Using Deep Features for Scene Detection and Annotation

Stanislav Protasov
Innopolis University,
Innopolis, Russia
stanislav.protasov@gmail.com

Adil Mehmood Khan
Innopolis University,
Innopolis, Russia
a.khan@innopolis.ru

Konstantin Sozykin
Innopolis University,
Innopolis, Russia
k.sozykin@innopolis.ru

Muhammad Ahmad
Innopolis University,
Innopolis, Russia
m.ahmad@innopolis.ru

Abstract—**Semantic video indexing problem is still underdeveloped. Solutions for the problem will significantly enrich experience in video search, monitoring and surveillance. This paper approaches scene detection and annotation, specifically, the task of video structure mining for video indexing using deep features. The paper proposes and implements a pipeline that consists of feature extraction and filtering, shot clustering and labeling stages. Deep convolutional network is used as a source of features. The pipeline is evaluated using metrics for both scene detection and annotation. Obtained results show high scene detection and annotation quality estimated with various metrics. Additionally, we performed overview and analysis of contemporary segmentation and annotation metrics. The outcome of this work can be applied for semantic video annotation in real time.**

I. Introduction

In the last decade researchers working in area of video processing highlighted the problem of semantic gap [1]. On the low-level we have binary video representation (video files and streams), whereas on the high-level of video representation we use words and texts (movie genre and category, reviews, etc.). Semantic gap refers to lack of intermediate representations, abstraction levels between these extremes. Powerful video search engines supporting scene, object, person or event indexing do not exist due to this lack. Researchers approach the problem from both edges of the gap by implementing per-frame (image) recognition [2], and whole video classification (e.g. movie recommendation networks) [3]. But we still need to close the rest of the gap. Solving this task will probably provide industry a tool for large video datasets mining and search. In this work, we propose to start semantic video annotation process from mining video structure. Given a structure, we can iteratively enrich video annotation with scene types, events, present objects and persons and other data. Semantic contribution done by this work relates to scene type information.

Previously, scene detection and annotation tasks were studied and developed separately because they are used in different applied domains. Scene detection is important for video production, whereas annotation is done for semantic video analysis. Nonetheless, today things change. Contemporary automated semantic video analysis tasks involve complex pipelines. Among those are automated storyboard creation in film production, advanced video skimming algorithms and video processing for surveillance and manufacturing control.

During the last decade image and video recognition algorithms significantly improved. Deep learning techniques together with GPU computations allow training and serving very accurate real-time classifiers. Our intention is to exploit new image classification approaches to create representative feature vector for video structure mining as a part of semantic analysis.

Contribution of this paper includes but is not limited to the following results.

1) We propose a pipeline for automated video structure mining and labeling. This pipeline consolidates image classification using deep network and time-efficient scene detection approach. Pipeline consist of four major stages, which can be iteratively improved.
2) Within the pipeline, we implemented data filtering approaches to deal with impulse noise without affecting accuracy of scene edge detection. We performed analysis of two edge-preserving filters and came up with optimal filtering parameters.
3) We performed comparative analysis of different quality metrics for video structure mining and for video labeling. We implemented these metrics and studied their similarities and limitations.

The paper is organized as follows. In section II we discuss related works including contemporary approaches to image and video classification, and scene detection. Section III is devoted to our proposed methodology for scene detection and annotation. In section IV we present experimental results. These results are then discussed in section V, conclusions are made in section VI.

II. Relates works

Video understanding and search require retrieval and storage of structured semantic information, such as labels, annotations, objects and events. Providing granular search outputs assume that videos should be divided into semantically cohesive blocks. As we discussed in *Introduction*, usually researchers address semantic mining or structure mining problem separately.

Our work presents video structure mining approach based on video semantics, in this section we will overview existing works targeting image and video annotation, as well as scene detection methods.

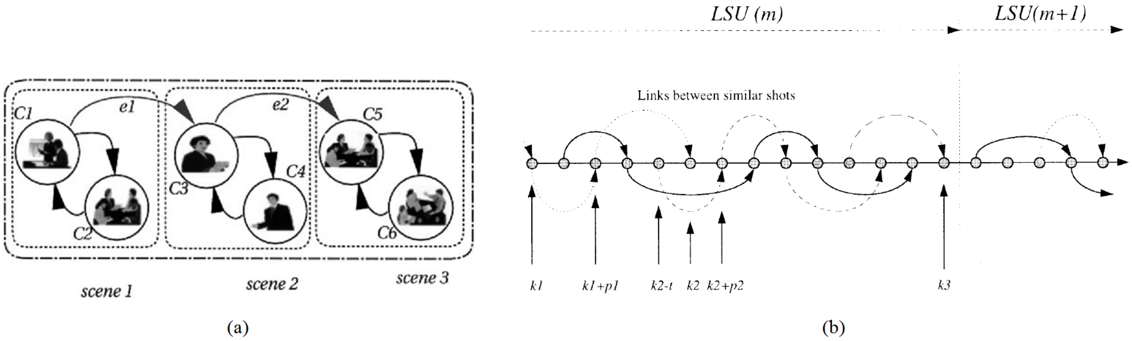


Figure 1. Graph-based (a) and overlapping link (b) approaches for scene detection presented in [12] [13]

