

Video2GIF: Automatic Generation of Animated GIFs from Video

Michael Gygli*
CVL, ETH Zurich
Zurich, Switzerland

gygli@vision.ee.ethz.ch

Yale Song
Yahoo Research
New York, USA

yalesong@yahoo-inc.com

Liangliang Cao
Yahoo Research
New York, USA

liangliang@yahoo-inc.com

Abstract

We introduce the novel problem of automatically generating animated GIFs from video. GIFs are short looping video with no sound, and a perfect combination between image and video that really capture our attention. GIFs tell a story, express emotion, turn events into humorous moments, and are the new wave of photojournalism. We pose the question: Can we automate the entirely manual and elaborate process of GIF creation by leveraging the plethora of user generated GIF content? We propose a Robust Deep RankNet that, given a video, generates a ranked list of its segments according to their suitability as GIF. We train our model to learn what visual content is often selected for GIFs by using over 100K user generated GIFs and their corresponding video sources. We effectively deal with the noisy web data by proposing a novel adaptive Huber loss in the ranking formulation. We show that our approach is robust to outliers and picks up several patterns that are frequently present in popular animated GIFs. On our new large-scale benchmark dataset, we show the advantage of our approach over several state-of-the-art methods.

1. Introduction

Animated GIF is an image format that continuously displays multiple frames in a loop, with no sound. Although first introduced in the late 80's, its popularity has increased dramatically in recent years on social networks, such as Tumblr and reddit, generating numerous famous Internet memes and creative Cinemagraphs [1]. In response, various websites have been created to provide easy-to-use tools to generate GIF from video, e.g., GIFSoup, Imgflip, and Ezgif. However, while becoming more prevalent, the creation of GIF remains an entirely manual process, requiring the user to specify the timestamps of the beginning and the end of a video clip, from which a single animated GIF is generated. This way of manually specifying the exact time range makes existing solutions cumbersome to use and re-

*This work was done while the author was an intern at Yahoo! Inc.

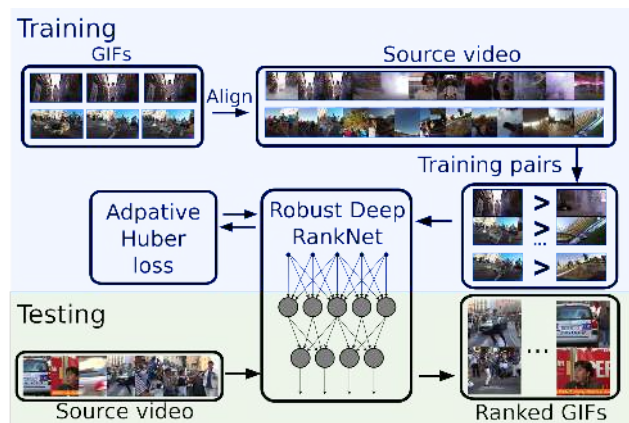


Figure 1. Our goal is to rank video segments according to their suitability as animated GIF. We collect a large-scale dataset of animated GIFs and the corresponding video sources. This allows us to train our Robust Deep RankNet using over 500K pairs of GIF and non-GIF segment pairs, learning subtle differences between video segments using our novel adaptive rank Huber loss.

quires extensive human effort.

In this paper, we introduce the novel problem of automatically generating animated GIFs from video, dubbed Video2GIF. From the computer vision perspective, this is an interesting research problem because GIFs have some unique properties compared to conventional images and videos: A GIF is short, entirely visual with no sound, expresses various forms of emotions, and sometimes contains unique spatio-temporal visual patterns that make it appear to loop forever. The task has some connections to existing computer vision problems – such as visual interestingness [16, 11], creativity [31], video highlights [35, 40] and summarization [13, 33] – but differs from them due to the unique properties described above. Apart from research interest, the task is supported by real-world demand and has many practical application scenarios including photojournalism, advertising, video sharing and preview, as well as video promotion on social media.

To handle this task, we propose a novel RankNet that, given a video, produces a ranked list of segments accord-

ing to their suitability as animated GIF. Our framework has several novel components designed to learn what content is frequently selected for popular animated GIFs. First, to capture the highly dynamic spatio-temporal visual characteristics of GIFs, we use 3D convolutional neural networks [36] to represent each segment of a video. Second, to unravel the complex relationships and learn subtle differences between segments of a given video, we construct a ranking model that learns to compare pairs of segments and find the ones that are more suitable as GIF. Third, to make our learning task robust to the noisy web data, we design a new robust adaptive Huber loss function in the ranking formulation. Lastly, to account for different degrees of quality in user generated content, we encode the popularity measure of GIFs on social media directly into our loss.

