Differentiating Smooth Pursuit from Fixation in Video Saliency Prediction

John Ridley*
TU München
john.ridley@tum.de

Deepan Das*
TU München
deepan.das@tum.de

Mikhail Startsev
TU München
mikhail.startsev@tum.de

Abstract

The prediction of salient regions in videos strives to mirror the perception process of humans and has the potential to provide useful information for a variety of further applications. However, until recently, most of the predictive models failed to account for the various types of eye movements present in human perception and the pertinent information about focus and attention these processes can yield. Fixations and smooth pursuit are two such disparate eye movement behaviours which have only recently been explored for application with predictive models. Furthermore, it has been shown that smooth pursuit can provide more reliable saliency details for dynamic motion in videos, despite it being generally less common in comparison to fixations. We propose and examine a variety of different deep learning architectures and approaches to deal with these two types of eye movements, based on various commonly-used pre-trained image and video feature extractors. As a part of our approach, we compare models that deal with the eye movement types simultaneously as well as those that are specialised for specific types. In addition, we also examine various augmentations to the architectures and loss functions to improve certain performance aspects.

1. Introduction

The identification of salient regions in imagery, those which are the focus of human observers, can yield valuable information for a wide range of further applications, including compression, transmission, object detection and tracking.

A multitude of predictive saliency models have been explored in previous approaches, both in image and video contexts. However, given the diverse range of techniques, it is often difficult to quantify the performance of such models' predictive abilities, especially in comparison with a human observer. These challenges are compounded when considering the representation of saliency.

Saliency is often abstracted into a single map of focal regions (fixation onsets) for a given observed image or video frame. This generalisation stems as a result of the pre-processing of ground-truth eye-tracking data, where data is often conceptualised in a more straightforward form with a focus on instantaneous locations. Having been developed on perception of static images, this generalisation does not account for the more dynamic elements of eye movement.

Two common subclasses of eye movement, often overlooked during saliency prediction, are fixation (*Fix*) and smooth pursuit (*SP*). Fixations represent more static representations of visual attention, and can sometimes be unreliable due to their spurious nature. Smooth pursuit, in contrast, occurs in a more dynamic sense as the eye tracks a moving target or object. SP is a more reliable indicator of attention while trying to anticipate observed motion [1, 2], but is limited to its dynamic context and hence not always present.

Neither fixation nor smooth pursuit shows a clear benefit for sole use in a video saliency prediction application. Therefore our approach focuses on the application of both of these eye movement types in a predictive pipeline. We propose and examine various deep learning architectures for predicting both the fixation and smooth pursuit present in videos. We also compare approaches that combine the prediction of both eye movement types to those specialised on a single type as well as potential augmentations to improve specialisation.

To facilitate more complex architectures, we also utilise and compare various common pre-trained temporal and single-frame feature extraction architectures as a means to provide suitable foundations for saliency prediction. This is in contrast to previous works, which have constructed and trained custom feature extractors.

2. Existing Approaches & Dataset

With deep-learning becoming the new norm for saliency detection approaches, there have been various techniques proposed to output individual or collections of saliency maps, given a video input sequence. As more powerful hardware becomes readily available, the technical challenges inherent to performing predictions with temporal video information become less of an obstacle to more substantial and descriptive models.

One of the most popular datasets used for video saliency prediction is Hollywood2 [3], accompanied by a leaderboard [4] that lists the best performing models on its test set. The maintainers of the leaderboard also introduced ACLNet [5] for the video saliency detection task. The design used a CNN encoder to create a feature map, which was then modulated by an attention map created from a self-attention module. The output, in line with the popular visual question answering architectures using attention, was then fed into a convolutional long short-term memory (*ConvLSTM*) module to process the temporally varying feature maps jointly.

SalSAC [6], the current best performer on the Hollywood2 test set, used a relatively straightforward encoder-decoder structure, with a temporal component at its core. To solve the challenge of variations in object scale, it used self-attention between various depths in its encoder to extract features from various scales. A FlowNet-style correlation layer was used to consider the difference from the previous frame in the sequence, which was also fed into the ConvLSTM.

STRA-Net [7] took a different approach in consideration of temporal features. It used two branches, a regular image processing branch (similar to existing approaches) and a motion branch which utilised pre-generated inter-frame optical flow. The authors demonstrated strong performance on scenes with motion after using a feature fusion process to combine the two branches. The technique also used ResNet-based feature extractors and a gated recurrent unit (GRU) for temporal features, rather than the more common LSTM found in other works.

SalEMA [8] used either a ConvLSTM or an Exponential Moving Average layer as its temporally aware module. As observed previously, the temporal module was placed between the CNN encoder and decoder. In the EMA layer, weighted feature maps of previous frames were added to that of the current frame, thus ensuring that the impact of a specific frame has spread temporally.

Most of the approaches to video saliency detection employed recurrent units (often LSTM or GRU) for temporal considerations across a given video sequence. TASED-Net [9], in contrast, utilised only convolutions along both the spatial and temporal dimensions of the input. A contemporary pooling-unpooling technique was used as a means of feature selection and upscaling, with the authors also proposing 'auxiliary pooling' skip connections to provide scale-specific detail.

There were a few common design considerations noted in the architectures discussed above.

- Temporal modules were generally located in the middle of the network, often as a bridge between encoder and decoder.
- Simpler architectures could often achieve state-of-art performance, often rendering convoluted extensions futile.
- Off-the-shelf networks (primarily VGG) were commonly employed as a means to initialise feature extractors, but their pre-trained versions were not used out-of-the-box.
- ConvLSTM units were often employed as a means to learn the temporal behaviours of processed features.

A drawback of the approaches discussed was that the authors focused on predicting the conventional form of saliency, which is not eye movement-aware. This was most likely a result of the default lack of movement-type labels for most datasets. Furthermore, the addition of movement-type labels does increase the difficulty of making certain predictions. While fixation prediction can be made independently per video frame, smooth pursuit detection is more dependent on the identification of a temporally contiguous saliency trajectory/volume. This would tend to indicate the necessity of temporally-aware modules like ConvLSTM, as utilised in previous works.

However, the simple inclusion of such modules does not ensure appropriate handling of temporal context. This becomes more prominent in the case of SP since it occurs far less frequently than Fix, often following a small target across vast stretches of the display area. Thus, current work should focus on handling the absence of SP, tracking small salient regions demonstrating significant spatial variance and across longer time durations.

