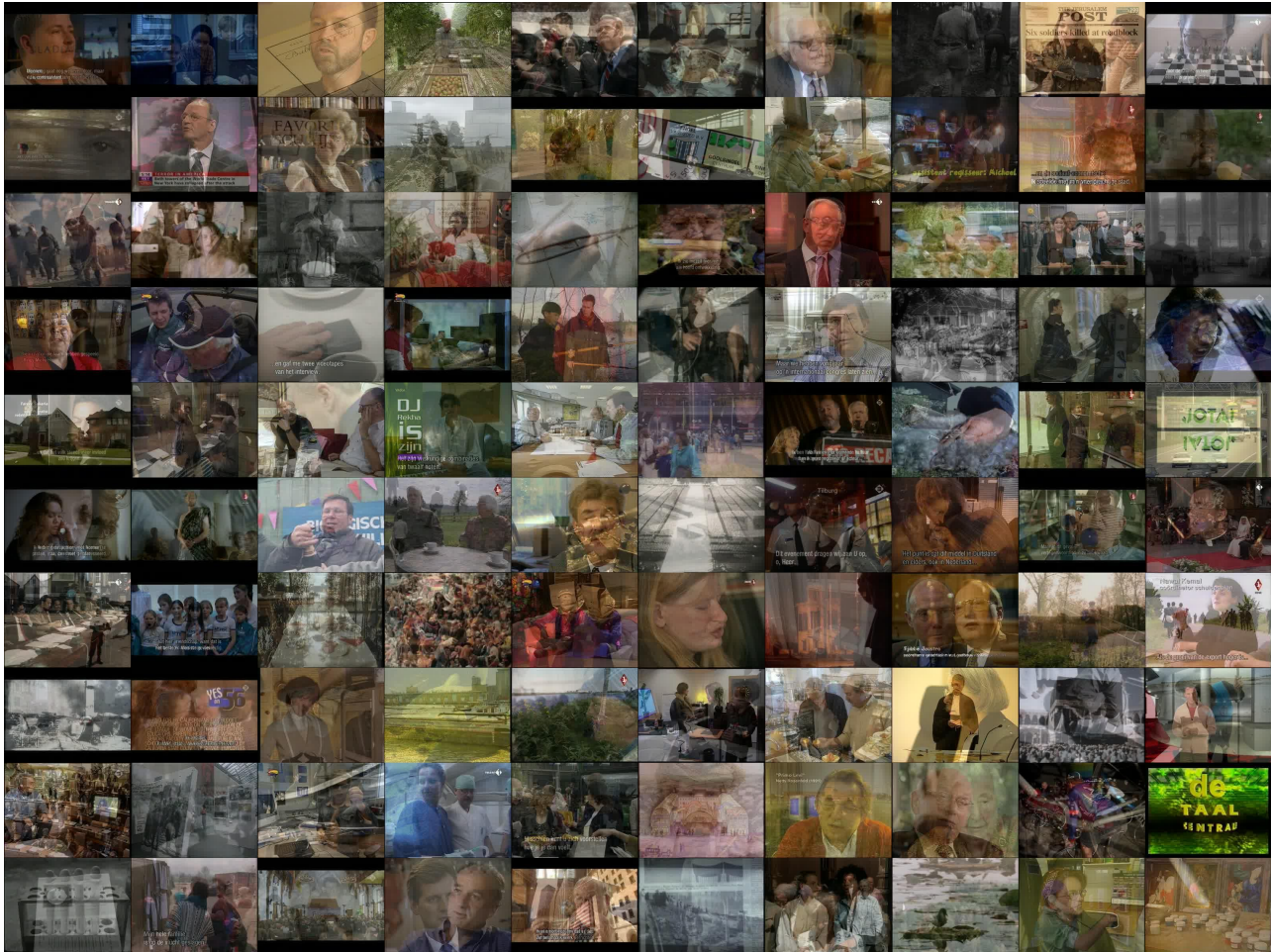
Fig. 1. 100 frames from the gradual transition class of our dataset. This data is generated synthetically through image compositing.
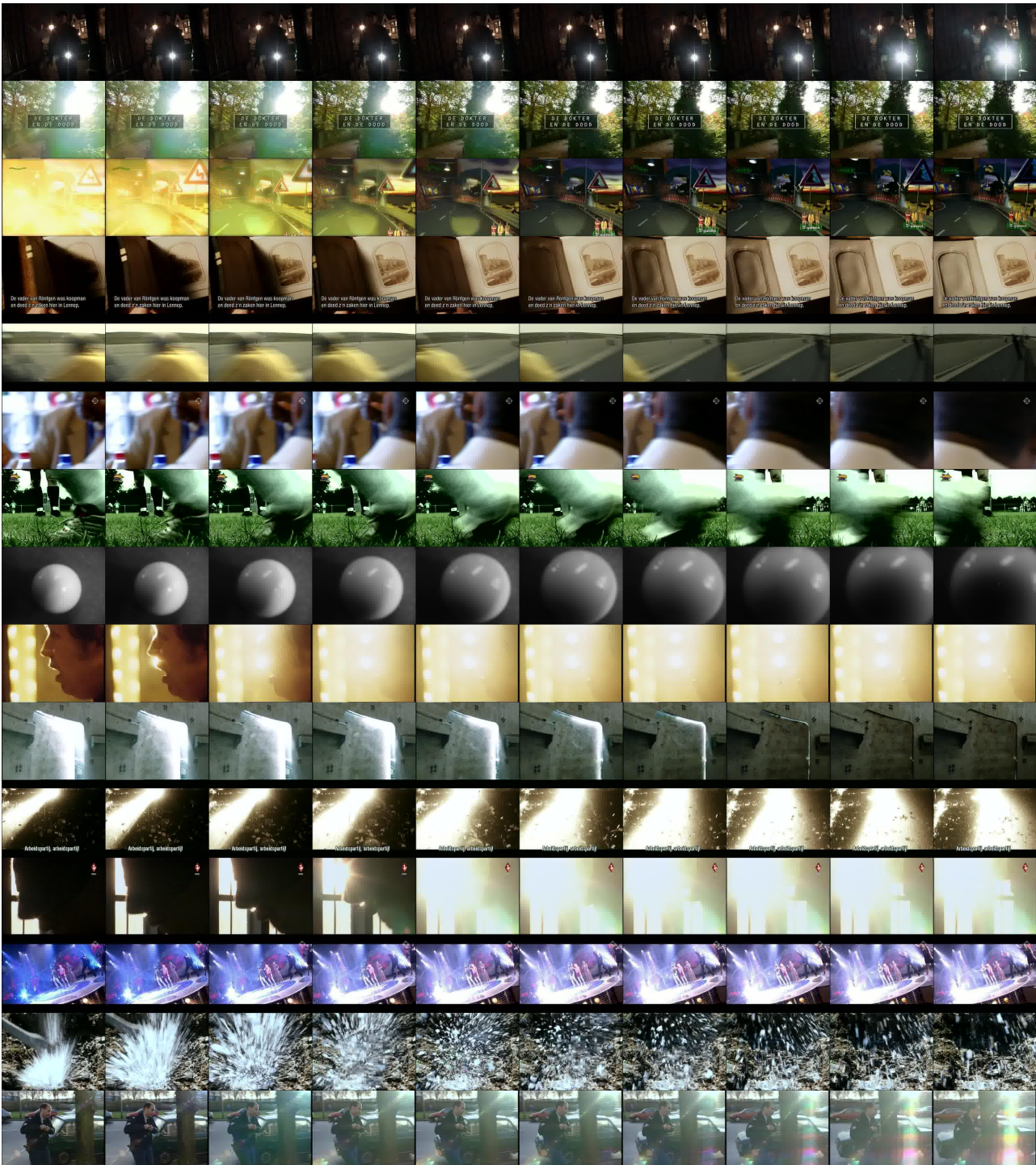
Fig. 2. Hard negative samples from our bootstrapping dataset. We carefully selected these samples through a semi-automated process. They represent complicated cases such as illumination variation, fast motion, occlusion and so on.
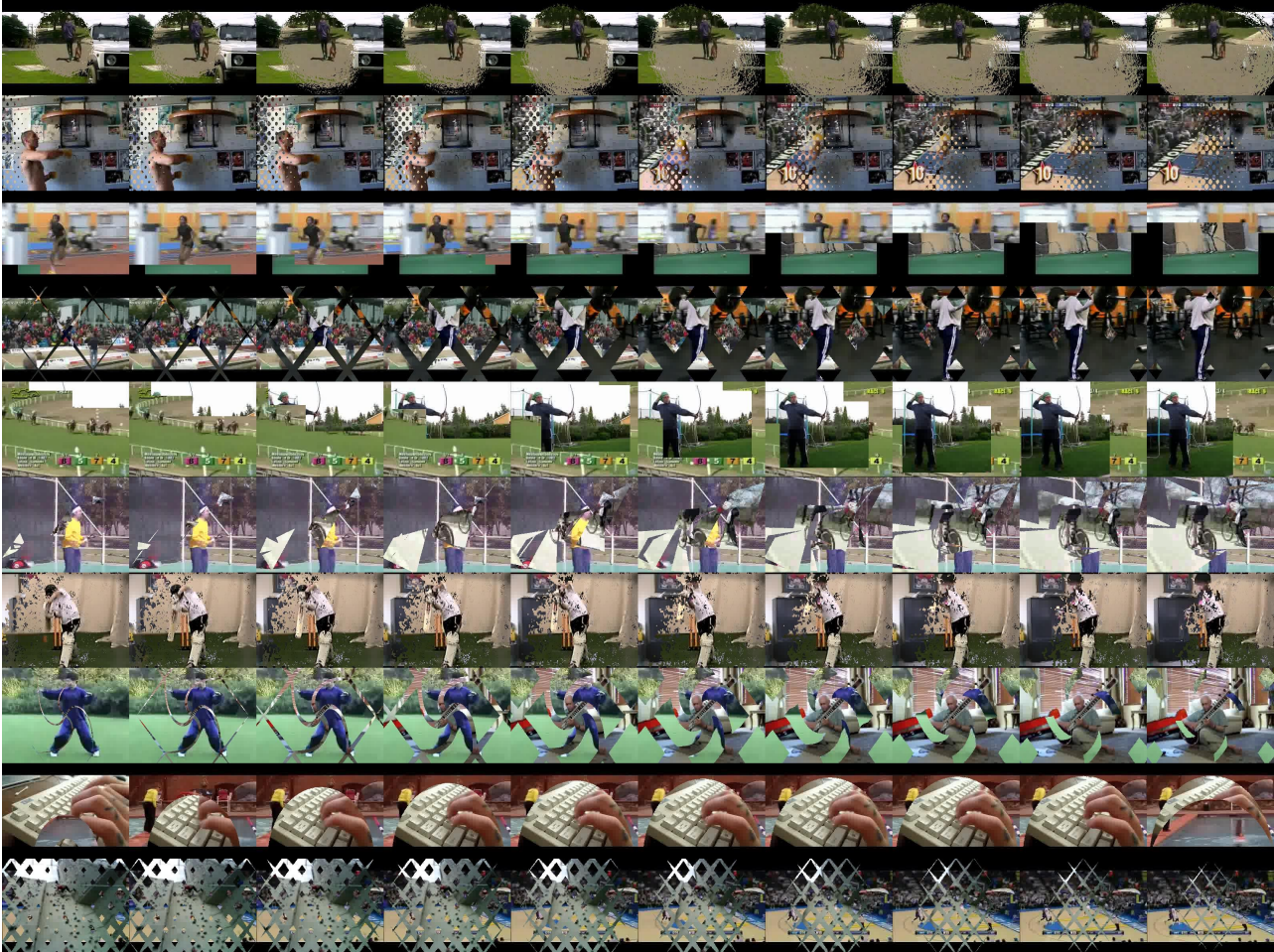
Fig. 3. 10 sequences from our synthetically generated wipes data-set. Each row shows frames from one sequence. The dataset is generated using image compositing models with a wide variety of alpha mattes.