Crucial to the success of our approach is our new large-scale animated GIF dataset: We collected more than 100K user generated animated GIFs with their corresponding video sources from online sources. There are hundreds of thousands of GIFs available online and many provide a link to the video source. This allows us to create a dataset that is one to two orders of magnitude larger than existing datasets in the video highlight detection and summarization literature [35, 33, 12]. We use this dataset to train our deep neural network by making comparisons between more than 500K GIF and non-GIF pairs. Experimental results suggest that our model successfully learns what content is suitable for GIFs, and that our model generalizes well to other tasks, namely video highlight detection [35].

In summary, we make the following contributions:

1. We introduce the task of automatically generating animated GIFs from video. This is an interesting computer vision research problem that, to the best of our knowledge, has not been addressed before.
2. We propose a Robust Deep RankNet with a novel adaptive Huber loss in the ranking formulation. We show how well our loss deals with noisy web data, and how it encodes the notion of content popularity to account for different degrees of content quality.
3. We collect a new, large-scale benchmark dataset of over 100K user generated animated GIFs and their video sources. The dataset is one to two orders of magnitude larger than existing video highlighting and summarization datasets. The dataset is publicly available¹.

2. Related Work

Our work is closely related to image aesthetics and interestingness, as well as video highlight detection and summarization. We review some of the most relevant work and discuss the differences. We also review and make connections to recent efforts on learning deep neural networks for

ranking and trained on large-scale weakly-labeled data.

Image aesthetics and interestingness. Finding the best images in a collection has been studied from several angles. Early approaches aimed at predicting the quality [19] or aesthetics [4, 5] of an image. More recently, several approaches for predicting visual interestingness of an image have been proposed [10, 11, 7]. While interestingness is a subjective property assessed by the viewer, there is considerable consistency across annotated ratings [11]. This makes it possible to model interestingness with computational means, but ground truth is typically noisy. Fu *et al.* [7] propose an approach accounting for this, by learning a ranking model and removing outliers in a joint formulation. Khosla *et al.* [20] analyze the related property of image popularity. Using a large-scale dataset of Flickr images, they analyze and predict what types of images are more popular than others, surfacing trends similar to those of interestingness [11]. In a similar direction is the work of Redi *et al.* [31], which analyzes creativity. Rather than analyzing images, however, they focus on Vines videos, whose lengths are restricted to 6 seconds.

Video summarization. A thorough discussion of earlier research can be found in [37]. Here, we discuss two recent trends, (i) using web-image priors [21, 20, 33, 27] and (ii) supervised learning-based methods [25, 8, 13, 27]. Methods using web-image priors are based on the observation that web images for a specific topic or query are often canonical visual examples for the topic. This allows one to compute frame scores as the similarity between a frame and a set of web images [21, 20, 33]. Learning-based methods, on the other hand, use supervised models to obtain a scoring function for frames [25, 8, 27] or segments [13]. Lee *et al.* [25] learn a regression model and combine it with a clustering approach to diversify the results. Instead, [8, 13] directly learn an objective function that scores a set of segments, based on relative importance between different aspects of a summary (e.g. balancing highlights and diversity). Crucial to these learning-based methods is some notion of importance or interestingness of a segment. Next, we will discuss methods focusing only on this part while ignoring diversity and information coverage.

Video highlights. The definition of highlight is both subjective and context-dependent [37]. Nevertheless, it has been shown that there exists some consistency among human ratings for this task [33, 12]. Several methods exploit, for example, that close-ups of faces are generally of interest [37, 25, 12]. But these approaches are limited in that they rely on a few hand-crafted features for capturing highlights in highly diverse settings. Instead, several approaches for domain-specific models have been proposed. In particular, in sport games highlight is more clearly defined (e.g. scoring a goal) which has been exploited in many works (see [37] for an overview). Recently, Sun *et al.* [35]

¹https://github.com/gyglim/video2gif_dataset

and Potapov *et al.* [30] proposed a more general approach. Based on annotated videos for a specific topic (*e.g.* surfing), they use machine learning on top of generic features to train a highlight predictor. In order to train their model, [30] uses a large, manually annotated dataset for action recognition. Instead, [35] use a smaller dataset obtained by crawling YouTube data. They find pairs of raw and edited videos, used in training, by matching all pairs of videos within a certain category (*e.g.* gymnastics). The size of their dataset is, however, limited by the availability of domain-specific videos in both raw and edited forms.

Obtaining a large-scale video highlight dataset is difficult. Thus, Yang *et al.* [40] propose an unsupervised approach for finding highlights. Relying on an assumption that highlights of an event category are more frequently captured in short videos than non-highlights, they train an auto-encoder. Our work instead follows a supervised approach, introducing a new way to obtain hundreds of thousands of labeled training videos (10x larger than the unlabeled dataset of [40]), which allows us to train a deep neural network with millions of parameters.

