Binary convolutional neural network features off-the-shelf for image to video linking in endoscopic multimedia databases

Stefan Petscharnig¹ · Klaus Schöffmann¹

Received: 29 May 2017 / Revised: 5 April 2018 / Accepted: 16 April 2018 /

Published online: 8 May 2018

© The Author(s) 2018

Abstract With a rigorous long-term archival of endoscopic surgeries, vast amounts of video and image data accumulate. Surgeons are not able to spend their valuable time to manually search within endoscopic multimedia databases (EMDBs) or manually maintain links to interesting sections in order to quickly retrieve relevant surgery sections. Enabling the surgeons to quickly access the relevant surgery scenes, we utilize the fact that surgeons record external images additionally to the surgery video and aim to link them to the appropriate video sequence in the EMDb using a query-by-example approach. We propose binary Convolutional Neural Network (CNN) features off-the-shelf and compare them to several baselines: pixel-based comparison (PSNR), image structure comparison (SSIM), hand-crafted global features (CEDD and feature signatures), as well as CNN baselines Histograms of Class Confidences (HoCC) and Neural Codes (NC). For evaluation, we use 5.5 h of endoscopic video material and 69 query images selected by medical experts and compare the performance of the aforementioned image matching methods in terms of video hit rate and distance to the true playback time stamp (PTS) for correct video predictions. Our evaluation shows that binary CNN features are compact, yet powerful image descriptors for retrieval in the endoscopic imaging domain. They are able to maintain state-of-the-art performance, while providing the benefit of low storage space requirements and hence provide the best compromise.

Keywords Content-based video retrieval · Endoscopic multimedia · CNNs

✉ Stefan Petscharnig
stefan.petscharnig@itec.aau.at

Klaus Schöffmann
ks@itec.aau.at

¹ Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt am Wörthersee, Austria

1 Introduction

Minimally invasive surgery (MIS) methods benefit the patients' well-being, as they aim at reducing wound healing time, associated pain and risk of infections. These methods have been enabled by the advance of several medical and technical technologies, such as electro-surgery, precision instruments, and imaging technology. Via a small incision, a camera is introduced in the human body enabling surgeons to perform the MIS. During the MIS the surgeon is able to record video as well as image data. The recorded data can later be used for documentation purposes, medical research, education of young surgeons, and for the improvement of surgery techniques. Apparently, huge amounts of data are produced in MIS contexts. Considering full surgeries lasting for hours of routine work and only minutes of medically relevant events [22], surgeons do not have the resources to cope with these huge amounts of data. For documentation purposes, surgeons use recorded images - they thus play a central role for quick assessment of a medical case. There are cases such as education of young surgeons, when a still image does not contain a sufficient amount of information, as the temporal context cannot be mapped on a still image. In all of these cases, the surgeons need to reflect on a video sequence.

The main problem in getting these sequences is that manual browsing in such endoscopic multimedia databases (EMDBs) is a tedious and time-consuming task. We assume that the automatic creation of bookmarks - linking the externally captured images to the correct videos and playback time stamps (PTS) - saves the surgeons' time at tasks such as education or documentation, thus enabling them to spend more time on tasks such as medical research. It is important to note that in practice there is no way for a trivial interlinking of captured images and recorded videos (as also described in related work [3, 7, 27]). This problem originates from the use of different systems with distinct encoders. Images and video frames suffer from different encoding errors. Moreover, image and video encoding run on different systems with diverging time stamps and clock synchronization is practically hardly feasible. Hence, current systems do not provide the functionality to automatically link the external images to the appropriate video positions, in order to be able to perform a navigation in the video using images of interest. On the other hand, surgeons are not able to manually edit and maintain image to video links themselves, as this task is a tedious process wasting the already short time resources: One full surgery alone can easily last for four hours and more. So, if we consider a surgeon performing a single surgery per day, there accumulate 1000 hours of video material per year and surgeon.

An example use-case is illustrated in Fig. 1. Assume a surgeon wants to revisit a particular scene of a surgery for presentation at a medical congress. Currently, the surgeon has to search for the specific video scene manually within the EMDB. In order to do this manually, this would include a two-step procedure: (1) identifying the correct video and (2) navigating to the desired playback time. Together with the considerations above, this example illustrates that there is a need for an automatic way to retrieve the correct video and PTS. We consider this as a query-by-example problem in the field of image/video retrieval and evaluate several image descriptors for their performance in terms of both, retrieval efficiency computational resource requirements. As already mentioned above, images showing the most important information about particular important scenes are available for documentation purposes. We aim at retrieving the desired information from such a representational image, i.e. we query the EMDB by an image in order to retrieve a video id and PTS. From a practical point of view, our work is driven by the following problems:

P1 Given a query image, what is the probability to retrieve the correct video?

Fig. 1 Linking externally recorded endoscopic images to videos and video playback time stamps within an endoscopic multimedia database. The problem is to assign a video id and playback time stamp to a query image which is assumed to exist as near-duplicate as a video frame

P2 Given the video is retrieved correctly, how close is the predicted PTS to the true PTS?

In order to cope with these problems, we propose a binarization scheme for Convolutional Neural Network (CNN) features off-the-shelf reducing the storage space cost while still maintaining a high predictive performance. As baselines for comparison, we evaluate various approaches which are inspired from different points of view. From an *encoding perspective*, we exploit the fact that image and video are encoded from the same raw image stream. We assume that query image and the video frame of interest only differ by (a) compression loss, or (b) the difference in time between an image and two video frames is that small that there is no significant visual difference. Hence, we use the full-reference visual quality metrics PSNR as distance metric from the query image to each video frame. The same considerations lead to the use of SSIM [35] as similarity measure for the video. From a *visual similarity perspective*, we model the visual similarity between image and video frames. Using image descriptors, we are able to ignore compression artifacts and map the video frames and query images to a feature space. Of course, the main advantage of such approaches is the computational efficiency: we have to calculate the descriptors for the EMDB only once, followed by a computational cheap distance calculation - while on the other hand, we have to calculate PSNR or SSIM against every frame in the EMDB. As image descriptors, we evaluate CEDD [9] and feature signatures, since these descriptors have shown good performance in related work [3, 7, 27]. Recent literature also shows that CNN features provide an astounding baseline for visual recognition [26]. We include features from the top layers of the network, Neural Codes (NC) [2] as well as Histograms of Class Confidences (HoCC) as baseline for our comparative evaluation.

Furthermore, we assume a very practical situation where videos are prepared for adaptive streaming (c.f. [30]). We therefore split the videos of the EMDB into segments of 1s, 2s, 4s, 8s, and 16s duration. The aforementioned approaches are evaluated using only the first frame of these segments acting as a representative frame for this segment against a dense sampling approach. In order to answer P1, we evaluate the approaches on a video-level as a retrieval problem with a single correct video and measure video hit rate, i.e. how often the prediction was correct. We answer P2 on a frame-level basis, by measuring the distance between the predicted PTS and the true PTS if the video was predicted correctly and calculate an average temporal deviation. This work is novel, as to the best of our knowledge, we are the first to provide such a statistically motivated binarization scheme for CNN features off-the-shelf. Moreover, this work extends the work of Carlos et al. [7], Beecks et al. [3], and

Schöffmann et al. [27] which also tackle this problem for the domain of endoscopic videos. In contrast to these works, we do not only evaluate whether the video was hit, but we also the correct frame position. Therefore, we have to use a different dataset. Furthermore, our evaluation is more rigorous, as we also thoroughly investigate dense sampling (considering each video frame) against sparse sampling (considering 1 frame per time interval).

The remainder of the paper is structured as follows: Our work continues with Section 2 giving an overview on related work. In Section 3, we propose our approach of binary CNN features off-the-shelf. Section 4 deals with the methods used in this paper, while the evaluation of these methods is described in Section 5. We conclude and point at future work in Section 6.