In contrast to this trend, the authors of [10] explored the application of deep-learning models to the prediction of specific forms of saliency. A slicing CNN architecture was proposed, which performed a series of convolutions across both spatial and temporal dimensions. The authors demonstrated that networks trained on specific forms of eye movement did show a tendency to specialise, and introduced a new metric named Cross

AUC to quantify this. Furthermore, their model was able to outperform other state-of-art approaches in predicting Fix or SP precisely.

The authors went on to demonstrate that models trained additionally with smooth pursuit data were also able to perform better on fixation tasks. Based on this and the aforementioned benefits of SP, a significant focus of the current work was to develop models that are better SP-detectors than they are Fix-detectors.

3. Proposed Approach

We started with a basic spatio-temporally aware architecture to establish a benchmark, then developed a series of refinements. We discuss these refinements, architectural variations and data handling in the subsequent sections.

3.1. Dataset

There are apparent challenges with the availability of datasets that have labels beyond basic fixation onsets. This results in limited options for dataset selection in our context. We chose the Hollywood2 and GazeCom datasets since they had appropriate movement-type labels, as provided in [10].

Hollywood2 consists of ∼20 hours of video excerpts from 69 Hollywood movies. The train and test splits respectively consist of 823 clips (from 33 movies) and 884 clips (from 36 movies), with no overlap of movies between them. Eye-tracking ground truth was obtained from 16 individuals viewing the clips. As the Hollywood2 dataset is significantly larger than GazeCom (1707 vs 18 videos), we also chose to utilise the former's training split as a basis to train all of our models. We further split the train set into separate train and evaluation subsets, amounting to approximately 90% and 10% of the original set respectively. For testing, we use a 50 video subset of the test set, which is identical to the set used for testing by [10]. This enables direct comparison with other eye movement-aware saliency prediction approaches.

The content of the Hollywood2 and GazeCom datasets provide a notable juxtaposition of saliency contexts. The cinematic scenes in Hollywood2 are constructed scenes with apparent human influence in the framing, positioning and motion of objects. The GazeCom dataset, however, provides a broader range of more natural 'in the wild' scenes which are not predisposed to direct the observers' attention. While it contains significantly fewer videos (only the real-world videos are used) than Hollywood2, the sequences are on average longer and contain better examples of motion as a result of the static camera viewpoints.

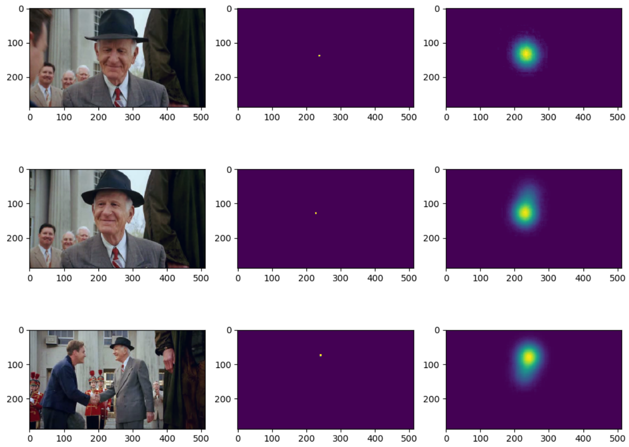


Figure 1: Original frames (left), point map *ptGT* (middle, exaggerated for visibility) and smoothed map *smGT* (right) showing effects of temporal smoothing

Hence, the more diverse GazeCom dataset serves as a challenging baseline to explore the degree to which our models have generalised.

3.2. Data Preparation

The 24fps videos of Hollywood2 were segmented into samples lasting 24 frames, with no overlap. Zero-padding was applied when necessary to keep the number of frames consistent across all samples. Since the train/eval splitting occurred at the video-level, it ensured that samples from the same video remained in the same spilt. All videos were resized to 288×512 pixels, in keeping with the recommended input dimension of ResNet. The GazeCom videos were down-sampled in FFmpeg from 30fps to 24fps to maintain consistent frame rates between the two datasets.

The ground truth was provided as a list of salient pixels recorded at 250 Hz. This was converted to 24fps point maps, where each pixel in a frame denoted the number of times its corresponding pixel in the video was recorded to be a salient region. The count was obtained from all observers at all timesteps of the eye-tracker data aligned to a particular video frame. The point maps (*ptGT*) were separated into two sets of ground truth corresponding to classes SP and Fix respectively.

We also smoothed the point maps spatio-temporally to obtain *smGT*. The Gaussian kernel for temporal smoothing had a sigma of 8.25 (1/3 seconds of footage) to redistribute the saliency value over a single second. The spatial sigmas were set to 26.178, which equates to approximately 1° of visual angle. These values were adopted from [10]. As before, the processed maps were also split according to classes Fix and SP. Figure 1 depicts example outputs of the pre-processing stages.