Learning to rank with deep neural networks. Several works have used CNNs to learn from ranking labels. The loss function is often formulated over pairs [27, 9] or triplets [38, 39, 14, 23]. Pairwise approaches typically use a single CNN, while the loss is defined relatively over the output. For example, Gong *et al.* [9] learn a network to predict image labels and require the scores of correct labels to be higher than the scores of incorrect labels. Triplet approaches, on the other hand, use Siamese networks. Given an image triple (query, positive, negative), a loss function requires the learned representation of the query image to be closer to that of the positive, rather than the negative image, according to some metric [38, 39, 14, 23].

Supervised deep learning from noisy labels. Several previous works have successfully learned models from weak labels [18, 38, 27]. Liu *et al.* [27] considers the video search scenario. Given click-through data from Bing, they learn a joint embedding between query text and video thumbnails in order to find semantically relevant video frames. In contrast, [18, 38] use labels obtained through automatic methods to train neural networks. Karpathy *et al.* [18] train a convolutional neural network for action classification in videos. Their training data is obtained from YouTube where it is labeled automatically by analyzing meta data associated with the videos. Wang *et al.* [38] learn a feature representation for fine-grained image ranking. Based on existing image features they generate labels used for training the neural network. Both approaches obtain state-of-the-art performance, showing the strength of large, weakly-labeled datasets in combination with deep learning.

Property	Quantity
Total number of animated GIFs	121,647
Mean GIF duration	5.8 sec
Total number of videos	84,754
Total video duration	7,379 hr
Mean video duration	5.2 min
Total number of videos (CC-BY)	432
GIFs used in experiment	100,699
Videos used in experiment	70,456

Table 1. Statistics on the Video2GIF dataset. We show numbers for the complete dataset and for the one after discarding too short or too long videos (see text). We also show the number of videos that come with the Creative Commons license (CC-BY).

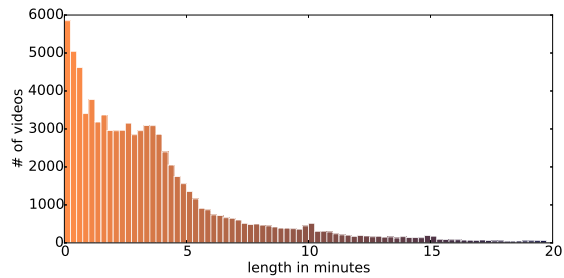


Figure 2. Length distribution of the input videos.

3. Video2GIF Dataset

Inspired by the recent success with large, weakly-labeled datasets applied in combination with deep learning, we harvest social media data with noisy, human generated annotations. We use websites that allow users to create GIFs from video (Make-a-GIF and GIFSoup). Compared to edited videos used in [35], GIFs have the intriguing property that they are inherently short and focused. Furthermore they exist in large quantities and typically come with reference to the initial video, which makes alignment scale linearly in the number of GIFs. Aligning GIFs to their source videos is crucial, as it allows us to find non-selected segments, which serve as negative samples in training. In addition, videos provide a higher frame-rate and fewer compression artifacts, ideal for obtaining high quality feature representations.

Using these GIF websites, we collected a large-scale dataset with more than 120K animated GIFs and more than 80K videos, with a total duration of 7,379 hours. This is one to two orders of magnitude larger than the highlight datasets of [40] and [35]. We will show further statistics on the dataset after discussing the alignment process.

Alignment. We aligned the GIFs to their corresponding videos using frame matching. In order to do this efficiently, we encoded each frame with a perceptual hash based on the discrete cosine transform [41]. The perceptual hash is

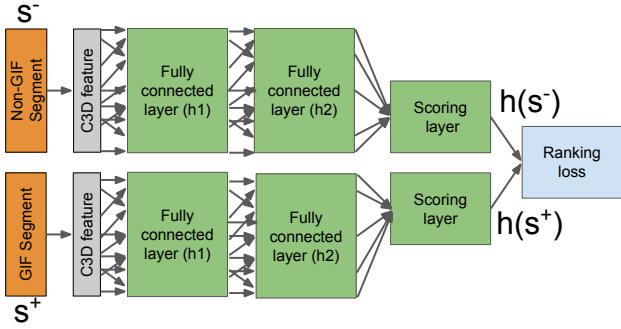


Figure 5. The architecture of our Robust Deep RankNet. We train the green-colored layers from scratch. Each hidden layer is followed by a ReLU non-linearity [29]. The final scoring layer is a linear function of the last hidden layer. The rank loss acts on pairs of segments and is non-zero, unless s^+ scores higher than s^- by a margin of 1. To emphasize that the loss acts on pairs of segments, we show the two passes separately, but we use a single network.

segment and computes its GIF-suitability score using the learned scoring function. We compute the score $h(s)$ for all segments $s \in \mathcal{S}$ and produce a ranked list of the segments for their suitability as an animated GIF.

