Hedging Static Saliency Models to Predict Dynamic Saliency

Yasin Kavak[a], Erkut Erdem[a,*], Aykut Erdem[a]

[a]*Department of Computer Engineering, Hacettepe University, Ankara, Turkey*

Abstract

In recent years, many computational models for saliency prediction have been introduced. For dynamic scenes, the existing models typically combine different feature maps extracted from spatial and temporal domains either by following generic integration strategies such as averaging or winners take all or using machine learning techniques to set each features importance. Rather than resorting to these fixed feature integration schemes, in this paper, we propose a novel weakly supervised dynamic saliency model called HedgeSal, which is based on a decision-theoretic online learning scheme. Our framework uses two pretrained deep static saliency models as experts to extract individual saliency maps from appearance and motion streams, and then generates the final saliency map by weighted decisions of all these models. As visual characteristics of dynamic scenes constantly vary, the models providing consistently good predictions in the past are automatically assigned higher weights, allowing each expert to adjust itself to the current conditions. We demonstrate the effectiveness of our model on the CRCNS, UCFSports and CITIUS datasets.

Keywords: Dynamic saliency, hedge algorithm, decision theoretic online learning, feature integration

*Corresponding author at the Department of Computer Engineering, Hacettepe University, Beytepe, Cankaya, Ankara, Turkey, TR-06800. Tel: +90 312 297 7500, 146. Fax: +90 312 297 7502.

Email addresses: yasinkavak@cs.hacettepe.edu.tr (Yasin Kavak), erkut@cs.hacettepe.edu.tr (Erkut Erdem), aykut@cs.hacettepe.edu.tr (Aykut Erdem)

1. Introduction

Visual saliency estimation, the task of predicting where humans look at images, has been an active research area in the computer vision community [1] over the past 40 years. Especially with the introduction of new benchmark datasets and new methods such as deep learning, we are witnessing increasingly more sophisticated models. Despite this surge of interest, however, saliency prediction has not been solved yet as the existing saliency models are not fully capable of describing all of the phenomena observed in the visual attention studies [2].

Most of the existing approaches for saliency prediction focus primarily on static images and thus predict eye fixations without considering dynamic scene characteristics such as apparent motion. For instance, the early static models (e.g. [3, 4]) encode local contrast information based on differences of very low-level visual features like intensity, color and orientation. Some more complex models (e.g. [5]) employ features that encode faces and pedestrians in order to include some known top-down factors, yet these features are also based on static object detectors. Another key issue is the so-called feature integration problem. The early models consider very simple fusion strategies such as taking the mean or the product of the individual feature channels, however the progress within the last decade has led to more sophisticated solutions which aim at learning an optimal strategy from training data. While the first group of such approaches uses hand-crafted features and shallow machine learning techniques like SVM [6], AdaBoost [7], the current state-of-the-art models for static saliency prediction are all based on convolutional neural networks and trained in an end-to-end fashion [2]. These deep saliency models, however, require a vast amount of data either in their pre-training or training phases.

As compared to its static counterpart, dynamic saliency prediction addresses the problem of estimating where humans fixate their eyes in videos, and is a far more challenging problem. In literature, the number of studies on dynamic saliency is far smaller than that on static saliency. The majority of the dynamic

models commonly consider separate appearance (spatial) and motion (temporal) streams, extract features from these streams and finally combine them to obtain a final saliency map (e.g. [8, 9, 10, 11, 12, 13]). In this sense, the models mainly differ from each other by the features they use and their feature integration strategies. Regarding feature integration in dynamic saliency, existing models employ either very generic integration strategies like the ones for static saliency such as averaging or winners take all, or consider very ad hoc solutions to combine features from the appearance and motion streams. But, these naive approaches greatly limit the overall performance. As a remedy, recent models try to solve this issue by learning each feature's contribution to the overall saliency directly from training data. However, for dynamic scenes, this is still not sufficient since these methods still associate a (learned) constant weight with each feature. On the other hand, in regards to human visual system, visual attention mechanisms exhibit completely different, more complex behavior in dynamic scenes than static scenes. For instance, in [14], the authors showed that humans fixate their eyes at different people or objects on videos and static images, or according to the camera motion, the fixations on videos and images are not on the objects exist in the scenes but rather lies on the anticipated directions. Moreover, the central bias which has a strong effect on static images has lesser impacts in dynamic scene fixations. All these observations suggest that to fully deal with the challenges of dynamic scenes and to achieve better prediction accuracies, the weights of the visual features should be defined in a more flexible way and should change over time. That is, it is important to consider integration schemes that can adapt themselves according to changes in the visual content to combine different features in the best possible way.

In this paper, we propose a novel weakly supervised dynamic saliency model which is built upon a set pretrained deep static saliency models processing the appearance and motion streams. In short, we use these static models as experts within our framework and combine their results by considering a decision theoretic formulation to infer the master saliency map. Using decision theory helps us to define certain reliability scores to each one of our expert models

according to some optimality conditions with respect to the end results and accordingly allows the integration step to be carried out in an adaptive manner. In literature, there are several decision-theoretic solutions exists for defining these optimality conditions such as minimum probability of error. Within our formulation, we specifically follow the decision theoretic online learning scheme known as the Hedge algorithm [15, 16], hence, we refer to our proposed approach as HedgeSal throughout our paper. Specifically, we first extract appearance and motion streams of a given video, and then run SALICON [17] and SalNet [18], two deep static saliency models, on individual frames, and generate the final saliency map by the weighted decisions of all these models. Each one of our experts captures different visual characteristics of the scene, the ones which provide consistently good predictions in the previous frames are given higher weights in the current frame, increasing the prediction accuracy. Here, it is important to mentioned that a recent trend in the dynamic saliency literature is to employ deep learning to train saliency models in an end-to-end manner [19, 20, 21]. However, all these models are trained in a supervised manner and need a large amount of annotated video data with the groundtruth eye fixations, which is in general very hard to obtain.

In summary, our main contributions in this paper are as follows:

1. We propose a novel weakly supervised dynamic saliency model that integrates the results of several deep static saliency models to predict where humans look at videos.
2. We develop an adaptive feature integration scheme which depends on a decision theoretic online learning mechanism.
3. We perform an extensive set of experiments on three different benchmark datasets to demonstrate the effectiveness of the proposed models against the state-of-the-art models.

The paper is structured as follows: In Section 2, we give a brief discussion about the existing saliency models in the literature. In Section 3, we introduce our adaptive dynamic saliency model. After that, in Section 4, we present

our experimental results together with the details of the benchmark datasets, evaluation metrics used in the experiments. Finally, in Section 5, we provide a summary of our work and discuss possible future research directions.

2. Related Work

2.1. Deep Learning-Based Static Saliency Models

With the introduction of large-scale benchmark datasets such as SALICON [22], quite effective deep neural networks based models were proposed in the past couple of years for saliency prediction in static images. In [23], Vig et al. proposed one of the first deep learning based static saliency prediction model named eDN where a set of CNNs to learn features for visual saliency. A linear SVM is then employed to integrate resulting feature vectors into a final map. In [24], Kümmerer et al. introduced the DeepGaze model which adapts a deep model pre-trained for image classification to a new deep architecture with five convolutional layers. Kruthiventi et al. [25] proposed a novel 20 layered fully convolutional neural network, named DeepFix, specifically designed for saliency prediction. This network simply learns features for saliency prediction in a multi-scale fashion. Liu et al. [26] assembled a group of CNNs with 3 layers, in order to build a multi-resolutional saliency model to handle image patches with different scales. This scale oriented CNNs are robust against exploiting low and high level salient features. In [17], Huang et al. proposed another deep architecture that is called SALICON which consists of two subnetworks that depend on pre-trained models for image classification to include coarse and fine-scale analysis in their formulation. In addition to this multi-scale approach, they also investigated different loss functions that are based on evaluation metrics commonly used in saliency prediction. Pan et al. [18] proposed to use shallow (3 layers) and deep (10 layers) convolutional neural networks referred as SalNet. As the loss function they used Euclidean distance between the predicted saliency map and ground truth human density maps. Bruce et al. [27], proposed a fully convolutional networks based saliency model, which they refer

to FUCOS. Lastly, Jetley et al. [28] proposed a deep model which formulates saliency maps as probability distributions. They added 3 new layers to eliminate features after the convolutional layers of the VGGNet model. In addition, they investigated some probability distances as loss functions and reported that Bhattacharyya distance is the best performing one amongst them.