A. Image classification

Image classification and object recognition tasks even being studied for a long time, still don't have good general solutions. One of the reasons is that classification accuracy estimation depends on number and quality of classes. Human experience and natural language limit both exactness and size of label sets. Thus, Places205-AlexNet [4] deep network shows only 50% accuracy for Places database with 205 categories and up to 95% for Scene15 dataset [5]. The difference in the results were discussed in paper [4]. Authors showed that increasing dataset diversity and size improve classification accuracy. Also there is a problem with label set size: even being smaller than Places205, Scene15 provides better results given smaller number of labels [6]. Despite of this biased behavior, image classification task is well-studied.

Public contests on image classification provide good overview for state-of-the-art methods. TRECVID [2] contests are devoted to video analysis. The most relevant section is semantic indexing (SIN) where algorithms detect concepts on per-frame basis. Large Scale Visual Recognition Challenge (ILSVRC) [7], specifically detection and object localization contest ("Taster challenge"), shows that top performing algorithms use region proposal networks (RPN) implemented with region-based convolutional network (R-CNN). For example, the solution proposed by Oxford Vision Geometry Group (VGG) [8]. These methods can be used for both image annotation and semantic video annotation. Development of novel methods is influenced by new datasets built by labs and communities: Places [4] and recent Places2 form MIT, Scene15 [5] from the Ponce group (University of Illinois at Urbana-Champaign), ImageNet [9] from Stanford University and others. Together with datasets labs and teams provide most accurate deep networks for detection, localization and recognition: real-time semantic search solution from VGG [10], deep network Places205-CNN based on AlexNet from MIT Computer Science and Artificial Intelligence Laboratory [4], DenseCap network for dense capturing task (simultaneous localization and tagging) from Stanford Computer vision lab [11].

B. Video annotation

There are also works related to video annotation. In [14] authors use an event-based approach. They created rich text annotations trying to infer high-level descriptions of events. This method does not account time interrelations between events. That is, whether events coincide, interact or alternate; events can fuse into complex activities and events. Such interrelation can help in building richer annotations and exact video structure. We believe that semantic search outputs should provide continuous blocks of video that are usually referenced as *scenes* or *story units*. Thus, we can treat scenes as a fusion of tightly connected events.

Torralba et al. proposed solution for semantic video annotation in their paper [6]. They filmed different indoor and outdoor locations with predefined transition map. Algorithm result consisted of per-frame annotations including exact (*office 400/628*) and grouped (*office*) classes. Result accuracy relies on a priori transition information and their approach perform much works without this information. In our work we implemented annotation quality metric proposed in this paper.

C. Scene detection

A comprehensive overview of scene detection algorithms is provided by *Del Fabro et al.* [15]. We follow the same major terminology as in the paper. *Frame* – single image of the video sequence. *Shot* – continuous sequence of frames that are similar with respect to selected feature space and distance metric. *Scene (also story unit, LSU)* – continuous sequence of shots that represents logically solid part of the video.

As per survey [15], scene detection task is understood as a three-staged problem. At the first step (shot detection) frames are grouped into shots. At the second step *key frames* are selected to represent the shot. This step is done to reduce computational cost. At the third stage shots are clustered into scenes using similarity metric and assumptions on the film structure. Survey authors state that it is commonly decided to use feature deviation for shot edge detection. Features are selected and adjusted with respect to problem domain. Widely used features include RGB, HSV or LUV color histograms [13] [16] [17], backgrounds similarity [18] [12], motion features [19], edge ratio change and SIFT [20], spectral features [21]. We also follow feature deviation method for our proposed feature vector.

For key frame selection we can apply different techniques. An overview of existing key frame selection methods is given in [22]. The paper covers 4 methods: two edge frames selection, first frame selection, middle frame selection and the chain method. In the *chain method* we always pick first frame of a shot. Next key frame is added if we reach threshold distance from previous key frame within the shot [12]. We prefer this method, because it produces more cohesive shots: wider range of shots can be clustered using this approach. This is important because: (i) it reduces oversegmentation, and (ii) it preserves smooth feature changes inside single shot. Authors of the paper suggest to use empirically determined share of key frames around 2-5%. It is important to mention that for contemporary videos like sitcoms and action movies shot length is usually less then 5 seconds. For such kind of videos we will have $[1-5]s*30fps*0.05 = [1.5-7.5]frames$ for a shot at most, which is close to edge frame or middle frame selection approaches.

As for shot clustering, authors of [15] discussed two widely used approaches: graph-based method [12] and overlapping links method [13]. Methods illustration presented at figure 1.