Feature representation. Animated GIFs contain highly dynamic visual content; it is crucial to have feature representation that captures this aspect well. To capture both the spatial and the temporal dynamics of video segments, we use C3D [36] pretrained on the Sports-1M dataset [18] as our feature extractor. C3D extends the image-centric network architecture of AlexNet [22] to the video domain by replacing the traditional 2D convolutional layers with a spatio-temporal convolutional layer, and has been shown to perform well on several video classification tasks [36].

Inspired by previous methods using category specific models [35, 30], we optionally add contextual features to the segment representation. These can be considered meta-information, supplementing the visual features. They have the potential to disambiguate segment rankings and allow a model to score segments conditioned on the semantic category of a video. The features include the category label, a semantic embedding of the video tags (mean over their word2vec representation [28]) and positional features. For positional features, we use the timestamp, rank and the relative position of the segment in the video.

Problem formulation. A straightforward way to formulate our problem is by posing it as a classification problem, i.e., treat GIF and non-GIF segments as positive and negative examples, respectively, and build a binary classifier that separates the two classes of examples. This formulation, however, is inadequate for our problem because there is no clear cut definition of what is a good or a bad segment. Rather, there are various degrees of GIF suitability that can

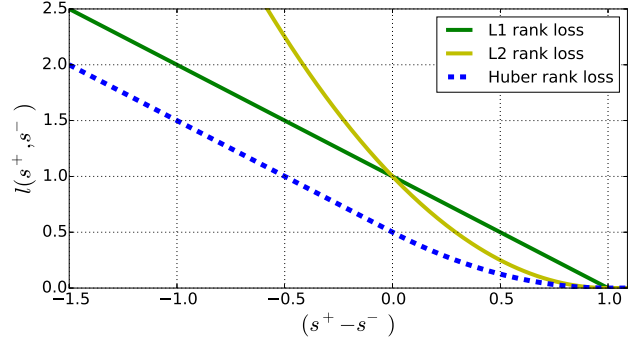


Figure 6. **Rank loss comparison.** Ours Huber rank loss combines the robustness w.r.t. to small margin violations of the l_2 loss with the robustness to outliers of the l_1 loss.

only be inferred by comparing GIF and non-GIF pairs.

A natural formulation is therefore posing it as a ranking problem. We can define a set of rank constraints over the dataset \mathcal{D} , where we require GIF segments s^+ to rank higher than non-GIF segments s^- , i.e.

$$h(s^+) > h(s^-), \quad \forall (s^+, s^-) \in \mathcal{D}.$$

This formulation compares two segments even if they are from different videos. This is problematic because a comparison of two segments is meaningful *only within the context of the video*, e.g., a GIF segment in one video may not be chosen as a GIF in another. To see this, some videos contain many segments of interest (e.g. compilations), while in others even the selected parts are of low quality. The notion of GIF suitability is thus most meaningful only within, but not across, the context of a single video.

To account for this, we revise the above video-agnostic ranking formulation to be video-specific, i.e.

$$h(s^+) > h(s^-), \quad \forall (s^+, s^-) \in S.$$

That is, we require a GIF segment s^+ to score higher than negative segments s^- that come from the same video only. Next we define how we impose the rank constraints.

Loss function. One possible loss function for the ranking problem is an l_p loss, defined as

$$l_p(s^+, s^-) = \max(0, 1 - h(s^+) + h(s^-))^p, \quad (1)$$

where $p = 1$ [17] and $p = 2$ [35, 24] are the most popular choices. The l_p loss imposes the ranking constraint by requiring a positive segment to score higher than its negative counterpart by a margin of 1. If the margin is violated, the incurred loss is linear in the error for the l_1 loss, while for the l_2 loss it is quadratic. One drawback of the l_1 loss, compared to the l_2 loss, is that it over-penalizes small margin violations. The l_2 loss does not have such problem, but it quadratically penalizes margin violations, and thus is more affected by outliers (see Figure 6).

Our dataset contains animated GIF contents created by online users, so some of the contents will inevitably be of low quality; these can be considered as outliers. This motivates us to propose a novel robust rank loss, which is an adaption of the Huber loss formulation [15] to the ranking setting. This loss gives a low penalty to small violations of the margin (where the ranking is still correct), and is more robust to outliers compared to the l_2 loss. We define our loss as

$$l_{\text{Huber}}(s^+, s^-) = \begin{cases} \frac{1}{2}l_2(s^+, s^-), & \text{if } u \leq \delta \\ \delta l_1(s^+, s^-) - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (2)$$

where $u = 1 - h(s^+) + h(s^-)$. Thus, if the margin is violated, the loss corresponds to a Huber loss, which is squared for small margin-violations and linear for stronger violations. The parameter δ defines the point at which the loss becomes linear. We illustrate the three different forms loss functions in Figure 6.