2 Related work

Video hyper-linking is the more general task to our studied problem. It was already topic to TRECVID [1] and MediaEval Challenges [12]. The main difference to video hyper-linking is essential, as the problem differs in input and desired output dramatically. Instead of a video sequence, the query is an image known to exist as a near-duplicate in a video database as a frame of a video. Moreover, the desired output is a video id and PTS instead of multiple similar video sequences. There is only one correct answer to this retrieval problem (the video sequence the image was shot in). Moreover, our use case is delimited to use visual media (image and video) only, whereas hyper-linking approaches often are able to fall back on other features such as text or audio, e.g. Cheng et al. [10] study the performance of video hyper-linking methods using multiple input modalities including subtitle, meta data, audio, visual features, and their combinations. Other state-of-the-art approaches in hyper-linking are multi-modal and cross-modal approaches based on machine learning: bimodal LDA, deep auto-encoders, bidirectional deep neural networks, and generative adversarial networks. Deep auto-encoders (as proposed by Vukotic et al. [32]) and bimodal LDA approaches (as proposed by Simon et al. [29]) for video hyper-linking are evaluated by Bois et al. [4] with respect to relevance and diversity. The baseline for this evaluation uses a bag-of-words representation for each segment with tf-idf-weighting. Vukotic et al. [33] approach the problem of video hyper-linking with bidirectional deep neural networks. The same authors propose in their further work [34] to use generative adversarial networks. All these learning based state-of-the-art approaches utilize multi-modal or cross-modal input data, e.g. they make use of video subtitles or automatic audio transcriptions. In contrast to these use cases, in the domain of endoscopic video no such additional input modalities are available. Galuščáková et al. [13] investigate visual descriptors for the task of video hyper-linking within a multi-modal approach, i.e. they use visual (feature signatures, AlexNet fc7 CNN features, concept detection, and face recognition) and text-based (subtitles and automatic transcripts) input modalities.

The problem of image to video linking for endoscopic surgeries (strictly speaking: retrieval of a video from an endoscopic video database based on a query image) has already been topic to recent research [3, 7, 27] which are the works most related to our research. In Carlos et al. [7], retrieval based on global features (Color and Edge Directivity Descriptor CEDD [9], auto color correlogram [16], and pyramid histogram of oriented gradients PHOG [5]) fusion methods by rank and score as well as a localized version of CEDD using the SIMPLE approach [17] on every fifth video frame is evaluated. They report that they were able to correctly map 78.3, 78.5, and 79.8% of the queries to the correct video sequence using the approaches Sum of Ranks, Sum of Scores, and SIMPLE-CEDD respectively. The

work done by Beecks et al. [3] was done on the same dataset and also utilized a description for every fifth frame. The authors approached this problem by using feature signatures and also compared various signature matching distances to the baseline of Carlos et al. [7]. Their results imply that feature signatures using a unidirectional approach based on L1 distance improved the state-of-the-art by approximately 10%. Schöffmann et al. [27] improved on this work and investigated further distance measures including signature matching distance (SMD), signature quadratic form distance (SQFD), and the earth movers distance (EMD) over various signature sizes. Their results imply that on the used dataset, the correct video segment can be retrieved for more than 88% of the query images.

Ercoli et al. [11] propose a quantization process that is based on the distance of a feature vector to precomputed cluster centroids which are obtained by k-means. They evaluate the threshold measures as geometric mean, arithmetic mean, and n^{th} -nearest distance. Guo and Li [14] use a CNN for supervised learned hashing. They binarize the activation value of a specific fully connected layer with threshold 0. Opposed to this work (which is intended for a slightly different use-case), our approach does not require CNN training. We use a pre-trained CNN model for feature extraction. Moreover, we show that our approach is working well with very similar images, i.e. consecutive video frames. Last, we propose non-zero thresholds for binarization and analyze their impact on retrieval.

3 Binary CNN features off-the-shelf

CNN features off-the-shelf have an excellent record as simple, yet good working approach for content-based description of images. As they are real-valued and high-dimensional image descriptors, they have the drawback of consuming a huge amount of storage space. In recent literature, it was proposed that their dimensionality should be reduced using PCA [26]. The dimensionality reduction could also be achieved using auto-encoders (which has been investigated for global features by our research group [25]). In this work, we propose a method for binarization of CNN features off-the-shelf which was intentionally designed for the task of image to video linking. Instead of reducing the feature dimensionality, we aim at binarization of the feature values. We use basic statistical methods, in order to encode real-valued CNN features off-the-shelf into binary-valued feature vectors of the same dimension.

3.1 Problem formulation

As already mentioned above, the problem of using CNN descriptors off-the-shelf for image linking is the fact that this approach requires a huge amount of storage space. We aim at reducing this storage space requirements by quantization of real-valued CNN feature vectors into binary vectors. As an example, a single AlexNet FC7 descriptor can be represented by 4096 float values resulting in a descriptor of 16KB size. A binary version of the AlexNet FC7 descriptor thus could reduce the storage requirements to an equivalent of 128 integer values resulting in a descriptor size of 512Byte. Hence, our approach aims at solving the following research question:

RQ How can we binarize real-valued CNN features while still maintaining high retrieval performance?

We define a CNN feature $\mathbf{f}$ as an n -dimensional vector $\mathbf{f} \in \mathbb{R}^n$, where n denotes the number of feature components. A feature set $F \in \mathbb{R}^{k \times n}$ of k individual feature observations is

represented by a $k \times n$ matrix. Each row $F_{i,\cdot}$ represents an individual CNN feature observation and each column represents a feature component $F_{\cdot,i}$. Within the context of this work, we state that the problem of binarization of a feature vector is to encode each feature component individually depending on its observed value into a binary value. Hence, our research question aims at finding a pair of intervals I for each feature component mapping the observed value to such a binary value. Hence, the problem reduces to providing a mapping $q(f; I) \rightarrow \mathcal{Q}^n$ from a CNN feature f to a binarized feature with binary values $b_i \in \mathcal{Q}$ depending on parametrized intervals I . The approach for calculating these intervals is described in the following. Please note that for the purposes of our work, we set $\mathcal{Q} := \{0, 1\}$, as we consider a binarization scheme. We want to emphasize that one could use a general quantization scheme by using a set $\mathcal{Q}$ consisting of more than two different values. The benefit of choosing a binarization scheme over a quantization with an arbitrary number of values is that the resulting binary feature vectors can be stored in a very compact representation requiring n bit storage only where n denotes the number of dimensions of the real-valued CNN feature which has to be binarized. Moreover, these binary feature vectors can be compared efficiently using computationally inexpensive distance metrics such as hamming distance or jaccard distance which only require bit-wise operations and bit-counting operations.

3.2 Binarization for CNN features off-the-shelf

We assume that for each individual feature component, the observations $F_{i,\cdot}$ (for the i -th feature component) follow a random distribution drawn from an unknown population X_i . Furthermore, we assume that the observations for an individual feature component are identically and independently distributed. We do not impose additional assumptions for the relations between individual feature components. The idea behind the binarization process is to assign the real-valued feature observation to one of two categories. We say that an observation for a feature component is *higher* than the norm when its observed value exceeds the expected value for the respective feature component. Otherwise, the observation is *lower* than the norm. Based on an available feature set F and the law of large numbers, we estimate the expected value of the feature component j using the arithmetic mean $\mu_j = \frac{1}{k} \sum_{i=1}^k F_{i,j}$. Using the calculated intervals $(-\infty, \mu_j]$ and (μ_j, ∞) , we are able to transform feature f into a binary feature representation. We call a value μ_j which defines our two intervals of interest a binarization threshold.

Algorithm 1 shows the straightforward binarization algorithm for an EMDB. Given is a real-valued feature set F with k observations of n real-valued feature components. Output is a binary feature set F_Q consisting of k binary feature vectors of dimension n . Please note that if we do not want to restrict ourselves on a binary feature vector, we could use empirical quantiles in order to define the intervals for quantization. Furthermore, the algorithm results in transformation thresholds T which are used to transform further features (e.g. new video frames, or query images). For each feature component, we estimate the corresponding arithmetic mean μ_i . For each observation j , we now set the binary feature value accordingly to these mean values. If the feature value is smaller than or equal to the estimated arithmetic mean, we clear the bit for observation at position i . Otherwise, if the feature value is greater than the estimated arithmetic mean, we set the bit for observation at position i . Whenever new videos are added to the EMDB or additional query images have to be transformed, we use the calculated transformation thresholds T to repeat the binarization for these descriptors. The transformation thresholds remain static until the EMDB has grown significantly. In this case, we suggest applying Algorithm 1 to the new, much larger EMDB. Please note that

we use the same transformation parameters for images and videos, as they were captured from the same device and only differ in compression.