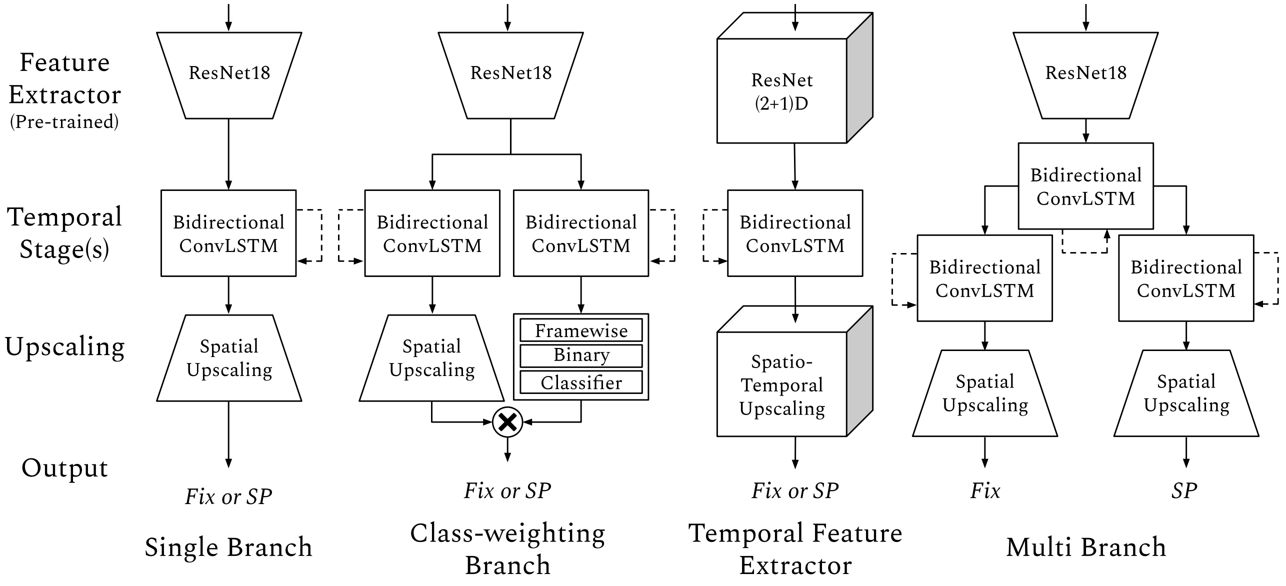


Figure 2: Proposed Architectures

3.3. Architectures

We devised a range of architectures to compare different saliency prediction approaches, their capability of specialisation towards either Fix or SP and to explore the feasibility of model combinations. The goal of each of the models was to output a series of fixation and/or smooth pursuit maps, given a sequence of input video frames.

In the feature extraction stages of previous approaches, most relied either on training from scratch or on updating pre-trained weights. In contrast, the weights of our ResNet-based feature extractors, used to provide initial features to be utilised in later stages, were never updated throughout the training process. This solution allowed us to consider more elaborate temporal architectures without running into hardware limitations. Additionally, differences in architecture design did not influence the performance of the feature extractor in any way, thus enabling fairer comparisons.

An overview of our proposed architectures is illustrated in Figure 2 and elaborated upon in the following subsections.

3.3.1 Single Branch (*Br1*)

In line with many conventional approaches to predicting fixation onset, we first proposed a model that predicts a map with only one type of saliency and utilises a bidirectional ConvLSTM (*ConvBLSTM*) to enable temporally-aware predictions. Being limited to a single output map, the network had to be trained separately for Fix and SP, with the goal being that each network becomes ‘specialised’ to its own type of eye movement.

Image features were first extracted for every video frame using a pre-trained ResNet18, which had its average pooling and fully connected layers removed. The 512-channel features were then passed through a ConvBLSTM, which carried spatial features both forward and backward along the temporal axis. This was followed by another frame-wise CNN decoder stage that simultaneously reduced the channel count and increased the output size in stages. The result was a single-channel saliency map at the original size of the input frames, such that each frame in the sequence corresponded to a frame of the output saliency map. Kullback-Leibler Divergence (KLD) was the loss function used to train the network (Section 3.4).

3.3.2 Class-weighting Branch (*CW*)

One challenge when considering smooth pursuit in comparison to fixation is its relative scarcity. The implication of this data imbalance is that a model might predict smooth pursuit in situations where it does not occur. To better assist the network with discerning presence or absence of SP (or Fix), our next proposal was to add another branch to perform binary classification on whether smooth pursuit (or fixation) is present.

This branch accepted the same ResNet features and had its own dedicated ConvBLSTM. Similar to conventional classification approaches, the features taken from the convolutional stage were flattened into a feature vector which was fed through a fully-connected stage. The resulting fully-connected bottleneck had a single neuron output. A sigmoid function provided softened binary-classification between zero (video frame does not expect SP from viewer) and one (certainty of SP).

The advantage of this representation is that it can be directly used to weight the smooth pursuit map, similar to a self-attention approach. To better impose the branch's role as a 'smooth pursuit detector', we trained the branch directly with a standard binary cross-entropy loss. This was in addition to the training it received as a result of being connected to the regular KLD loss.

3.3.3 Temporal Convolution in Feature Extractor (*Temp*)

Instead of the ResNet used in *Br1*, this architecture employs a ResNet(2+1)D as a feature extractor. The extractor, pre-trained on the Kinetics-400 dataset [11], was retained till before its spatio-temporal pooling layer. The idea behind adding a 1D convolution element was to better exploit the temporal features present in the data.

As a spatio-temporal feature extractor, ResNet(2+1)D was preferable to 3D or mixed convolutions. The decomposition of 3D convolution into (2+1)D results in fewer trainable parameters. Despite that, ResNet(2+1)D outperforms the 3D and mixed variants in the action classification task of Kinetics-400.

Given the input dimension of 24×3×112×112 (frames × channels × recommended height × recommended length), the ResNet(2+1)D output a 5×512×7×7 feature map. This was passed through a ConvBLSTM, then upsampled to 25×3×18×32, to maintain parity with the remaining architectures. The upsampling step was placed after the ConvBLSTM, so that it could process the extracted features while they were still undistorted by upsampling. This network was also trained using KLD.

3.3.4 Multi Branch (*Br2*)

The networks above were primarily designed to, trained to and specialised for a specific eye movement type. The desire to predict both smooth pursuit and fixations simultaneously would require two appropriately specialised subnetworks. Thus, we proposed a multi-branch approach to provide simultaneous predictions.

Each 'specialised' branch has a ConvBLSTM for extracting temporal features specific to the target saliency type. Both branches also share a common ConvBLSTM to enable the shared training to construct a robust temporal feature extractor.

Each branch was trained exactly as was Br1 and backpropagation occurred over the sum of the branches' KLD losses. As a means to further improve the training approach, we then proposed various modifications to the loss formulation.