Considering the source of our dataset (social media), not all GIFs are expected to be of equal quality. Some might be casually created by beginners and from mediocre videos, while others are carefully selected from a high quality source. Thus, some GIFs can be considered more reliable as positive examples than others. We take this into account by making the parameter δ GIF dependent: We assign a higher value to δ to more popular GIFs. Our intuition behind this adaptive scoring scheme is that popular GIFs are less likely to be outliers and therefore do not require a loss that becomes linear early on.

Objective function. Finally, we define our objective as the total loss over the dataset \mathcal{D} and a regularization term with the squared Frobenius norm on the model weights \mathbf{W} :

$$L(\mathcal{D}, \mathbf{W}) = \sum_{S_i \in \mathcal{D}} \sum_{(s^+, s^-) \in S_i} l_{\text{Huber}}(s^+, s^-) + \lambda \|\mathbf{W}\|_F^2, \quad (3)$$

where λ is the regularization parameter.

4.3. Implementation Details

We experimented with various network architectures. While the loss function turned out to be crucial, we empirically found that performance remains relatively stable for different depths of a network. Thus, we opt for a simple 2 hidden layer fully-connected model, where each hidden unit is followed by a ReLU non-linearity [29]. We use 512 units in the first and 128 in the second hidden layer. The final prediction layer, which outputs $h(s)$, is a simple single linear unit, predicting an unnormalized scalar score. The final network has 2,327,681 parameters.

We minimize the objective in Eq. 3 using mini-batch stochastic gradient descent with backpropagation [32]. We use mini-batches of 50 pairs. In order to accelerate convergence, we apply Nesterov’s Accelerated Momentum [2] for

updating the weights. The momentum is set to 0.9 and $\lambda = 0.001$ (weight decay). We initialize training with a learning rate of 0.001 and reduce it every 10th epoch. The learning is stopped after 25 epochs. We apply dropout [34] regularization to the input (0.8) and after the first hidden layer (0.25). Dropout is a simple, approximate way to do model averaging that increases robustness of the model [34].

We obtain the training set segment pairs (s^+, s^-) by using all positive segments, randomly sampling $k = 4$ negatives per video, and combining them exhaustively. We limit the negatives in order to balance the positive-negative pairs per video. Finally, we obtain 500K pairs for training. For the Huber loss with a fixed δ we set $\delta = 1.5$ based on the performance on the validation set. For the adaptive Huber loss, we set $\delta = 1.5 + p$, where p is normalized viewcount proposed in [20].

In order to further decrease the variance of our model, we use model averaging, where we train multiple models from different initializations and average their predicted scores. The models were implemented using Theano [3] with Lasagne [6].

5. Experiments

We evaluate our method against several state-of-the-art methods on our dataset. In Section 5.3 we further evaluate cross-task performance on the highlight dataset of [35].

Evaluation metrics. Two popular performance metrics used in video highlight detection are mean Average Precision (mAP) [35] and average meaningful summary duration (MSD) [30]. Both mAP and MSD are, however, sensitive to video length: the longer the video is, the lower the score (think about finding the needle in the haystack). To compensate for a variety of video lengths in our dataset (see Figure 2), we propose a normalized version of MSD.

The normalized MSD (nMSD) corresponds to the relative length of the selected GIF at a recall rate of α . We define it as:

$$nMSD = \frac{|G^*| - \alpha|G^{gt}|}{|\mathcal{V}| - \alpha|G^{gt}|}, \quad (4)$$

where $|\cdot|$ denotes the length of a GIF or video, and G^* is the GIF with α recall w.r.t. the ground truth GIF G^{gt} . The score is normalized with the length of the ground truth GIF and the video \mathcal{V} , such that it is 0 if the selection equals to G^{gt} (is perfect), and 1 if the ground truth has the lowest predicted score. The added normalization helps make the scores of different videos more comparable, in contrast to mAP, which is strongly affected by the length of the ground truth GIF, relative to the video. To account for inaccuracies in segmentation we set $\alpha = 0.5$. For videos with multiple GIFs we use their mean nMSD as the video score. In addition to nMSD, we also evaluate performance using the traditional mAP.

5.1. Compared Methods

We compare our method to three state-of-the-art methods in highlight detection. We also provide an approximate upper bound that puts the results in perspective.