2.2. Existing Dynamic Saliency Models

The early examples of the dynamic saliency models are mostly built upon existing static saliency studies, and extend them to work in spatiotemporal domain. A second line of dynamic saliency models transform visual features or learn weights for feature integration according to spatial and temporal information. Among these studies, Itti and Baldi [3] proposed a model in which salient regions in video frame are extracted with intensity and color contrast features using Bayesian surprise theory. Seo and Milanfar [29] employed local steering kernels over center-surround neighborhood differences within both spatial and temporal dimensions. Cui et al. [30] used Fourier transformation for spectral residual analysis on temporal slices of video frames over X-T and Y-T planes to find the salient areas. Leboran et al. [11] formulated a computational saliency model that employs high-order statistical structures to extract the relevant information from video frames. In particular, they used chromatic representations of input frames in Fourier transformations in order to generate saliency maps of spatial and temporal streams. Guo and Zhang [31] used quaternion replica of a frame including two color channels, motion and intensity within Fourier transformation to build their phase spectrum model. Hou and Zhang [10] used an objective function with incremental coding length to find the maximum entropy difference over rare visual features in space-time. In [32], Fang et al. used discrete cosine transform (DCT) over motion, color, luminance and texture to build DCT blocks and calculated the Hausdorff distance between these blocks to estimate saliency. Mahadevan and Vasconcelos [33] used motion and center surround differences to build dynamic textures (DTs) and proposed to predict saliency over these structures using KL-divergence. Fang et al. [9]

used uncertainty weighting to fuse compressed domain features from space and time channels. Zhou et al. [8] calculated the displacement vector of moving objects against dynamic backgrounds using Fourier transformation. Mathe and Sminchisescu [12] presented a Multiple Kernel Learning framework for dynamic saliency estimation. Liu et al. [34] used a conditional random field model to integrate dynamic and static feature channels. Li et al. [35] built a Bayesian learning based model to combine high level (task related) and low level (stimulus driven) features in order to predict dynamic saliency. Rudoy et al. [36] employed random forest regression to learn salient fixation locations in space and time. In order to do so, they used motion, semantic and static visual features to estimate candidate gaze locations. Lastly, Nguyen et al. [37] used two neural networks to find optimum weights for static and dynamic feature channels using linear regression. Recently, we carried a comparative study of feature integration strategies for dynamic saliency in [38] where we considered several different low and high-level visual features such as static saliency, motion, faces, humans and text.

Recently, deep learning based dynamic saliency models have been also proposed in the literature [19, 20, 21]. All these models employ neural architectures that simultaneously process spatial and temporal cues either by considering two-stream models [20] or LSTM based recurrent connections [19, 21]. Our proposed model, however, differs from these models in that it does not require a training phase. Since all the aforementioned deep dynamic saliency models involve a fully-supervised learning setting, they need a huge amount of videos with the associated groundtruth eye fixation data.

3. Adaptive Feature Integration for Dynamic Saliency

Previous studies have shown that both bottom-up features such as color, intensity, orientation and top-down factors like faces, text, people are linked to static saliency [2]. For dynamic scenes, however, the influence of these features on attentional mechanisms in human visual system are much more complex.

For instance, a high contrast region might attract few people's attention as compared to a low contrast object that is in motion. Hence, temporal factors like motion and actions become highly important. In dynamic scenes, there might be countless shifts in the attention, even in small time periods, and semantic and low-level features are all in competition. To handle these complex relationships, we propose to use an online decision theoretic algorithm called the Hedge algorithm [15, 16] to design an adaptive dynamic saliency model. Specifically, in our model named HedgeSal, we employ deep networks proposed for static saliency as experts and then hedge their decisions to infer a master saliency map. An illustration of the proposed framework is given in Figure 1. Each component will be discussed in detail in the subsequent sections.

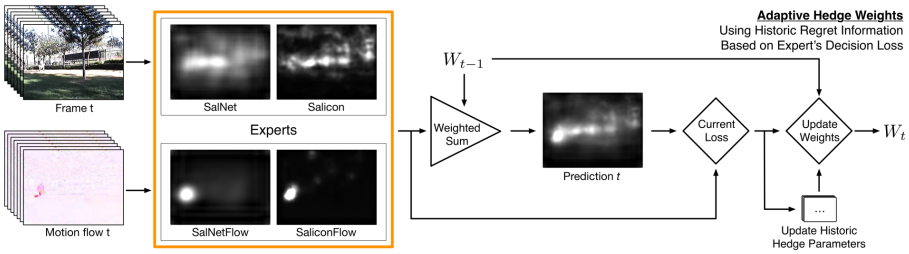

Figure 1: An illustration of the proposed hedge-based adaptive dynamic saliency (HedgeSal) model.

3.1. Hedged Saliency Prediction

Each individual frame is first processed by the deep static saliency models SALICON [17] and SalNet [18], which correspond to the experts used in our hedge model. These saliency methods take RGB images (encoding appearance) or optical flow images (encoding motion) as inputs and predict different saliency maps which we refer to as their decisions. We give a detailed description of our experts in Section 3.3 but we should note that our algorithm is in fact agnostic to the experts and any other saliency models can be employed.

Treating the aforementioned static saliency models as our experts, the master saliency map s_t for each frame t is estimated by combining the individual

decisions via weighted averaging where the weights denote the time-varying reliability scores of the saliency experts, as follows:

$$s_t = \sum_{k=1}^{K} w_t^k f_t^k \ . \tag{1}$$

Here, K represents the total number of experts ($K = 4$ in our case), and f_t^k and w_t^k respectively denote the decision (individual saliency map) and the reliability score of the kth expert.

An expert that provides good predictions in the previous frames in a consistent manner is given a higher reliability score for the current frame. Adaptively integrating experts' decisions is carried out by performing a loss and regret-based analysis. In short, at each frame t, the decision of each expert returns a specific loss value ℓ_t^k. In our experiments, we tested several alternative loss functions, which will be explained in detail in Section 3.2. These loss functions are either based on measures used in evaluating saliency models or defined in a way reflecting the characteristics of human fixations in dynamic scenes.

Once loss of each expert is estimated, a regret value r_t^k for each particular expert can be computed by inspecting the difference between its current loss ℓ_t^k and the expected loss $\overline{\ell}_t^k$ estimated from all of the experts, as follows:

$$r_t^k = \overline{\ell}_t^k - \ell_t^k \ , \tag{2}$$

with $\overline{\ell}_t^k$ being computed as:

$$\overline{\ell}_t^k = \sum_{k=1}^{K} w_t^k \ell_t^k \ . \tag{3}$$

The current reliability of an expert can be then calculated based on the regret by following the strategy in [15], which uses a cumulative regret value R_t^k estimated from the initial frame to the current frame, as follow:

$$R_t^k = \sum_{\tau=1}^{t} r_\tau^k \ . \tag{4}$$

In particular, the combined model aims at minimizing the cumulative regret of all the experts, especially that of the best performing expert giving the lowest

cumulative loss. Hence, a low loss value for an expert means that the expert is reliable. Since we try to achieve the lowest cumulative regret, this leads the combined model to assign higher weights to the reliable experts in the future. On the contrary, if an expert has a high loss value at a frame, the tendency is to decrease its reliability for the future decisions. Here, it is important to note that while giving a feedback to the experts in regards to their performances, we actually do not consider any groundtruth fixation information during testing. How well an expert's decision is measured either by analyzing it according to some prior knowledge or by comparing it with that of the combined model which utilizes the predictions of all of the experts in an unified manner.

To perform the aforementioned adaptive updates, the Hedge framework considers a potential function of the form:

$$\phi(x, c) = \exp\left(\frac{([x]_+)^2}{2c}\right) \quad \text{for } x \in \mathbb{R}, c > 0 \ , \tag{5}$$

with $[x]_+$ denoting the function $\max\{0, x\}$. While keeping track of the cumulative regrets R_t^k, the framework also maintains a scale parameter c_t to keep the average potential of the experts always constant at e:

$$\frac{1}{K}\sum_{k=1}^{K} \exp\left(\frac{([R_t^k]_+)^2}{2c_t}\right) = e \ . \tag{6}$$

Since the potential function $\phi(x, c)$ is a convex function, this scale parameter c_t can be easily determined by performing line search. The weights of the experts are then proportional to the first-derivative of the potential, as given in the following formula:

$$w_{t+1}^k \propto \frac{[R_t^k]_+}{c_t} \exp\left(\frac{([R_t^k]_+)^2}{2c_t}\right). \tag{7}$$

As described above, the original Hedge algorithm [15] considers all the regret values till the current timeframe t in estimating the cumulative regret, and thus in updating the experts' reliabilities. However, as discussed in [16], this might be problematic when each expert captures a different aspect of the data or when the characteristics of the data changes a lot over time. For dynamic saliency

prediction, this is exactly the issue since the observers can shift their focus rapidly and fixate their eyes on different locations, and our experts, i.e. the static saliency models that we use, capture different aspects of salient image regions as they use different architectures and loss functions.

