Technische Hochschule Mittelhessen (THM)

Master Thesis

Gaze recognition:

Current research and development of an AI based prototype

Severin Erasmus Stahl

supervised by
Prof. Dr.-Ing. Hartmut Weber

co-supervised by
Prof. Dr.-Ing. Martin Gräfe

Sunday 23$^{\text{rd}}$ August, 2020

Declaration Of Academic Integrity

I hereby declare that this master thesis has been written only by the undersigned and without any assistance from third parties. Furthermore, I confirm that all sources are listed and no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

Friedberg, the   Signature

Abstract

Gaze tracking systems are being researched for more than a hundred years. Yet, there is still more to be learned and improved upon. They are at this time mostly used in the medical and scientific fields. There has been recent research in less confined methods of usage for these systems. The least confined method of gaze tracking, having a camera placed independently from the observed, is probably the least researched method. If this method would achieve high degrees of accuracy even people who would act unusually while wearing an eye tracker could be have their gaze tracked easily. Therefore, this method is suitable for analyzing the gaze of the severely psychologically impaired under natural circumstances.

In this master thesis currently existent methods of gaze tracking are going to be compared against one another. There will be a focus on gaze tracking methods utilizing cameras placed independently from the observed. Further several machine learning-based prototypes designed for this situation will be presented.

The development of gaze tracking methods utilizing cameras placed independently from the observed is a complex issue. None of the in this thesis developed prototypes give decent results in their analysis of images. There are however other systems presented in this thesis where the best has a mean angular error of 17,6° on the chosen dataset.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Goals

Human communication uses a variety of ways including speech, facial expression, gestures and the direction of gaze. Eye contact being a particularly important communication through the direction of gaze.

The way a person communicates can be used to evaluate mental wellbeing. Because of these psychological tests in which the person to be examined communicates with a test supervisor in a standardized interview situation are used to classify human behavior when a psychiatric disorder is suspected. Typical for certain psychiatric disorders in such tests is, for example, a conspicuously low level of eye contact between the client and the examiner.

The number of eye contacts during a conversation is measurable and is an important diagnostic criterion among other factors. At the time of the conversation there is usually no extra evaluator available, so that the determination of the number of eye contacts has to be determined either from a subsequently created protocol of the examining person, which can only be done with limited accuracy due to the time delay between conversation and protocol creation and also due to the large number of observations in the conversation situation. Alternatively, the number of eye contacts can be determined from a video recording of the conversation situation. An automated evaluation of the video recording is desirable in order not to tie up personnel capacity for video evaluation.

The automated recognition of the direction of gaze in different personal situations is a current research topic [11, 13, 14]. Concerning the recognition of psychiatric disorders, the direction of gaze has already been examined for some time [15]. In this master's thesis, current research sources on the recognition of the direction of gaze have to be investigated and systematized. Furthermore, an AI-based approach for the recognition of gaze direction has to be implemented as a prototype. Based on the directions of gaze, the prototype should ideally also be able to detect eye contacts in two-person-conversation videos.

1.2 Foundations and Backgrounds

1.2.1 The human eye

The human eyes are being studied since ancient times. The ancient Greek philosopher Aristotle for example researched binocular vision and made assumptions on the working of the eyes [16].

Nowadays the working of the eyes is considerably better understood. In Figure 1 a labeled human eye can be seen. The pupil of the eye is the opening where the light enters the eye. It is similar to the aperture of a camera. The size of the pupil is regulated by two muscles in the iris. The cornea above it provides together with the lens the ability to focus the light on to the retina. The sclera forms the tough shell of the eye. At the sclera a group of three pairs of extraocular muscles is connected. These muscles give the eye the ability to rotate [1].

