TASED-Net: Temporally-Aggregating Spatial Encoder-Decoder Network for Video Saliency Detection

Kyle Min Jason J. Corso
University of Michigan
Ann Arbor, MI 48109
{kylemin,jjcorso}@umich.edu

Abstract

TASED-Net is a 3D fully-convolutional network architecture for video saliency detection. It consists of two building blocks: first, the encoder network extracts low-resolution spatiotemporal features from an input clip of several consecutive frames, and then the following prediction network decodes the encoded features spatially while aggregating all the temporal information. As a result, a single prediction map is produced from an input clip of multiple frames. Frame-wise saliency maps can be predicted by applying TASED-Net in a sliding-window fashion to a video. The proposed approach assumes that the saliency map of any frame can be predicted by considering a limited number of past frames. The results of our extensive experiments on video saliency detection validate this assumption and demonstrate that our fully-convolutional model with temporal aggregation method is effective. TASED-Net significantly outperforms previous state-of-the-art approaches on all three major large-scale datasets of video saliency detection: DHF1K, Hollywood2, and UCFSports. After analyzing the results qualitatively, we observe that our model is especially better at attending to salient moving objects.

1. Introduction

Video saliency detection aims to model the gaze fixation patterns of humans when viewing a dynamic scene. Because the predicted saliency map can be used to prioritize the video information across space and time, this task has a number of applications such as video surveillance [12, 41], video captioning [27], video compression [11, 13], etc.

Previous state-of-the-art approaches for video saliency detection [3, 19, 39] largely depend on LSTMs [16] to aggregate information temporally. For example, OM-CNN [19] feeds spatial features from YOLO [31] and temporal features from FlowNet [10] into a two-layer LSTM. The leading state-of-the-art model, ACLNet [39], also uses a LSTM to aggregate spatial features guided by frame-wise image saliency maps. The strong performance of LSTM-based approaches over non-LSTM based ones suggests that aggregating information temporally boosts performance on video saliency detection.

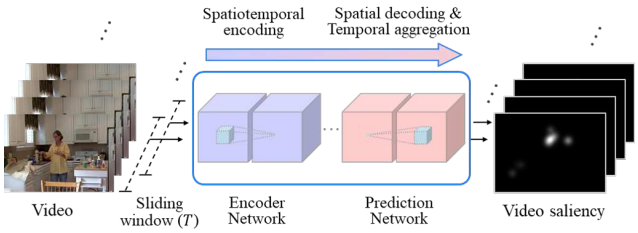

Figure 1: An illustration for the overall flow of TASED-Net. The encoder network extracts spatiotemporal features from an input clip of T frames. The prediction network decodes spatially and also aggregates temporally the features to produce a single saliency map of the last input frame. This process is applied in a sliding window fashion with a window size of T.

However, all of these LSTM-based, existing video saliency models fail to jointly process spatial and temporal information when predicting a saliency map from the extracted features. Specifically, either spatial decoding and temporal aggregation are performed separately, or only one of these two processes is considered for the final prediction. The existing works are hence unable to leverage the collective spatiotemporal information, which is expected to be important to video saliency [9, 25].

To this end, we propose a novel 3D fully-convolutional encoder-decoder network architecture for video saliency detection, which we call the Temporally-Aggregating Spatial Encoder-Decoder Network (TASED-Net). As described in Figure 1, TASED-Net progressively reduces the temporal dimensionality within both the encoder and the decoder subnetworks, which enables it to spatially upsample the encoded features and temporally aggregate all the information

as well. Similarly to other architectures designed for pixel-level tasks [1, 28, 33], TASED-Net compresses the spatial dimensions to extract high-level features at a low resolution, then upscales them to produce a full-resolution prediction map. On top of that, the decoder subnetwork performs temporal aggregation; we refer to it as the prediction network in our architecture since it jointly processes spatial and temporal information in a fully-convolutional way. TASED-Net predicts a single saliency map conditioned on a fixed number of previous frames, thus we apply it in a sliding-window fashion to predict a saliency map for every frame in the video.

Just as numerous 2D encoder-decoder architectures adopt VGG-16 [34] pre-trained on ImageNet [8] as their encoder network, we choose S3D [40] pre-trained on the Kinetics dataset [21] as the encoder network for TASED-Net. It has been shown by Xie *et al.* [40] that S3D is efficient and effective in extracting spatiotemporal features, and by Hara *et al.* [14] that the Kinetics dataset is sufficiently large for effective transfer-learning. Therefore, we expect that the encoder network of TASED-Net can fully benefit from the successful 3D convolutional network architecture and extremely large-scale video dataset.

For the prediction network, we first place a series of transposed convolution layers and max-unpooling layers for spatial upscaling, and then we use convolution layers for temporal aggregation. The tricky part is that the max-unpooling layers cannot reuse the pooling indices or switches [42] from the corresponding max-pooling layers since they have larger temporal receptive field than the max-unpooling layers. We introduce a new type of pooling operation, which we call *Auxiliary pooling*, that overcomes this non-trivial problem by adding extra max-poolings that can produce the properly-sized switches. *Auxiliary poolings* first reduce the temporal dimension of the input feature maps, and then obtain the appropriate switches for the matching max-unpooling layers. We compare *Auxiliary pooling* with two common upsampling operations, which are interpolation and transposed convolution (deconvolution), to demonstrate its effectiveness and necessity.