Algorithm 1 Binarization for CNN features off-the-shelf

Given:

▷ Feature set

$$\mathbf{F} := \begin{bmatrix} f_{1,1} & f_{1,2} & \dots & f_{1,n} \\ f_{2,1} & f_{2,2} & \dots & f_{2,n} \\ \dots & \dots & \dots & \dots \\ f_{k,1} & f_{k,2} & \dots & f_{k,n} \end{bmatrix} \text{ with } f_{i,j} \in \mathbb{R}$$

Output:

▷ Binarized feature set

$$\mathbf{F}_Q := \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \dots & \dots & \dots & \dots \\ b_{k,1} & b_{k,2} & \dots & b_{k,n} \end{bmatrix} \text{ with } b_{i,j} \in \{0, 1\}$$

▷ Transformation thresholds

$$\mathbf{T} := (\mu_1, \mu_2, \dots, \mu_n) \text{ with } \mu_i \in \mathbb{R}$$

```

1: for  $i \in \{1 \dots n\}$  do
2:    $\mu_i \leftarrow \frac{1}{k} \sum_{o=1}^k F_{o,i}$ 
3:   for  $j \in \{1 \dots k\}$  do
4:      $b_{j,i} \leftarrow \begin{cases} 1 & \text{if } f_{j,i} \leq \mu_i \\ 0 & \text{else} \end{cases}$ 
5:   end for
6: end for

```

Example Assume a feature set consisting of 2-dimensional features. For the feature components $F_{\cdot,1}$ and $F_{\cdot,2}$, we estimate the arithmetic means on a big multimedia database: $\mu_1 = .15$ and $\mu_2 = .25$. Assume, we now want to binarize the first feature observation $F_{1,\cdot} = (f_{1,1}, f_{1,2}) = (.14, .39)^T$. The result is a binary vector of length 2: $b := (b_1, b_2)^T$. The first feature component $f_{1,1}$ is smaller than μ_1 , thus we set b_1 to zero. The second feature component $f_{1,2}$ is greater than μ_2 , thus we set b_2 to one. The resulting binary feature vector thus is $Q(F_{1,\cdot}) = (0, 1)^T$.

3.3 Usage of other binarization thresholds

Please note that the arithmetic mean is not the only possible choice for binarization threshold. From our point of view, there are two more options, which may be considered: median-value and a constant value of zero. Before we delve deeper into these options, we first analyze CNN features off-the-shelf. As mentioned above, these off-the-shelf descriptors are real-valued vectors with n feature components. Moreover, we observe that the values are sparse, i.e. most values are zero. This originates mostly in the fact that we extract CNN-features after a ReLU layer (which we do throughout this manuscript), which has

non-negative output. In our opinion, this sparsity is the key that these features achieve great performance for retrieval. A proposed binarization scheme now should preserve this sparsity, in order to maintain as much retrieval performance as possible. Now, the logical conclusion is to set a constant threshold of 0 for each feature component. This means we set all values which are greater than zero to 1 and every other value to 0. This approach bases on the idea that whenever a component of an image's feature vector is greater than zero, it is highly descriptive for this image. However, there arises the drawback that small deviations from zero in a single feature component might instantly flip the bit of the binary feature vector at this component. While small deviations do not add much to a distance metric for real-valued CNN features, such as the euclidean metric, a flip of a bit in the binary CNN descriptors of length n means a metric change of 1 within a possible interval of integers between 0 and n . Additionally considering that within our preliminary experiments, binary CNN descriptors (with arithmetic mean) two random frames have an average distance of around 850, such a flip (which may be caused by a very small change of CNN descriptor value) has higher impact on the distance of two frames for binary CNN features than when using real-valued features. This fact is relevant considering consecutive frames, when a feature component rises from zero. For larger sampling intervals we aim to stall this flip in the binary feature vector, as we thus keep being similar to the previous frames for more consecutive frames. Hence, the need non-zero thresholds arises. Furthermore, a single threshold per individual feature component should be used, as individual feature components may behave differently. We earlier stated that the other two other candidates for binarization thresholds are the median value and the arithmetic mean. Analysis of medians of each feature component show that approximately 90% of medians are zero. Thus, binarization with non-zero medians make the feature vector even sparser. However, median calculation is computationally expensive, which is contrary to the aim of low-space usage with our proposed binary features. Considering computation of the arithmetic mean is feasible by using an one-pass algorithm. This has the advantage that not all CNN features of the entire EMDB have to be stored at the same time. In contrast to the arithmetic mean, for the median we would need to hold all CNN descriptors in memory during its computation. Furthermore, arithmetic means of the feature components will be non-zero and greater than or equal to the medians, as we have non-negative values only. This adds to even more sparse feature vectors. Hence, from a theoretical point of view, the additional sparsity in combination with the computational efficiency (compared to median values) justifies our choice for the arithmetic mean over medians and a global zero threshold. Preliminary experiments have shown that the arithmetic mean can not be outperformed by the other two choices of binarization thresholds. Hence, we evaluate our proposed approach with arithmetic means only.

4 Methodology

This section presents the used dataset and an overview on the used methods to link images to videos and PTS in the EMDB. Furthermore, we present the results of the methods on this dataset that has been generated in cooperation with medical experts at the regional hospital (LKH) Villach.

4.1 Ground truth generation

Our medical partners have provided 221 short video sequences and 115 images of gynecologic interventions. The images were selected due to medical considerations, e.g. showing

pathology. The video sequences and images were produced before our cooperation on this linking use-case. Hence, no selection bias is introduced. The video sequences feature a spatial resolution of 1920×1080 pixel at 25 frames per second and are H.264 encoded. On average, a video is 1m28s long with a standard deviation of 1m12s. The median video is of 1m9s length. In total, the video sequences feature a length of approximately 5h27m (490720 frames in total), whereas the individual video length ranges from 1s to 4m11s. For some images there exists no corresponding video, because the scene the image was taken in has not been recorded. We exclude these images, as well images that have been recorded out-of-patient (for considerations of privacy issues), in the final evaluation dataset. Examples for the query images are given in Fig. 2 showing an image during a laser treatment, two images of pathology examination in the begin of the surgery (situs) and during the surgery. The laser treatment image is considered to be *easier* to link correctly, as for the application of the laser a different optic is inserted in the endoscope and the image is darker in general. Often it also features a black circle while other scenes don't. The situs images are most relevant as they allow for a begin and end examination of the patient. We assess their difficulty as *medium*. Images during the surgery are considered as *difficult* to link, as they have the most candidate frames. For annotation, we calculate frame-per-frame PSNR of video frames against the images serving as a reference. For each image, we extract 3 top candidates and manually compare these candidates to the reference image. This procedure allows for frame-accurate linking of the images to the video and has the advantage that it is executable by a non-expert user. Using this procedure, we have annotated 69 images together with their true PTS from 38 different videos. The dataset comprises 64 annotations matching a video frame exactly and 5 annotations of images, which have been shot 'between' two video frames. The correctness of these 5 frames has been inferred on the basis of instrument positioning, tissue, and reflections in retrospective. Our explanation for the latter is as follows: The videos are stored at 25 frames per second and the recording camera (which is also responsible for recording the images) has an output of 50 frames per second. For this fact, we consider that arbitrary frames are taken, which are not in the video, but temporally 'between' two frames. The dataset is a little smaller in total playback duration compared to former works in this field and much smaller than the amount of data which would pile-up in a real-world application considering that a single intervention may easily last hours. Nevertheless, our dataset is representative. The benefit of the smaller dataset is that its size allows for evaluating a dense sampling method, i.e. a search for each image frame, and compare its performance against sparse sampling of frames, as well as full-scale PSNR comparison in reasonable time.

4.2 Encoding perspective approaches

During endoscopic surgery, raw image data is delivered to a so-called frame-grabber. All video data is shown to the surgeons in real time on a monitor in the operating room.

Fig. 2 Examples for query images: the leftmost image shows laser treatment, while the other three images show pathology examination in the begin and during the surgery

Additionally, the surgeon is able to control the frame grabber in order to record and store video and image files. We therefore model the video frame f_V and image f_I as the base image f distorted by an additive compression error ε_V and ε_I respectively. The different compression errors are introduced by different compression methods for video and image compression. *PSNR* and *SSIM* are full-reference image quality metrics, aiming at modeling the distortion of frames suffering from such compression errors. While *PSNR* is based on a per-pixel mean square error on a logarithmic scale, *SSIM* exploits image structure [35]. For two $m \times n$ images x and y we compute *PSNR* as follows:

$$PSNR(x, y) = 10 \log_{10} \left(\frac{255^2}{\frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |x(i, j) - y(i, j)|^2} \right)$$

SSIM for two images x and y is computed as follows:

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ denotes the average value, σ^2 the variance, and $\sigma_{x,y}^2$ the covariance. We assume *PSNR* and *SSIM* to model compression errors in a monotonic way: the more the distortion, the worse the metric value. For our case of two distorted frames, we conclude that the near duplicate images f_I and f_V have a better metric value than dissimilar video frames. Thus, we calculate the metric value between the query image and each frame in the database and select the frame with the best metric value as result. We expect these methods to provide good results and act as a baseline for further experiments. Nevertheless, these methods are not scalable with respect to resolution and computationally very expensive, as every query is followed by a metric calculation for each frame in the EMDB.