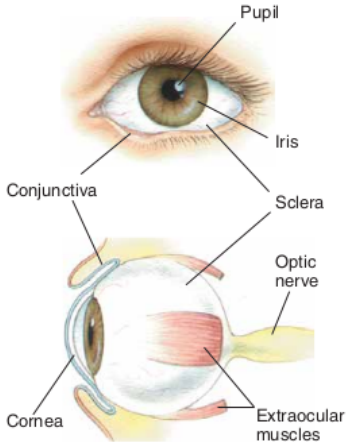

Figure 1: Gross anatomy of the human eye [1]

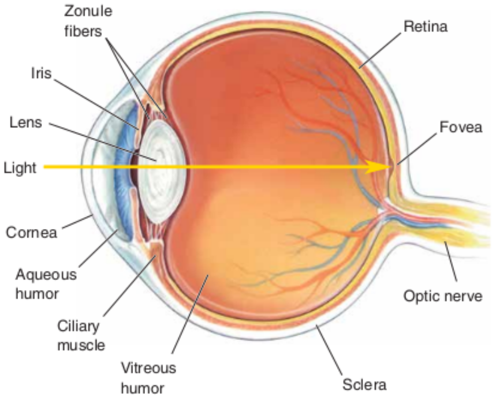

Figure 2: Cross section of the human eye [1]

In Figure 2 the labeled cross section of the human eye can be seen. Behind the pupil is the lens of the eye. The lens is held suspended by the zonule fibers. With the help of the ciliary muscle the lens can be focused to different distances allowing crisp images at different distances. The retina is the sensor element of the eye. The neural signal of the retina is sent to the brain through the optical nerve [1].

The eye is only capable of fixating on a small area. To facilitate a comprehensive visual perception the eye makes use of the ocular muscles. The movements between fixation points are called saccade. A saccade takes 10 to 100 ms. During a saccade the eye move to fast for visual processing. The target point of a saccade is assumed to be fixed and it is assumed it cannot be changed during the saccade [2]. In Figure 3 the major known elements of the oculomotor system are visualized.

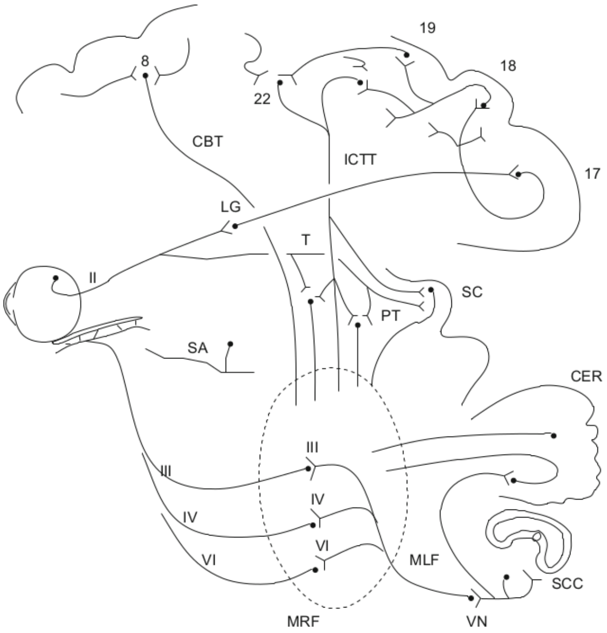

Figure 3: Schematic of the major known elements of the oculomotor system [2]. Adapted from Robinson [17] CBT, corticobulbar tract; CER, cerebellum; ICTT, internal corticotectal tract; LG, lateral geniculate body; MLF, medial longitudinal fasciculus; MRF, mesencephalic and pontine reticular formations; PT, pretectal nuclei; SA, stretch afferents from extraocular muscles; SC, superior colliculi; SCC, semicircular canals; T, tegmental nuclei; VN, vestibular nuclei; II, optic nerve; III, IV, and VI, the oculomotor, trochlear, and abducens nuclei and nerves; 17, 18, 19, 22, primary and association visual areas, occipital and parietal (Brodmann); 8, the frontal eye fields

Once the eyes have reached their rough fixation point the eyes fixate on the target. At a fixation the eye focuses on a target. After focusing on a target, the eye has a neigh zero velocity. The eye only moves for tremor, drift, and microsaccades. The changes are relatively small ranging in 12 min of arc in amplitude [2].
Additionally, the eyes are capable of pursuit movements. During these the matches their angular velocity to keep the fixation point in center and focus. This works up to a certain angular velocity. Beyond this the eyes make catchup saccades to keep the target in focus. The nystagmus of the eyes is a conjugated movement of the eyes to filter out the movement of the head and target position. The movement of the nystagmus is made of sawtooth like movements followed by a saccade [2].