In our HedgeSal framework, to properly handle these challenges, we consider an adaptive hedge strategy [16] that considers a historic regret which is defined over a specific time period Δt and which is used in updating the reliabilities of the experts. Moreover, a stability score is estimated for each expert, reflecting how consistent its decisions over time. This makes the combined model more robust against the rapidly changing dynamic data and the instabilities of the experts.

The stability of an expert p_t^k at time t is measured by modeling the loss of each expert during Δt with a Gaussian distribution $\mathcal{N}(\mu_t^k, \sigma_t^k)$:

$$\mu_t^k = \frac{1}{\Delta t} \sum_{\tau=t-\Delta t+1}^{t} \ell_\tau^k \,, \tag{8}$$

$$\sigma_t^k = \sqrt{\frac{1}{\Delta t - 1} \sum_{\tau=t-\Delta t+1}^{t} (\ell_\tau^k - \mu_t^k)^2} \,. \tag{9}$$

The stability of expert k is then computed by using the formula:

$$p_t^k = \frac{|\ell_t^k - \mu_t^k|}{\sigma_t^k} \,. \tag{10}$$

The larger the value of p_t^k, more consistent the expert in its decisions (in terms of its loss values over the specified time period Δt). Using this observation, the cumulative regret is defined as:

$$R_t^k = (1 - \alpha_t^k) R_{t-1}^k + \alpha_t^k r_t^k \,, \tag{11}$$

$$\alpha_t^k = \min(g, \exp(-\gamma p_t^k)) \,, \tag{12}$$

where γ is a scale factor and g is a scalar denoting the maximum ratio on historic regret to avoid situations that no historic regret is taken into account. For a more stable expert its cumulative regret become close to its current regret value. On the other hand, if an expert has a low stability value, then its cumulative regret highly depends on the whole historic information.

3.2. Loss Functions

As mentioned previously, each deep saliency model contributes to the decision of the combined model in proportion to its reliability (Equation 1), which is mainly determined by a loss function. We tested five different loss functions which are defined by considering distance measures used in evaluating saliency estimation and a density-based measure.

The first two loss functions, namely Kullback-Leibler Divergence (KLdiv) [39] and Earth Mover's Distance (EMD) [40] measures, calculate how similar an expert's decision is to the final decision set by the combined model:

$$\ell_{KL} = KLdiv(f_t^k, s_t^k) \,, \tag{13}$$

$$\ell_{EMD} = EMD(f_t^k, s_t^k) \,. \tag{14}$$

Both of these measures treat the saliency maps as probability mass functions. While KLdiv finds the difference between two probability distributions by measuring their entropies, EMD estimates the distance by finding the minimum cost required to transform one input distribution to the other. Hence, with this formulation, an expert which produces a decision similar to the final prediction has given a low loss value, which in return increases that expert's reliability.

As our third loss function, we consider a density-based measure. It treats the saliency map of an expert as a probability mass function but instead of comparing it to the final saliency map, here, we perform a statistical analysis directly on the expert's own decision map. The loss function assumes that an expert focusing on a single salient image region is more likely to miss the whole complexity of the data, and thus an expert which produces more sparse density maps has given a high loss value, which in return decreases that expert's reliability. In particular, we first select top 30% salient pixels from the estimated individual saliency map, and use the Mean Shift algorithm to cluster these highly salient pixels. Then, we count the number of local modes returned by the clustering algorithm, and define our density loss $\ell_{Density}$ inversely proportional

to this number, as follows:

$$\ell_{Density} = \frac{1}{n_t^k} \ , \tag{15}$$

with n_t^k denoting the number of cluster centers, i.e. the total number of modes.

Finally, in the last two loss functions, we combine our density based loss function $\ell_{Density}$ and distance based loss functions ℓ_{KL} and ℓ_{EMD} as follows:

$$\ell_{KL+Density} = \frac{1}{n_t^k} KLdiv(f_t^k, s_t^k) \ , \tag{16}$$

$$\ell_{EMD+Density} = \frac{1}{n_t^k} EMD(f_t^k, s_t^k) \ . \tag{17}$$

These joint loss functions take advantage of both the distance and density-based loss functions by considering the similarity between the expert's decision and the final prediction, along with the overall sparseness of the saliency map of the expert.

3.3. Saliency Experts

As in most of the existing dynamic saliency approaches, we separately take into account appearance and motion streams. While we employ RGB video frames as the source for the appearance information, we extract optical flows[1] from subsequent frames to encode the motion information. We then generate optical flow images by stacking horizontal and vertical flow components and the magnitude of the flow together. We use these RGB and flow images as inputs to our experts, two recently proposed deep static saliency models, namely SALICON [17] and SalNet [18]. Each expert processes motion and appearance streams separately and to obtain the final saliency map, we use a total of four saliency maps extracted by these deep saliency networks. In Figure 2, we provide some sample individual saliency maps obtained by our experts.

[1] We use the implementation publicly available at `http://people.csail.mit.edu/celiu/OpticalFlow`

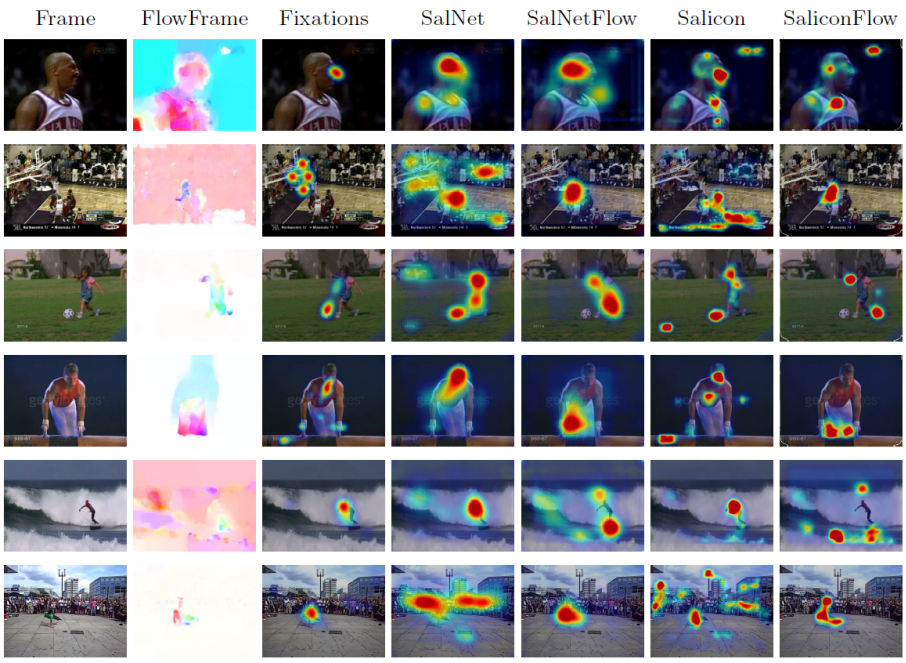

Figure 2: Saliency maps extracted by our experts (SalNet, SalNetFlow, SALICON and SALICONFlow) on some sample video frames along with the extracted optical flow images and the groundtruth eye fixations. The first two frames are from CRCNS, the next two are from UCF-Sports, and the last two are from CITIUS.

SALICON. The model proposed by Huang et al. [17], which is referred to as SALICON, considers a multi-stream deep network architecture where each stream processes the input from a different scale (coarse or fine) in parallel. Hence, the final saliency model learn features in multi-scale fashion. In their work, the authors evaluate different base network models such as AlexNet, VGG-16 and GoogLeNet to encode a kind of implicit semantic knowledge through pre-training. Moreover, they consider various loss functions which depend on KL-div, CC, NSS, Sim, AUC and sAUC evaluation measures to optimize these models. Among these configurations, the best performing model is found to be a VGG-Net-based model that is trained using the KL-div-based loss function.