4.3 Visual similarity perspective approaches

These approaches leverage visual similarity of the images in order to link a query image to a specific video frame in the EMDB. As baseline, we use the global image feature *CEDD* [9]. *CEDD* uses a color unit for extracting color information and a texture unit for texture information. The image is separated into 1600 image blocks. Each of these blocks is classified into one of six texture bins by the texture unit. The color unit provides a fuzzy classification of each block into 24 bins. Finally, the results for each block is aggregated into a $6 \times 24 = 144$ bin histogram. This histogram is quantized into three bits per bin resulting in a descriptor of 432 bit size. For retrieval of the correct video frame with *CEDD*, we calculate the Tanimoto distance between two descriptors (as suggested for *CEDD*) and retrieve the frame with minimal distance to the query image. For metric and distance calculation, we use the *LIRE* [21] library.

As second approach, we use feature signatures. Feature signatures are obtained by clustering image features, such as color, position, or texture. The cluster representatives and weights are then stored and used as descriptor. We calculate feature signatures using *OpenCV* [23] and default parameters. As these signatures do not necessarily have the same dimension, we have to use a special distance metric for the distance calculation. In particular, we use *SQFD* (signature quadratic form distance; as investigated in [20]).

We also investigate the deep learning methods based on CNN features off-the-shelf, as well as binary CNN features as described in Section 3. For these CNN-based approaches, we use the CNN models *AlexNet* [19] and *GoogLeNet* [31] initialized with weights from the Caffe Model Zoo [6] (berkeley-trained). We choose these two network architectures,

as we already know from our previous work [24] that they perform well on the domain of endoscopy. Furthermore such computationally inexpensive networks have the benefit that the processing is finished shortly after the end of the surgery. For a real-life scenario, this means a surgeon performing an intervention before noon is able to use CNN feature-based methods in the afternoon to write a case documentation. Furthermore, we use the *ResNet* [15] models (ResNet50, ResNet101, and ResNet 152) initialized with the provided weights in order to get an comparison with (on ImageNet) even better performing models.

The approach *CNN features off-the-shelf* uses features extracted from the aforementioned CNNs. Specifically, we use the results of the CNN output layer, in order to get (probability) Histograms of Class Confidences (HoCC). From deep, fully-connected layers near to the top of the CNN architecture, we extract the neurons activations. We henceforth denote these activation values as *Neural Codes (NC)* (c.f [2], [8]). We use these uncompressed CNN features off-the-shelf HoCC and NC as image descriptors. We compare these feature vectors using three different distance functions: max, euclidean, and manhattan distance. For CNN feature extraction, we use the Caffe framework [18]. In particular, we extract these CNN features from the last three layers of the AlexNet architecture: *prob* for HoCC, *fc7*, and *fc6* for NC. We use the extracted descriptors for NC and HoCC in an uncompressed way. They feature a dimension of 4096 (*fc6*), and 4096 (*fc7*). The HoCC approach features a dimensionality of 1000, as the models were trained for ImageNet classification into 1000 different classes. As the network has an input size of 224×224 pixel, we resize the input images such that the smaller side has a length of 224 pixel. We then use the center crop as input for the CNN. This is reasonable in the endoscopic domain as the camera always focuses on the most important aspects and thus they are centered in the image as well as results from [8] indicate that this method is superior to padding and naive scaling. We follow the same approach for GoogLeNet, but use the 1024-dimensional bottleneck features extracted from layer *pool5/7x7_s1* as NC. For the ResNet models we use the 2048-dimensional bottleneck features extracted from layer *pool5* as NC.

Fig. 3 Binarization of CNN features: extraction of real-valued CNN features and binarization using pre-calculated binarization thresholds

For the last approach, we transform uncompressed CNN features off-the-shelf into binary CNN features using Algorithm 1. The envisioned binarization process is illustrated in Fig. 3 and consists of two steps: CNN feature extraction and binarization.

We found in our preliminary studies that various distance metrics (i.e. hamming, dice, and jaccard) for binary vectors perform equally good in the case of binary CNN features. Hence, for the purposes of this work, we present the evaluation of the hamming distance only.

4.4 Evaluation methodology

We propose to evaluate the performance of the individual approaches in two different dimensions relevant to the task of image to video linking: video hit rate v_{HR} as well as average distance to the correct PTS d_{PTS} . In order to thoroughly explain the performance metrics, we formulate the goal behind the use case: Provide a function l that links a query image $q \in I$ to a single video $v \in V$ and a PTS $p \in \mathbb{R}$ within an EMDB with image set I and video set $l(q) = (v, p)$. For evaluation, we have a ground truth $g(q)$ available which was generated as described in Section 4. We define a set of correctly classified images:

$$C := \{i \mid i \in I \wedge g(i) = (v, p_1) \wedge l(i) = (v, p_2)\}$$

We define video hit rate (or recall) as number of images which were linked to the same video as in the ground truth relative to the number of all query images:

$$v_{HR} := \frac{|C|}{|I|}$$

Based on the same notation, we calculate the average distance to the corrects PTS as following. We define the set of PTS deviations D as:

$$D := \{|p_1 - p_2| \mid i \in C \wedge l(i) = (v, p_1) \wedge g(i) = (v, p_2)\}$$

For evaluation of average distance to the correct PTS, we calculate the mean of the aforementioned set:

$$d_{PTS} := \frac{1}{|D|} \sum_{d \in D} d$$

Please note that for the calculation of average distance to the correct PTS, only images which were linked correctly are used.

In contrast to previous work, which used a constant sub-sampling of the surgical videos, we evaluate dense and sparse sampling for the methods described in Section 4. For dense sampling, we link a query image to a video and PTS and consider every frame as possible candidate. For sparse sampling, we divide the videos in segments and use the first frame of each segment as key frame. The segment sizes were chosen according to commonly used segment sizes in adaptive streaming: 1s, 2s, 4s, 8s, and 16s, as already mentioned in Section 1.

5 Evaluation and discussion

A graphical overview on the performance of the individual approaches is given in Figs. 4, 5, 6, and 7. A comparative overview between the best performing instances of these approaches is given in Fig. 8. The top rows of these figures illustrate the video hit rate. For the queries which were answered correctly, the standard deviation to the true PTS in

Fig. 4 An overview on the performance of the hand-crafted approaches: PSNR, SSIM, CEDD, and feature signatures. The X-axes denote sampling strategies: Dense, 1s, 2s, 4s, 8s, and 16s. The Y-axis of the top diagram denotes video hit rate, the bottom diagram illustrates the average distance from the found video position to the correct playback time in seconds, whenever the correct video was hit. Error bars denote standard deviations

seconds was evaluated and illustrated together with the appropriate standard deviations to the true PTS. Please note that in some cases, the standard deviation is very high and thus cut off in the figures, this is for the sake of readability. For an extensive overview on the retrieval performance, please refer to Table 1 for video hit rate and Table 2 for deviations to the true PTS. In the following, we discuss the results for hand-crafted and CNN-based approaches individually. We compare these groups of approaches and point at advantages and drawbacks while comparing their retrieval performances. Eventually, we compare the approaches on the dimension of computational requirements.