	P	R	F	P	R	F
T2001a						
R_3-5	0.693	0.78	0.734	0.863	0.691	0.768
R_3-6	0.762	0.814	0.787	0.93	0.891	0.91
R_3-6+BT	0.917	0.753	0.827	0.96	0.923	**0.941**
S+r	0.782	0.851	0.815	0.926	0.92	0.923
S+r+BT	**0.951**	0.861	0.904	0.927	**0.936**	0.931
S+BT	0.934	**0.912**	**0.923**	**0.979**	0.904	**0.94**
T2006						
R_3-5	0.641	0.747	0.69	0.691	0.838	0.758
S+r	0.834	0.744	0.786	0.86	0.873	0.866
S+r+BT	**0.888**	0.804	**0.844**	0.863	**0.93**	**0.895**
S+BT	0.827	**0.834**	0.83	**0.876**	0.869	0.872
T2007						
R_3-5	0.495	0.665	0.568	0.894	0.872	0.883
R_3-6 +	0.683	0.683	0.683	0.957	0.95	0.953
R_3-6+BT	0.755	0.705	0.729	0.961	0.961	0.961
S+r	0.722	0.63	0.673	0.979	0.955	**0.967**
S+r+BT	**0.799**	**0.753**	**0.776**	**0.973**	**0.969**	**0.971**
S+BT	0.779	0.714	0.745	0.969	**0.966**	0.968
T2003						
S+r	0.735	0.703	0.718	**0.899**	0.837	0.867
S+r+BT	**0.779**	0.741	0.759	0.892	**0.842**	0.866
S+BT	0.741	**0.804**	**0.771**	0.898	0.846	**0.871**
T2004						
S+r	0.868	0.774	0.818	0.928	**0.929**	**0.929**
S+r+BT	**0.918**	0.819	0.866	0.923	**0.929**	0.926
S+BT	0.888	**0.884**	**0.886**	0.941	0.918	0.929
T2005						
S+BT	0.791	0.866	0.827	0.927	0.941	0.934