Domain-specific highlights [35]. We learn a domain-specific rankSVM per video category. Sun *et al.* [35] use an EM-like approach to handle long, loosely selected highlights. For our dataset, this problem does not occur because the GIFs are already short and focused. We therefore simply train a rankSVM [24] per video category using C3D features. We set $C = 1$ for all models.

Deep visual-semantic embedding [27]. We train a network using triplets of segment, true and random titles (s^+, t^+, t^-) . The titles are embedded into \mathbb{R}^{300} using word2vec [28]. In contrast to our method, the loss of [27] is defined over positive and negative titles and uses only positive segments (or images in their case) for training.

Category-specific summarization [30]. This approach trains a one-vs-all SVM classifier for each video category. Thus, the classifier learns to separate one semantic class from the others. At test time it uses the classifier confidence to assign each segment an importance score, which we use to obtain a ranked list.

Approximate upper bound. This bound provides a reference for how well an automatic method can perform. To obtain the upper bound, we first find all videos in our dataset that have animated GIFs from multiple creators. We then evaluate the performance of one GIF w.r.t. the remaining ones from the same video. Thus, the approximate upper bound is the performance users achieve in predicting the GIFs of other users. And it allows us to put the performance of automatic methods in perspective. We note, however, that this bound is only approximate because it is obtained in a very different setting than other methods.

5.2. Results and Discussions

Table 2 summarizes the results. Figure 7 shows qualitative results obtained using our method. As can be seen, our method (“Ours” in Table 2) outperforms the baseline methods by a large margin in terms of nMSD. The strongest baseline method is domain-specific rankSVM [35]. Their learning objective is similar to ours, *i.e.*, they use pairs of positive and negative segments from the same video for training. In contrast, two other baselines [30, 27] use a “proxy” objective, *i.e.*, learn semantic similarity of segments to video category [30] or segments to video title [27]. We believe this different training objective is crucial, allowing both our method and rankSVM [35] to significantly outperform the two baselines.

Domain-specific rankSVM [35] with C3D features performs fairly well; but our method outperforms it. We believe the reason for this performance difference is two-fold: (1) the l_2 loss in [35] is not robust enough to outliers; and

Method	nMSD ↓	mAP ↑
Joint embedding [27]	54.38%	12.36%
Category-spec. SVM [30]	52.98%	13.46%
Domain-spec. rankSVM [35]	46.40%	16.08%
Classification	61.37%	11.78%
Rank, video agnostic	53.71%	13.25%
Rank, l_1 loss	44.60%	16.09%
Rank, l_2 loss	44.43%	16.10%
Rank, Huber loss	44.43%	16.22%
Rank, adaptive Huber loss	44.58%	16.21%
Rank, adaptive Huber loss + context (Ours)	44.19%	16.18%
Ours + model averaging	44.08%	16.21%
Approx. bounds	38.77%	21.30%

Table 2. Experimental results. A lower nMSD and higher mAP represent better performance.

(2) the learning capabilities of [35] are limited by the use of a linear model, compared to highly nonlinear neural nets. Next, we analyze different configurations of our method in greater detail and discuss impacts of each design choice. The configurations differ in terms of used inputs, network architecture and objective.

What loss function is most robust? We analyze performance with different loss functions and training objectives discussed in Section 4.2. As expected, classification models always performs poorly compared to ranking models. Also, using video agnostic training data performs poorly. This indicates that the definition of a highlight is most meaningful within the video. When comparing l_1 loss and l_2 loss, we find that l_1 loss penalizes small margin violations (*i.e.*, $0 < h(s^+) - h(s^-) < 1$) too strongly, while the l_2 loss is affected by outliers. Our Huber rank loss avoids the two issues by combining the robustness to outliers (l_1 loss) and the decrease in the gradient for small margin violations (l_2 loss); it thus performs better than the other losses.

The role of context. Inspired by previous methods using category specific models [35, 30] we used contextual information as input to our model (category label, a semantic of the video tags and positional features). When comparing the performance with and without context, we find that they perform similarly (Table 2). We believe that most of the information about the context is already present in the segment representation itself. This is supported by [26] who show that the context can be extracted from the segment itself with high accuracy.