5.1 Hand-crafted approaches

In this section, we take a detailed look at the hand-crafted approaches PSNR, SSIM, CEDD and feature signatures. An overview on their video hit rate and temporal deviation performance is depicted in Fig. 4. The two visual quality metrics *PSNR* and *SSIM* deliver reasonable results for this kind of retrieval task. At dense sampling, both provide good results. PSNR even manages to perform perfectly in terms of video hit rate, whereas SSIM delivers a slightly better performance in terms of temporal deviation from the true PTS. Considering larger sampling intervals, the video hit performance of SSIM drops more drastically than PSNRs. Interestingly, SSIM is very precise within a found video until a sampling size of 8 s: whenever SSIM identifies the correct video, the true PTS is (on average) also retrieved within the sampling interval. On the other hand, PSNR achieves high video hit rates until a sampling rate of 4s (where still 78.3 percent of the query images are linked to the correct

Fig. 5 An overview on the performance for CNN features off-the-shelf: Histogram of Class Confidences (HoCC) and Neural Codes (NC) extracted from layers f_{c6} and f_{c7} for the AlexNet model as well as HoCC and NC of GoogLeNet initialized with weights from the Caffe model zoo. The X-axes denote sampling strategies: Dense, 1s, 2s, 4s, 8s, and 16s. For the top row, the Y-axis denotes video hit rate. Higher is better. The bottom row illustrates the average distance from the found video position to the correct playback time in seconds, whenever the correct video was hit. Smaller is better. Error bars denote standard deviations

video). The average distance to the true PTS is very small compared to the sample size, but has a standard deviation of approximately 20 s for 2, 4, 8, and 16 s sampling intervals. We conclude that the dramatic performance decrease of SSIM originates in the fact that the structural features of consecutive frames change a lot. PSNR on the other hand uses a pixel-based averaging which proves to be more resilient in terms of larger sampling intervals. A clear drawback of these two approaches is the high query complexity: a full-resolution calculation of PSNR and SSIM is much more expensive than distance calculations between feature vectors.

In contrast to PSNR and SSIM, the *CEDD* descriptor is a precomputed feature vector representing a single image. Despite its computational simplicity, this method features a maximum video hit rate of .855 which is approximately the same performance as PSNR at a sampling rate of 2s. The performance drop with growing sampling intervals is comparable to PSNRs. With an average distance of roughly 10 seconds CEDD is not that precise in determining PTS. Within our evaluation this approach also exhibits a high standard deviation while determining the PTS. Compared to PSNR and SSIM, CEDD is able to quickly find the correct video when applied with dense sampling, because only 432 bit descriptors have to be compared. However, CEDD does not provide a precise localization of the PTS within the correctly retrieved video.

Feature signatures with signature quadratic form distance as distance function is the last hand-crafted approach that is evaluated in this work. For dense sampling, feature signatures surpass the performance of CEDD and nearly achieve the performance of SSIM and PSNR. With medium sampling intervals (1 to 4 s), feature signatures' performance is comparable to CEDDs. For higher sampling intervals than 4s, CEDD provides better results, as we conclude that the abstraction of CEDD works better for such big intervals. In order to be able to make a sanity check compared to related work, we evaluate a sampling interval

Fig. 6 An overview on the performance for the CNN features off-the-shelf: Histogram of Class Confidences (HoCC) and Neural Codes (NC) for the ResNet models ResNet-50, ResNet-101, and ResNet-152. The X-axes denote sampling strategies: Dense, 1s, 2s, 4s, 8s, and 16s. For the top row, the Y-axis denotes video hit rate. Higher is better. The bottom row illustrates the average distance from the found video position to the correct playback time in seconds, whenever the correct video was hit. Smaller is better. Error bars denote standard deviations

Fig. 7 An overview on the performance of binary CNN features: Histogram of Class Confidences (HoCC) and Neural Codes (NC) for AlexNet, GoogLeNet, and ResNet. The X-axes denote sampling strategies: Dense, 1s, 2s, 4s, 8s, and 16s. For the top graph, the Y-axis denotes video hit rate. Higher is better. The bottom graph illustrates the average distance from the found video position to the correct playback time in seconds, whenever the correct video was hit. Smaller is better. Error bars denote standard deviations

Fig. 8 An overview on the performance of PSNR, feature signatures, AlexNetFC7 with euclidean distance and Binay AlexNet FC6 features. The X-axes denote sampling strategies: Dense, 1s, 2s, 4s, 8s, and 16s. For the top graph, the Y-axis denotes video hit rate. Higher is better. The bottom graph illustrates the average distance from the found video position to the correct playback time in seconds, whenever the correct video was hit. Smaller is better. Error bars denote standard deviations

of 5 frames for feature signatures as it was used in previous work. We achieve similar performance as [3] and [27] with a video hit rate of .899 compared to .88 on the other dataset. In this case, the average distance to the true PTS for found videos is 3.41s with a standard deviation of 7.96s.

We conclude that for a dataset, where dense sampling or small sampling intervals are feasible, feature signatures are the hand-crafted approaches of choice because of their trade-off between performance and speed. They are faster than PSNR and SSIM but a bit slower than CEDD, but outperform CEDD in terms of video hit rate and temporal deviation from the PTS.

5.2 CNN-based approaches

Overviews on the results for CNN-based approaches are given in Fig. 5 for CNN features off-the-shelf from the AlexNet and GoogLeNet architectures, Fig. 6 for CNN features off-the-shelf from the ResNet architectures, and Fig. 7 for binary CNN features. Advantages of using CNN-based approaches is their easy applicability, as a plethora of deep learning frameworks and pre-trained models allow for easy use. A drawback in comparison to the hand-crafted methods is that on systems without GPU support, CNN-based methods are of exorbitant computational cost. In the following, we deal with CNN features off-the-shelf and binary CNN features separately.

5.2.1 CNN features off-the-shelf

The *HoCC* approach using AlexNet model achieves a recall of up to 97%. Generally, their performance drops with larger sampling intervals. From Fig. 5, it is apparent that

Table 1 Experimental results for video hit rate v_{HR} of the various approaches, using different sampling intervals and distance metrics

		Dense	1s	2s	4s	8s	16s
SSIM		.986	.449	.333	.333	.261	.188
PSNR		1.00	.942	.855	.783	.623	.449
CEDD		.855	.754	.725	.594	.551	.464
Feature		.957	.739	.739	.565	.478	.391
Signatures							
CNN features AlexNet							
HoCC	d_{Max}	.754	.304	.304	.232	.116	.087
	d_{Euc}	.899	.435	.420	.333	.232	.145
	d_{Man}	.971	.739	.667	.565	.391	.333
NC_{fc7}	d_{Max}	1.00	.899	.826	.681	.536	.435
	d_{Euc}	1.00	.971	.928	.826	.696	.609
	d_{Man}	1.00	.942	.884	.812	.681	.594
NC_{fc6}	d_{Max}	.942	.899	.783	.696	.594	.449
	d_{Euc}	.942	.899	.870	.841	.768	.667
	d_{Man}	.942	.913	.870	.812	.754	.609
CNN features GoogLeNet							
HoCC	d_{Max}	.565	.333	.261	.232	.188	.116
	d_{Euc}	.754	.406	.304	.261	.232	.159
	d_{Man}	.913	.565	.464	.406	.319	.246
NC	d_{Max}	.942	.768	.681	.551	.406	.391
	d_{Euc}	.942	.913	.812	.739	.681	.565
	d_{Man}	.942	.899	.812	.739	.667	.536
CNN features ResNet-50							
HoCC	d_{Max}	.435	.159	.130	.087	.058	.087
	d_{Euc}	.565	.232	.203	.188	.116	.101
	d_{Man}	.725	.304	.261	.275	.217	.130
NC	d_{Max}	.942	.826	.739	.667	.565	.478
	d_{Euc}	.942	.899	.870	.826	.783	.623
	d_{Man}	.942	.884	.884	.855	.768	.638
CNN features ResNet-101							
HoCC	d_{Max}	.435	.232	.174	.159	.159	.130
	d_{Euc}	.507	.304	.261	.217	.159	.145
	d_{Man}	.667	.435	.348	.261	.188	.188
NC	d_{Max}	.942	.855	.826	.739	.623	.449
	d_{Euc}	.942	.899	.913	.884	.783	.667
	d_{Man}	.942	.899	.913	.855	.797	.638

Table 1 (continued)