Graph-based method works as follows. Shots are clustered by similarity to represent vertices of the graph. Edges represent transitions between shot clusters. Scene borders are defined as *bridges* in this graph. *Overlapping links* method works as follows. Shots are arranged in a list. Shot link represent similarity between shots. Such link overlaps a set of shot transition (at least one). If shot transition is not overlapped by any link, then it is a scene border. It can be shown, that overlapping links method is equivalent to graph-based method. We will use this proven fact to select computationally more efficient algorithm.

Theorem 1. *Overlapping links method is equivalent to graph scene detection method. Formally, given a connected undirected graph $G = (V, E)$ representing video, where V is a set of shot clusters and E is a set of transitions between these clusters, in such a graph bridge is equivalent to shot* ***transition*** *$T_a = (s_a, s_{a+1})$ which is not* ***overlapped*** *by any* ***link*** *$L_{ij} = (s_i, s_j)$ such that $i \leq a$ and $j > a$.*

Proof:

Cluster set V is a partitioning P of shot set: $V = [\{s_n\}]_P$.

Link $L_{ab} = (s_a, s_b)$ defines a circuit C_{ab} or a loop that starts in $V_x = [s_a]_P$ and ends in the same cluster $V_x = [s_b]_P$.

Overlapping of links $L_{ab} = (s_a, s_b)$ and $L_{cd} = (s_c, s_d)$ is sharing non-trivial chain $(V_j, ...V_k)$ between circuits corresponding to links. If link L_{ab} overlaps transition T_c then $T_c \in C_{ab}$.

Statement 1. If in graph G every transition T_a is overlapped by some link L_{ij}, this graph doesn't have bridges.

Overlapping of two links corresponds to the fact that two circuits share a chain. Union of two intersecting circuits is a circuit. By induction we can show that set of mutually overlapping links $\{L_{ij}, L_{kl}\}$ produce a circuit on G, which is Eulerian circuit (contains all transitions). Circuit is at least 2-edge connected graph. Thus, G doesn't have bridges.

Statement 2. If T_a is not overlapped by any link $L_{ij}, (i \leq \; a < j)$, then edge $E_a = ([s_a]_P, [s_{a+1}]_P)$ is a bridge.

By statement, there is no link L_{ij}, in other words none of the shot pairs (s_i, s_j) belong to the same equivalence class if $i \leq a < j$. Thus T_a creates a bi-partitioning Q of V into sets of clusters $[V]_Q = \{W_1, W_2\}$ such that $\cup W_1 = \{s_1, ...s_a\}$ and $\cup W_2 = \{s_{a+1}, ...s_n\}$. These partitions are connected by edge E_a.

By contradiction suppose E_a is not a bridge: there exists $E_q = ([s_q]_P, [s_{q+1}]_P)$ such that cluster $V_i = [s_q]_P \in W_2$ and $V_j = [s_{q+1}]_P \in W_1$ and $q > a$. As cluster $[s_{q+1}]_P$ belongs to W_1, there already exists at least one shot s_r (with maximum r index) in $V_j = [s_{q+1}]_P$ such that $r \leq a$. Accordingly there exists $L_{r(q+1)} = (s_r, s_{q+1})$, where $r \leq a < q + 1$. *Contradiction.* Similarly proven for $q < a$.

Combining statements 1 and 2 we show that non-overlapped transition is equivalent to a bridge and they both represent scene transition. ■

Overlapping links method is a stage of linear chain decomposition method [23]. Although Tarjan algorithm for bridge detection [24] also has linear time complexity, for the task of scene clustering we can avoid building a graph, that significantly improves method performance. For our work we selected overlapping links method. In *Results* section IV we discuss optimal parameter selection for this method.

III. Methodology

In this paper we propose novel approach for video structure mining, which involves semantics obtained with deep image classification network. An overview of the proposed approach is shown in figure 2. Presented pipeline is composed of feature estimation, feature filtering, and shot and scene clustering.

For feature estimation, we use Places205-AlexNet image classification network as a feature source for scene detection task. AlexNet classification network is ILSVRC-winning topology that can be quickly trained compared to other deep networks [8]. Reasons for using Places205-trained classifier are discussed in subsection III-A.

We introduced feature filtering to reduce impulse noise. Our tests showed that feature value sequences can have sudden outbreaks of 1-2 frames caused by camera aberrations, sudden scene changes or other factors. Having such outbreaks and following gradient-based shot detection approach we increase shot oversegmentation. Oversegmentation does not significantly lower accuracy, but it affects run time for scene detection algorithms. Thus we added the edge-preserving impulse filtering to our pipeline.

For scene recognition we implemented *overlapping links* method. In section II we showed that it is equivalent to graph-based method, and faster.