5.3. Cross Dataset Performance

As discussed, automatic GIF creation is related to video highlight detection. Of course, they are not identical: GIFs have a different focus and often depict funny or emotional content rather than highlights only. Nonetheless, it is inter-

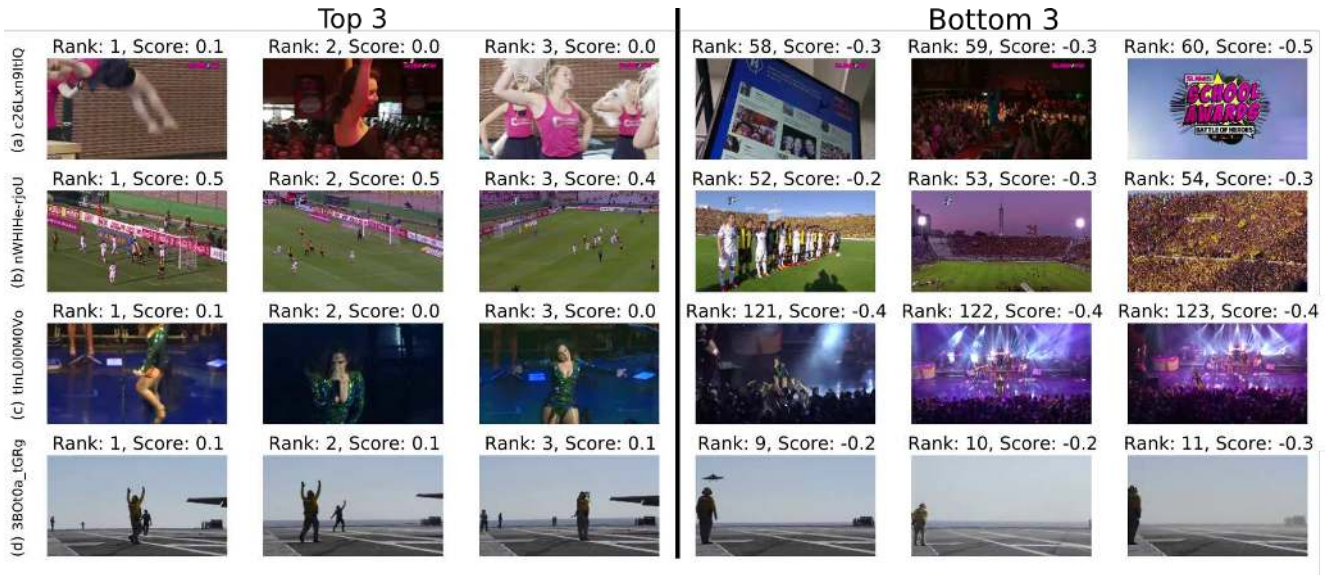


Figure 7. **Qualitative results.** Examples of top 3 and bottom 3 predicted segments. Our approach picks up aspects that are related to GIF suitability. For example, it learns that segments with people in motion are suitable for GIFs (e.g., (a) and (c)), while low contrast segments without any (main) objects are not (e.g., (a) the 4th image). It also scores segments showing the goal area of soccer games higher than the crowd in the stadium (b). We show a failure case (d): the network scores the segments with people on the ground higher than the landing plane (4th image). We provide more examples in GIF format on <http://video2gif.info>

esting to see how well our method generalizes to this task.

We evaluate our model on the dataset of [35], which contains videos from hand-selected categories such as surfing and skiing. We also evaluate the best performing baseline, domain-specific rankSVM [35], trained on our dataset and tested on the highlight dataset. The results are summarized in Table 3 (we borrow previously reported results [40, 35]).

Our method outperforms rankSVM by a large margin, which suggests that our model generalizes much better than the baseline method. It also significantly outperforms the method of Yang *et al.* [40], which trains an auto-encoder model for each domain. Sun *et al.* [35] tops the performance, but they use video category labels (which are hand-picked) and learn multiple models, one per category, directly on the highlight dataset. Instead, our method learns a single global model on the GIF data, with much more diverse video categories. Nonetheless, it shows competitive performance.

6. Conclusion

We introduced the problem of automatically generating animated GIFs from video, and proposed a Robust Deep RankNet that predicts the GIF suitability of video segments. Our approach handles noisy web data with a novel adaptive Huber rank loss, which has the advantage of being robust to outliers and able to encode the notion of content quality directly into the loss. On our new dataset of animated GIFs we showed that our method successfully learns to rank seg-

Category	Ours	rankSVM	Yang [40]	Sun [35]
skating	55.4%	26.2%	25%	61%
gymnastics	33.5%	25.5%	35%	41%
surfing	54.1%	45.0%	49%	61%
dog	30.8%	47.3%	49%	60%
parkour	54.0%	44.7%	50%	61%
skiing	32.8%	35.6%	22%	36%
Total	46.4%	37.9%	41.2%	53.6%

Table 3. Cross-dataset results (mAP). We train on our dataset and test on the video highlight dataset of [35]. Our method outperforms rankSVM and [40], which learns an unsupervised model for each domain. Sun *et al.* [35] performs best, but it is directly trained on their dataset and learns multiple models, one per category. Instead, we learn a single global model for GIF suitability.

ments with subtle differences, outperforming existing methods. Furthermore, it generalizes well to highlight detection.