		Dense	1s	2s	4s	8s	16s
CNN features ResNet-152							
HoCC	d_{Max}	.420	.203	.145	.087	.087	.043
	d_{Euc}	.507	.261	.188	.101	.087	.087
	d_{Man}	.638	.377	.290	.174	.145	.145
NC	d_{Max}	.942	.855	.797	.696	.551	.406
	d_{Euc}	.942	.913	.855	.812	.754	.609
	d_{Man}	.942	.913	.870	.826	.768	.609
Binary CNN features AlexNet							
HoCC		.928	.696	.594	.420	.406	.290
fc7		.942	.913	.812	.812	.652	.580
fc6		.942	.928	.884	.841	.783	.638
Binary CNN features GoogLeNet							
HoCC		.870	.710	.565	.464	.420	.362
NC		.942	.899	.826	.768	.638	.507
Binary CNN features ResNet-50							
HoCC		.812	.493	.449	.319	.275	.203
NC		.942	.870	.855	.797	.739	.609
Binary CNN features ResNet-101							
HoCC		.783	.478	.377	.362	.261	.174
NC		.942	.913	.870	.841	.739	.551
Binary CNN features ResNet-152							
HoCC		.797	.565	.493	.406	.362	.290
NC		.942	.913	.870	.812	.739	.536

manhattan distance works best for this approach in terms of video hit rate. Considering the standard deviations of the average deviation from the true PTS, we can not state that one distance metric works better than another for this dimension. The only exception from this observation is that manhattan distance seems to work better for dense sampling. The retrieved PTS with the Manhattan distance, which achieves the best results for HoCC, feature an average value of up to 5 seconds and a standard deviation of less than 10 seconds for sampling intervals of up to 4 seconds. For GoogLeNet and the three ResNet models, we see the same trend in terms of used distance function. Manhattan dominates euclidean distance which dominates maximum distance. We also observe the same behavior for PTS retrieval as when using AlexNet. While comparing HoCC for the CNN architectures AlexNet and GoogLeNet across sampling intervals, we observe that AlexNet outperforms GoogLeNet. This observation is especially visible in terms of deviation to true PTS. We observe the same for the ResNet models, whose HoCC features perform worse than AlexNet and GoogLeNet. We think this originates in the fact that GoogLeNet and ResNet architectures are more specialized for general images such as they occur in the ILSVRC dataset.

Table 2 Average deviation d_{PTS} (in seconds) from the true playback position for successfully retrieved videos for the various approaches, using different sampling intervals and distance metrics

		Dense	1s	2s	4s	8s	16s
SSIM		.001	.301	.697	1.88	2.85	9.23
PSNR		.007	.633	5.08	7.61	10.8	10.4
CEDD		7.80	8.07	10.2	11.0	15.7	10.2
Feature		.453	7.99	6.58	12.8	21.1	23.9
Signatures							
CNN features AlexNet							
HoCC	d_{Max}	2.10	3.21	6.01	9.13	12.9	22.7
	d_{Euc}	1.57	1.28	3.27	9.08	17.1	28.3
	d_{Man}	.156	1.93	3.65	4.82	8.59	19.2
NC_{fc7}	d_{Max}	.034	.475	1.70	1.78	5.22	13.7
	d_{Euc}	.019	.568	4.35	4.89	7.34	13.7
	d_{Man}	.024	.539	4.44	4.83	6.50	16.3
NC_{fc6}	d_{Max}	.050	1.48	3.27	8.84	10.8	18.0
	d_{Euc}	.005	.455	4.59	4.78	10.2	12.3
	d_{Man}	.005	.479	4.49	5.04	9.16	7.45
CNN features GoogLeNet							
HoCC	d_{Max}	3.43	11.3	7.69	8.30	13.4	26.4
	d_{Euc}	2.73	9.49	6.17	6.70	11.6	24.0
	d_{Man}	.266	7.62	4.39	5.55	7.91	15.6
NC	d_{Max}	.210	1.70	2.78	5.16	7.94	14.0
	d_{Euc}	.106	1.28	2.07	5.85	10.1	18.1
	d_{Man}	.094	.724	3.41	6.46	9.39	16.5
CNN features ResNet-50							
HoCC	d_{Max}	2.84	4.96	5.51	5.05	2.12	51.0
	d_{Euc}	1.36	3.60	5.34	7.93	9.73	25.5
	d_{Man}	2.08	6.03	4.85	9.87	9.99	20.5
NC	d_{Max}	.089	4.08	2.28	4.88	7.30	20.8
	d_{Euc}	.021	.630	1.25	4.21	9.33	13.0
	d_{Man}	.022	.690	1.21	2.84	7.95	13.6
CNN features ResNet-101							
HoCC	d_{Max}	3.15	12.2	13.5	21.7	25.0	30.3
	d_{Euc}	.671	7.90	9.06	11.4	18.8	27.4
	d_{Man}	.544	9.77	11.2	14.0	20.9	25.1
NC	d_{Max}	.110	1.22	6.64	9.76	11.2	13.5
	d_{Euc}	.052	.668	4.03	6.92	13.1	15.2
	d_{Man}	.026	.626	3.91	6.20	13.9	14.9

Table 2 (continued)

		Dense	1s	2s	4s	8s	16s
CNN features ResNet-152							
HoCC	d_{Max}	3.05	5.27	5.76	12.8	15.7	9.41
	d_{Euc}	6.22	5.13	3.40	3.81	15.9	12.2
	d_{Man}	1.50	1.93	5.61	3.97	11.8	29.9
NC	d_{Max}	.095	2.30	5.56	7.91	11.7	14.5
	d_{Euc}	.029	1.57	3.77	8.20	11.5	17.0
	d_{Man}	.026	1.48	4.16	8.29	12.6	17.5
Binary CNN features AlexNet							
HoCC		.439	1.39	5.94	8.21	10.0	20.5
fc7		.023	1.08	4.39	8.94	8.46	17.2
fc6		.008	.533	4.09	4.96	9.21	9.29
Binary CNN features GoogLeNet							
HoCC		.468	5.24	6.44	7.60	11.5	12.0
NC		.110	3.21	3.89	6.34	9.64	15.6
Binary CNN features ResNet-50							
HoCC		1.43	9.67	11.2	13.0	9.72	11.2
NC		.039	.778	1.21	3.64	11.1	18.6
Binary CNN features ResNet-101							
HoCC		.218	7.91	4.33	3.41	8.49	17.0
NC		.033	.678	3.54	5.48	13.4	12.3
Binary CNN features ResNet-152							
HoCC		2.68	3.76	10.2	15.9	17.3	17.9
NC		.084	1.37	4.32	8.47	14.2	19.5

Neural Code approaches for AlexNet provide best results for dense sampling out of all CNN-based approaches. Using features extracted from layer fc7 and dense sampling, the maximum video hit rate of 100% is achieved regardless of the distance metric. All in all, the Euclidean distance works well for fc7, the Manhattan distance is able to compete with and occasionally surpass the Euclidean distance using features from fc6. Regarding distance to the true PTS, the standard deviations of Manhattan and Max distances are comparable. The Max distance works worst across each test case. For GoogLeNet, euclidean distance works best over all sampling intervals in terms of video hit rate. With small sampling intervals, the manhattan distance provides better results in terms of temporal deviation to the true PTS, while the max distance works best for large sampling intervals in this dimension. With NC, we observe that GoogLeNet features perform slightly worse than AlexNet features, especially over larger sampling intervals. We think that originates in the fact that feature vectors from AlexNet are with 4096 dimensions four times larger than GoogLeNet's with a dimensionality of 1024. With ResNet, we observe that euclidean and manhattan distances work best. For dense sampling, there is no difference in the performance of the three ResNet models, but with larger sampling intervals, ResNet-101 outperforms the other two model variants in terms of v_{HR} , while there is no clear best

model in terms of d_{PTS} . Compared to AlexNet, we observe that the best ResNet model (ResNet-101 with euclidean distance) performs approximately as good as AlexNet features across all sampling intervals. We conclude that this originates in the fact that 2048 dimensional ResNet-101 features are as discriminative as 4096 dimensional AlexNet features for this use case. However, AlexNet still performs better than ResNet in terms of d_{PTS} .

5.2.2 Binary CNN features

The evaluation results imply that for video hit rate, binary neural codes slightly outperform binary HoCC features by at least 1.4% (94.2% against 92.8% for dense sampling). We furthermore observe over growing sampling intervals, that binary neural codes' performance decreases slower than binary HoCC's performance. The binarization process leads to a increase in performance of GoogLeNet relative to AlexNet. Binary neural codes and dense sampling work equally good in terms of video hit rate using AlexNet and GoogLeNet. At higher sampling intervals, the higher-dimensional features of AlexNet perform better. The observation that AlexNet's features provides better results in terms of temporal deviation to the true PTS also holds for the binary version. All in all binary AlexNet features from f_{c6} achieve best performance within this category. For the three ResNet models, the experiments show that NC works better than HoCC over all evaluated sampling intervals. As with the real-valued CNN features off-the-shelf, ResNet-101 performs best, with the exception of 16s sampling intervals, where it is outperformed by ResNet-50. Compared to AlexNet, binary ResNet features perform slightly worse. When compared to GoogLeNet they perform approximately equal for small sampling intervals and slightly better for larger sampling intervals. We conclude that this performance hierarchy for large sampling intervals originates in the number of dimensions (1024, 2048, and 4096 for GoogLeNet, ResNet, and AlexNet respectively) of the used features.