In this section we provide further details on the feature estimation, filtering and scene recognition stages. Also we

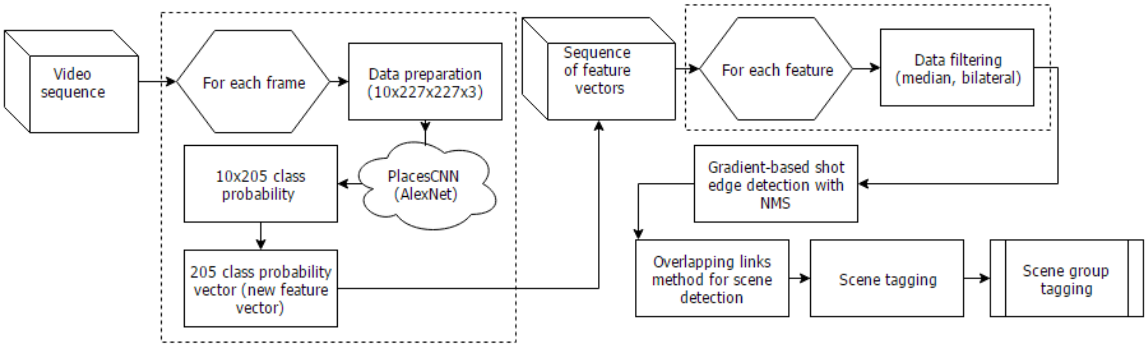

Video sequence
For each frame
Data preparation (10x227x227x3)
PlacesCNN (AlexNet)
10x205 class probability
205 class probability vector (new feature vector)
Sequence of feature vectors
For each feature
Data filtering (median, bilateral)
Gradient-based shot edge detection with NMS
Overlapping links method for scene detection
Scene tagging
Scene group tagging

Figure 2. Video scene estimation and tagging pipeline

discuss accuracy estimation approaches in the end of the section.

A. Feature estimation

According to the Film Encyclopedia [25] scene is a section of a motion picture with unified time and space. Researchers treat *unified time* as continuous frame sequence. For *unified space* we propose idea of fixed surrounding. Thus, we think that understanding *image background* is a good way to describe a place where scene is filmed. Therefore, for this task we proposed to use a network trained with a diverse set of scene types. Places205 dataset consists for 205 classes of different location types. Places205-AlexNet CNN [4] is Caffe-based AlexNet CNN trained on the 205 scene categories of the Places Database with 2.5 million images. Pre-trained AlexNet network for Places205 database was obtained from MIT Computer Science and Artificial Intelligence Laboratory's website [1] [4]. We chose this solution because it provides large variety of classes that can be utilized for hierarchical classification. Also, this network returns fuzzy class vector that brings a lot of implicit information rather than assigned discrete classes. AlexNet CNN architecture takes an input of 10 color image patches of size $227px * 227px$. These patches should be four corners and center of the image in straight and vertically flipped version. We used image pyramid [26] to select best image scale where patches cover all image area.

AlexNet output is 10 fuzziness vectors with 205 values each – one vector for each input image patch. We used median value selection for the resulting probability. Maximum or average selection method can be also used, but they potentially increase impulse noise. At the end of feature estimation stage, we have a 205-dimensional vector of raw features that is passed to filtering step.

B. Shot detection and feature filtering

We detect shot edges using feature values deviation in adjacent frames. Norm of feature time-derivative $\|dF/dt\|$ is used to estimate feature F deviation as proposed in [22]. If this value exceeds threshold Th_n, we detect shot change.

Our experiments showed that deviation amplitude significantly changes from video to video. On the other hand, within single video derivative norm values fit normal distribution. We use this fact to model threshold parameter with a single value N_σ. In our work we approximate expected value with average value of deviation. Thus, we define threshold as

$$Th_n = \overline{\left\| \frac{dF}{dt} \right\|} + \sigma N_\sigma, \tag{1}$$

where σ is standard deviation of $\|dF/dt\|$. For edge thinning we used non-maximum suppression (NMS) method [27].

We discovered, that impulse noise can lead to false shot edge detection. Since feature vector has 205 dimensions, joint value fluctuations can lead to significant shot oversegmentation. To reduce this effect we proposed data preprocessing before starting edge detection.

We tested two local edge-preserving filters for one-dimensional data. Firstly, we applied median filter with odd window size. Median filter is tuned with a single window W parameter. Secondly, to smooth the data within the shots, we applied 1-dimensional bilateral filtering [28]. The algorithm is a combinations of two Gaussian filters, thus it is managed with two parameters (σ_g, σ_f): σ_g tunes time window of interest (more distant in time the data, less the influence, it behaves like normal Gaussian filter); σ_f tunes value range of interest – more the difference, less the impact (it allows to preserve shot edges, significant value changes are preserved).

C. Scene recognition

Next step of our pipeline is to cluster shots into scenes. In [12] authors proposed to estimate shot similarity as minimal distance between frames of two shots. We also applied this approach, as it allows to build larger equivalence classes and reduce oversegmentation. Representing shots with smaller number of frames reduces computational costs. In the same work authors presented key frames selection method, that maximizes feature value range among key frames. They showed that for good scene recognition it is enough to keep only 2-5% of frames. We applied 3% threshold for our algorithm. It is important to mention that *Yeung et al.* [12] reported that

[1] http://places.csail.mit.edu/downloadCNN.html

”given inaccurate segmentation, they preferred oversegmentation rather than under-segmentation”. We would discuss this statement in section V.