We comprehensively evaluate our architecture on three large-scale video saliency datasets: DHF1K [39], Hollywood2 [23, 24], and UCFSports [24, 32, 35]. Our results demonstrate that TASED-Net significantly outperforms previous state-of-the-art baselines on all three datasets. We believe that our novel architecture is effective in predicting video saliency because it jointly performs spatial decoding and temporal aggregation in a fully-convolutional way, instead of using separate recurrent units such as LSTM.

In summary, our main contributions are threefold:

- We develop a powerful end-to-end 3D fully-convolutional network for video saliency detection, comprised of an encoder network followed by a prediction network, which we name TASED-Net.
- We propose the novel concept of *Auxiliary pooling* which obtains switches with reduced temporal dimension so that max-unpooling layers of the prediction network can properly work.
- We comprehensively evaluate our proposed network on three large-scale datasets for video saliency and show the effectiveness of our joint modelling of spatial decoding and temporal aggregation.

2. Related Work

Recent Video Saliency Detection Models. Previous state-of-the-art video saliency models rely on optical flow or LSTM to utilize temporal information. STSConvNet [2] adopts a two-stream architecture where temporal information from optical flow is processed independently by a temporal stream. RMDN [3] uses spatiotemporal features extracted from C3D [37] and then aggregates temporal information in the long term with a subsequent LSTM. OM-CNN [19] first extracts spatial and temporal features from YOLO [31] and FlowNet [10] subnets, which represent objectness and motion respectively, and feed them into a two-layer LSTM. ACLNet [39] implements an attention module pre-trained on SALICON [20], a large-scale dataset for image saliency, and uses the frame-wise attention mask to encourage an LSTM to better capture dynamic saliency in the long term. Comparative results of these previous models are reported in Wang *et al.* [39]. Image saliency detection models can also be used to predict video saliency if used in a frame-wise manner for each frame of a video. However, unsurprisingly, even state-of-the-art image saliency detection models such as SalGAN [29], DVA [38], Deep Net [30], and SALICON [17] are significantly outperformed by ACLNet because they does not consider any temporal information.

Relevant 2D ConvNets. Deep 2D ConvNets have achieved great success in diverse areas of image analysis beyond image classification for the last few years, including object detection, instance segmentation, and image saliency detection. Among such successes, VGG-16 [34] pre-trained on ImageNet [8] has played a key role as an effective feature extractor for transfer learning. Another success in 2D ConvNets has been encoder-decoder networks [1, 28, 33]. For example, SegNet [1] improves a single-stream encoder-decoder architecture by upsampling the feature maps through max-unpooling with switches from the encoder network. Switches [42] are latent variables which record the locations of maximum activation. These variables are used by unpooling layers to partially-inverse the max-pooling operation. This method shows that max-unpooling is more suitable for decoding than other upsampling operations such as linear upsampling or even

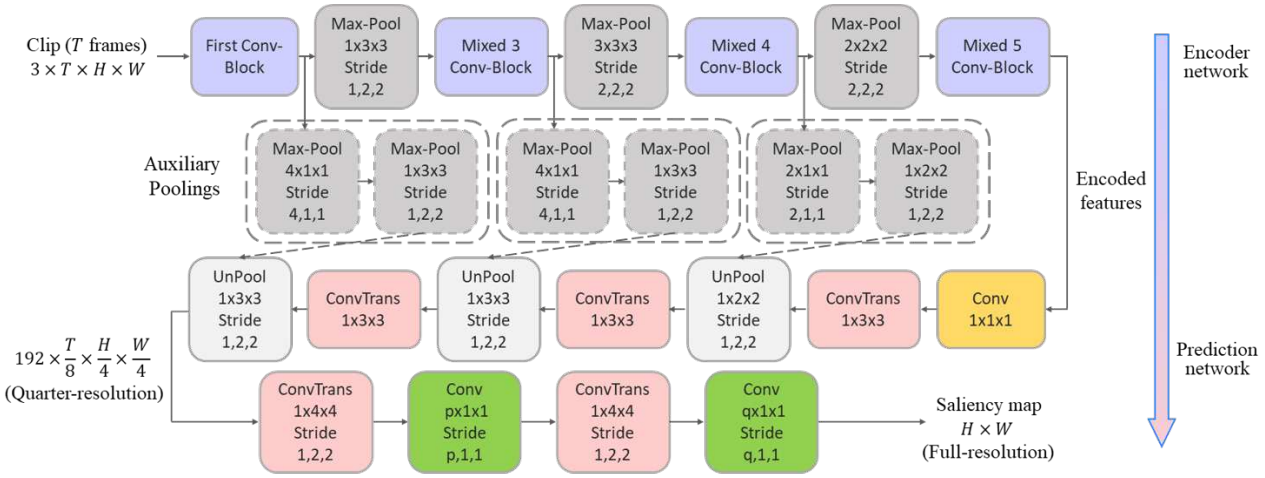

Figure 2: A detailed illustration of our proposed TASED-Net architecture. Violet boxes are convolutional operation blocks taken from the S3D [40] network pre-trained on the Kinetics dataset [21]. Pink boxes represent spatial decoding blocks. Green boxes are temporal convolutions that reduce the temporal dimension; within these blocks, p and q are set to reduce the temporal size of the output to 1. The $1 \times 1 \times 1$ convolutional operation in orange re-distributes the channel information of the encoded features. Because the unpooling layers operate only in spatial dimensions, switches [42] from the pooling layers cannot be reused. *Auxiliary poolings* are used as extra poolings to obtain properly-sized switches for the unpooling layers. Dashed arrows represent switch transfer. Note that *Auxiliary poolings* are not included in the main data stream.

learnable upsampling method through transposed convolution, which inspires our *Auxiliary pooling*.