1.2.2 Gaze tracking

There are three commonly used methods of tracking the eyes and estimating the gaze. The first method is the electrooculography (EOG). In the EOG electrodes are placed around the eye to measure the skins potential to estimate the gaze [18, 19]. In Figure 5 the electrodes for the EOG placed on a person can be seen. The second method are the eye-attached tracking methods (EAT). In the EAT methods an easy to track object is placed on the eye. An EAT method for example utilizes the scleral search coil. The scleral search coil is made of small wires inside a contact lens [20]. In Figure 4 a scleral search coil is depicted. The third method are camera-based designs to estimate the gaze.

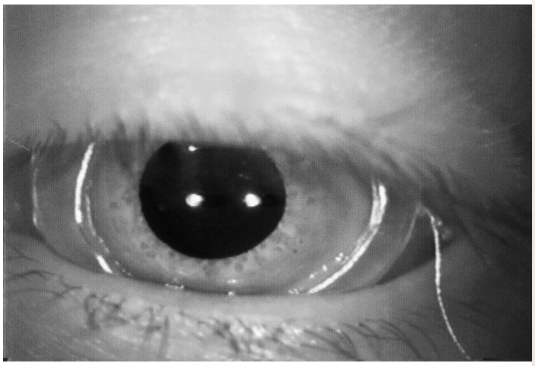

Figure 4: The scleral search coil [3]

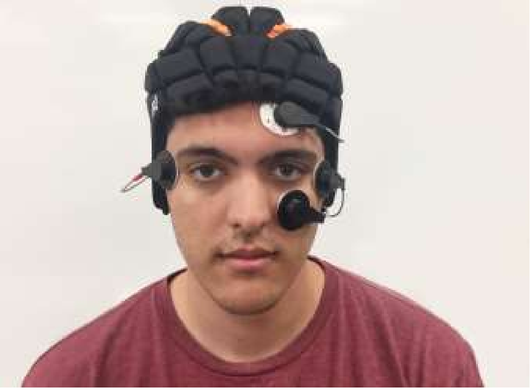

Figure 5: The electrooculography [4]

The EOG method measures a skin potential in the range of 15-200 µV. The number of electrodes varies. To accurately determine the position of focus it is further necessary to determine the position of the head [2]. It is a method with little delay, which works without any light source. However, it requires angular changes of more the 1 degree [21] to work and the cables of electrodes need to be attached. It is used in medical diagnostics.

The scleral search coil measures the electric generation within a magnetic field. It is the most accurate method of measuring the angle of gaze of a person having an error of 5 to 10 arc-seconds [2]. Like the EOG method it is a method with little delay, which works without any light source. However, it is a very uncomfortable method even risking the health of the cornea of the eye [22]. It is mostly used in research.

The camera-based designs date back to the early 20th century. The first camera-based design was made by Dodge in 1901 using the cornea reflection to detect the movement of the eyes [23]. This was also the first contact free gaze tracking device [16].

The camera-based methods can be divided into two different ways of application. The remote and the mobile eye tracking methods. The remote eye tracking methods use statically placed cameras. This category can be subdivided into two categories. The table-mounted designs and the free camera designs. In table-mounted camera setup the camera is placed on a structure where it is focused at close distance on the user. In Figure 6 a commercial application of a table-mounted camera can be seen. The free camera on contrast records a scene at greater distance and the people are not always facing the camera. The mobile eye tracking methods utilize head-mounted cameras. The mobile eye tracking arose from miniaturization of cameras and previous research on remote eye tracking methods. In Figure 7 a commercial grade head-mounted gaze tracker is shown.

Figure 6: "EyeAsteroids", an eye-controlled arcade game by Tobii Technology [5]

Figure 7: The Dikablis Eye Tracking Glasses by Ergoneers [6]

In both the head-mounted and table-mounted eye tracking designs the cornea reflection of light is commonly used. In many commercial products infrared light is used for the reflection. SR Research utilizes infrared light cornea reflection for their eye tracking devices [24, 25]. Ergoneers product Dikablis uses infrared light cornea reflection, too [26]. Tobii makes use of near-infrared light cornea reflection for their Tobii Pro Glasses 2 [27, 25]. Machine learning approaches using visible light are also considered for head-mounted and table-mounted eye tracking designs [28, 29].