SalNet. As compared to SALICON, the SalNet model proposed by Pan et al. [18] is a single-stream convolutional neural network that processes the input

data, performing a single scale analysis. In their paper, the authors propose one shallow and one deep architecture for saliency prediction, which mainly differ from each other by the number of considered layers. While the shallow network consists of 5 layers (3 convolutional and 2 fully connected layers), the deep network includes 10 layers (9 convolutional and 1 deconvolution layers) in which the first three layers are initialized with the VGG-Net model. Both of these models are trained by formulating saliency estimation as a regression problem and by using Euclidean distance between the current prediction map and ground truth eye fixations from training data as the loss function. In our work, we use the deep version of the SalNet model as our second expert.

Adapting SALICON and SalNet to Motion Saliency. In addition to the appearance stream, in our HedgeSal model, we also consider the motion stream. In particular, to include the motion stream into our model, we first encode the motion information inherent to dynamic scenes in terms of optical flow images. Then, we let the image saliency models SALICON and SalNet process these images. We refer to these experts that process the motion stream as SALICONFlow and SalNetFlow in our paper. These models provide fairly good predictions since the moving regions results high contrast regions in these images as we have investigated in our experiments. Directly using image saliency models on optical flow images has certain advantages. First and foremost, we do not need to perform any training or fine-tuning on the image saliency network models. Hence, our model simply transfers the knowledge acquired from static images regarding the saliency prediction task to dynamic scenes.

4. Experiments

4.1. Baseline Models

Our approach assumes that some initial weights have been assigned to each expert. A straightforward choice is to use uniform weights for the first frame and make the framework to adapt itself according to the visual content present in the current scene. Another alternative could be to learn these initial weights by

using a supervised learning strategy. For instance, as in [41, 38], the estimation of these weights can be formulated as a non-negative least squares (NNLS) problem [2]:

$$\underset{x}{argmin} \|Ax - y\|_2 \text{ where } x \geq 0 \, . \tag{18}$$

with A denoting the matrix composed of the experts' decisions (individual saliency maps), y representing the ground truth eye fixation responses and x is the weights to be learned. As our first baseline method, we use this strategy to form a model, which we refer to NNLS, which utilizes these learned (fixed) weights to combine the decisions of the experts without adaptively changing them over time. In training this baseline model, we used five fold cross validation to learn the optimum set of these initial weights.

In addition to the former NNLS-based baseline model, we also consider two basic transformation based models, namely Mean and Max. In the Mean model, the prediction for a pixel is obtained by directly averaging decisions of all of the experts. The Max model, on the other hand, takes the maximum of all the responses of all the experts for a pixel and employs that value as the final saliency score for that pixel.

4.2. Dynamic Saliency Datasets

CRCNS [42]. CRCNS is one of the first and commonly used dynamic saliency datasets. There are 50 videos with different contents like streets, video games and TV shows. The originally recorded videos are referred as ORIG and their randomly mixture of frames are called MTV. In our study, we only used ORIG videos and report results on the original recordings. There are 8 different observers participated in the eye tracking experiments where the recordings were done under free-viewing scenario, i.e. no prior information or instructions are

[2] In our previous work [38], we found out that NNLS, in general, gives the best results over other supervised learning fusion strategies such as SVM, Boosting, and Random Forest. Hence, we include it to our evaluation as a strong baseline.

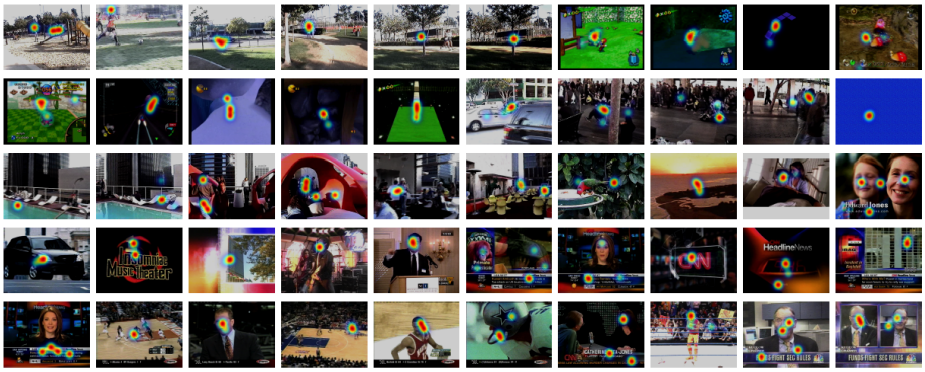

Figure 3: Sample frames from CRCNS dataset together with the superimposed eye fixation heatmaps.

given to the participants. There exist camera movements and scenes combined with different shots in the videos. In Figure 3, we show some sample frames from CRCNS dataset overlaid with the eye fixation heatmaps.

UCF-Sports action dataset [43]. UCF-Sports dataset is originally collected for action recognition and contains 150 videos from 13 different action classes. This dataset is used in dynamic saliency estimation with eye fixations provided by Mathe and Sminchisescu [12] where 16 subjects participated in the task-specific and task-free observations. In our experiments we only employed the fixation data from task-free viewing. There are some camera movements in the videos but each sequence is recorded in a single shot and there exists no scene shift. Figure 4 shows a few sample frames and the corresponding ground truth eye fixation heatmaps.

CITIUS dataset [11]. CITIUS is a recently proposed dataset for dynamic saliency estimation. It contains 72 videos including static and dynamic camera shots where 22 subjects from different ages participated the experiments. In order to better model the static visual effects in dynamic scenes, 27 of these videos are synthetic videos with dynamic movement effects. The videos are presented in random order to each observer with no prior instructions. Figure 5

Figure 4: Sample frames from UCF-Sports dataset overlaid with ground truth saliency heatmaps.

shows some sample frames with superimposed eye fixation heat maps.

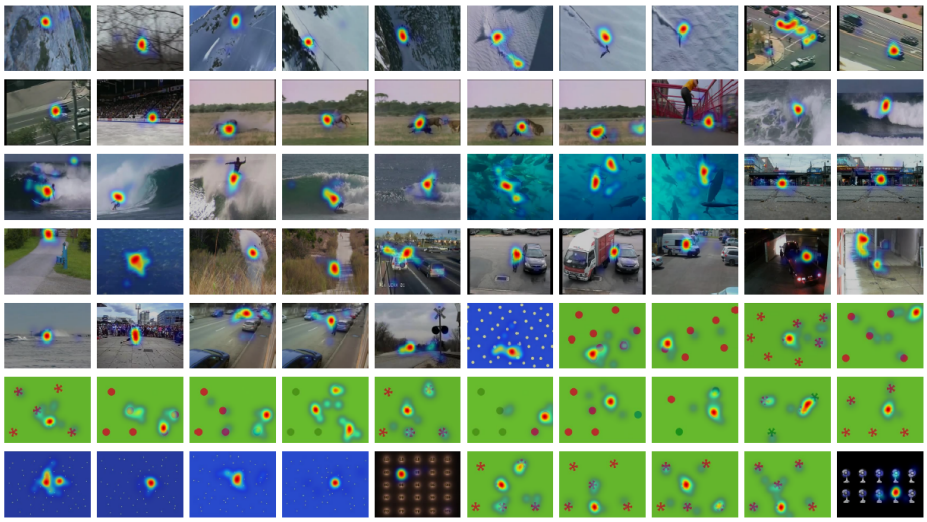

Figure 5: Sample frames from CITIUS dataset, together with the superimposed eye fixation heatmaps.

4.3. Evaluation Measures

For performance comparison, we compute Area under the ROC Curve (AUC), shuffled AUC (sAUC), Pearson's Correlation Coefficient (CC), Normalized Scanpath Saliency (NSS) and INformation Gain (IG) and report the results averaged over all videos of the given dataset.

AUC measure [44] thresholds a given saliency map at various threshold levels and accordingly labels each pixel as fixated or not. Then, these predictions are compared against results from the ground truth eye fixation density maps and the success is measured as the area under curve (AUC). While the perfect AUC is 1, a score of 0.5 indicates the chance performance. In our experiments, we use the AUC implementation of [6].

While AUC score is one of the widely used meausures for visual saliency, it is affected by the the so-called center bias. To tackle this issue, shuffled AUC (sAUC) metric is proposed by Zhang et al. [45]. Here, the negative examples are randomly sampled from fixation points from other frames, rather than selecting them from the current frame in consideration. In our study, we used sAUC implementation of [46].

Pearson's Correlation Coefficient (CC) considers the saliency map S and the fixation map H as random variables and calculates the linear relationship between them using a Gaussian kernel density estimator, as $CC(S,H) = \frac{cov(S,H)}{\sigma_S \sigma_H}$. While a CC score of 1 indicates a perfect correlation, 0 indicates no correlation and -1 denotes that they are perfectly negatively correlated.