Recent 3D ConvNets. 3D ConvNets have achieved state-of-the-art results in the action recognition task. Above all, 3D ConvNets inflated from 2D ConvNets are leading the field by leveraging successful 2D network architectures as well as their parameters. Carreira and Zisserman [5] propose I3D, which inflates the 2D convolutional filters of Inception [36] to produce a 3D ConvNet with strong performance. Xie *et al.* [40] further explore inflated 3D ConvNets by proposing a more computationally-efficient architecture called S3D. Hara *et al.* [14] experimentally show that various other inflated 3D ConvNets are also effective and predict that 3D ConvNets pre-trained on the Kinetics dataset [21] can retrace the success story of 2D ConvNets, i.e. that they can be used to initialize models for many other fields of video analysis, just as VGG-16 [34] has been applied to diverse image-based problems. We adopt S3D as the encoder network for our approach with the hope that it takes advantage of the successful architecture and the large-scale video dataset for effective transfer learning.

3. Approach

3.1. Architecture Overview

The overall flow of our proposed architecture is illustrated in Figure 1. We choose this design based on three assumptions: (i) saliency detection of any frame can be done well by only considering a fixed number of consequent past frames (we will call this number T throughout this paper); (ii) given an input of T frames, predicting a single saliency map for one specific time step is better than predicting maps for two or more steps at once; and (iii) there are enough number of frames in a video (specifically, the total number of frames of a video is not less than $2T - 1$).

The encoder network first encodes an input clip of T frames spatiotemporally; this provides a deep low-resolution feature representation. Then, the following prediction network decodes the features spatially while jointly aggregating temporal information to produce a full-resolution prediction map for a single time step. We note that unlike the previous state-of-the-art models that use LSTM, our method is conditioned on a fixed number of previous frames when predicting a saliency map. The prediction network is devised to coincide with the second assumption by predicting a single saliency map corresponding to the last frame of an input clip. Frame-wise saliency maps are predicted by applying the architecture in a sliding window fashion. In other words, S_t, a saliency map at t, is predicted given an input clip $(I_{t-T+1}, ..., I_t)$ for any $t \in \{T, ..., N\}$, where I_t is the frame at time step t and N is the total number of frames in the video.

The problem with this configuration is that the first $T - 1$ saliency maps are not predicted. Our workaround is to reverse the chronological order of the first $T-1$ input clips. That is, S_t for $t \in \{1, ..., T-1\}$ is predicted by conditioning on $(I_{t+T-1}, ..., I_t)$. As a result, our architecture can predict a frame-wise saliency map for every frame as long as our

third assumption that $N >= 2T - 1$ is satisfied.

TASED-Net has a common property with well-known image encoder-decoder networks that reduce and then upsample the spatial resolution [1, 28, 33]. The core difference of our model comes from temporal aggregation inside the prediction network, which requires extra operations that we call *Auxiliary pooling*. The architecture of TASED-Net, along with *Auxiliary pooling*, is explained in detail in the following sections.

3.2. Architecture specification

A detailed illustration of TASED-Net is depicted in Figure 2. An input clip is spatiotemporally encoded by 3D convolutional operation blocks of the encoder network taken from the S3D [40] network pre-trained on the Kinetics dataset [21]. The encoder network takes advantage of the successful 3D ConvNet architecture and the large-scale video dataset to extract rich encoded feature maps. We add a $1 \times 1 \times 1$ convolution after the convolutional blocks from S3D to re-distribute encoded information across the channel dimension.

Next, we describe the prediction network. We spatially upsample the encoded spatiotemporal features, leaving the time dimension alone, with a series of transpose convolutional layers and max-unpooling layers. At this point, we have only upsampled to a quarter of the original spatial resolution (quarter-resolution). Afterwards, we apply spatial transposed convolutions interspersed with temporal convolutions, which finally results in a full-resolution saliency map. The stride for these transposed convolution layers is $1 \times 2 \times 2$, so they double the spatial dimensions of the feature maps. The kernel sizes of the two temporal convolutions are $p \times 1 \times 1$ and $q \times 1 \times 1$, where p and q are set to 2 and $\frac{T}{16}$ respectively to aggregate all temporal information. Batch normalization [18] and ReLU [26] come after all the convolutional operations except the last layer. After the last convolution layer, a sigmoid function is applied to produce an intensity map of saliency. A more thorough description of the architecture can be found in Supplementary material.