TABLE I

TRAINING OUR TECHNIQUE DEEPSBD WITH DIFFERENT DATASETS. R_3-5 REPRESENT ALL TRECVID VIDEOS EXCEPT 2001A, 2006 AND 2007. RESULTS SHOW THAT THE BEST PERFORMANCE IS ALWAYS GENERATED WHEN BOTH OUR SYNTHETIC (S) AND BOOTSTRAPPING (BT) DATASETS ARE USED (SEE S+R+BT AND S+BT). HERE, R IS A VERY SMALL PORTION OF REAL VIDEOS (T2005 AND BARALDI). THE ADVANTAGE OF USING S+BT IS ALLOWING US TO TEST ON ALL TRECVID VIDEOS, INCLUDING T2005. FINALLY, OUR BOOTSTRAPPING DATA BT CLEARLY IMPROVES THE PRECISION AND OVERALL PERFORMANCE.

	Gradual and Sharp					
Video	TP	FP	FN	P	R	F
BOR03	237	32	5	0.881	0.979	0.928
BOR08	456	8	75	0.983	0.859	0.917
BOR10	58	84	94	0.408	0.382	0.395
BOR12	117	5	19	0.959	0.86	0.907
BOR17	77	137	171	0.36	0.31	0.333
Total	945	266	364	0.78	0.722	0.75

TABLE II

DETAILED PER VIDEO RESULTS OF T2001B. HERE, WE USE S+R+BT FOR TRAINING OUR MODEL. WE REPORT THE COMBINED RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. WE SHOW THE TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual and Sharp					
Video	TP	FP	FN	P	R	F
BOR03	240	30	2	0.889	0.992	0.938
BOR08	500	7	31	0.986	0.942	0.963
BOR10	54	82	98	0.397	0.355	0.375
BOR12	114	5	22	0.958	0.838	0.894
BOR17	66	106	182	0.384	0.266	0.314
Total	974	230	335	0.809	0.744	0.775

TABLE III

DETAILED PER VIDEO RESULTS OF T2001B. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE COMBINED RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. WE SHOW THE TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Fig. 4. Processing time in seconds of our technique. We report the results for different batch sizes as input.

Fig. 5. Real-time speed factor of our technique. We report the results for different batch sizes as input.