Our novel Video2GIF task, along with our new large-scale dataset, opens the way for future research in the direction of automatic GIF creation. For example, more sophisticated language models could be applied to leverage video meta data, as not all tags are informative. Thus, we believe learning an embedding specifically for video tags may improve a contextual model. While this work focused on obtaining a meaningful ranking for GIFs, we only considered single segments. Since some GIFs range over multiple shots, it would also be interesting to look at when to combine segments or even do joint segmentation and selection.

References

- [1] S. Bakhshi, D. Shamma, L. Kennedy, Y. Song, P. de Juan, and J. J. Kaye. Fast, cheap, and good: Why animated GIFs engage us. In *CHI*, 2016. 1
- [2] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. *ICASSP*, 2013. 6
- [3] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of SciPy*, 2010. 6
- [4] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In *ECCV*. 2006. 2
- [5] S. Dhar, V. Ordonez, and T. Berg. High level describable attributes for predicting aesthetics and interestingness. *CVPR*, 2011. 2
- [6] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, et al. Lasagne: First release. 2015. 6
- [7] Y. Fu, T. Hospedales, T. Xiang, S. Gong, and Y. Yao. Interestingness Prediction by Robust Learning to Rank. *ECCV*, 2014. 2
- [8] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*. 2014. 2
- [9] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *CoRR*, abs/1312.4894, 2013. 3
- [10] H. Grabner, F. Nater, and L. Van Gool. Visual Interestingness in Image Sequences. *ACM MM*, 2013. 2
- [11] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. Van Gool. The Interestingness Of Images. *ICCV*, 2013. 1, 2
- [12] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating Summaries from User Videos. *ECCV*, 2014. 2
- [13] M. Gygli, H. Grabner, and L. Van Gool. Video Summarization by Learning Submodular Mixtures of Objectives. In *CVPR*, 2015. 1, 2
- [14] Y. Huang and L. Wang. Deep semantic ranking based hashing for multi-label image retrieval. *CVPR*, 2015. 3
- [15] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964. 6
- [16] Y. Jiang, Y. Wang, R. Feng, X. Xue, Y. Zheng, and H. Yang. Understanding and Predicting Interestingness of Videos. *AAAI*, 2013. 1
- [17] T. Joachims. Optimizing search engines using clickthrough data. *ACM SIGKDD*, 2002. 5
- [18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. *CVPR*, 2014. 3, 5
- [19] Y. Ke, X. Tang, and F. Jing. The design of high-level features for photo quality assessment. In *CVPR*, 2006. 2
- [20] A. Khosla, R. Hamid, C. Lin, and N. Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. *CVPR*, 2013. 2, 6
- [21] G. Kim, L. Sigal, and E. P. Xing. Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. *CVPR*, 2014. 2
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 5
- [23] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, 2015. 3
- [24] C.-P. Lee and C.-J. Lin. Large-scale linear rankSVM. *Neural computation*, 2014. 5, 7
- [25] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. *CVPR*, 2012. 2
- [26] Y.-I. Lin, V. I. Morariu, and W. Hsu. Summarizing While Recording : Context-Based Highlight Detection for Egocentric Videos. *ICCV Workshop*, 2015. 7
- [27] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo. Multi-Task Deep Visual-Semantic Embedding for Video Thumbnail Selection. In *CVPR*, 2015. 2, 3, 7
- [28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 2013. 5, 7
- [29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 5, 6
- [30] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. *ECCV*, 2014. 3, 5, 6, 7
- [31] M. Redi, N. O. Hare, R. Schifanella, M. Trevisiol, and A. Jaimes. 6 Seconds of Sound and Vision : Creativity in Micro-Videos. *CVPR*, 2014. 1, 2
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 1986. 6
- [33] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. TVSum : Summarizing Web Videos Using Titles. *CVPR*, 2015. 1, 2, 4
- [34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014. 6
- [35] M. Sun, A. Farhadi, and S. Seitz. Ranking Domain-Specific Highlights by Analyzing Edited Videos. *ECCV*, 2014. 1, 2, 3, 4, 5, 6, 7, 8
- [36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. *ICCV*, 2015. 2, 5
- [37] B. T. Truong and S. Venkatesh. Video abstraction. *ACM TOMCCAP*, 2007. 2
- [38] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning Fine-grained Image Similarity with Deep Ranking. *CVPR*, 2014. 3
- [39] X. Wang and A. Gupta. Unsupervised Learning of Visual Representations Using Videos. *ICCV*, 2015. 3
- [40] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised Extraction of Video Highlights Via Robust Recurrent Auto-encoders. *ICCV*, 2015. 1, 3, 8
- [41] C. Zauner. Implementation and benchmarking of perceptual image hash functions. 2010. 3