NSS measure is defined as the response value at eye fixation points in an estimated saliency map which has been normalized to have zero mean and unit standard deviation [47], i.e. $NSS = \frac{1}{n} \sum_{i=1}^{n} \frac{S(x_H^i, y_H^i) - \mu_S}{\sigma_S}$ where n is the number of fixation points in the ground truth data. While a negative NSS indicates a bad prediction, a non-negative NSS denotes the correspondence between the saliency map and the eye fixations is above chance.

Since the fixations among observers might be inconsistent, Kümmerer et al. [48] proposed a formulation based on probabilities of eye fixations to unify the existing saliency metrics. Current prediction map and a baseline saliency map are log transformed and then compared against the current image's ground truth fixations. An IG score above 0 suggests that the saliency model in question performs better that the chosen baseline. In our experiments, we used a baseline map built by applying Gaussian blur over eye fixations from 50 other random frames of the same video.

4.4. Comparison of Loss Functions

In this section, we quantitatively analyze the effectiveness of the loss functions introduced in Section 3.2 for our HedgeSal model. In Table 1, we report AUC, sAUC, NSS, CC and IG scores averaged over all video sequences and all video frames for all three benchmark datasets. As can be seen from these results, the distance based loss functions provide the worst performances, with ℓ_{KL} performing a little better than ℓ_{EMD}. Our combined loss functions $\ell_{KL+Density}$ and $\ell_{EMD+Density}$, in general, provide results quantitatively better than both distance and density-based loss functions, demonstrating the importance of deciding reliability of an expert both in isolation and in relation to others. Among these combined losses, the best performance is achieved with the loss function that combines the density and KLdiv based distance based loss functions ($\ell_{KL+Density}$). Hence, in the remaining, we report the results obtained with $\ell_{KL+Density}$.

Table 1: Performance evaluation of the proposed loss functions.

	Loss Function	**AUC**	**sAUC**	**NSS**	**CC**	**IG**
CRCNS	ℓ_{KL}	0.890	0.726	1.721	0.329	-0.767
	ℓ_{EMD}	0.889	0.711	1.727	0.331	-0.766
	$\ell_{Density}$	0.893	0.728	1.783	0.340	-0.741
	$\ell_{KL+Density}$	**0.897**	**0.733**	**1.875**	**0.355**	**-0.670**
	$\ell_{EMD+Density}$	0.892	0.725	1.770	0.337	-0.765
UCF-Sports	ℓ_{KL}	0.870	0.720	1.915	0.482	-1.471
	ℓ_{EMD}	0.862	0.703	1.864	0.474	-1.501
	$\ell_{Density}$	0.882	0.739	2.051	0.514	-1.396
	$\ell_{KL+Density}$	**0.885**	**0.746**	**2.121**	**0.526**	**-1.354**
	$\ell_{EMD+Density}$	0.883	0.745	2.105	0.524	-1.365
CITIUS	ℓ_{KL}	0.874	0.765	2.021	0.460	-0.931
	ℓ_{EMD}	0.880	0.758	2.051	0.479	-0.899
	$\ell_{Density}$	0.883	0.766	2.027	0.476	-0.855
	$\ell_{KL+Density}$	**0.894**	**0.787**	**2.328**	**0.527**	**-0.768**
	$\ell_{EMD+Density}$	0.892	0.784	2.303	0.522	-0.781

4.5. Comparison to State-of-the-Art

We compare our approach with five state-of-the-art dynamic saliency models, Seo and Milanfar [29], Zhou et al. [8], Fang et al. [9], Hou and Zhang [10], and AWS-D [11]. In addition, we provide the results of three baseline models (NNLS, Max and Mean) and the Center Map that is defined as a single Gaussian blob, and the individual performances of our experts (SALICON, SalNet, SALICONFlow and SalNetFlow). In Tables 2, 3 and 4, we present the performances of the evaluated models on the CRCNS, UCF-Sports and CITIUS datasets, given by the five evaluation measures, respectively. Regarding the individual performances of our experts, SalNet has a better prediction accuracy than all the other experts on all datasets in terms of four out of five evaluation measures. The second best expert is SalNetFlow, which is followed by SALICON. It is interesting to note that our experts gives results highly competitive to the recently proposed AWS-D model even if they did not specifically trained for dynamic saliency. Our hedge method, on the other hand, provides the best results on all datasets when compared to our experts, the baseline models and all the other previous models. Hence, it can be said that the proposed decision-theoretic online learning has a key role in achieving this superior performance. In our framework, our experts work in harmony and complement each other's decisions, and thus provide more accurate predictions when integrated in the proposed adaptive way.

Moreover, to demonstrate the effectiveness and genericness of our framework, we define a second version of our model which we refer to HedgeSal*. In this model, we employ two additional saliency experts, namely the AWS-D model and the Center. We found that this second model, in general, gives better results that our original HedgeSal model on all of the datasets when all five evaluation metrics are considered. We also compare this model with other feature integration models, the supervised NNLS* model, and basic transformation based models Mean* and Max*, which are extended by the additional AWS-D and Center maps. We have two key observations. First of all, our model HedgeSal* achieves a performance on par with NNLS* model. However, it is

Table 2: Quantitative results on CRCNS dataset. The best and the second best results are given in boldface and underlined, respectively.

	AUC	sAUC	NSS	CC	IG
Existing Models					
Center Map	0.748	0.525	1.091	0.189	-4.672
Seo Milanfar [29]	0.636	0.559	0.263	0.063	-2.743
Zhou et al. [8]	0.783	0.657	1.046	0.174	-1.293
Fang et al. [9]	0.820	0.587	1.200	0.220	-1.143
Hou Zhang [10]	0.808	0.686	1.004	0.176	-1.350
AWS-D [11]	0.816	0.718	1.239	0.226	-1.140
Features					
SALICON	0.839	0.729	1.339	0.229	-1.025
SalNet	0.884	0.719	1.703	0.327	-0.753
SALICONFlow	0.771	0.588	0.929	0.162	-1.409
SalNetFlow	0.841	0.594	1.371	0.271	-0.998
Integration Models					
HedgeSal	<u>0.897</u>	0.733	1.875	0.355	<u>-0.670</u>
NNLS	0.895	0.725	1.809	0.347	-0.720
Max	0.879	0.710	1.566	0.303	-0.851
Mean	0.892	0.730	1.785	0.337	-0.740
HedgeSal*	<u>0.897</u>	**0.756**	<u>1.897</u>	<u>0.361</u>	-0.683
NNLS*	**0.903**	0.721	**1.928**	**0.364**	**-0.618**
Max*	0.874	0.722	1.496	0.286	-0.925
Mean*	0.895	<u>0.740</u>	1.840	0.344	-0.716

important to note that while our model is weakly supervised, NNLS* model is a fully supervised saliency model and requires a heavy training on eye fixation data collected for dynamic stimuli. That is, our HedgeSal model is able to achieve identical performances to NNLS* without any training on dynamic data by adaptively updating the weights of the experts. Interestingly, on the UCF-sports dataset, Mean* model gives slightly better results as compared to all the remaining models including ours. We conjecture that this is simply because, the sequences in the UCF-Sports datasets are much more simpler than those of CRCNS and CITIUS datasets, and more importantly the important actions in the videos are mostly at the center.

Table 3: Quantitative results on UCF-Sports dataset. The best and the second best results are given in boldface and underlined, respectively.