	Gradual							Sharp						
Video	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**
BOR10_001	11	11	0	0	1	1	1	0	0	0	0	-	-	-
BOR10_002	11	9	0	2	1	0.818	0.9	0	0	0	0	-	-	-
NAD57	25	22	1	3	0.957	0.88	0.917	45	45	4	0	0.918	1	0.957
NAD58	44	37	0	7	1	0.841	0.914	40	33	0	7	1	0.825	0.904
anni001	8	6	0	2	1	0.75	0.857	0	0	1	0	0	-	-
anni005	27	26	2	1	0.929	0.963	0.945	39	36	13	3	0.735	0.923	0.818
anni006	31	27	3	4	0.9	0.871	0.885	42	41	0	1	1	0.976	0.988
anni007	5	4	0	1	1	0.8	0.889	5	5	0	0	1	1	1
anni008	13	12	0	1	1	0.923	0.96	2	2	0	0	1	1	1
anni009	64	57	3	7	0.95	0.891	0.919	40	37	0	3	1	0.925	0.961
anni010	56	50	11	6	0.82	0.893	0.855	98	84	1	14	0.988	0.857	0.918
Total	295	261	20	34	0.929	0.885	0.906	311	283	19	28	0.937	0.91	0.923

TABLE IV

DETAILED PER VIDEO RESULTS OF T2001A. HERE, WE USE S+R+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).
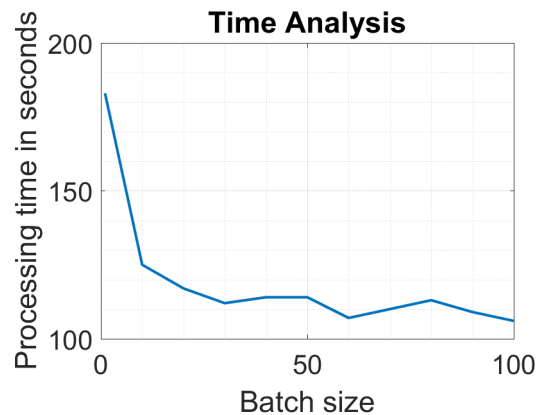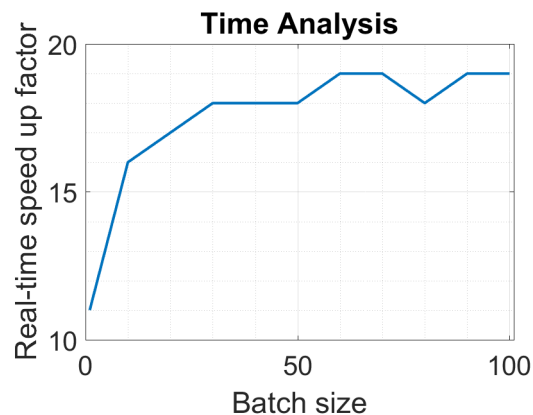
	Gradual							Sharp						
Video	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**
BOR10_001	11	11	0	0	1	1	1	0	0	0	0	-	-	-
BOR10_002	11	10	0	1	1	0.909	0.952	0	0	0	0	-	-	-
NAD57	25	21	2	4	0.913	0.84	0.875	45	45	1	0	0.978	1	0.989
NAD58	44	39	0	5	1	0.886	0.94	40	35	0	5	1	0.875	0.933
anni001	8	6	0	2	1	0.75	0.857	0	0	0	0	-	-	-
anni005	27	27	1	0	0.964	1	0.982	39	35	5	4	0.875	0.897	0.886
anni006	31	27	5	4	0.844	0.871	0.857	42	39	0	3	1	0.929	0.963
anni007	5	5	0	0	1	1	1	5	5	0	0	1	1	1
anni008	13	13	0	0	1	1	1	2	2	0	0	1	1	1
anni009	64	60	2	4	0.968	0.938	0.952	40	36	0	4	1	0.9	0.947
anni010	56	50	9	6	0.847	0.893	0.87	98	84	0	14	1	0.857	0.923
Total	295	269	19	26	0.934	0.912	0.923	311	281	6	30	0.979	0.904	0.94

TABLE V

DETAILED PER VIDEO RESULTS OF T2001A. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual and Sharp					
Video	**TP**	**FP**	**FN**	**P**	**R**	**F**
01811a	60	7	4	0.896	0.938	0.916
6011	40	96	81	0.294	0.331	0.311
8024	85	22	21	0.794	0.802	0.798
8386	113	10	5	0.919	0.958	0.938
8401	26	5	5	0.839	0.839	0.839
10558a	122	1	8	0.992	0.938	0.964
23585a	149	10	16	0.937	0.903	0.92
23585b	103	3	1	0.972	0.99	0.981
34921a	70	4	5	0.946	0.933	0.94
34921b	91	10	8	0.901	0.919	0.91
36553	200	21	14	0.905	0.935	0.92
50009	44	28	14	0.611	0.759	0.677
50028	81	17	12	0.827	0.871	0.848
UGS01	164	8	12	0.953	0.932	0.943
UGS04	218	25	5	0.897	0.978	0.936
UGS05	21	6	9	0.778	0.7	0.737
UGS09	169	12	24	0.934	0.876	0.904
Total	1756	285	244	0.86	0.878	0.869

TABLE VI

DETAILED PER VIDEO RESULTS OF T2002. HERE, WE USE S+R+BT FOR TRAINING OUR MODEL. WE REPORT THE COMBINED RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. WE SHOW THE TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual and Sharp					
Video	**TP**	**FP**	**FN**	**P**	**R**	**F**
01811a	60	7	4	0.896	0.938	0.916
6011	39	96	82	0.289	0.322	0.305
8024	96	29	10	0.768	0.906	0.831
8386	114	5	4	0.958	0.966	0.962
8401	30	8	1	0.789	0.968	0.87
10558a	125	1	5	0.992	0.962	0.977
23585a	159	8	6	0.952	0.964	0.958
23585b	103	4	1	0.963	0.99	0.976
34921a	71	6	4	0.922	0.947	0.934
34921b	91	11	8	0.892	0.919	0.905
36553	202	26	12	0.886	0.944	0.914
50009	53	29	5	0.646	0.914	0.757
50028	89	18	4	0.832	0.957	0.89
UGS01	171	12	5	0.934	0.972	0.953
UGS04	222	15	1	0.937	0.996	0.965
UGS05	26	21	4	0.553	0.867	0.675
UGS09	176	17	17	0.912	0.912	0.912
Total	1827	313	173	0.854	0.913	0.883