3.3. Auxiliary pooling

In our architecture, we wish to leverage the effective reconstruction ability of max-unpooling layers, which have been used in state-of-the-art pixel-level segmentation models [1, 28]. However, implementing this in our architecture is non-trivial because the decoder (prediction network) never upsamples along the temporal dimension, which makes the temporal dimensions of switches [42] from the encoder incompatible with those from the decoder. Specifically, switches of the max-unpooling layers and their corresponding max-pooling layers have different temporal sizes. In order to obtain switches with the proper sizes for the max-unpooling layers, extra processing steps are required.

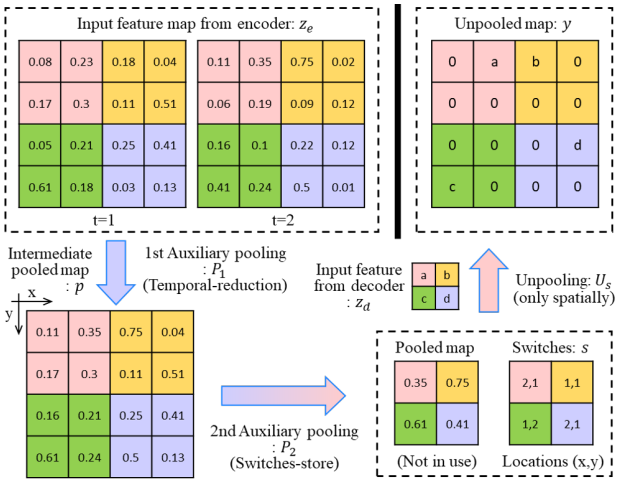

Figure 3: One example of how *Auxiliary poolings* work in $2 \times 2 \times 2$ input feature map from encoder z_e. The first *Auxiliary pooling* P_1 applies $2 \times 1 \times 1$ max-pooling to obtain temporally-reduced pooled map p. The second *Auxiliary pooling* P_2 applies $1 \times 2 \times 2$ max-pooling to store switches s in the reduced temporal dimension. As a result, the corresponding unpooling layer U_s with $1 \times 2 \times 2$ kernel can unpool the input feature map from decoder z_d spatially, which produces y.

For each max-unpooling layers, we add two sequential extra pooling layers, which we call *Auxiliary poolings*. The first *Auxiliary pooling* receives the input feature map from the encoder and reduces the temporal length of the feature map. Then, the following *Auxiliary pooling*, whose kernel works only spatially, stores the proper switches for the matched unpooling layer which also only works in spatial dimension. These blocks of two sequential *Auxiliary poolings* make it possible for the decoder to reconstruct spatial information effectively by using the stored switches. Note that *Auxiliary poolings* are only used for storing switches and are not included in the main data stream. A detailed illustration of how *Auxiliary poolings* truly work is described in Figure 3. A general pooling operation P takes an input feature map z and produces pooled map p with switches s which record the location of maximum activation within the input: $[p, s] = P(z)$. The first *Auxiliary pooling* is applied to obtain the intermediate temporally-reduced pooled map p: $[p, \text{-}] = P_1(z_e)$ (hyphen: variables not in use). The second *Auxiliary pooling* is applied to store switches in the reduced temporal domain: $[\text{-}, s] = P_2(p)$. The matched unpooling operation U_s unpools the input feature map from decoder only spatially using the switches s: $y = U_s(z_d)$. A more detailed input and output sizes can be found in Supplementary material. The necessity of *Auxiliary pooling* in TASED-Net and its variants are also further discussed in Section 4.4.

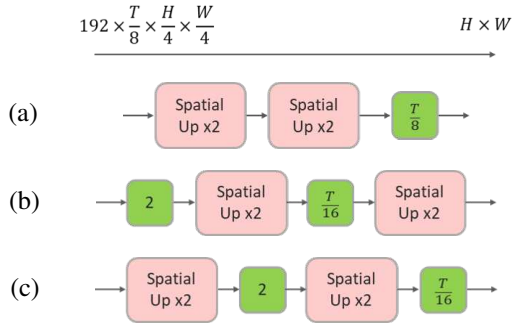

Figure 4: Different temporal aggregation strategies. Pink boxes are transposed convolutions that double each spatial dimension of the input feature maps. Green boxes are temporal convolutions that reduce the temporal dimension by a factor of the number written in each box.

3.4. Temporal aggregation strategy

Temporal aggregation takes a spatiotemporally encoded feature map, whose spatial resolution is a quarter of the full video resolution, and performs the following two operations: reducing the time dimension of the input features to 1, and upscaling the spatial dimensions to full-resolution. There exist a variety of strategies that perform the required spatial upsampling and temporal reduction operations in different orders; we depict a few in Figure 4. The first strategy, late aggregation, performs two spatial upsampling operations followed by one temporal convolutional operation that performs temporal dimension reduction. The second strategy, early two-step aggregation, performs one temporal convolution before each spatial upsampling operation. The final strategy, late two-step aggregation, performs one temporal convolution after each spatial upsampling operation. We found that late two-step aggregation performs best (see Section 4.2), so we implemented it in TASED-Net.