	AUC	sAUC	NSS	CC	IG
Existing Models					
Center Map	0.503	0.500	0.049	0.008	-38.349
Seo Milanfar [29]	0.806	0.721	1.373	0.314	-1.888
Zhou et al. [8]	0.817	0.729	1.710	0.365	-1.674
Fang et al. [9]	0.853	0.700	1.952	0.446	-1.441
Hou Zhang [10]	0.781	0.694	1.206	0.269	-1.940
AWS-D [11]	0.819	0.751	1.698	0.397	-1.709
Features					
SALICON	0.813	0.737	1.224	0.270	-1.833
SalNet	0.850	0.684	1.818	0.448	-1.529
SALICONFlow	0.761	0.634	1.275	0.296	-1.994
SalNetFlow	0.847	0.686	1.910	0.486	-1.722
Integration Models					
HedgeSal	**0.885**	0.746	**2.121**	**0.526**	**-1.354**
NNLS	0.881	0.733	2.042	0.516	-1.395
Max	0.866	0.724	1.797	0.457	-1.523
Mean	<u>0.883</u>	0.744	2.065	0.513	<u>-1.393</u>
HedgeSal*	0.881	<u>0.761</u>	2.055	0.508	-1.423
NNLS*	0.882	0.756	2.084	<u>0.521</u>	-1.420
Max*	0.857	0.743	1.720	0.434	-1.641
Mean*	**0.885**	**0.763**	<u>2.105</u>	0.519	-1.413

For qualitative analysis, in Figures 6-8, we present sample results of the existing dynamic saliency approaches, the best performing baseline model NNLS and our proposed model along with the ground truth density maps. In Figure 9, we demonstrate how our model adaptively alters the reliabilities of the experts over time on three sample sequences of various lengths from the CRCNS, UCF-Sports and CITIUS datasets, respectively. As it can be seen, our model can effectively handle both short and long sequences. Our historic regret based update mechanism adjusts these weights to capture the changes in the dynamic scenes. For instance, the `tv-sports-05` sequence from CRCNS dataset is among the most challenging sequences for saliency prediction, containing several scene shifts (sudden camera changes), and high camera motion. Moreover, the motion of objects within this sequence have different paces and the objects show different contrast characteristics. But yet, as it can be seen from Figure 10,

Table 4: Quantitative results on CITIUS dataset. The best and the second best results are given in boldface and underlined, respectively.

	AUC	sAUC	NSS	CC	IG
Existing Models					
Center Map	0.664	0.517	0.712	0.185	-24.551
Seo Milanfar [29]	0.790	0.740	1.474	0.297	-1.440
Zhou et al. [8]	0.800	0.748	1.829	0.357	-1.162
Fang et al. [9]	0.841	0.729	2.082	0.416	-0.926
Hou Zhang [10]	0.842	0.762	1.740	0.383	-1.098
AWS-D [11]	0.842	**0.811**	2.185	0.458	-0.980
Features					
SALICON	0.836	0.760	1.821	0.376	-1.014
SalNet	0.873	0.744	1.959	0.468	-0.888
SALICONFlow	0.792	0.699	1.778	0.360	-1.183
SalNetFlow	0.858	0.732	2.118	0.496	-0.985
Integration Models					
HedgeSal	<u>0.894</u>	0.787	2.328	0.527	-0.768
NNLS	0.893	0.774	2.221	0.523	-0.798
Max	0.876	0.763	1.931	0.458	-0.954
Mean	0.890	0.783	2.257	0.512	-0.803
HedgeSal*	<u>0.897</u>	**0.807**	**2.437**	<u>0.543</u>	**-0.700**
NNLS*	<u>0.897</u>	**0.807**	<u>2.428</u>	**0.550**	<u>-0.708</u>
Max*	0.876	0.774	1.906	0.448	-0.966
Mean*	<u>0.897</u>	0.799	2.394	0.538	-0.709

NNLS and our proposed hedge model share the best performance.

In Figures 10, 11 and 12, we present the IG scores of the evaluated models for each sequence of CRCNS, UCF-Sports and CITIUS datasets in the form of a heatmap, respectively. These figures demonstrate the groupings of the saliency models in terms of their performances and moreover how challenging a sequence is over the others in the corresponding dataset. For instance, we find that, on average, the integration based models perform better than the existing models or the expert models that we considered in our models on all of the datasets. Moreover, the results show that deep saliency models SALICON and SALICONFlow perform poorly nearly on half of the sequences of the UCF-Sports dataset as compared to our SalNet and SalNetFlow models. Even this is the case, our HedgeSal model that employs these models as experts is able

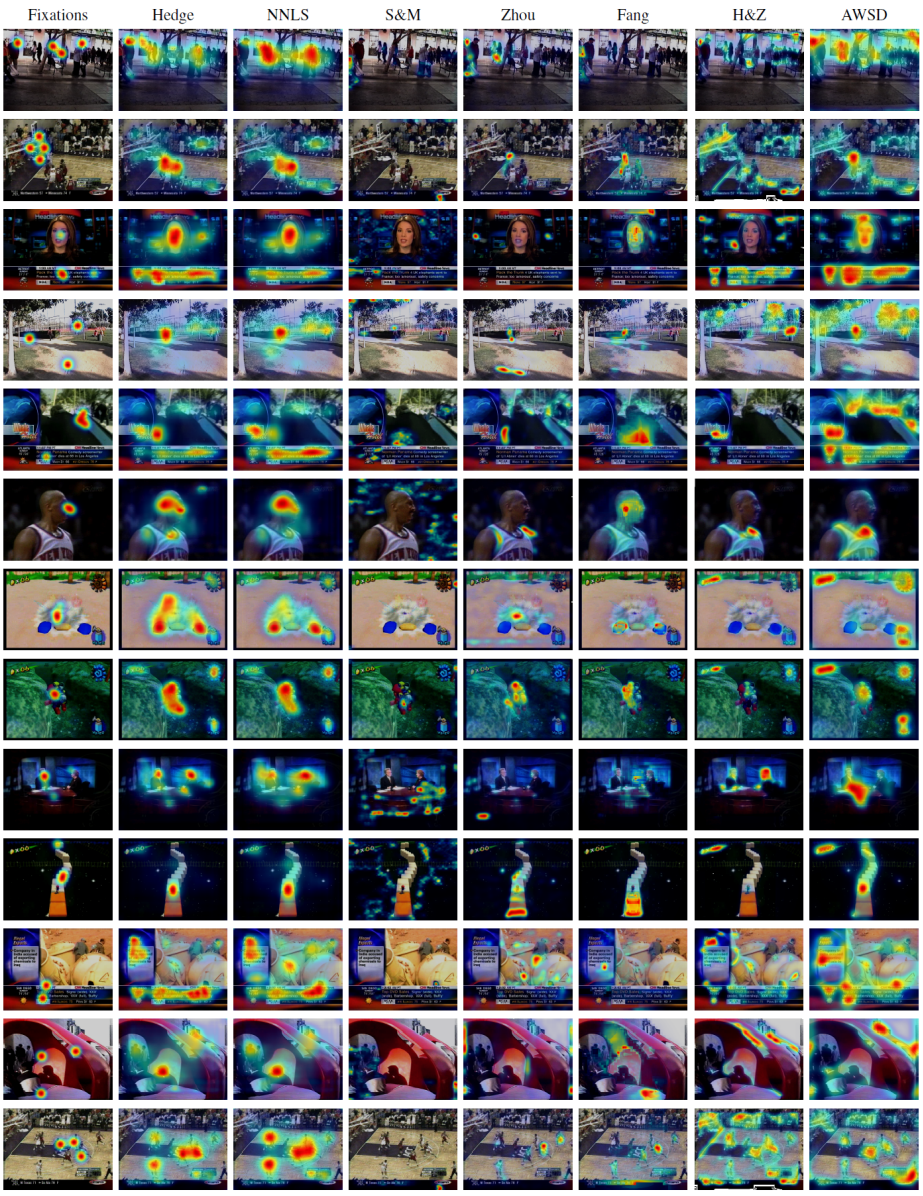

Figure 6: Sample results from CRCNS dataset. In each row, we show a video frame overlayed with the ground truth density map, the results of our hedge model, NNLS baseline and the competing dynamic saliency models.

to handle the poor performances of these models within its decision-theoretic framework and gives better results.

Figure 7: Sample results from UCF-Sports dataset. In each row, we show a video frame overlayed with the ground truth density map, the results of our hedge model, NNLS baseline and the competing dynamic saliency models.