D. Accuracy estimation

In survey [15] authors aggregated metrics used by research teams for scene detection and annotation accuracy estimation. In our work we estimated the quality of both scene detection and annotation.

For scene detection we used *coverage-overflow* [29] and *purity* [30] metrics. *Coverage* indicates to what extend frames are grouped together correctly to make scenes. A value of one indicates full coverage. *Overflow* expresses the amount of frames that are wrongly grouped to scenes. A value of zero indicates no overflow. *Purity* expresses intersection between a ground truth segmentation and an automatic segmentation. A value of one indicates maximum purity.

For scene annotation we used *differential edit distance* (DED) [31] and *per-frame* method proposed in [6]. *DED* is a minimum share of shots for which scene labels must be changed in order to match the ground truth labels. *Per-frame* metric is a ratio between correctly labeled frames and all frames. Scene annotation accuracy was estimated using both Top-1 (exact match) and Top-5 (ground truth label is within top-5 predicted labels) methods.

We also implemented tag grouping. As it is stated in [6], we can increase classification accuracy by grouping tags. This approach works because false detects more frequently occur inside a group of similar tags. For this we proposed tag hierarchy of Places205 tags. Our grouping and other materials are available online at http://sprotasov.ru/cv/. To avoid multiple group detection, we proposed strong threshold of 60% group probability.

In our experiments we performed exhaustive grid search for three parameters to achieve the best scene detection and annotation accuracy. These parameters are: N_σ – shot detection threshold defined in formula (1), D_{fwd} – shots inspected forward for *overlapping links* method, and ϵ – between-shot distance threshold for shot similarity match. In the paper we refer to these parameters as the tuple (N_σ, D_{fwd}, ϵ). Major pipeline steps are presented in algorithm 0[2]. We also did another run of grid search for selecting the best filtering strategy.

IV. Results

We tested our method on synthetic dataset consisting of MPEG-7 dataset and our own video. For testing we improved MPEG-7 markup by assigning Places205 tags to the shots. We considered only outputs of shot detection with E_{shot} accuracy at least 0.9 to reduce the parameter search space.

Our solution is implemented using python 2.7. The source code is available at http://sprotasov.ru/cv/. The network is implemented using Caffe framework by Berkeley Vision and Learning Center http://caffe.berkeleyvision.org/ [32]. We used Linux Mint x64 virtual machine with 4GB RAM. As our research is devoted to hypothesis testing, we concentrate on analysis of the best achieved results.

Algorithm 1 Scene detection and annotation pipeline

```
function $ShotDetector(features)$
    formula (1)
end function
function $Filter(frames)$
    $MedianFilter(frames, 5)$
end function
function $KeyFrames(shots)$
    $ChainMethod(shots, 0.03)$
end function
function $Annot(tags, groupTags, features, scenes)$
    function selects top tags with maximum probability within scene
end function
for $frame$ in $frames$ do
    $F[frame] \leftarrow Places205\_AlexNet.predict(frame)$
end for
for all ($N_\sigma =$ -0.2..1, $D_{fwd} = 1..\infty$, $\epsilon =$ 1e-4..1e-5) do
    $F_{filtered} \leftarrow filter(F)$
    $shots \leftarrow ShotDetector(F_{filtered}, N_\sigma)$
    for $shot$ in $shots$ do
        $keyFrames[shot] = KeyFrames(shot, \epsilon)$
    end for
    $scenes \leftarrow OverlappingLinks(keyFrames, D_{fwd}, \epsilon)$
    $labels \leftarrow Annot(PLACES205\_TAGS, F, scenes)$
    $save(labels, scenes)$
    compute metrics for $scenes$ and $shots$
end for
```

Table I
Oversegmentation with different filters for top achieved accuracies for 30fps dataset

Filter	**Parameters**	**Edge det. accuracy**	**Oversegm.**
No filter	–	0.929	10.571
Median	$W = 3$	**0.982**	13.5
Median	$W = 5$	0.964	**8.75**
Bilateral	$\sigma_g = 1, \sigma_f = 0.1$	0.964	12.5
Bilateral	$\sigma_g = 1, \sigma_f = 0.3$	0.964	13.732

A. Feature filtering

We performed grid search on both median and bilateral filter parameters and came up with the results. Experiments showed that for *30fps* 3-frames-sized median filter gives the best edge detection accuracy, whereas 5-frame-sized window for minimizes oversegmentation. We did not find significant positive influence of bilateral filter parameters on oversegmentation and edge detection accuracy. Results are shown in the table I.