TABLE VII

DETAILED PER VIDEO RESULTS OF T2002. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE COMBINED RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. WE SHOW THE TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Video	Gradual #T	TP	FP	FN	P	R	F	Sharp #T	TP	FP	FN	P	R	F
203_CNN	171	134	44	37	0.753	0.784	0.768	280	228	13	52	0.946	0.814	0.875
222_CNN	101	74	5	27	0.937	0.733	0.822	309	273	11	36	0.961	0.883	0.921
224_ABC	131	108	10	23	0.915	0.824	0.867	296	281	13	15	0.956	0.949	0.953
412_ABC	137	115	6	22	0.95	0.839	0.891	345	323	17	22	0.95	0.936	0.943
425_ABC	180	161	12	19	0.931	0.894	0.912	295	266	11	29	0.96	0.902	0.93
515_CNN	131	89	11	42	0.89	0.679	0.771	283	265	17	18	0.94	0.936	0.938
531_CNN	108	75	12	33	0.862	0.694	0.769	359	316	13	43	0.96	0.88	0.919
619_ABC	127	46	125	81	0.269	0.362	0.309	321	154	155	167	0.498	0.48	0.489
Total	1086	802	225	284	0.781	0.738	0.759	2488	2106	250	382	0.894	0.846	0.87

TABLE VIII

DETAILED PER VIDEO RESULTS OF T2003. HERE, WE USE S+r+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Video	Gradual #T	TP	FP	FN	P	R	F	Sharp #T	TP	FP	FN	P	R	F
203_CNN	171	143	57	28	0.715	0.836	0.771	280	230	9	50	0.962	0.821	0.886
222_CNN	101	80	24	21	0.769	0.792	0.78	309	275	11	34	0.962	0.89	0.924
224_ABC	131	116	14	15	0.892	0.885	0.889	296	282	8	14	0.972	0.953	0.962
412_ABC	137	122	11	15	0.917	0.891	0.904	345	323	11	22	0.967	0.936	0.951
425_ABC	180	170	28	10	0.859	0.944	0.899	295	265	12	30	0.957	0.898	0.927
515_CNN	131	105	16	26	0.868	0.802	0.833	283	259	15	24	0.945	0.915	0.93
531_CNN	108	85	24	23	0.78	0.787	0.783	359	316	18	43	0.946	0.88	0.912
619_ABC	127	52	131	75	0.284	0.409	0.335	321	154	155	167	0.498	0.48	0.489
Total	1086	873	305	213	0.741	0.804	0.771	2488	2104	239	384	0.898	0.846	0.871

TABLE IX

DETAILED PER VIDEO RESULTS OF T2003. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Video	Gradual #T	TP	FP	FN	P	R	F	Sharp #T	TP	FP	FN	P	R	F
1004_ABCa	203	166	13	37	0.927	0.818	0.869	224	213	22	11	0.906	0.951	0.928
1012_CNNa	170	136	13	34	0.913	0.8	0.853	215	194	15	21	0.928	0.902	0.915
1016_CNNa	150	119	9	31	0.93	0.793	0.856	242	214	13	28	0.943	0.884	0.913
1021_ABCa	175	154	13	21	0.922	0.88	0.901	240	230	18	10	0.927	0.958	0.943
1101_CNNa	204	172	20	32	0.896	0.843	0.869	191	187	11	4	0.944	0.979	0.961
1109_ABCa	170	151	10	19	0.938	0.888	0.912	257	246	15	11	0.943	0.957	0.95
1123_CNNa	126	93	29	33	0.762	0.738	0.75	236	214	10	22	0.955	0.907	0.93
1126_ABCa	189	168	12	21	0.933	0.889	0.911	273	261	23	12	0.919	0.956	0.937
1208_CNNa	137	112	15	25	0.882	0.818	0.848	212	196	17	16	0.92	0.925	0.922
1210_ABCa	159	140	8	19	0.946	0.881	0.912	271	252	14	19	0.947	0.93	0.939
1216_CNNa	153	119	11	34	0.915	0.778	0.841	197	187	26	10	0.878	0.949	0.912
1221_ABCa	195	149	14	46	0.914	0.764	0.832	217	197	27	20	0.879	0.908	0.893
Total	2031	1679	167	352	0.91	0.827	0.866	2031	2591	211	184	0.925	0.934	0.929