5.3 Retrieval performance comparison and computational analysis

The results for PSNR, feature signatures, CNN features off-the-shelf extracted from AlexNet f_{c7} and compared with euclidean distance, as well as binary CNN features from AlexNet f_{c6} are compared in Fig. 8. At dense sampling, uncompressed CNN features and PSNR outperform the other approaches. When the sampling interval is set to 1s, binary features catch up with and nearly reach PSNR's performance. At a sampling interval of 2s, binary features surpass PSNR in terms of video hit rate. At sampling interval 4s, we observe that binary features even outperform uncompressed CNN features. Binary CNN features also yield the best performance of all approaches compared in Fig. 8 for sampling intervals larger than 4s. Furthermore, we observe that no other approach performs significantly better than binary CNN features at PTS retrieval. Please note that binary CNN features do not outperform CNN features off-the-shelf for high sampling intervals in every combination of distance metric, CNN architecture and extraction point. Rather we conclude that binary CNN features perform on par with CNN features off-the-shelf for high sampling intervals. We state that preserving the sparsity of CNN features off-the-shelf and stalling binary vector components' flipping for consecutive frames with non-zero thresholds (see Section 3.3) is the key that binary CNN features maintain high retrieval performance. The results for average deviation from the playback time stamp indicate that at higher intervals, binary CNN

Table 3 Overview on computational requirements of selected approaches: SSIM, PSNR, CEDD, feature signatures, CNN features compared with euclidean distance, and binary CNN features compared with hamming distance, where the bit counting function is implemented in three different ways

Approach	t_C	s_D
SSIM	648.60 ms	0
PSNR	49.82 ms	0
CEDD	1.80 μ s	54 B
Feature signatures	9.72 μ s	1.53 KB ^a
CNN features (4096-dim, d_{Euc})	12.07 μ s	16 KB
Binary CNN features (4096-dim, d_H v1)	6.49 μ s	512 B
Binary CNN features (4096-dim, d_H v2)	0.94 μ s	512 B
Binary CNN features (4096-dim, d_H v3)	0.64 μ s	512 B

We present the average time for distance calculation (or comparison in the case of PSNR and SSIM) t_C , based on an experiment with 500000 observations, as well as storage space requirements per descriptor s_D

^aFeature signatures are variable length. Hence, the provided value is an estimation based on our database

descriptors are slightly worse than off-the-shelf descriptors for the very reason of stalling these flips.

Table 3 shows a comparison of the approaches from the viewpoint of their computational resource needs. For SSIM and PSNR, we compute the average comparison time based on with 500,000 individual comparisons. As SSIM and PSNR do not impose storage requirements, the entry for these categories is empty. SSIM and PSNR have an infeasible computational cost for comparison, so that they can only be used for very small databases. For a five hour surgery, the linking of a single image would take hours. For the measurement of similarity computation time of binary CNN features, we implement the bit counting function of hamming distance computations with three different versions. In version 1 (v1), we count the bits using Brian Kernighan's bit count algorithm. The algorithm uses a loop that clears the least significant set bit until no bits are set. The number of loop executions is the number of set bits. The loop is executed at most per once per bit, or put differently for a 4096 bit descriptor, the loop is executed at most 4096 times. Version 2 (v2) and version 3 (v3) use a precomputed lookup table of 8 bit and 16 bit respectively. The lookup table is realized as an integer array of size 2^8 for 8 bit approach and 2^{16} for 16 bit approach. The lookup table contains the number of set bits for the corresponding index. The input descriptors are sliced into parts of 8 and 16 bits respectively and the sum of set bits for each slice is the number of total bits set for a given binary number. Thus, for the 16 bit lookup table, a binary number of size $16n$ is sliced into n parts of 16 bits. Hence, we compute the number of set bits in a binary number using n lookups and additions. For example, a 4096 bit number (that is the result of an xor operation of two 4096 bit binary CNN descriptors) requires 256 lookups and additions. Binary features of dimension 4096 compared with hamming distance v1 (Brian Kernighan's bit count) show an average distance calculation in 6.49 μ s. Distance comparison of two 432 bit CEDD descriptors using the Tanimoto distance implemented in LIRE performs even better with an average of 1.80 μ s. Optimizing the hamming distance calculation for binary features with lookup tables leads to a performance increase: hamming distance v2 (8 bit lookup table) and hamming distance v3 (16 bit lookup table) on average only take 0.94 and 0.64 μ s respectively for distance computation. Please note that CEDD comparison was performed using the LIRE library while the other approaches have been implemented in C++. Furthermore we ignore descriptor extraction time (which

will be dealt with qualitatively) as well as pre-computations, e.g. calculating binarization thresholds, or initialization of lookup tables. Whereas modern GPUs allow for efficient extraction of CNN features, extraction of hand-crafted features is computationally expensive. Within the CNN-based approaches, descriptor extraction is most efficient for AlexNet and GoogLeNet. Furthermore, we observe that CEDD with its 432 bit descriptor requires very little storage space. Feature signatures are variable-length image descriptors. In our experiments the average size of a features signature is about 1.5KB. Binary CNN features require 1000, 1024, 2048, and 4096 bit storage space per descriptor for HoCC, GoogLeNet NC, ResNet NC, and AlexNet NC respectively. We want to emphasize that this is a huge decrease in required storage compared to real-valued CNN descriptors which take 4000, 4096, 8192, and 16384 Byte per descriptor for HoCC, GoogLeNet NC, ResNet NC, and AlexNet NC respectively.

Based on estimations from [28], we assume for an exemplary hospital with 5 departments and two operation rooms generates 60 h video per day. Over a year, that would be over 20,000 h of endoscopic video recordings. For 20,000 h of video and a sampling rate of 16 s, the hospital system is required to store 4.5 million image descriptors. For GoogLeNet NC features, this would mean 17.16 GB of image descriptors alone, while a binarized version of the same features (that would perform approximately 6% worse in terms of video hit rate: .565 vs .507) would take about 550MB. From another point of view and the same features, extraction of a binary descriptor every second would require 8.5GB descriptor size and those binary features would perform approximately 33% better (.564 vs .899) than the CNN features off-the-shelf at half the size.

6 Conclusion

In this work, we investigate the problem of linking endoscopic images and video sequences within an endoscopic multimedia database. Therefore, we propose a binarization scheme for off-the-shelf CNN features aiming at straightforward applicability and low computational resource cost. We perform an extensive evaluation against baselines of hand-crafted methods PSNR, SSIM, CEDD, and feature signatures, as well as uncompressed CNN features off-the-shelf features. Our evaluation shows that state-of-the-art methods have severe drawbacks in this task. PSNR and SSIM provide a solid baseline in terms of retrieval performance, but are computationally not feasible for large EMDBs. The content descriptor CEDD and feature signatures do not have these problems, but yield a mediocre performance at dense sampling of the EMDB frames. PSNR and off-the-shelf features work best in terms of video hit rate and distance to the true PTS. The proposed approach - binary CNN features off-the-shelf - maintains only a slight performance loss for dense sampling intervals compared to uncompressed CNN features. It reaches and partially outperforms the performance of other approaches (e.g. feature signatures) at sparse sampling intervals. Binary CNN features are able to achieve high retrieval performance, as our binarization scheme aims at preserving the sparsity of CNN features off-the-shelf. Moreover, it allows for similar feature vectors of consecutive video frames which is beneficial at larger sampling intervals. Furthermore, binary CNN features have a significantly reduced storage space requirement and distance calculation complexity compared to uncompressed CNN features off-the-shelf. Future work in this topic may consider reducing the dimensionality of already binarized features without loss of accuracy. Furthermore, for

future work we aim at automatically detect shot boundaries of relevant surgery scenes automatically.

Acknowledgements This work was supported by Universität Klagenfurt and Lakeside Labs GmbH, Klagenfurt, Austria and funding from the European Regional Development Fund and the Carinthian Economic Promotion Fund (KWF) under grant KWF 20214 u. 3520/ 26336/38165. Special thanks to Univ.-Prof. Dr. Jörg Keckstein, who supports us by providing data and relevant real-world use cases.