4. Evaluation

4.1. Experiments setup

Datasets. We evaluate our method on three standard datasets: DHF1K [39], Hollywood2 [23, 24], and UCFSports [24, 32, 35]. These datasets and some others are compared in terms of variety, scalability, and generality by Wang *et al.* [39], and we choose the DHF1K dataset as our main benchmark (i.e. we focus our analysis on this dataset) because it includes the most general and diverse scenes with various types of objects, motion, and backgrounds out of the aforementioned datasets. It consists of 1K videos with around 600K frames; 300 videos are preserved as a test set with no public ground-truth annotations of human eye fixation points. There is a public server for reporting results on the test set for fair evaluation. The Hollywood2 dataset contains 1,707 videos focusing on human actions in movie scenes, and the UCFSports dataset contains of 150 videos of human actions in sports. We believe that our selection of three datasets is sufficient to show the effectiveness and generality of our approach.

Training/testing process. For training TASED-Net, clips with T consequent frames are randomly but densely sampled from a video. Note that this sampling scheme is valid because our model predicts each saliency map independently. Each frame is resized to 224×384. We train our network with a batch size of 40 on 600 videos from the DHF1K training set through the SGD algorithm with 0.9 momentum in an end-to-end manner. The learning rate is fixed at 0.001 for the encoder network. For the prediction network, the learning rate starts at 0.1 and decays twice by a factor of 10 when the validation loss does not decrease for a certain number of steps that depends on T. For TASED-Net with $T = 32$, the first decaying point is at step 750, the second one is at step 950. The whole training process of 1K iterations takes less than 3 hours. Evaluation on the whole validation set takes a lot of time due to a large number of frames (60K in the validation set of the DHF1K dataset), so we uniformly sample 2K clips to approximate the validation loss. We choose Kullback-Leibler (KL) divergence as the loss function, which Jiang *et al.* [20] have shown to be effective for training saliency models. When testing, we apply TASED-Net in a sliding-window fashion to predict a frame-wise saliency map for every frame of all videos within the dataset. It takes around 0.06s to process each frame.

Evaluation metrics. Following prior work [39], we report our model's performance using the following metrics: (i) Normalized Scanpath Saliency (NSS), (ii) Linear Correlation Coefficient (CC), (iii) Similarity (SIM), (iv) Area Under the Curve by Judd (AUC-J), and (v) Shuffled-AUC (s-AUC). NSS and CC estimate a linear correlation between the prediction and ground-truth fixation map. SIM is for computing similarity between two histograms, and AUC-J and s-AUC are variants of the well-known AUC metric. Higher scores on each metric indicate better performance.

4.2. Evaluation on DHF1K

Since the ground-truth annotations for the test set of DHF1K [39] are hidden for fair comparison, we first evaluate variants of our model on the validation set. The performance of TASED-Net with different T and temporal aggregation strategies are compared in Table 1. The results indicate that TASED-Net with $T = 32$ and late two-step aggregation performs the best since this configuration achieves the best performance across most metrics (it has 21.2M Params and 63.2G FLOPs; more results on different T's are provided in Section 4.5). We believe that late two-step aggregation performs better than early two-step aggregation because the feature maps used in spatial upscaling have a

Aggregation strategy	NSS	CC	SIM	AUC-J	s-AUC
Late-aggregation (16)	2.555	0.460	0.340	0.892	0.712
Late-aggregation (32)	2.618	0.467	0.343	**0.897**	0.713
Early two-step (16)	2.591	0.464	0.343	0.894	0.708
Early two-step (32)	2.673	0.475	0.361	0.891	0.706
Late two-step (16)	2.622	0.469	0.349	0.892	0.713
Late two-step (32)	**2.706**	**0.481**	**0.362**	0.894	**0.718**

Table 1: Performance comparison of TASED-Net with different Ts (shown in parentheses) and temporal aggregation strategies on the validation set of DHF1K [39]. The late two-step approach performs the best since it utilizes temporally rich features while avoiding overfitting.

larger size in the temporal dimension. That is, late two-step aggregation performs better thanks to temporally richer feature maps. Interestingly, late aggregation performs poorly despite having the richest features, probably due to overfitting. In addition, we observe that the scores drop by 0.5 NSS (0.06 CC, 0.04 SIM, 0.015 AUC) without Kinetics pre-training for most cases. This shows the effectiveness of Kinetics pre-training. For the rest of the paper, we report the performance of TASED-Net with $T = 32$, late two-step aggregation, and pre-training.

Next, we submitted our results to the DHF1K online benchmark [39]. The performance of TASED-Net and previous state-of-the-art methods on the test set of DHF1K is reported in Table 2. Our model outperforms other methods by a wide margin across all evaluation metrics. We note that ACLNet [39], the leading state-of-the-art method, is arguably better-primed for saliency detection than TASED-Net—it has a component pre-trained on an image-saliency dataset, SALICON [20], whereas we pre-train the encoder network of TASED-Net on an action recognition dataset. The higher performance of TASED-Net suggests that pre-training on a large-scale video dataset plays a significant role in performing well on other tasks in general. We also want to point out that TASED-Net has a much smaller network size (82MB v.s. 252MB). Interestingly, our AUC-J score does not increase much compared to the other metrics. This phenomenon has already been reported by Bylinskii *et al.* [4], who suggest that AUC-J is less capable of discriminating between different high-performing saliency models because it is invariant to monotonic transformations.