TABLE X

DETAILED PER VIDEO RESULTS OF T2004. HERE, WE USE S+r+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Video	Gradual #T	TP	FP	FN	P	R	F	Sharp #T	TP	FP	FN	P	R	F
1004_ABCa	203	177	9	26	0.952	0.872	0.91	224	209	18	15	0.921	0.933	0.927
1012_CNNa	170	149	23	21	0.866	0.876	0.871	215	191	13	24	0.936	0.888	0.912
1016_CNNa	150	122	13	28	0.904	0.813	0.856	242	211	12	31	0.946	0.872	0.908
1021_ABCa	175	154	22	21	0.875	0.88	0.877	240	227	14	13	0.942	0.946	0.944
1101_CNNa	204	187	13	17	0.935	0.917	0.926	191	180	12	11	0.938	0.942	0.94
1109_ABCa	170	159	12	11	0.93	0.935	0.933	257	241	11	16	0.956	0.938	0.947
1123_CNNa	126	99	32	27	0.756	0.786	0.77	236	206	8	30	0.963	0.873	0.916
1126_ABCa	189	179	16	10	0.918	0.947	0.932	273	260	14	13	0.949	0.952	0.951
1208_CNNa	137	117	22	20	0.842	0.854	0.848	212	192	17	20	0.919	0.906	0.912
1210_ABCa	159	148	21	11	0.876	0.931	0.902	271	251	7	20	0.973	0.926	0.949
1216_CNNa	153	137	25	16	0.846	0.895	0.87	197	184	21	13	0.898	0.934	0.915
1221_ABCa	195	168	18	27	0.903	0.862	0.882	217	195	13	22	0.938	0.899	0.918
Total	2031	1796	226	235	0.888	0.884	0.886	2775	2547	160	228	0.941	0.918	0.929

TABLE XI

DETAILED PER VIDEO RESULTS OF T2004. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual						**Sharp**						
Video	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**
LNA	198	147	11	51	0.93	0.742	0.826	45	31	26	14	0.544	0.689	0.608
NFC	77	57	11	20	0.838	0.74	0.786	121	115	8	6	0.935	0.95	0.943
NEC	94	88	3	6	0.967	0.936	0.951	74	57	5	17	0.919	0.77	0.838
HNA	124	107	4	17	0.964	0.863	0.911	24	21	8	3	0.724	0.875	0.792
3PGC	228	171	32	57	0.842	0.75	0.794	132	112	48	20	0.7	0.848	0.767
CLE	123	98	22	25	0.817	0.797	0.807	244	236	33	8	0.877	0.967	0.92
CDC	302	231	27	71	0.895	0.765	0.825	139	129	65	10	0.665	0.928	0.775
8NNE	214	184	44	30	0.807	0.86	0.833	424	418	36	6	0.921	0.986	0.952
CLE	37	28	7	9	0.8	0.757	0.778	57	54	7	3	0.885	0.947	0.915
5PGC	190	155	44	35	0.779	0.816	0.797	81	75	30	6	0.714	0.926	0.806
MNE	181	156	11	25	0.934	0.862	0.897	339	323	21	16	0.939	0.953	0.946
CLE	27	25	4	2	0.862	0.926	0.893	44	42	0	2	1	0.955	0.977
1NNE	146	134	11	12	0.924	0.918	0.921	120	118	5	2	0.959	0.983	0.971
Total	1941	1581	231	360	0.873	0.815	0.843	1844	1731	292	113	0.856	0.939	0.895

TABLE XII

DETAILED PER VIDEO RESULTS OF T2006. HERE, WE USE S+R+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual						**Sharp**						
Video	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**
LNA	198	150	16	48	0.904	0.758	0.824	45	39	31	6	0.557	0.867	0.678
NFC	77	56	22	21	0.718	0.727	0.723	121	115	8	6	0.935	0.95	0.943
NEC	94	81	8	13	0.91	0.862	0.885	74	46	6	28	0.885	0.622	0.73
HNA	124	113	33	11	0.774	0.911	0.837	24	22	6	2	0.786	0.917	0.846
3PGC	228	168	38	60	0.816	0.737	0.774	132	105	41	27	0.719	0.795	0.755
CLE	123	110	28	13	0.797	0.894	0.843	244	223	14	21	0.941	0.914	0.927
CDC	302	241	40	61	0.858	0.798	0.827	139	119	53	20	0.692	0.856	0.765
8NNE	214	183	53	31	0.775	0.855	0.813	424	372	28	52	0.93	0.877	0.903
CLE	37	35	7	2	0.833	0.946	0.886	57	51	3	6	0.944	0.895	0.919
5PGC	190	149	42	41	0.78	0.784	0.782	81	72	17	9	0.809	0.889	0.847
MNE	181	168	26	13	0.866	0.928	0.896	339	294	17	45	0.945	0.867	0.905
CLE	27	26	5	1	0.839	0.963	0.897	44	28	0	16	1	0.636	0.778
1NNE	146	138	21	8	0.868	0.945	0.905	120	116	3	4	0.975	0.967	0.971
Total	1941	1618	339	323	0.827	0.834	0.83	1844	1602	227	242	0.876	0.869	0.872

TABLE XIII

DETAILED PER VIDEO RESULTS OF T2006. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual						**Sharp**						
Video	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**	**#T**	**TP**	**FP**	**FN**	**P**	**R**	**F**
BG_11362	4	2	2	2	0.5	0.5	0.5	104	95	14	9	0.872	0.913	0.892
BG_14213	61	49	0	12	1	0.803	0.891	106	106	3	0	0.972	1	0.986
BG_2408	20	17	4	3	0.81	0.85	0.829	101	100	5	1	0.952	0.99	0.971
BG_34901	16	9	4	7	0.692	0.562	0.621	224	215	4	9	0.982	0.96	0.971
BG_35050	4	1	0	3	1	0.25	0.4	98	98	0	0	1	1	1
BG_35187	23	19	3	4	0.864	0.826	0.844	135	125	2	10	0.984	0.926	0.954
BG_36028	0	0	0	0	-	-	-	87	86	9	1	0.905	0.989	0.945
BG_36182	14	3	0	11	1	0.214	0.353	95	95	1	0	0.99	1	0.995
BG_36506	6	4	1	2	0.8	0.667	0.727	77	76	0	1	1	0.987	0.993
BG_36537	30	24	13	6	0.649	0.8	0.716	259	243	0	16	1	0.938	0.968
BG_36628	10	5	3	5	0.625	0.5	0.556	192	187	2	5	0.989	0.974	0.982
BG_37359	6	6	1	0	0.857	1	0.923	164	158	1	6	0.994	0.963	0.978
BG_37417	12	9	2	3	0.818	0.75	0.783	76	73	2	3	0.973	0.961	0.967
BG_37822	10	9	1	1	0.9	0.9	0.9	119	115	3	4	0.975	0.966	0.97
BG_37879	4	2	1	2	0.667	0.5	0.571	95	91	0	4	1	0.958	0.978
BG_38150	4	4	0	0	1	1	1	215	213	2	2	0.991	0.991	0.991
BG_9401	3	3	0	0	1	1	1	89	88	0	1	1	0.989	0.994
Total	227	166	35	61	0.826	0.731	0.776	227	2164	48	72	0.978	0.968	0.973