There are at the time of writing no commercially available eye tracker using a free camera design. This topic is currently part of ongoing research. In the research machine learning methods are commonly used [10, 12, 11]. For more information see Chapter 2.

The camera-based methods are less accurate than the EOG or the EAT methods. The head-mounted and table-mounted eye tracking designs achieving errors as low as 1 to 3 degrees of angular error [30]. The free camera designs achieve errors of 18 degrees of angular error [12]. They are however more flexible than the EOG or the EAT methods. The table-mounted and free camera designs allowing for contact free measuring, while the free camera even allows for free movement in view of the camera. The head-mounted in contrast allows for free movement in every situation. The head-mounted however require a constant wearing of the camera.

1.2.3 Artificial Neural Networks

Artificial Neural Networks or ANN's are made of artificial neurons. These artificial neurons are can be mathematically described as Equation 1. In the equation y_k is the output of the neuron k of the current layer y, x_j is the output of the neuron j of the previous layer x, w_{kj} the weight from the neuron x_j to the neuron y_k and φ is the activation function. The activation function is commonly a threshold function.

$$y_k = \varphi(\sum_{j=0}^{m} w_{kj} x_j) \tag{1}$$

The artificial neurons are placed in layers. The layer where the input information is supplied is called input layer and the layer where the result is extracted is called output layer. All other layers are referred to as hidden layers. A simple neural network can be seen in Figure 8.

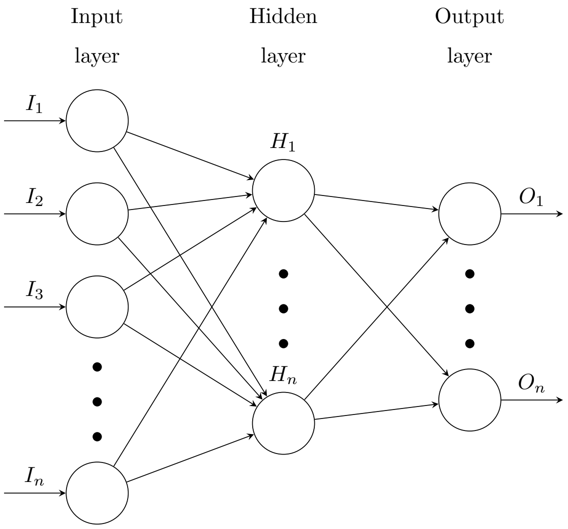

Figure 8: A simple ANN [7]

In visual analysis convolutional neural network or CNN's are widely spread. A CNN is defined by having convolution layers. A convolution layer is a neuron layer where a neuron is only connected to a convolution area on a previous layer, thus they mathematically perform a sliding dot product. This pattern requires less neuron connections than a fully connected network would require, making it easier to learn for the ANN and thus leading to better results. Due to keeping the spatial structure of the image, the kernel-based processing does cause a loss in information when processed compared to a fully connected ANN System. After the initial convolution layers a pooling layer usually follows. The pooling layer extracts the dominant feature of a convolution result. This is to improve performance with the reduction of the dimensionality of the result. In Figure 9 a sketch of a CNN can be seen.

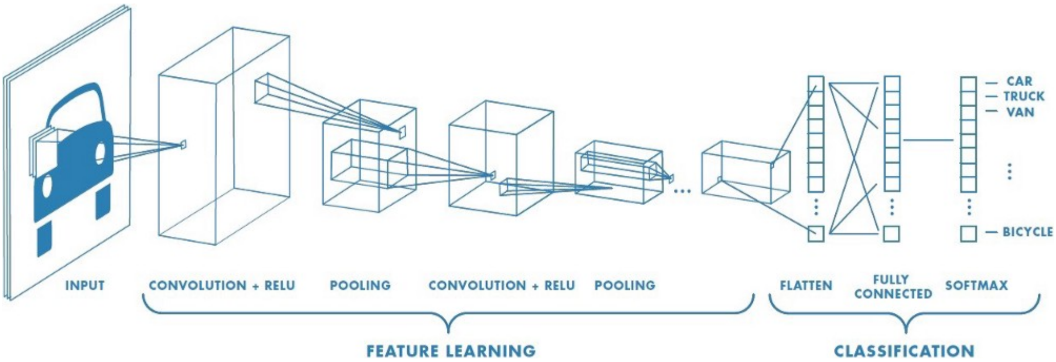

Figure 9: A sketch of a CNN [8]