The majority of the synthetic videos of the CITIUS dataset stand out as the easiest sequences among all the others, with all models performing quite well on

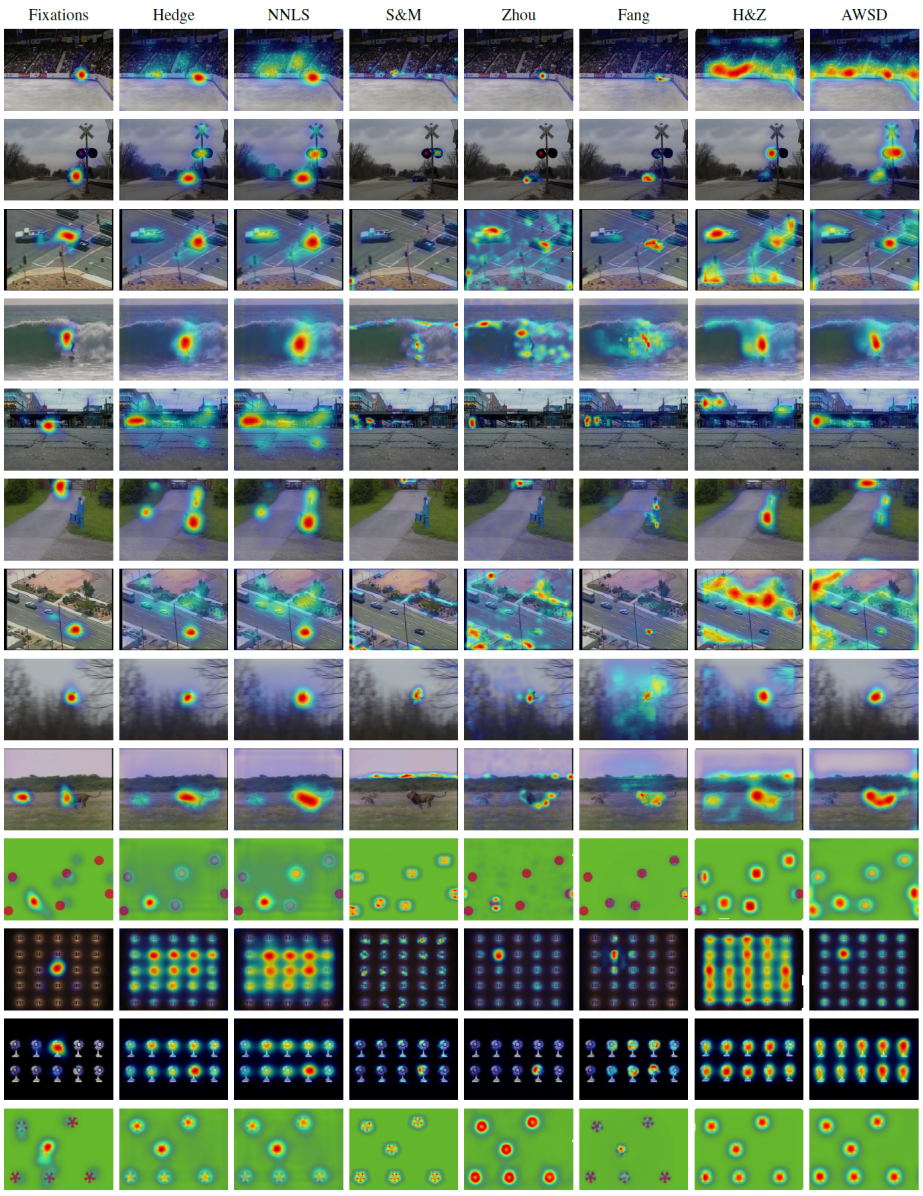

Figure 8: Sample results from CITIUS dataset. In each row, we show a video frame overlayed with the ground truth density map, the results of our hedge model, NNLS baseline and the competing dynamic saliency models.

these sequences. However, on the synthetic sequences which demonstrate low motion contrast and where the semantic content change rapidly, nearly all the

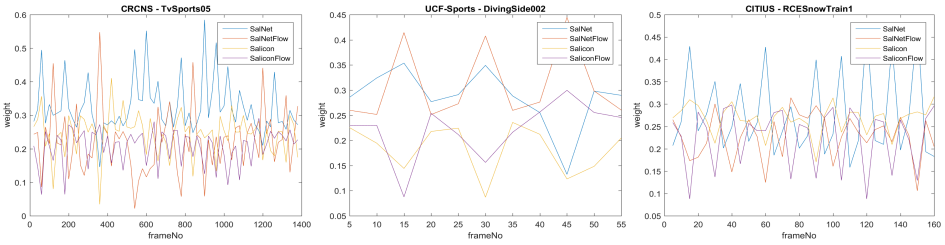

Figure 9: Plots demonstrating how the reliabilities of our experts vary over time in our adaptive hedge model.

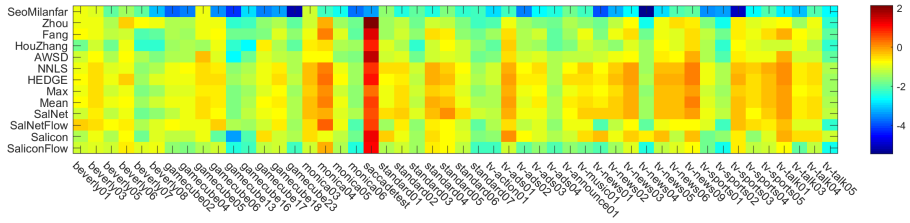

Figure 10: Information Gain scores of the models for each sequence in the CRCNS dataset.

models perform poorly. In the UCF-Sports dataset, the important actions in the videos are mostly at the center and the videos have high quality. The videos demonstrating actions like riding or walking have high contrast and continuous motion characteristics, hence the performances in these videos are fairly good. On the contrary, the videos including actions like kicking or playing golf have high camera motion, which decreases the overall performances. According to the average CRCNS results, the highest performance belongs to `saccadetest` sequence illustrating a synthetic example with high color contrast. Besides that, the overall performance of all models are slightly lower on the CRCNS dataset since most of the videos in this dataset were recorded with a low resolution under high camera motion. Nevertheless, our approach performs reasonably well on these highly challenging sequences as compared to the previous dynamic saliency approaches.

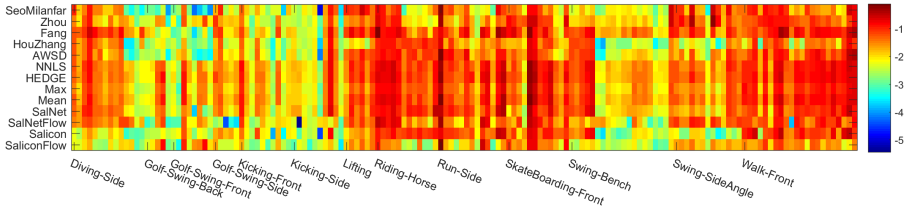

Figure 11: Information Gain scores of the models for each sequence in the UCF-Sports dataset.

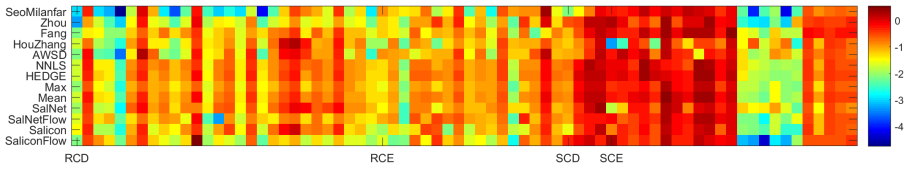

Figure 12: Individual video scores on CITIUS dataset according to Information Gain metric.

The hierarchical feature learning mechanisms in deep image saliency models enable extracting low-level and semantic features from training data. However, we still need a through analysis on how to integrate these deep features, which was the main motivation for this work. We should state that, despite their superior performances over shallow models, using the deep saliency models as our experts introduces some limitations to our method. As pointed in [27], the performances of these deep models highly depend on the training data in that they might suffer from overfitting especially when the training data is scarce. Moreover, as explored in [2], they might also fail to reason about the relative importance of deep features during saliency prediction. For instance, the third row of Figure 7 demonstrates a sequence which includes a man kicking a ball. Here, both the low level features such as motion and color contrast, and the high level features such as pedestrians and faces are all available. However, when the groundtruth fixations are examined, it can be seen that the main focus is on the ball. Since we expect that the man is going to kick the ball, most of the humans look at it. Likewise, in the second row of Figure 7, the attention is mainly on the golf ball rather than the man or the regions showing high contrast information as humans are curious about the result of the golf shot, leading to follow the ball.

Table 5: Running times of dynamic saliency models used in our evaluation (in seconds).

Seo Milanfar [29]	Zhou et al. [8]	Fang et al. [9]	Hou Zhang [10]	AWS-D [11]
0.87	0.04	16.20	0.33	4.53

NNLS	NNLS*	HedgeSal	HedgeSal*
1.00	1.02	1.06	1.08

Finally, we perform a running-time evaluation of the saliency prediction methods that we considered in our experimental analysis. Table 5 presents the results of this evaluation that we carried on a machine with 2×Intel Xeon E5-2640 2.00GHz CPU, 48 GB RAM and NVidia Titan X GPU. For each model, we report the computation time required to process a single video frame. As can be seen from the table, the dynamic saliency method by Zhou et al. [8] is the fastest model since it is the only one that works in the frequency domain. That is being said, its prediction performance is, in general, lower that those of other saliency models as given in Table 2-4. Two versions of our proposed approach, HedgeSal and HedgeSal* is nearly four times faster than the recently proposed state-of-the-art model AWS-D [11] model. We also observe that our models are a bit slower than NNLS and NNLS* as these models use a fixed weighting scheme for feature integration. However, it is important to emphasize again that both NNLS and NNLS* are fully supervised models which require training on dynamic stimuli.