TABLE XIV

DETAILED PER VIDEO RESULTS OF T2007. HERE, WE USE S+R+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

	Gradual							Sharp						
Video	#T	TP	FP	FN	P	R	F	#T	TP	FP	FN	P	R	F
BG_11362	4	0	2	4	0	0	-	104	81	13	23	0.862	0.779	0.818
BG_14213	61	45	2	16	0.957	0.738	0.833	106	106	3	0	0.972	1	0.986
BG_2408	20	18	5	2	0.783	0.9	0.837	101	100	7	1	0.935	0.99	0.962
BG_34901	16	8	2	8	0.8	0.5	0.615	224	219	6	5	0.973	0.978	0.976
BG_35050	4	0	1	4	0	0	-	98	98	0	0	1	1	1
BG_35187	23	19	1	4	0.95	0.826	0.884	135	125	3	10	0.977	0.926	0.951
BG_36028	0	0	0	0	-	-	-	87	86	14	1	0.86	0.989	0.92
BG_36182	14	3	0	11	1	0.214	0.353	95	95	3	0	0.969	1	0.984
BG_36506	6	4	2	2	0.667	0.667	0.667	77	76	1	1	0.987	0.987	0.987
BG_36537	30	24	20	6	0.545	0.8	0.649	259	244	0	15	1	0.942	0.97
BG_36628	10	6	3	4	0.667	0.6	0.632	192	191	5	1	0.974	0.995	0.985
BG_37359	6	6	1	0	0.857	1	0.923	164	157	3	7	0.981	0.957	0.969
BG_37417	12	10	2	2	0.833	0.833	0.833	76	72	4	4	0.947	0.947	0.947
BG_37822	10	9	1	1	0.9	0.9	0.9	119	115	5	4	0.958	0.966	0.962
BG_37879	4	3	1	1	0.75	0.75	0.75	95	92	0	3	1	0.968	0.984
BG_38150	4	4	2	0	0.667	1	0.8	215	214	1	1	0.995	0.995	0.995
BG_9401	3	3	1	0	0.75	1	0.857	89	89	0	0	1	1	1
Total	227	162	46	65	0.779	0.714	0.745	2236	2160	68	76	0.969	0.966	0.968

TABLE XV

DETAILED PER VIDEO RESULTS OF T2007. HERE, WE USE S+BT FOR TRAINING OUR MODEL. WE REPORT THE RESULTS FOR BOTH GRADUAL AND SHARP TRANSITIONS. FOR EACH CLASS WE SHOW THE NUMBER OF TRANSITIONS (#T), TRUE POSITIVES (TP), FALSE POSITIVES (FP), FALSE NEGATIVES (FN), PRECISION (P), RECALL (R) AND F-MEASURE (F).

Video	#T	TP	FP	FN	P	R	F
V1	80	66	12	14	0.846	0.825	0.835
V2	146	123	8	23	0.939	0.842	0.888
V3	112	101	1	11	0.99	0.902	0.944
V4	60	59	2	1	0.967	0.983	0.975
V5	104	101	3	3	0.971	0.971	0.971
V6	54	51	4	3	0.927	0.944	0.936
V7	109	105	3	4	0.972	0.963	0.968
V8	196	172	5	24	0.972	0.878	0.922
V9	61	60	1	1	0.984	0.984	0.984
V10	63	59	0	4	1	0.937	0.967
Overall	985	897	39	88	0.958	0.911	0.934

TABLE XVI

Detailed per video results of the RAI dataset. Here, we use S+BT for training our model. We report the combined transition results. For each video we show the total number of transitions (#T), true positives (TP), false positives (FP), false negatives (FN), precision (P), recall (R) and F-measure (F).

Batch size	Starting Time	End Time	# Seconds	Memory	# Iterations	Faster than real time by
1	19:08:23	19:11:26	183	69413912	6394	11.18076503
1	16:13:03	16:16:06	184	69413912	6394	11.12
10	21:16:31	21:18:37	125	694139048	640	16.36864
10	21:20:45	21:22:53	128	694139048	640	15.985
20	14:55:16	14:57:13	118	1388278088	320	17.33966102
20	14:59:21	15:01:19	117	1388278088	320	17.48786325
30	21:06:39	21:08:30	112	2082417128	214	18.26857143
30	21:10:44	21:12:36	112	2082417128	214	18.26857143
40	15:04:59	15:06:52	114	2776556168	160	17.94807018
40	15:09:55	15:11:56	120	2776556168	160	17.05066667
50	11:00:04	11:01:59	115	3470695208	128	17.792
50	14:50:18	14:52:11	114	3470695208	128	17.94807018
60	21:25:47	21:27:34	107	4164834248	107	19.12224299
60	16:07:51	16:09:44	113	4164834248	107	18.10690265
70	15:18:27	15:20:19	112	4858973288	92	18.26857143
70	15:22:29	15:24:20	110	4858973288	92	18.60072727
80	10:49:16	10:51:09	113	5553112328	80	18.10690265
80	10:54:25	10:56:19	114	5553112328	80	17.94807018
90	15:50:29	15:52:19	109	6247251368	72	18.77137615
90	15:55:18	15:57:08	111	6247251368	72	18.43315315
100	21:32:57	21:34:43	106	6941390408	64	19.30264151
100	21:36:45	21:38:32	106	6941390408	64	19.30264151