3.4. Loss Formulation

Kullback-Leibler Divergence (KLD) formed the basis for the training loss for all of our approaches. The smoothed ground truth saliency ($smGT$) and the predicted network outputs (Out) were both converted into three-dimensional probability distributions to exploit KLD's ability to capture the disparity between two distributions. Let a pixel at location (h, w) at time t in the predicted output M be denoted by m_{thw}. The elements of the three-dimensional maps were then divided by the sum of the map, which yielded a probability distribution while retaining the relative scales of the map.

$$p(m_{thw}) = \frac{m_{thw} + \epsilon}{\sum_{t'} \sum_{h'} \sum_{w'} (m_{t'h'w'} + \epsilon)}$$

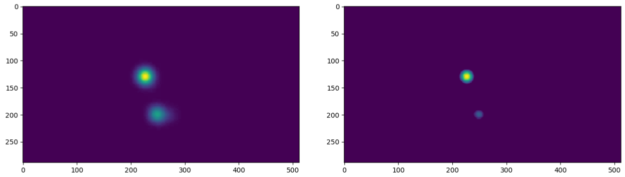
Care had to be taken to handle cases where no salient regions were recorded, which was often the case for the more infrequent smooth pursuit instances. These were handled by the ϵ term, which replaced an absence of SP with a uniform distribution. This had a negligible numerical impact on both the KLD loss and the relevant performance metrics.

$$\mathcal{L}_{KL}(smGT, Out) = D_{KL}(\mathcal{P}(M_{smGT})||\mathcal{Q}(M_{Out}))$$

For network Br2, where two KLD losses were present at training, we considered various loss augmentations.

- L1 Component (*L1*): An L1-norm of the predicted saliency map was applied to both of the losses to impose sparsity on the maps. Since our pre-trained image feature extractors were originally trained for object classification, all those regions of *Out* tended to get activated that corresponded to large, easily recognisable objects in the video. However, a human's observations may be directed more specifically to parts of particular objects (only eyes instead of the whole face) or to non-dominant objects (subtitles, images of text). The purpose of the L1 term was to weight against marking large regions as salient in preference of smaller regions, providing a sharper contrast between the salient and non-salient regions in the map. A factor of 10 was applied to the L1 component to bring it into the same order of magnitude as the initial KLD loss.

$$\mathcal{L}_{L1} = \mathcal{L}_{KL}(smGT, Out) + 10||Out||_1$$



(a) Normal Map *smGT* (b) Sharpened Map *ceGT*

Figure 3: Comparison of processed maps

- Contrast Enhancement (*CE*): With similar motivation as behind the L1 approach, we applied an additional loss based on a 'contrast-enhanced' ground truth. By weighting this loss more, more weight was assigned to the centre of saliency regions. Thus, the more tightly the predicted saliency regions conformed to the saliency regions of ground truth, the further this loss decreased. A floor of 0.2 was empirically selected to provide sharper regions, the threshold being applied after normalising the map using the sum of its elements. The effect this creates is apparent in Figure 3. This loss was also weighted by a factor of 2 to make its magnitude comparable to the initial KLD loss. Let the contrast-enhanced ground truth map be $ceGT$, with each pixel denoted by c_{thw}. Let that pixel in $smGT$ be m_{thw}.

$$c_{thw} = \begin{cases} 0.2 & \text{where } m_{thw} \leq 0.2 \\ m_{thw} & \text{else} \end{cases}$$

$$\mathcal{L}_{CE} = \mathcal{L}_{KL}(smGT, Out) + 2\mathcal{L}_{KL}(ceGT, Out)$$

To analyse the effect of selected losses, we compare the approaches, along with the aforementioned network architectures across a range of standard saliency performance metrics.

3.5. Performance Metrics

Performance metrics allow us to compare network performances independently of their loss values, providing the ability to directly compare various architectures and training approaches (i.e. loss functions). We utilise a suite of standard metrics, mostly adopted from [5, 12, 13]. These metrics have been modified to handle saliency volumes rather than 2D saliency maps, using the approach defined in [10]. Of the large collection of metrics utilised across previous approaches, we employ a subset outlined follows.

- Normalised Scanpath Saliency (NSS): The individual pixels of *Out* are z-score normalised and multiplied element-wise to *ptGT*. A positive value indicates correspondence between prediction and the ground truth point map. NSS has been widely accepted as the primary metric used for comparing saliency predictors, according to [12].

- AUC-Borji (AUC-B): Random pixels are sampled from within the salient locations of *ptGT* as positive samples, while random pixels from outside the salient locations are considered negative samples. The same pixels from *Out* are counted as positive predictions if they are above a certain threshold. This allows estimating the area under an ROC curve.

  We decided to drop AUC-Judd as a metric since AUC-B is a discrete approximation of AUC-Judd [13] and it is not as sensitive to changes in model performance as other metrics [4]. This lack of sensitivity was also observed across another metric, Shuffled AUC. It focuses on penalising centre bias by taking negative samples from salient regions in other videos, even if it correctly identifies salient regions in central locations. This is not ideal for Hollywood2, which does exhibit a strong bias towards a central location [5]. Additionally, the intensive sampling regiment requires significant resources to enable application across larger datasets, which makes testing multiple models less feasible.

- Similarity (SIM): The similarity between *smGT* and *Out*, measured using the histogram intersection between the two maps after normalization. A similarity of 1 indicates identical distributions, and 0 indicates no overlap whatsoever.

- Correlation Coefficient (CC): The Pearson's correlation coefficient between *smGT* and *Out*. Positive values indicate strong correlation and negative values indicate anti-correlation. The values range from -1 to 1, where close to 0 indicates no correlation.

- Cross AUC (xAUC) [10]: In this work, xAUC is always defined to be between $ptGT_{SP}$ and Out_{SP}. The difference from AUC-Borji is that the negative samples are taken from the positive fixation locations of $ptGT_{Fix}$. Since a major focus of this work is to distinguish between the two different kinds of eye movement, we are interested in determining how tuned a network (or part thereof) is to detecting SP. For example, an xAUC score of 0.5 between $ptGT_{SP}$ and Out_{SP} (with fixation as negative samples) indicates that the network can detect smooth pursuit and fixation equally well.

	SP Train	**Fix Train**	**Model**
SP Test	3.07	3.07	Br1
Fix Test	2.64	2.94	
SP Test	3.10	3.09	CW
Fix Test	2.51	2.94	
SP Test	3.00	2.97	Temp
Fix Test	2.48	2.71	
SP Test	**3.21**	3.21	Br2
Fix Test	2.76	2.76	
SP Test	**3.21**	3.10	Br2-L1
Fix Test	2.75	**2.95**	
SP Test	2.78	2.78	Br2-CE
Fix Test	2.11	2.10	

Table 1: NSS Specialisation Matrices (Hollywood2)

A higher score indicates that an SP predictor specialises the way it was intended, being more sensitive to SP than to Fix. It follows that a score lower than 0.5 is undesirable for an SP predictor, but desirable for a Fix predictor.