To perform a qualitative analysis, we compare the performance of TASED-Net to the leading state-of-the-art method, ACLNet [39], on videos from the validation set of the DHF1K dataset. We observe that we can easily recognize the differences between the results of each model when the difference of NSS scores between the two is greater than 0.5. Based on this gap, TASED-Net outperforms ACLNet on 37 out of the 100 videos in the validation set, while ACLNet outperforms TASED-Net only on 7 videos. Qualitative results of our model and ACLNet for the better and worse cases are given in Figure 5 (see Supplementary material for more examples of qualitative results). As shown in (a) and (b) in Figure 5, TASED-Net seems highly sensitive to salient moving objects and less sensitive to background objects, which is consistent with the goal of video saliency in general. On the other hand, ACLNet seems to put more weight on spatially conspicuous objects, so sometimes it attends to distracting background objects. This makes the saliency map predicted by ACLNet a lot blurrier than ours in many cases.

Method \ Metric	NSS	CC	SIM	AUC-J	s-AUC
GBVS [15]	1.474	0.283	0.186	0.828	0.554
STSConvNet [2]	1.632	0.325	0.197	0.834	0.581
Deep Net [30]	1.775	0.331	0.201	0.855	0.592
SALICON [20]	1.901	0.327	0.232	0.857	0.590
OM-CNN [19]	1.911	0.344	0.256	0.856	0.583
DVA [38]	2.013	0.358	0.262	0.860	0.595
SalGAN [29]	2.043	0.370	0.262	0.866	0.709
ACLNet [39]	2.354	0.434	0.315	0.890	0.601
TASED-Net	**2.667**	**0.470**	**0.361**	**0.895**	**0.712**

Table 2: Comparison of TASED-Net with other state-of-the-art methods on the test set of DHF1K. TASED-Net significantly outperforms all the previous methods across all the evaluation metrics by a large margin.

We have observed that for videos where the ground-truth fixation points are scattered across a large area, our model quantitatively performs worse than ACLNet. This is because ACLNet generally predicts blurrier maps that better fit highly-scattered fixation points. However, we also find that ground-truth fixation points are unstable for these videos. For example, in (c) of Figure 5, the fixation points do not smoothly follow the carp, but instead flicker and jump between different carp. In (d), because the foreground object is so large, fixation points tend to move around the object. Furthermore, different subjects do not fixate on the same part of a large object. In these cases, it is hard to say that the ground-truth fixation points represent general human gaze behavior well. Therefore, we strongly believe that a larger number of human subjects is needed to properly annotate videos where the fixation points are frequently scattered across a large area. We also believe that a larger and more comprehensive dataset with more diverse scenes is needed to cover general situations where the salient moving objects are not the only dominant information. More qualitative results can be found in Supplementary material.

4.3. Performance on other datasets

We further test our model on two commonly used public datasets, which are Hollywood2 [23, 24] and