The weights between the artificial neurons of the ANN's are stepwise approximated for a specific problem. This process is commonly referred as training. Common training algorithms are various kinds of gradient descent, the Newton method or Levenberg-Marquardt algorithm. All of them work with iteratively adapting the ANN's weights to solve the given training data. If too much training is done relative to the amount of training data and network size, the ANN will be to strongly adapted for the training data. This will prevent it from solving other problems of similar kind. This effect is called overfitting. In Figure 10 the fitting of training is illustrated.

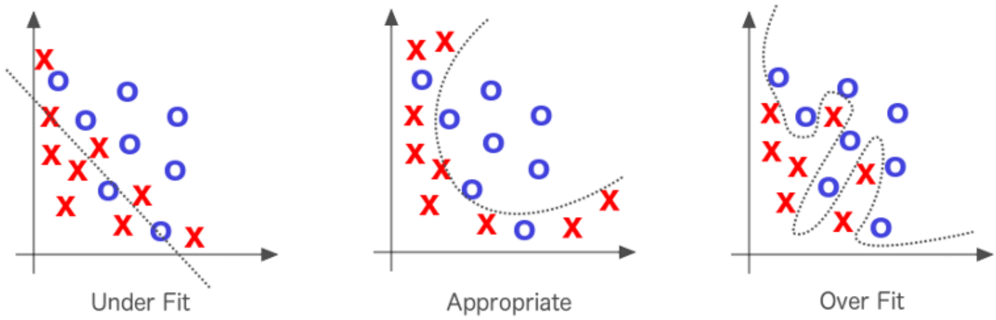

Figure 10: ANN fitting [9]

For the development of ANN's specialized frameworks are usually being used. Common ones are Caffe, Tensorflow and Torch. They provide prebuilt layer structures and learning algorithms for easy and generalized use. Further they allow deployment and training on various systems, as well as importing and exporting of ANN models.

1.2.4 Frameworks

Caffe

Caffe (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework first released in 2014 by Berkeley Artificial Intelligence Research (BAIR). The Caffe framework is made for high speed neural network processing and training. The Caffe framework is written in C++, but provides interfaces for both Matlab and Python. It is completely open-source and expandable [31].
Caffe is not only able to process an ANN on a CPU, but also capable running the ANN on a GPU. This can be achieved with a NVIDIA GPU and the NVIDIA CUDA platform. It is also able to use the CUDA Deep Neural Network library called cuDNN to speed up processing on a GPU [32].
To run Caffe on a system it needs to be compiled on the target system. On Linux a suitable Compiler is the GCC and a Compiler for Windows is the Visual Studio. The Matlab interface requires a C++ Compiler capable of producing a compatible .mex file for the installed version of Matlab [33]. The Python interface requires an installation of Python 2.7.X or Python 3.3+ [32]. The Caffe framework has been cited more than 5000 times since its publication in November 2014 to time of writing [31].

Keras

Keras is a machine learning framework first released in 2015 by Chollet et al. on Github. It is written exclusively in Python focusing on being user-friendly, modular and extensible. It was integrated into Tensorflow [34]. Keras is not only able to process an ANN on a CPU, but also capable running the ANN on a GPU. This can be achieved with a NVIDIA GPU and the NVIDIA CUDA platform. It is also able to use the CUDA Deep Neural Network library called cuDNN to speed up processing on a GPU [35]. On various package managers the installation of NVIDIA CUDA and the cuDNN library is handled automatically. It is also capable to run on Tensor Processing Units (TPUs) [35, 36].

PyTorch

PyTorch is an open source machine learning framework first released in 2016 by Facebook's AI Research lab (FAIR). It is based on the Torch framework. It is commonly used with a Python interface however the framework is written in C++ to provide high speed neural network processing and training [37]. PyTorch is not only able to process an ANN on a CPU, but also capable running the ANN on a GPU. This can be achieved with a NVIDIA GPU and the NVIDIA CUDA platform. It is also able to use the CUDA Deep Neural Network library called cuDNN to speed up processing on a GPU [38]. On various package managers the installation of NVIDIA CUDA and the cuDNN library is handled automatically.