4. Results

Based on the metrics above, the following results have been generated on the Hollywood2 50-video test set. We tested all of our approaches, which were trained on the full Hollywood2 train set. Where applicable, we trained individual networks specialised on smooth pursuit and fixations.

Since NSS is the metric of choice for architecture comparison, we use this to compare the methods' specialisation to specific eye movement types. Table 1 compares the NSS values for both eye movement types and across branches/specialisation.

Tables 2 and 3 contain the results for the other metrics, focusing on networks trained and tested on fixations and smooth pursuit respectively.

4.1. Interpreting the Specialisation Matrix

In the Specialisation Matrix, 'SP Train' corresponds to a model trained on the smooth pursuit data only from train set, whereas 'SP Test' refers to a model being used to predict only smooth pursuit from the test set. 'Fix Train' and 'Fix Test' have similar interpretations. Thus, the cell at SP Train / Fix Test under model Br1 is the NSS value computed between the output *Out* of Br1 trained using smooth pursuit train data and the ground truth *ptGT* of fixation test data.

The smooth pursuit and fixation ground truths are not identical and their occurrence is imbalanced. Thus, metrics should not be compared across SP Test and Fix Test. However, SP Train and Fix Train values under the same Test (SP or Fix, i.e. the same ground truth) can be compared as a means to examine specialisation.

Model	**AUC-B**	**SIM**	**CC**	**xAUC**
Br1	**0.918**	0.368	**0.452**	0.481
CW	0.917	0.360	0.436	**0.479**
Temp	0.904	0.343	0.424	0.524
Br2	0.916	**0.369**	0.451	0.520
Br2-L1	0.916	0.338	0.387	0.520
Br2-CE	0.882	**0.369**	0.451	0.490

Table 2: Trained & tested on Fix (Hollywood2)

Model	**AUC-B**	**SIM**	**CC**	**xAUC**
Br1	0.860	0.243	0.333	0.517
CW	0.869	0.247	0.343	**0.546**
Temp	**0.872**	0.232	0.340	0.543
Br2	0.871	**0.254**	**0.357**	0.522
Br2-L1	0.871	**0.254**	0.352	0.522
Br2-CE	0.856	0.228	0.295	0.530

Table 3: Trained & tested on SP (Hollywood2)

4.2. Eye Movement Specialisation

Specialisations of the networks/branches to certain eye movement types yield insight into their utilisation of training data. In the case of smooth pursuit prediction, it would seem that there is often negligible improvement when a network is trained on SP instead of Fix ground truths. This is evidenced by SP Test / SP Train being only marginally higher than SP Test / Fix Train in all approaches.

The lack of difference potentially stems from the selection of NSS as a comparison metric. NSS primarily evaluates the network's ability to localise on the salient video regions, and it is this localisation which would appear to be consistent between the two eye movement types. This behaviour can be expected in the context of cinematographic scenes, in which particular objects are deliberately framed to draw the viewer's attention.

The effect of this is also apparent when looking at the specialisation in the fixation case. In almost every approach, there is usually a marked performance decline in fixation prediction when trained on SP (i.e. Fix Test / Fix Train is usually much higher than Fix Test / SP Train). Since smooth pursuit occurs more infrequently than fixations, the networks are trained on significantly fewer 'localised' objects than with the fixations training set.

However, based on this observation, one would expect that a network trained on Fix would perform better for a smooth pursuit task (since there are more samples to learn localisation from). The fact that this is not the case would tend to indicate that there is a