TABLE XVII

THE PROCESSING TIME OF OUR TECHNIQUE. WE REPORT DETAILED ANALYSIS OF DIFFERENT BATCH SIZES AS INPUT. THE BIGGER THE BATCH SIZE, THE LESS PROCESSING TIME IS REQUIRED. THIS, HOWEVER, REQUIRES MORE GPU MEMORY. EXPERIMENTS SHOWS THAT THE PROCESSING SPEED GAIN FROM 10 TO 100 BATCH SIZE IS NOT SIGNIFICANT. THATS BETWEEN 16-19.3 REAL-TIME SPEED UP FACTOR.
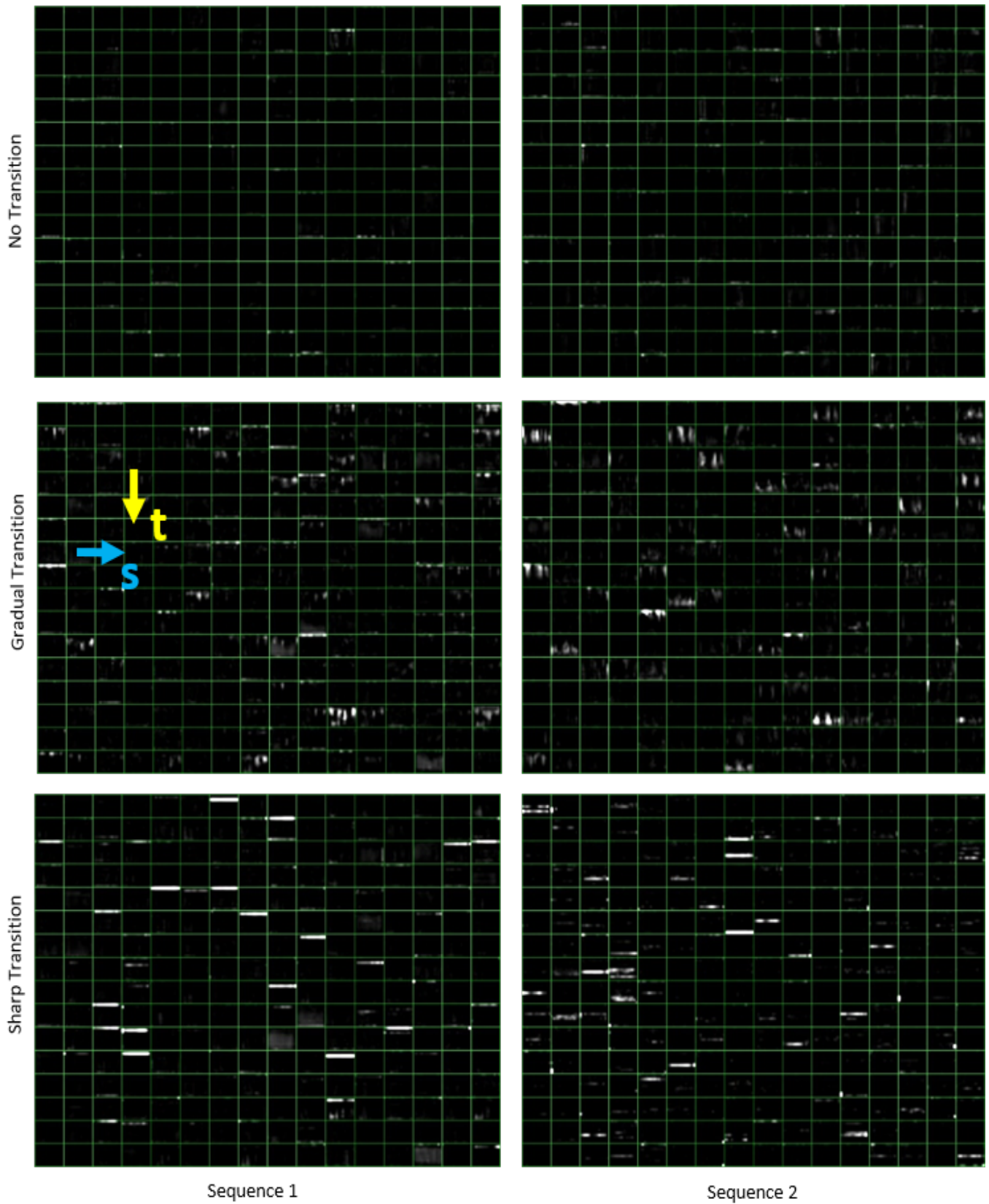
Fig. 6. Filter responses of our technique DeepSBD stacked next to each other. The green grid shows filters' borders. Here, y-axis is time (see blue arrow) and x-axis is space (see yellow arrow). Sharp transitions have an abrupt response in time (bright horizontal lines). Gradual transitions have blurred responses in time. No transition do not show specific patterns.