B. Achieved results

Scene detection. We achieved the best *purity* value of 0.755 with parameter values $(0.9, 17, 0.0001)$. In the results of [30] that values of purity were clustered around 0.75 and did not

[2] Source code is available by the link: https://bitbucket.org/compvisioniu/human-activity-recognition/src

Table II
PARAMETERS AND METRICS SIGNIFICANT CORRELATIONS

name	D_{fwd}	E_{scene}	coverage	overflow	DED (Top-5)	$MITG_1$
DED (Top-5)	-0.292	0.654	**-0.830**	-0.253	-	0.380
overflow	**0.878**	-0.592	0.534	-	-0.253	-0.631
coverage	0.509	**-0.845**	-	0.534	**-0.830**	-0.644
E_{scene}	-0.510	-	**-0.845**	-0.592	0.654	**0.819**
D_{fwd}	-	-0.510	0.509	**0.878**	-0.292	-0.545

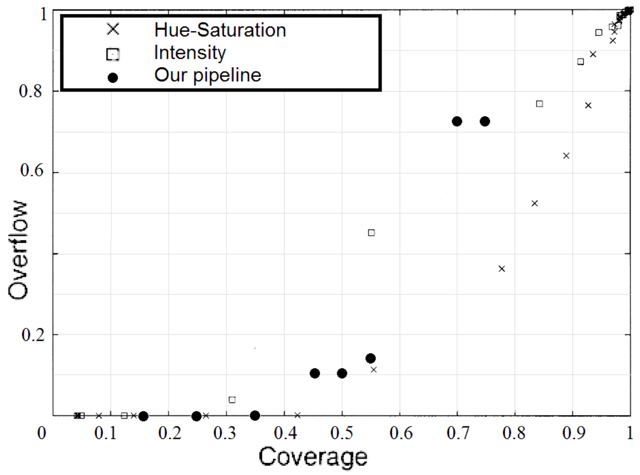


Figure 3. Comparison of coverage and overflow values with original paper

vary significantly while changing the parameters. This makes the usage of purity for accuracy estimation questionable.

The authors of [29] introduced *coverage* and *overflow* to check the extent to which scenes are different and shots within one scene are similar, respectively. They evaluated different shot segmentation approaches with proposed metrics. Comparison of our results and results provided in paper is represented on figure 3. For the hue-saturation histogram feature space, coverage-overflow curve is slightly better than our results. For the intensity histogram feature space our results are better for smaller coverage. Since these metrics are used by authors for feature testing, the results obtained could indicate that our feature space is not perfect. We assume that our features have redundant dimensionality.

For DED (closer to 0 is better) we achieved 0.732 and 0.625 with parameter values $(-0.2, 1, *)$ for Top-1 and Top-5 respectively. In the original paper [31] authors achieved 0.16 DED score which is much better than our results. Possible reasons for this are discussed in section V.

Scene annotation. As mentioned earlier, for scene annotation accuracy estimation, we used per-frame accuracy of tagging proposed in [6]. The best results achieved for this metric, along with their respective parameter values and methods, are as follows:

- 0.662 with (0.0, 4, 0.0001) - Top-1
- 0.915 with (0.9, 17, 0.00007) - Top-5
- 0.600 with (-0.2, 8, 0.0001) - Top-1 (groups)
- 0.927 with (0.4, 13, 0.0001) - Top-5 (groups)

Compared with original results our method performs significantly better for both basic and group classification.

C. Metrics correlation

Experiments showed that some of the selected metrics depend on initial or internal algorithm parameters. To test this statement we introduced two additional simple metrics. *Scene/shot edge detection accuracy* is used to check if we accurately detected all scene/shot borders from ground truth. Values closer to one are better. *Scene/shot oversegmentation* shows how many redundant scene/shot edges we generated in addition to correctly detected. Again, values closer to one are better.

- scene edge detection accuracy

$$E_{scene} = \frac{||scene\_edges\_detected \cap scene\_edges||}{||scene\_edges||} \tag{2}$$

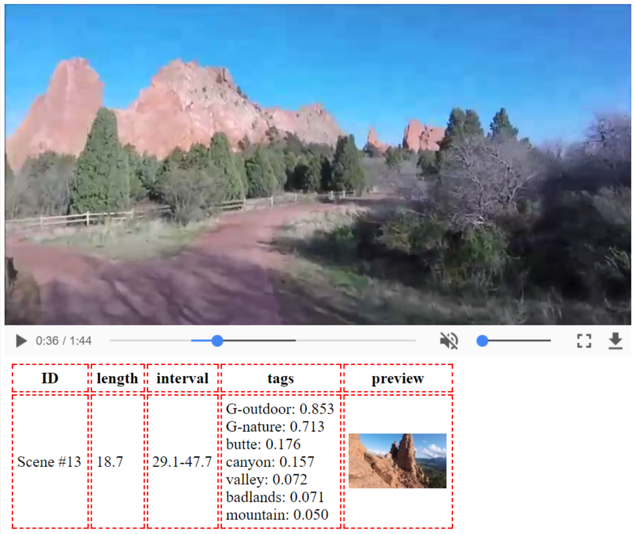
- oversegmentation

$$OVS_{scene} = \frac{||scene\_edges\_detected||}{||scene\_edges||} \tag{3}$$

We estimated Pearson product-moment correlation between different metrics and parameters. Major results are presented in the table II. We can divide values into 3 groups:

- *adjustable parameters*: N_σ, shots forward chains length D_{fwd}, and between-shot distance ϵ.
- *shot detection metrics*: shot edge detection accuracy E_{shot}, and shot oversegmentation OVS_{shot}.
- *quality metrics*: E_{scene}, OVS_{scene}, *purity*, *coverage*, *overflow*, *DED* (Top-1, Top-5), per-frame MIT_1 and MIT_5 for Top-1 and Top-5, and similar values $MITG_1$ and $MITG_5$ for class groups.