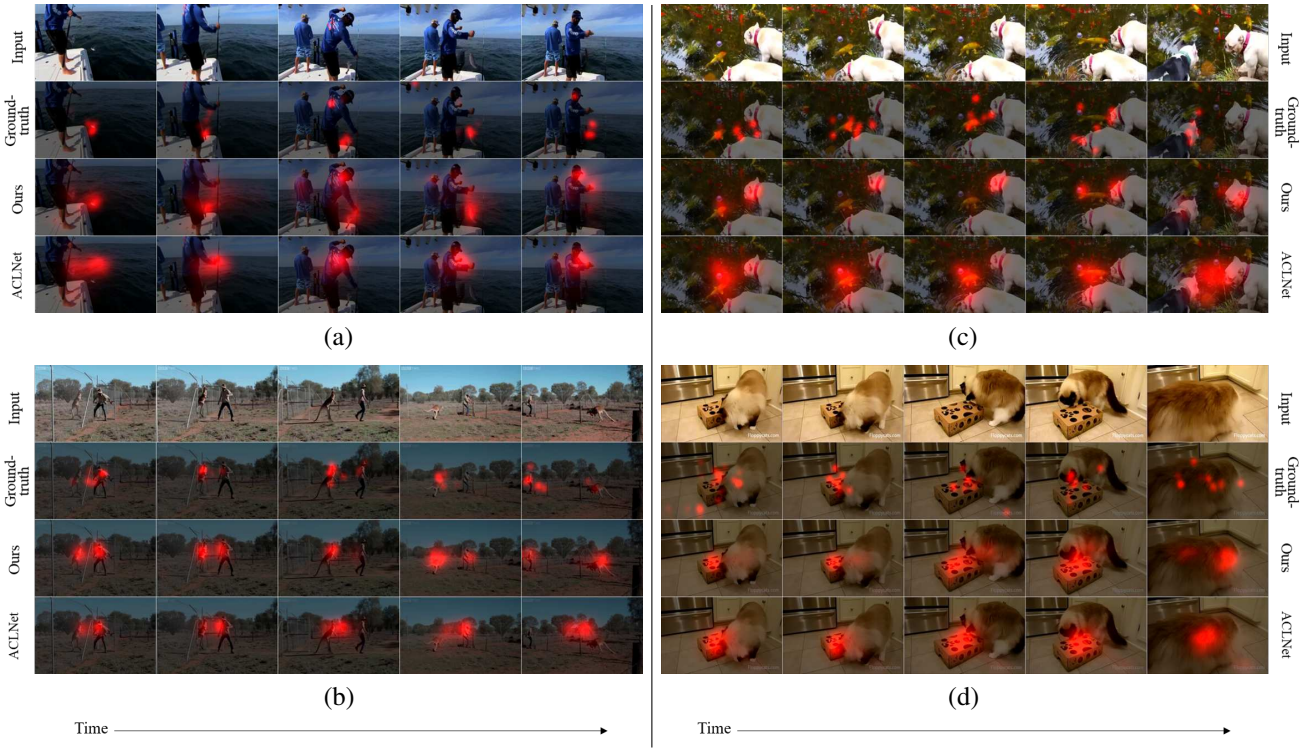

Figure 5: Qualitative results of our TASED-Net and the main competitive model ACLNet [39] on the DHF1K validation set. We observe that the differences between the two results are easily identified when the difference between NSS scores is greater than 0.5. Our method beats ACLNet by this margin on 37 videos, and ACLNet beats our method by this margin on 7 videos. We show improved results on two clips from the 37 videos ((a) and (b)), and worse results on two clips from the 7 videos ((c) and (d)). As seen in (a) and (b), TASED-Net attends to the salient moving objects very well, even when there are many background objects. In (c) and (d), it seems that the ground-truth fixation points do not represent general human gaze behavior well. For example, in (c), the fixation points flicker and jump around on different carp. In (d), only small parts of the foreground objects (the body of the cat) are fixated on. More examples are available in Supplementary material.

UCFSports [24, 32, 35]. To leverage the relatively large scale of the DHF1K dataset, we first pre-train TASED-Net on DHF1K, and then fine-tune on Hollywood2 or UCFSports. For short videos with fewer than $2T - 1 = 63$ frames, we simply loop those videos to fit in with our method. Table 3 compares our model with various previous state-of-the-art approaches. TASED-Net again achieves the best performance on each dataset across most of the metrics.

4.4. Necessity of Auxiliary pooling

As discussed earlier, *Auxiliary poolings* are needed for the max-unpooling layers to work in our proposed architecture. Here, we compare two possible variants of *Auxiliary pooling*. The first variant, which we call TASED-Net-tri, replaces all the max-unpooling layers with trilinear upsampling (interpolation). The second variant, which we name TASED-Net-trp, replaces the max-unpooling layers with transposed convolutions (deconvolution). Note that these two variants do not require *Auxiliary poolings*. Table 4

	Method \ Metric	NSS	CC	SIM	AUC-J	s-AUC
Hollywood2	STSConvNet [2]	1.748	0.382	0.276	0.863	0.710
	SALICON [20]	2.013	0.425	0.321	0.856	0.711
	Deep Net [30]	2.066	0.451	0.300	0.884	0.736
	OM-CNN [19]	2.313	0.446	0.356	0.887	0.693
	DVA [38]	2.459	0.482	0.372	0.886	0.727
	ACLNet [39]	3.086	0.623	**0.542**	0.913	0.757
	TASED-Net	**3.302**	**0.646**	0.507	**0.918**	**0.768**
UCFSports	GBVS [15]	1.818	0.396	0.274	0.859	0.697
	Deep Net [30]	1.903	0.414	0.282	0.861	0.719
	OM-CNN [19]	2.089	0.405	0.321	0.870	0.691
	DVA [38]	2.311	0.439	0.339	0.872	0.725
	ACLNet [39]	2.567	0.510	0.406	0.897	0.744
	TASED-Net	**2.920**	**0.582**	**0.469**	**0.899**	**0.752**

Table 3: Comparison of TASED-Net to state-of-the-art methods on the test sets of Hollywood2 and UCFSports. High scores for our model across most of the metrics prove the effectiveness of our model.

Method \ Metric	NSS	CC	SIM	AUC-J	s-AUC
TASED-Net-tri	2.452	0.448	0.337	0.891	0.702
TASED-Net-trp	2.598	0.470	0.353	0.894	0.707
TASED-Net	2.706	0.481	0.362	0.894	0.718

Table 4: Comparison of variants of *Auxiliary pooling* on the validation set of DHF1K. TASED-Net-tri and TASED-Net-trp do not utilize *Auxiliary pooling* because they replace unpooling layers with trilinear upsampling (interpolation) and transposed convolution (deconvolution), respectively. TASED-Net perform better, which demonstrates the effectiveness of *Auxiliary pooling*.

Method \ Metric	NSS	CC	SIM	AUC-J	s-AUC
TASED-Net (4)	2.434	0.441	0.327	0.887	0.689
TASED-Net (8)	2.585	0.460	0.348	0.889	0.696
TASED-Net (16)	2.622	0.469	0.349	0.892	0.713
TASED-Net (32)	**2.706**	**0.481**	**0.362**	**0.894**	**0.718**
TASED-Net (48)	2.636	0.472	0.348	**0.894**	0.708
TASED-Net (64)	2.554	0.459	0.336	0.893	0.702

Table 5: Performance of TASED-Net with different T's (number in bracket) on the validation set of DHF1K. The clear trend is observed. TASED-Net performs well when $T = 32$.

compares these variants and shows that TASED-Net without *Auxiliary pooling* operations performs poorly. In other words, we discover that replacing max-unpooling layers does not work well although TASED-Net-tri and TASED-Net-trp may seem more straightforward. This proves the effectiveness and necessity of *Auxiliary pooling* in TASED-Net.