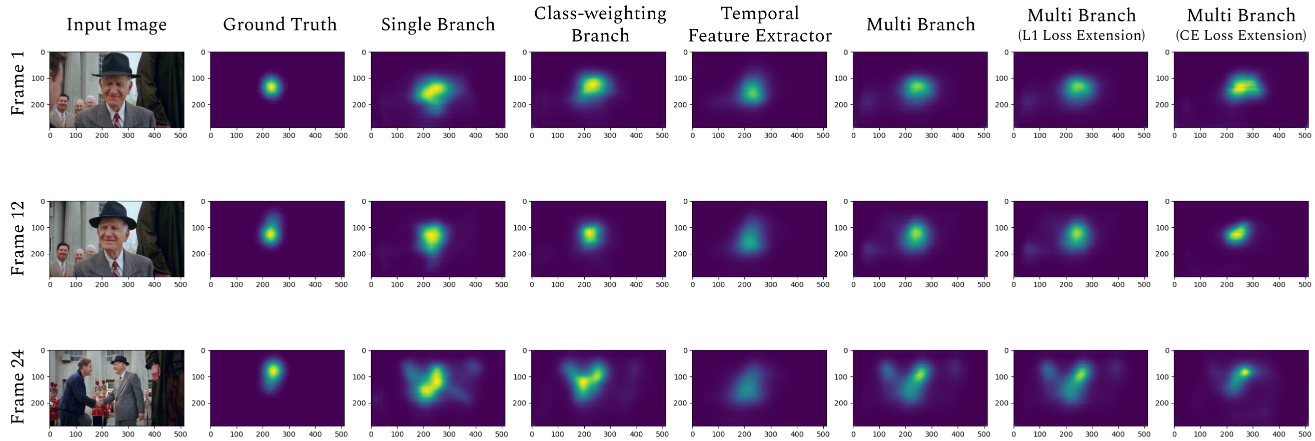


Figure 4: Fixation Output

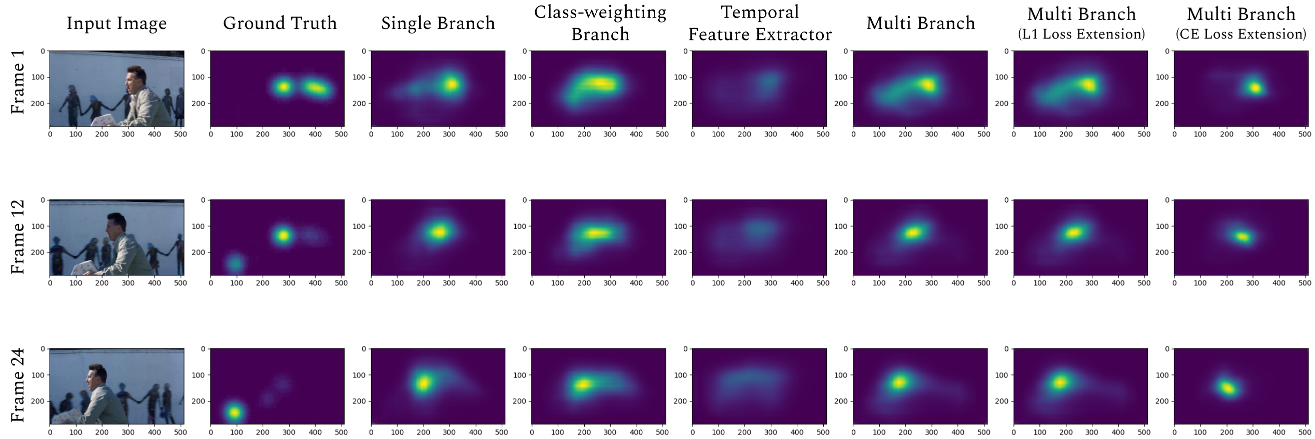


Figure 5: Smooth Pursuit Output

certain degree of specialisation occurring.

The work of [10] noted that training a detector on SP has potential benefits for more conventional forms of saliency in an unseen dataset. While we did not achieve a clear benefit on fixation performance when training on smooth pursuit, Table 4 does show similar Fix NSS values for some models (CW, Temp, Br2), irrespective of their training on SP or Fix.

The measure of specialisation is better quantified in the xAUC metric, which measures the tendency of a network to favour smooth pursuit over fixation. Table 3 indicates that all architectures have scored above the baseline of 0.5 (statistical indifference), meaning that they have become more sensitive to smooth pursuit in comparison with fixations. This displays the clear benefit of training smooth pursuit detectors on SP ground truths, notably in the single branch networks.

Despite the tendency to specialise, the xAUC values do indicate that the degree of specialisation is not particularly significant since the values are close to the 0.5 threshold. Looking at the same metric for fixations (in Table 2), it is clear that the single branch networks specialise better for fixations (less than 0.5 xAUC). However, networks with certain modifications are still more sensitive to smooth pursuit. We explore the effects of different architectures/modifications in their following relevant sections.

4.3. Feature Extractor Comparison

The pre-trained ResNet18 extractor we employed for most of our approaches generated a feature map on a frame-wise basis, which implies the features were generated without temporal consideration. On the other hand, the pre-trained ResNet(2+1)D is a temporally aware feature extractor, for which we also had to reduce the input image size. Thus, Br1 and Temp are identical single branch networks with the only difference being the feature extractor. This enables us to compare the two extractors directly.

As evident in Table 1, Br1 outperforms Temp across every NSS metric, although the results are very similar. This would seem to indicate the pre-trained temporal features are not providing any benefit over the temporal features already extracted by our ConvBLSTM stages. However, the xAUC metric in Tables 2 and 3 would suggest that the temporal extractor would tend to bias the network more towards smooth pursuit features, even when trained on fixations. One further observation from Figures 4 and 5 is that the predicted maps of Temp tend to be more uniform and lacking in clear focal points, compared to the other models.

Overall, in the case of Hollywood2, the use of a temporal feature extractor does not appear to provide a clear benefit over a more conventional ResNet feature extractor.

4.4. Class-weighting

In an effort to increase a single branched network's performance on the less common smooth pursuit instances, we proposed a class-weighting branch to detect the presence of smooth pursuit and then weight the map accordingly. Our motivation toward this approach is supported in the previous results, where fixation and smooth pursuit would seem to localise similarly (in Hollywood2).

Given our class-weighted network (CW), we again compare with our standard single branch (Br1) approach. Looking at Table 1, we see a modest improvement in SP performance, even when trained on fixations. The fact both SP Test values improve would tend to indicate the network does weight down maps to the point that poor results are dampened (leading to a small improvement in NSS). However, the fact that smooth pursuit-trained approaches perform better would tend to indicate a degree of smooth pursuit specialisation, even though the difference is quite minimal.

This is supported when considering the xAUC for CW, which is the lowest and highest value in Tables 2 and 3 respectively. What this shows is that the class-weighting branch is able to bias the network towards the desired type of saliency. Furthermore, this is achieved without having to predict the other saliency type at the same time.

4.5. Single-branch vs. Multi-branch

The goal of Br2 approach was to be able to utilise and predict both forms of eye movement simultaneously. As shown in Table 1, it achieves the highest NSS of all networks on SP, especially in comparison with the single-branch variants. However, it does not perform as well in the fixation cases.

Another notable observation for Br2 was that it achieves almost identical NSS values in both branches, despite single branches only being trained on specific eye movement types. This would suggest that (like Br1) the localisation process of saliency detection in Hollywood2 is similar between the two eye movement types. Despite this similarity, the xAUC values would suggest that both branches have become sensitive to smooth pursuit. This is potentially a result of the shared ConvBLSTM, even though each branch has an additional ConvBLSTM.

Overall, Br2 (with regular KLD loss) is the best (or close to the best) performer across the similarity and correlation metrics, suggesting that a combined multi-branch approach is better able to generate similar saliency maps across both eye movement types.

4.6. Loss Formulation

We devised various loss function augmentations to assist in 'tightening up' salient region predictions, particularly in the smooth pursuit cases. The high NSS of Br2 indicates it is better able to localise than the other networks. However, the comparison with the groundtruth (Figures 4 and 5) indicates that the maps are lacking specificity with respect to the exact salient regions.

To attempt to address this, we first explored adding the proposed L1 component (Br2-L1) to weight against maps that predicted larger saliency regions. Table 1 shows a clear decrease in NSS on the fixation branch and very little change on the smooth pursuit branch. Given the larger and more diverse fixation regions, this is to be expected (since all regions are punished to a certain extent). However, the lack of improvement in smooth pursuit performance shows no benefit to balance the costs of this augmentation. These results are consistent across the other metrics in Tables 2 and 3. Furthermore, the output maps (Figures 4 and 5) show very little difference when compared with the standalone KLD loss.

We explored adding another KL divergence loss that dealt with a more contrasting map, where salient regions were smaller and tighter. It is worth noting that this is similar to selecting smaller sigma values during pre-processing. We weighted the contrast-enhanced (Br2-CE) KLD loss more to provide further reward to regions that fell into tighter bounds. In practice, however, the approach delivers significantly degraded performance across almost all metrics. The output maps show that the network is indeed outputting 'sharper' focal regions, but these regions are not particularly better localised than their smoother counterparts.