Among adjustable metrics we found strong positive correlation between shot-forward chain length D_{fwd} and $overflow$ metric. That means the longer tail we inspect, the worse is the overflow metric. This can be explained by false connections between remote shots or by weak metric design. Also we detected obvious influence of N_σ on OVS_{shot} (formula 3) parameter, where the latter itself does not affect any quality metric individually, and can be considered only together with other metrics.



ID	length	interval	tags	preview
Scene #13	18.7	29.1-47.7	G-outdoor: 0.853 G-nature: 0.713 butte: 0.176 canyon: 0.157 valley: 0.072 badlands: 0.071 mountain: 0.050	

Figure 4. Algorithm output

V. Discussion

We will discuss results of our work from two points of view. Firstly we analyze existing and proposed metrics. We discuss how expressive are the quality metrics. Second part presents our findings: best algorithm parameters, performance and improvements.

A. Analysis of metrics

In [12] authors wrote that given inaccurate segmentation, they preferred oversegmentation rather than undersegmentation. We showed in the previous section, there is no significant correlation between oversegmentation and output quality metrics. On the other hand, oversegmentation leads to growth of complexity for the later stages, for example overlapping links method. We state that oversegmentation does not significantly affect the accuracy, but consumes more CPU time at succeeding stages. To reduce performance overhead we recommend to have this parameter closer to 1.

In [31] authors achieved the best DED value of 0.16, whereas our best result is only 0.45. One of the reasons for this is that DED metric strongly correlates with edge detection accuracy, when other metrics do not. In our solution this is significant, because Places205 features are invariant to scale, angle and camera disposition, that leads to missing of shot edges. Other problem with DED metric is the assumption that ground truth and detected shots match in time. For most approaches this statement is false. We also found strong correlation between DED and the coverage. We state that both these metrics are not very suitable for scene recognition algorithms that can potentially produce different shot splitting.

B. Analysis of the proposed metrics

We found strong correlation between the following metrics:

- E_{shot}, $coverage$ and DED
- E_{scene} (formula 2), $MITG$ and $coverage$.

Correlation shows that both DED and $coverage$ strongly depend on how accurately we detect shot transitions: by construction both metrics assume that detected shot edges coincide with ground truth. This make these metrics hardly applicable to methods with different shot definitions and feature spaces. For example, deep features will not react to camera motion, zoom or illumination changes. Also correlation between $MITG$ and E_{scene} can be explained by coarse-grained algorithm output. $MITG$ relies on per-frame matches and if we mistakenly merge two scenes, we can fail all matches for both scenes. We think this metric should be applied only for non-structured annotations.

Output of our solution is a list of scenes annotated with tags. Tags are taken from Places205 dataset and from proposed group tags set. Typical output is presented in the image 4.

C. Findings

Proposed pipeline allows to achieve state-of-the-art accuracy. Thus, we state that output of Places205 network provide sufficient features for scene detection. For most of the results we used parameters $N_\sigma = 0.9$ and $\epsilon = 0.0001$.

Our test setup was implemented in Python and was executed on a single 1.7 GHz CPU core, showing $2 fps$ processing speed. Thus, it leaves the room for further speed improvement. It is worth mentioning that all algorithms in the pipeline (frame processing, feature filtering, shot detection, overlapping links) are *local* algorithms. This means they can be run in parallel with a lag in both shared and distributed memory systems. Lag size depends on filter parameters and is mostly influenced by D_{fwd} (shots inspected forward) parameter of the overlapping links algorithm.

We also introduced two quality metrics: OVS_{scene} and E_{scene}. We found that *coverage* quality metric correlates with edge detection accuracy, whereas there is no exact correlation between oversegmentation and other metrics. Finally, the feature vector produced by Places205 is high-dimensional, so in future we will try dimensionality reduction approaches. We hope that this would increase both the speed and accuracy of our solution.

VI. Conclusion

In this paper we proposed an approach to semantic video indexing. We implemented a pipeline for video structure mining and annotation, which consists of feature extraction and filtering, shot clustering and labeling stages. Deep convolutional network output was used as feature source for scene detection. We used the same output data for scene annotation. Then we evaluated our pipeline quality using metrics for both scene detection and annotation. We obtained results that are similar to or exceed results achieved in original papers. Thus, we can state that features obtained from deep network not only can be integrated into classical video processing pipelines, but also provide additional semantic information. We assume that our work result can be applied in video search engines and other video semantic mining tasks.