In addition, we apply our temporally-aggregating scheme to many other powerful architectures including FCN [22], U-Net [33], Deeplab [6, 7], which have achieved great success in dense prediction tasks. The results are reported in Supplementary material. The unsatisfying results justify our architecture with the proposed *Auxiliary pooling*.

4.5. Other observations

We observe that stacking multiple transposed convolution layers with stride $1 \times 1 \times 1$ within each spatial decoding block in the prediction network does not boost performance. To demonstrate this, we augment TASED-Net by adding two more transposed convolutional layers to each spatial decoding block. This denser (or deeper) version approximately increases the network size by 40%, so we expect that it would yield better performance by finely decoding spatial information. However, we found that it actually yields slightly worse performance (see Supplementary material). This might be because spatial decoding is of less importance in video saliency detection than in other tasks where more precise pixel-wise outputs are required (e.g. video segmentation). Therefore, video saliency models may not necessarily benefit from stronger spatial decoding capabilities. Otherwise, it may be due to overfitting. To better understand how this phenomenon is affected by dataset size and task formulation, we would have to test the denser TASED-Net on larger datasets and alternative tasks like video segmentation.

It is also observed that predicting multiple saliency maps all at once for each sliding window decreases the overall performance when compared to predicting a single saliency map. We believe that this is because increasing the prediction space makes it harder for the decoder (prediction network) to be optimized. It shows that our temporally-aggregating scheme is more appropriate for the video saliency detection.

Furthermore, we observe that TASED-Net with T larger than 32 performs worse than when $T = 32$ (see Table 5). These results may indicate that it is sufficient to consider a fixed number of past frames for video saliency detection. However, they could also be a result of overfitting. TASED-Net with T smaller than 32 also performs worse than when $T = 32$, which implies that it is necessary to consider enough number of past frames with a duration of about one second for video saliency detection. We believe that further optimization on T is not necessary for this paper.

5. Conclusion

We have presented TASED-Net as a novel fully-convolutional architecture for video saliency detection. The main idea is simple but effective: spatially decoding the features extracted by the encoder while jointly aggregating all the temporal information in order to produce a single full-resolution prediction map. We also propose the new concept of *Auxiliary pooling*, which enables our architecture to leverage the benefits of max-unpooling layers for reconstruction. TASED-Net significantly outperforms previous state-of-the-art methods on major video saliency detection datasets, which demonstrates the benefits of performing spatial decoding and temporal aggregation in a fully-convolutional way, as well as the benefits of conditioning on a limited amount of past information when predicting video saliency. Finally, we comprehensively analyze TASED-Net with many variants, and show that our proposed *Auxiliary pooling* is necessary and effective.

Acknowledgement. We thank Ryan Szeto for his valuable feedback and comments. We also thank Stephan Lemmer, Mohamed El Banani, and Luowei Zhou for their discussions. This research was, in part, supported by NIST grant 60NANB17D191.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[2] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, 20(7):1688–1698, 2018.

[3] Loris Bazzani, Hugo Larochelle, and Lorenzo Torresani. Recurrent mixture density network for spatiotemporal visual attention. *arXiv preprint arXiv:1603.08199*, 2016.

[4] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[9] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[11] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Trans. Image Processing*, 19(1):185–198, 2010.

[12] Fahad Fazal Elahi Guraya, Faouzi Alaya Cheikh, Alain Tremeau, Yubing Tong, and Hubert Konik. Predictive saliency maps for surveillance videos. In *Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on*, pages 508–513. IEEE, 2010.

[13] Hadi Hadizadeh and Ivan V Bajic. Saliency-aware video compression. *IEEE Transactions on Image Processing*, 23(1):19–33, 2014.

[14] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pages 18–22, 2018.

[15] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Lai Jiang, Mai Xu, and Zulin Wang. Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm. *arXiv preprint arXiv:1709.06316*, 2017.

[20] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015.

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009.

[24] Stefan Mathe and Cristian Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1408–1424, 2015.

[25] Hermann J Müller and Jemima Maxwell. Perceptual integration of motion and form information: Is the movement filter involved in form discrimination? *Journal of Experimental Psychology: Human Perception and Performance*, 20(2):397, 1994.

[26] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[27] Tam V Nguyen, Mengdi Xu, Guangyu Gao, Mohan Kankanhalli, Qi Tian, and Shuicheng Yan. Static saliency vs. dynamic saliency: a comparative study. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 987–996. ACM, 2013.

[28] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation.

In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[29] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[30] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[32] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer vision in sports*, pages 181–208. Springer, 2014.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[38] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE Trans. Image Process*, 27(5):2368–2378, 2018.

[39] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018.

[40] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[41] Tong Yubing, Faouzi Alaya Cheikh, Fahad Fazal Elahi Guraya, Hubert Konik, and Alain Trémeau. A spatiotemporal saliency model for video surveillance. *Cognitive Computation*, 3(1):241–263, 2011.

[42] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.