	SP Train	Fix Train	Model
SP Test	0.99	0.97	Br1
Fix Test	0.83	0.94	
SP Test	1.00	0.75	CW
Fix Test	0.91	0.95	
SP Test	**1.42**	1.37	Temp
Fix Test	1.08	**1.09**	
SP Test	1.03	0.89	Br2
Fix Test	0.95	0.95	
SP Test	0.89	0.87	Br2-L1
Fix Test	0.95	0.89	
SP Test	0.65	0.55	Br2-CE
Fix Test	0.67	0.69	

Table 4: NSS Specialisation Matrices (GazeCom)

Net.	AUC-B	SIM	CC	xAUC
Br1	0.779	0.044	0.051	0.498
CW	0.745	**0.046**	0.055	0.478
Temp	**0.790**	**0.046**	**0.070**	0.544
Br2	0.754	0.044	0.053	0.497
Br2-L1	0.749	0.044	0.049	0.497
Br2-CE	0.697	0.037	0.035	**0.467**

Table 5: Trained & tested on Fix (GazeCom)

Net.	AUC-B	SIM	CC	xAUC
Br1	0.760	0.095	0.105	0.526
CW	**0.765**	**0.098**	0.107	0.487
Temp	0.749	**0.098**	**0.119**	**0.568**
Br2	0.749	0.096	0.100	0.508
Br2-L1	0.749	0.096	0.105	0.508
Br2-CE	0.697	0.089	0.079	0.478

Table 6: Trained & tested on SP (GazeCom)

4.7. Generalisation

As previously mentioned, Hollywood2 deals with saliency in a very controlled and manipulated context. Hence, to explore the generalisation performance of the models, we tested against the GazeCom dataset. The 'in the wild' scenes present a much more realistic environment for saliency prediction and enables us to test whether our previous observations hold in a more general context.

We took the same networks trained on Hollywood2, then used GazeCom for testing and calculating the same metrics as before.

Initial examination of the values (Tables 4, 5 and 6) shows significant decreases in all metrics in comparison with the Hollywood2 test numbers. This is evident in the networks' predisposition towards the very structured forms of saliency evident in cinematography. The 'in the wild' scenes in GazeCom present more of a challenge as there are often a larger number of discrete salient objects, unlike in Hollywood2. Thus the metrics are much more indicative of the networks' performance in unfamiliar saliency prediction scenarios. However, the predictors are to a certain degree still able to recognise both types of eye movement.

In most of the metrics, the trends between the different network types do not change. The most notable anomaly is the performance of Temp. It would seem the pre-training of its ResNet(2+1)D extractor (on video action classification) yields more appropriate features on this dataset, especially in SP. This results in the network performing significantly better than the others in smooth pursuit cases. However, the network maintains its muted feature maps (Figures 6 and 7).

There is also a shift in trends with the xAUC metric. Most networks show less sensitivity to specific eye movement types. Br2 shows better specialisation to GazeCom fixation than Hollywood2 fixation, but at the cost of smooth pursuit specialisation. As before, the performance between the two branches remained remarkably consistent. Temp is a notable exception as it becomes significantly more sensitive to smooth pursuit.

This effect is also indicated by its NSS performance, which again shows a clear bias toward smooth pursuit, even when trained with fixations. This indicates that the extractor's suitability to more general motion scenes would seem to assist in the general motion present in GazeCom.

In contrast to what was previously observed, CW reduces the hindered smooth pursuit sensitivity. This is likely a compound effect of weighting with a smooth pursuit classifier that has no experience with 'in the wild' scenes. Furthermore, Figure 7 indicates that both the Br1 and Br2 (with CE loss) develop fairly muted maps in the smooth pursuit case, similar to those produced by Temp.

Overall, training on cinematographic scenes and testing on 'in the wild' scenes highlights the explicit biases that controlled saliency imposes. As previously mentioned, scenes in GazeCom have a wider variety of variable and independent salient regions. However, an average Hollywood2 scene contains only one or two reasonably large salient regions. Thus, one general observation made when testing on GazeCom is that clusters of independent salient regions are often combined into much larger regions, similar to what is observed in Hollywood2. This is evident on examination of the feature maps in Figures 6 and 7. As expected, this only serves to further degrade the measured performance of these networks across most of the metrics.