We also studied different metrics used for video segmentation and annotation quality estimation, and proposed our own metrics.

In future we plan to improve our pipeline carefully working with features: using other deep networks as well as dimensionality reduction techniques. We are also going to extend our pipeline with object and human detection algorithms to enrich semantical information.

References

[1] A. D. Bagdanov, M. Bertini, A. D. Bimbo, G. Serra, and C. Torniai, "Semantic annotation and retrieval of video events using multimedia ontologies," in *International Conference on Semantic Computing (ICSC)*, Sept 2007, pp. 713–720.

[2] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Queenot, and R. Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[3] X. Amatriain and D. Agarwal, "Tutorial: Lessons learned from building real-life recommender systems," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 433–433. [Online]. Available: http://doi.acm.org/10.1145/2959100.2959194

[4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 487–495. [Online]. Available: http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[6] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ser. ICCV '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 273–. [Online]. Available: http://dl.acm.org/citation.cfm?id=946247.946665

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei, "Construction and Analysis of a Large Scale Image Ontology." Vision Sciences Society, 2009.

[10] K. Chatfield, R. Arandjelović, O. M. Parkhi, and A. Zisserman, "On-the-fly learning for visual search of large-scale image and video datasets," *International Journal of Multimedia Information Retrieval*, 2015.

[11] J. Johnson, A. Karpathy, and F. Li, "Densecap: Fully convolutional localization networks for dense captioning," *CoRR*, vol. abs/1511.07571, 2015. [Online]. Available: http://arxiv.org/abs/1511.07571

[12] M. Yeung, B.-L. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Comput. Vis. Image Underst.*, vol. 71, no. 1, pp. 94–109, Jul. 1998. [Online]. Available: http://dx.doi.org/10.1006/cviu.1997.0628

[13] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, Jun 1999.

[14] A. Altadmri and A. Ahmed, "Automatic semantic video annotation in wide domain videos based on similarity and commonsense knowledgebases," in *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*, Nov 2009, pp. 74–79.

[15] M. Del Fabro and L. Böszörmenyi, "State-of-the-art and future challenges in video scene detection: a survey," vol. 19, no. 5, pp. 427–454, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00530-013-0306-4

[16] B. T. Truong, S. Venkatesh, and C. Dorai, "Scene extraction in motion pictures," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 13, no. 1, pp. 5–15, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2002.808084

[17] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot, *Spectral Structuring of Home Videos*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 310–320. [Online]. Available: http://dx.doi.org/10.1007/3-540-45113-7_31

[18] A. Aner and J. R. Kender, *Video Summaries through Mosaic-Based Shot and Scene Clustering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 388–402. [Online]. Available: http://dx.doi.org/10.1007/3-540-47979-1_26

[19] Y.-M. Kwon, C.-J. Song, and I.-J. Kim, "A new approach for high level video structuring," in *IEEE International Conference on Multimedia and Expo*, 2000.

[20] D. Mitrović, S. Hartlieb, M. Zeppelzauer, and M. Zaharieva, *Scene Segmentation in Artistic Archive Documentaries*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 400–410. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16607-5_27

[21] L. Huayong and Z. Hui, *The Segmentation of News Video into Story Units*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 870–875. [Online]. Available: http://dx.doi.org/10.1007/11563952_95

[22] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, Feb. 2007. [Online]. Available: http://doi.acm.org/10.1145/1198302.1198305

[23] J. M. Schmidt, "A simple test on 2-vertex- and 2-edge-connectivity," *Information Processing Letters*, vol. 113, no. 7, pp. 241 – 244, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020019013000288

[24] R. Tarjan, "A note on finding the bridges of a graph," *Information Processing Letters*, vol. 2, no. 6, pp. 160 – 161, 1974. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020019074900039

[25] E. Katz, *The Film Encyclopedia: Third Edition*. HarperCollins, 1998. [Online]. Available: https://books.google.ru/books?id=jhx0QgAACAAJ

[26] P. J. Burt, "Fast filter transform for image processing," *Computer graphics and image processing*, 1981. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/0146664X81900927

[27] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[28] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: http://dl.acm.org/citation.cfm?id=938978.939190

[29] J. Vendrig and M. Worring, "Systematic evaluation of logical story unit segmentation," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 492–499, Dec 2002.

[30] A. Vinciarelli and S. Favre, "Broadcast news story segmentation using social network analysis and hidden markov models," in *Proceedings of the 15th ACM International Conference on Multimedia*, ser. MM '07. New York, NY, USA: ACM, 2007, pp. 261–264. [Online]. Available: http://doi.acm.org/10.1145/1291233.1291287

[31] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, and J. Kittler, "Differential edit distance: A metric for scene segmentation evaluation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 6, pp. 904–914, June 2012.

[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.