5. Conclusion

In this work, we have investigated a novel framework for predicting saliency in dynamic scenes. Rather than considering a fixed scheme to combine different feature maps extracted from appearance and motion streams, our HedgeSal model employs a decision-theoretic online learning algorithm. This allows our framework to integrate the appearance and motion saliency maps extracted by two different deep saliency models in an adaptive manner. In estimating a saliency map for a video frame, we combine the decisions of these deep models by considering their reliabilities, which vary over time. To obtain these reliabilities, we propose to use different loss functions. Since we directly use pretrained deep

image saliency models as experts, our framework requires no explicit training on dynamic stimuli, and is thus a weakly supervised approach. Our experiments on three benchmark datasets clearly demonstrate the effectiveness our approach to previous dynamic saliency models and the suggested adaptive feature integration strategy performs much better than the classical integration schemes. As a future work, we plan to investigate a deep dynamic saliency model, which consider similar adaptive feature integration schemes which can be trained in an end-to-end manner.

Acknowledgments

This work was supported in part by a grant from The Scientific and Technological Research Council of Turkey (TUBITAK) Career Development Award 113E497, and TUBA GEBIP fellowship awarded to E. Erdem.

[1] A. Borji, State-of-the-art in visual attention modeling, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 185–207.

[2] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, F. Durand, Where should saliency models look next?, in: ECCV, 2016, pp. 809–824.

[3] L. Itti, P. Baldi, A principled approach to detecting surprising events in video, in: CVPR, Vol. 1, 2005, pp. 631–637.

[4] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: NIPS, 2007, pp. 545–552.

[5] M. Cerf, E. Frady, C. Koch, Faces and text attract gaze independent of the task: Experimental data and computer model, Journal of Vision 9 (12) (2009) 1–15.

[6] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: ICCV, 2009, pp. 2106–2113.

[7] A. Borji, Boosting bottom-up and top-down visual features for saliency estimation, in: CVPR, 2012, pp. 438–445.

[8] B. Zhou, X. Hou, L. Zhang, A phase discrepancy analysis of object motion, in: ACCV, Springer, 2011, pp. 225–238.

[9] Y. Fang, Z. Wang, W. Lin, Z. Fang, Video saliency incorporating spatiotemporal cues and uncertainty weighting, IEEE Transactions on Image Processing 23 (9) (2014) 3910–3921.

[10] X. Hou, L. Zhang, Dynamic visual attention: searching for coding length increments., in: NIPS, Vol. 5, 2008, p. 7.

[11] V. Leboran, A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, Dynamic whitening saliency, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (5) (2017) 893–907.

[12] S. Mathe, C. 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) (2015) 1408–1424.

[13] T. Liu, X. T. Jian Sun, Nan-Ning Zheng, H.-Y. Shum, Learning to detect a salient object, in: CVPR, 2007, pp. 1–8.

[14] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, S. Yan, Static saliency vs. dynamic saliency: a comparative study, in: ACM-MM, 2013, pp. 987–996.

[15] K. Chaudhuri, Y. Freund, D. J. Hsu, A parameter-free hedging algorithm, in: NIPS, 2009, pp. 297–305.

[16] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, M.-H. Yang, Hedged deep tracking, in: CVPR, 2016, pp. 4303–4311.

[17] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks, in: ICCV, 2015, pp. 262–270.

[18] J. Pan, K. McGuinness, S. E., N. O'Connor, X. Giró-i Nieto, Shallow and deep convolutional networks for saliency prediction, in: CVPR, 2016, pp. 598–606.

[19] L. Bazzani, H. Larochelle, L. Torresani, Recurrent mixture densitynetwork for spatiotemporal visual attention, in: ICLR, 2017.

[20] C. Bak, A. Kocak, E. Erdem, A. Erdem, Spatio-temporal saliency networks for dynamic saliency prediction, IEEE Transactions on Multimedia 20 (7) (2018) 1688–1698.

[21] W. Wang, J. Shen, F. Guo, M.-M. Cheng, A. Borji, Revisiting video saliency: A large-scale benchmark and a new model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4894–4903.

[22] M. Jiang, S. Huang, J. Duan, Q. Zhao, Salicon: Saliency in context, in: CVPR, 2015, pp. 1072–1080.

[23] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: CVPR, 2014, pp. 2798–2805.

[24] M. Kümmerer, L. Theis, M. Bethge, Deep Gaze I: Boosting saliency prediction with feature maps trained on imagenet, in: ICLR Workshop, 2015.

[25] S. S. Kruthiventi, K. Ayush, R. V. Babu, Deepfix: A fully convolutional neural network for predicting human eye fixations, IEEE Transactions on Image Processing 26 (9) (2017) 4446–4456.

[26] N. Liu, J. Han, D. Zhang, S. Wen, T. Liu, Predicting eye fixations using convolutional neural networks, in: CVPR, 2015, pp. 362–370.

[27] N. D. Bruce, C. Catton, S. Janjic, A deeper look at saliency: Feature contrast, semantics, and beyond, in: CVPR, 2016, pp. 516–524.

[28] S. Jetley, N. Murray, E. Vig, End-to-end saliency mapping via probability distribution prediction, in: CVPR, 2016, pp. 5753–5761.

[29] H. J. Seo, P. Milanfar, Static and space-time visual saliency detection by self-resemblance, Journal of Vision 9 (12) (2009) 15.

[30] X. Cui, Q. Liu, D. Metaxas, Temporal spectral residual: fast motion saliency detection, in: ACM-MM, 2009, pp. 617–620.

[31] C. Guo, L. Zhang, A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, IEEE Transactions on Image Processing 19 (1) (2010) 185–198.

[32] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai, C.-W. Lin, A video saliency detection model in compressed domain, IEEE Transactions on Circuits and Systems for Video Technology 24 (1) (2014) 27–38.

[33] V. Mahadevan, N. Vasconcelos, Spatiotemporal saliency in dynamic scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 171–177.

[34] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, H.-Y. Shum, Learning to detect a salient object, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2) (2011) 353–367.

[35] J. Li, Y. Tian, T. Huang, W. Gao, Probabilistic multi-task learning for visual saliency estimation in video, International Journal of Computer Vision 90 (2) (2010) 150–165.

[36] D. Rudoy, D. B. Goldman, E. Shechtman, L. Zelnik-Manor, Learning video saliency from human gaze using candidate selection, in: CVPR, 2013, pp. 1147–1154.

[37] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, S. Yan, Static saliency vs. dynamic saliency: A comparative study, in: ACM-MM, MM '13, 2013, pp. 987–996.

[38] Y. Kavak, E. Erdem, A. Erdem, A comparative study for feature integration strategies in dynamic saliency estimation, Signal Processing: Image Communication 51 (2017) 13–25.

[39] S. Kullback, Information theory and statistics, Courier Corporation, 1997.

[40] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000) 99–121.

[41] Q. Zhao, C. Koch, Learning a saliency map using fixated locations in natural scenes, Journal of Vision 11 (3) (2011) 1–15.

[42] L. Itti, R. Carmi, Eye-tracking data from human volunteers watching complex video stimuli. crcns.org., http://dx.doi.org/10.6080/K0TD9V7F.

[43] M. D. Rodriguez, J. Ahmed, M. Shah, Action mach a spatio-temporal maximum average correlation height filter for action recognition, in: CVPR, 2008, pp. 1–8.

[44] B. W. Tatler, R. J. Baddeley, I. D. Gilchrist, Visual correlates of fixation selection: Effects of scale and time, Vision Research 45 (5) (2005) 643–659.

[45] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, G. W. Cottrell, SUN: A Bayesian framework for saliency using natural statistics, Journal of Vision 8 (7) (2008) 1–20.

[46] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, Y. Shan, How many bits does it take for a stimulus to be salient?, in: CVPR, 2015, pp. 5501–5510.

[47] R. J. Peters, A. Iyer, L. Itti, C. Koch, Components of bottom-up gaze allocation in natural images, Vision Research 45 (18) (2005) 2397–2416.

[48] M. Kümmerer, T. S. Wallis, M. Bethge, Information-theoretic model comparison unifies saliency metrics, PNAS 112 (52) (2015) 16054–16059.