Funding Information Open access funding provided by University of Klagenfurt.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Awad G, Fiscus J, Michel M, Joy D, Kraaij W, Smeaton AF, Quénot G, Eskevich M, Aly R, Ordelman R (2016) Trecvid 2016: evaluating video search, video event detection, localization, and hyperlinking. In: Proceedings of TRECVID, vol 2016
2. Babenko A, Slesarev A, Chigorin A, Lempitsky V (2014) Neural codes for image retrieval. Springer International Publishing, Cham, pp 584–599
3. Beecks C, Schoeffmann K, Lux M, Uysal MS, Seidl T (2015) Endoscopic video retrieval: a signature-based approach for linking endoscopic images with video segments. In: Del Bimbo A, Chen SC, Wang H, Yu H, Zimmermann R (eds) Proceedings of the IEEE international symposium on multimedia 2015 (ISM 2015). IEEE, Los Alamitos, pp 1–6
4. Bois R, Vukotić V, Simon AR, Sicre R, Raymond C, Sébillot P, Gravier G (2017) Exploiting multi-modality in video hyperlinking to improve target diversity. Springer International Publishing, Cham, pp 185–197
5. Bosch A, Zisserman A, Munoz X (2007) Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, pp 401–408
6. BVLC (2016) Caffe model zoo. http://caffe.berkeleyvision.org/model_zoo.html
7. Carlos JR, Lux M, Giro-i Nieto X, Munoz P, Anagnostopoulos N (2015) Visual information retrieval in endoscopic video archives. In: 2015 13th international workshop on content-based multimedia indexing (CBMI). IEEE, pp 1–6
8. Chandrasekhar V, Lin J, Morère O, Goh H, Veillard A (2016) A practical guide to CNNs and fisher vectors for image instance retrieval. Signal Process 128:426–439
9. Chatzichristofis SA, Boutalis YS (2008) Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In: International conference on computer vision systems. Springer, pp 312–322
10. Cheng Z, Li X, Shen J, Hauptmann AG (2015) Cmu-smu@ trecvid 2015: video hyperlinking
11. Ercoli S, Bertini M, Bimbo AD (2017) Compact hash codes for efficient visual descriptors retrieval in large scale databases. IEEE Trans Multimedia 19(11):2521–2532. <https://doi.org/10.1109/TMM.2017.2697824>
12. Eskevich M, Aly R, Racca D, Ordelman R, Chen S, Jones GJ (2014) The search and hyperlinking task at mediaeval 2014
13. Galuščáková P, Batko M, Čech J, Matas J, Novák D, Pecina P (2017) Visual descriptors in methods for video hyperlinking. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval, ICMR '17. ACM, New York, pp 294–300. <https://doi.org/10.1145/3078971.3079026>
14. Guo J, Li J (2015) Cnn based hashing for image retrieval. arXiv:1509.01354
15. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385
16. Huang J, Kumar SR, Mitra M, Zhu WJ, Zabih R (1997) Image indexing using color correlograms. In: 1997 IEEE computer society conference on computer vision and pattern recognition, 1997. Proceedings. IEEE, pp 762–768

17. Iakovidou C, Anagnostopoulos N, Kapoutsis AC, Boutalis Y, Chatzichristofis SA (2014) Searching images with mpeg-7 (& mpeg-7-like) powered localized descriptors: the simple answer to effective content based image retrieval. In: 2014 12th international workshop on content-based multimedia indexing (CBMI), pp 1–6
18. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on multimedia, MM '14. ACM, New York, pp 675–678
19. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Bartlett P, Pereira F, Burges C, Bottou L, Weinberger K (eds) Advances in neural information processing systems 25, pp 1106–1114
20. Lokoč J, Hetland ML, Skopal T, Beecks C (2011) Ptolemaic indexing of the signature quadratic form distance. In: Proceedings of the fourth international conference on similarity search and applications, SISAP '11. ACM, New York, pp 9–16
21. Lux M, Chatzichristofis SA (2008) Lire: lucene image retrieval: an extensible java cbir library. In: Proceedings of the 16th ACM international conference on multimedia. ACM, pp 1085–1088
22. Münzer B, Schoeffmann K, Böszörményi L (2013) Relevance segmentation of laparoscopic videos. In: 2013 IEEE international symposium on multimedia, pp 84–91. <https://doi.org/10.1109/ISM.2013.22>
23. OpenCV (2015) Open source computer vision library. <https://github.com/itseez/opencv>
24. Petscharnig S, Schoeffmann K (2017) Learning laparoscopic video shot classification for gynecological surgery. *Multimedia Tools and Applications*:1–19. <https://doi.org/10.1007/s11042-017-4699-5>
25. Petscharnig S, Lux M, Chatzichristofis S (2017) Dimensionality reduction for image features using deep learning and autoencoders. In: Proceedings of the 15th international workshop on content-based multimedia indexing, CBMI '17. ACM, New York, pp 23:1–23:6. <https://doi.org/10.1145/3095713.3095737>
26. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition workshops, CVPRW '14. IEEE Computer Society, Washington, pp 512–519
27. Schoeffmann K, Beecks C, Lux M, Uysal MS, Seidl T (2016) Content-based retrieval in videos from laparoscopic surgery, pp 97,861V–97,861V–10
28. Schoeffmann K, Münzer B, Riegler M, Halvorsen P (2017) Medical multimedia information systems (mmis). In: Proceedings of the 2017 ACM on multimedia conference, MM '17. ACM, New York, pp 1957–1958. <https://doi.org/10.1145/3123266.3130142>
29. Simon AR, Sicre R, Bois R, Gravier G, Sébillot P (2015) IRISA at TrecVid2015: leveraging multimodal LDA for video hyperlinking. In: TRECVID 2015 workshop, working notes of the TRECVID 2015 workshop. Gaithersburg, United States. <https://hal.archives-ouvertes.fr/hal-01403726>
30. Sodagar I (2011) The MPEG-DASH standard for multimedia streaming over the internet. *IEEE MultiMedia* 18(4):62–67
31. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: The IEEE conference on computer vision and pattern recognition (CVPR)
32. Vukotić V, Raymond C, Gravier G (2016) Bidirectional joint representation learning with symmetrical deep neural networks for multimodal and crossmodal applications. In: Proceedings of the 2016 ACM on international conference on multimedia retrieval, ICMR '16. ACM, New York, pp 343–346
33. Vukotić V, Raymond C, Gravier G (2016) Multimodal and crossmodal representation learning from textual and visual features with bidirectional deep neural networks for video hyperlinking. In: Proceedings of the 2016 ACM workshop on vision and language integration meets multimedia fusion, iV&L-MM '16. ACM, New York, pp 37–44. <https://doi.org/10.1145/2983563.2983567>
34. Vukotić V, Raymond C, Gravier G (2017) Generative adversarial networks for multimodal representation learning in video hyperlinking. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval, ICMR '17. ACM, New York, pp 416–419. <https://doi.org/10.1145/3078971.3079038>
35. Wang Z, Bovik A, Sheikh H, Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. In: *IEEE transactions on image processing*, vol 13, pp 600–612

Stefan Petscharnig started his studies of computer science in 2010 at AAU Klagenfurt. He received his bachelor's degree in 2014 and his master's degree in 2015 from AAU Klagenfurt. In his master's thesis he investigated the impact of asynchronism on the Quality of Experience in social TV like scenarios using methods from Crowdsourcing, Human Computation, and Gamification. He was occupied as student research assistant and after graduation as project assistant for the AdvUHD-DASH project in 2015. Since 2016, he works in the KISMET research project with a focus on the analysis of endoscopic video data using machine learning.

Dr. Klaus Schöffmann is an Associate Professor in the distributed multimedia systems research group at the Institute of Information Technology (ITEC) at Klagenfurt University, Austria. He received his Ph.D. in 2009 and his Habilitation (*venia docendi*) in 2015, both in Computer Science and from Klagenfurt University. His research focuses on human-computer interaction with multimedia data (e.g., video browsing), multimedia content analysis, and multimedia systems (particularly in the mobile and medical domain). He has co-authored more than 80 publications on various topics in multimedia and he has co-organized international conferences, special sessions and workshops (e.g., MMM 2012, CBMI 2013, VisHMC 2014, MMC 2014, MMC 2015, and MMC 2016). He is co-founder of the Video Browser Showdown (VBS), an editorial board member of the Springer International Journal on Multimedia Tools and Applications (MTAP), Springer International Journal on Multimedia Systems, and a steering committee member of the International Conference on MultiMedia Modeling (MMM). Additionally, he is member of the IEEE and the ACM and a regular reviewer for international conferences and journals in the field of multimedia. Prof. Schoeffmann teaches various courses in computer science, including mobile app development (interactive multimedia applications), video retrieval, media technology, distributed systems, distributed multimedia systems, and operating systems.