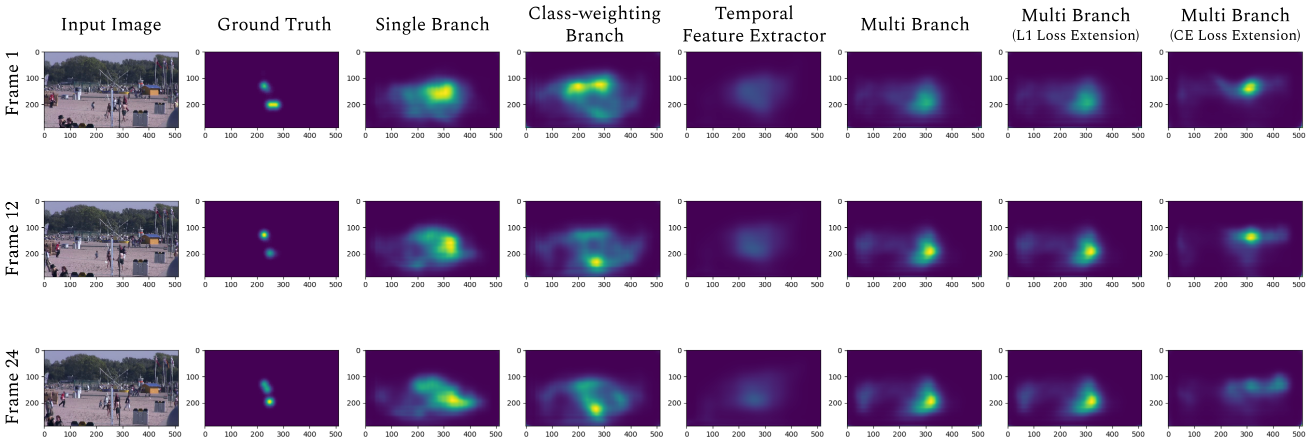


Figure 6: GazeCom Fixation Output

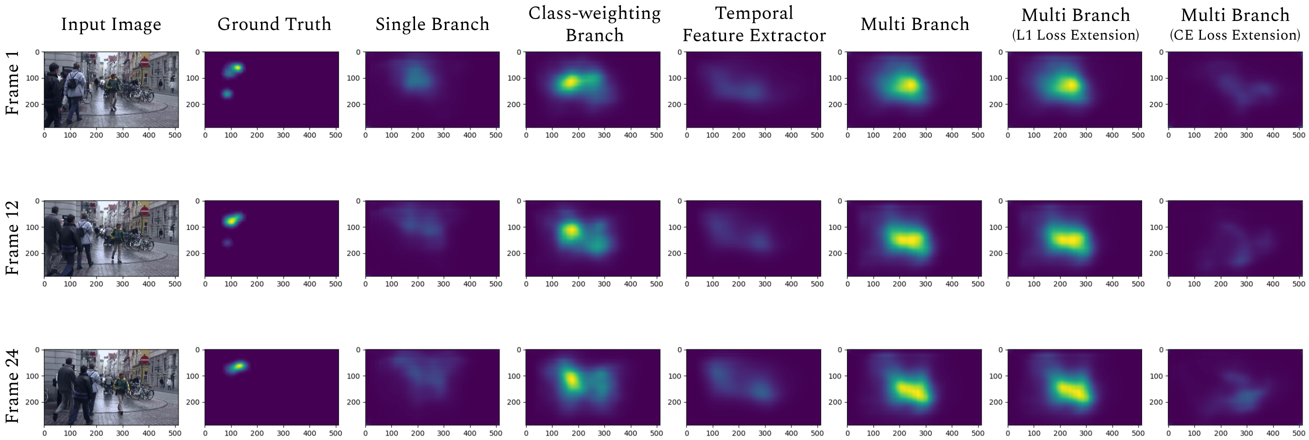


Figure 7: GazeCom Smooth Pursuit Output

Despite the performance reduction expected when applying saliency predictors to a more generalised case, specialisation to specific eye movement types is still a factor observed in most instances. However, the degree of specialisation is not particularly significant nor consistent between the two test sets.

5. Conclusion

The major contributions of this work can be summarised as:

- Instead of working on general saliency predictors for videos, the current work attempts to develop models capable of focusing on a specific type of saliency (smooth pursuit or fixation) over the others.

- In view of the previous goal, a theme common to all our explored approaches is the use of ConvLSTM modules since they can exploit both spatial and temporal features. Another feature shared by all our designs is the use of pre-trained networks as feature extractors.

- Hollywood2, characterised by a varying point-of-view observing directed scenes, is used to train and test all networks. GazeCom, characterised by a static point-of-view observing 'in the wild' activities, is used as an additional test set to determine robustness.

- One of our main contributions is a branched network design, in which each branch employs a ConvBLSTM tuned to either SP or Fix. This enables the same model to predict two eye movement classes simultaneously and can be easily extended to cover other types as well. This architecture design outperforms our others on Hollywood2.

- The other important contribution of this work is the use of a temporal component in the feature extractor, which supports the time continuity inherent in SP. The benefit of its inclusion was evident in the architecture's robustness to SP on GazeCom.

While our results do not indicate a considerable amount of specialisation to eye movement types, it is observed that certain architectures and techniques yield better specialisation to certain forms of eye movement. However, as indicated by variations between our testing on Hollywood2 and GazeCom, this is heavily dependent on the data used for training. Given richer datasets labelled with eye movements and improvements in hardware capabilities it would become possible to explore the potential for eye movement specialisation in greater detail, especially in the temporally-dependent smooth pursuit cases.

There are a few possible future directions to expand on saliency predictors sensitive to eye movement. Since audio signals temper how humans perceive actions, the inclusion of audio features of Hollywood2 can be potentially useful in SP detection. There are successful deep learning approaches capable of correlating visual objects with audio signatures [14], which might find an application here.

Another potential extension would be to extend the multi-branch approach with a single class weighting branch (which could not be explored due to hardware limitations). Rather than training the class weighting branch to detect whether a single class is present or not, this branch could be treated as a multi-class classifier. This would enable it to identify patterns in the co-occurrence and the correlation between the eye movement types, and has potential to utilise their underlying relationship.

References

[1] Miriam Spering, Alexander C Schütz, Doris I Braun, and Karl R Gegenfurtner. Keep your eyes on the ball: smooth pursuit eye movements enhance prediction of visual motion. *Journal of Neurophysiology*, 105(4):1756–1767, 2011.

[2] Aarlenne Z Khan, Philippe Lefèvre, Stephen J Heinen, and Gunnar Blohm. The default allocation of attention is broadly ahead of smooth pursuit. *Journal of Vision*, 10(13):7–7, 2010.

[3] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[4] Ming-Ming Cheng. Hollywood-2 video saliency leaderboard. `https://mmcheng.net/videosal/`, 2018. [Online; accessed 31-January-2020].

[5] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibin Ling, and Ali Borji. Revisiting video saliency prediction in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[6] Xinyi Wu, Zhenyao Wu, Jinglin Zhang, Lili Ju, and Song Wang. Salsac: A video saliency prediction model with shuffled attentions and correlation-based convlstm.

[7] Qiuxia Lai, Wenguan Wang, Hanqiu Sun, and Jianbing Shen. Video saliency prediction using spatiotemporal residual attentive networks. *IEEE Transactions on Image Processing*, 29:1113–1126, 2019.

[8] Panagiotis Linardos, Eva Mohedano, Juan Jose Nieto, Noel E O'Connor, Xavier Giro-i Nieto, and Kevin McGuinness. Simple vs complex temporal recurrences for video saliency prediction. *arXiv preprint arXiv:1907.01869*, 2019.

[9] Kyle Min and Jason J Corso. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2394–2403, 2019.

[10] Mikhail Startsev and Michael Dorr. Supersaliency: A novel pipeline for predicting smooth pursuit-based attention improves generalisability of video saliency. *IEEE Access*, 2019.

[11] 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.

[12] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.

[13] 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*, 41(3):740–757, 2018.

[14] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 570–586, 2018.