2 State of the Art of Gaze Recognition Systems

2.1 Datasets

There are several different datasets for gaze recognition systems as can be seen in Table 1. However only two of these datasets are for gaze tracking free camera setups. These are the GazeFollow dataset [10] and the VideoCoAtt dataset [11]. The TVHI dataset [39] does not have the correct labels for gaze point recognition. The VideoGaze dataset [40] annotates the neighboring frames for gaze predictions across frames. The EGTEA Gaze+ dataset [41] and the Gaze-in-wild dataset [42] are annotated for head-mounted cameras.

Dataset	Year	Format	Size	Annotation	Goal	Data Source
TVHI [39]	2012	Video	300 video clips, 30 to 600 frames per clip	Upper body bbx, discrete head orientations, interaction label	Human interaction learning in TV show	23 different TV shows
GazeFollow [10]	2015	Image	122.143 images, 130.339 people	Eye loc., head bbx and gaze loc.	Gaze following in images	Actions 40, MS COCO, SUN, PASCAL, etc.
VideoGaze [40]	2017	Video	140 movies, 6 frames per movie	Eye loc., head bbx and gaze loc.	Gaze following in videos	MovieQA
EGTEA Gaze+ [41]	2017	Video	86 unique sessions, 32 subjects.	frame-level action annots., pixel-level hand masks	ego-centric activity recognition	meal-preparations with SMI eye-tracking glasses
VideoCoAtt [11]	2018	Video	380 videos, 492.100 frames	Shared attention bbx, involved head bbx	Shared attention detection in videos	20 different TV shows
Gaze-in-wild [42]	2019	Video	140 minutes, 20.000 fixation events	Eye-In-Head vel., Absolute Head vel. gaze loc.	eye and head coord. in activities	Participants wearing eye tracker and a hardhat with sensors

Table 1: Comparison of several related datasets

2.1.1 GazeFollow

The GazeFollow dataset consists of 122.143 images with 130.339 people resulting in 130.339 rows of data. Of this data 4.782 rows are designated as test data. The rest is designated as training data. For the image scaled to a size of 1x1, the data row contains the position of gaze and of the eyes as a floating-point value. It additionally contains the head bounding box of the person seeing on the same scale as the eyes. In Figure 11 data from the GazeFollow dataset is visualized. The circles are centered on die eye positions, the points are centered on the gaze locations. The lines indicate the gaze from an eye position to a gaze position.

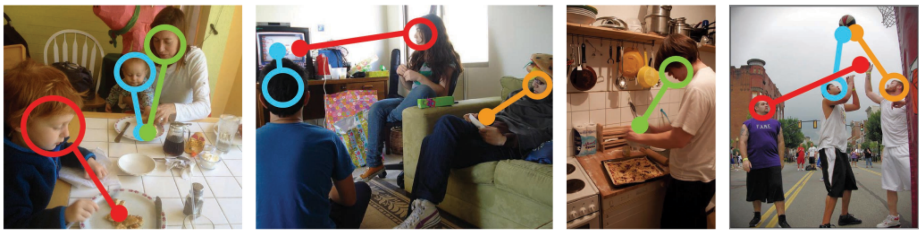

Figure 11: Sample data of the GazeFollow dataset [10]

The data has been built taking the images of several different datasets. These are 1.548 images from SUN [43], 33.790 images from Microsoft COCO [44], 9.135 images from Actions 40 [45], 7.791 images from PASCAL [46], 508 images from the ImageNet detection challenge [47] and 198.097 images from the Places dataset [48]. In these hired workers annotated the ground-truth for gaze tracking. As the result 122.143 images with 130.339 people have been annotated [10]. The GazeFollow dataset is publicly available online at http://gazefollow.csail.mit.edu/.

2.1.2 VideoCoAtt

The VideoCoAtt dataset consists of 492.100 frames taken from 380 different video clips from 20 different TV shows or movies. For each frame there are for every attention bounding box the participants bounding boxes stored inside the dataset. All bounding boxes are stored according to pixel position. In Figure 12 data from the VideoCoAtt dataset is visualized. The red bounding boxes are centers of attention. The differently colored bounding boxes are color coded attention groups.

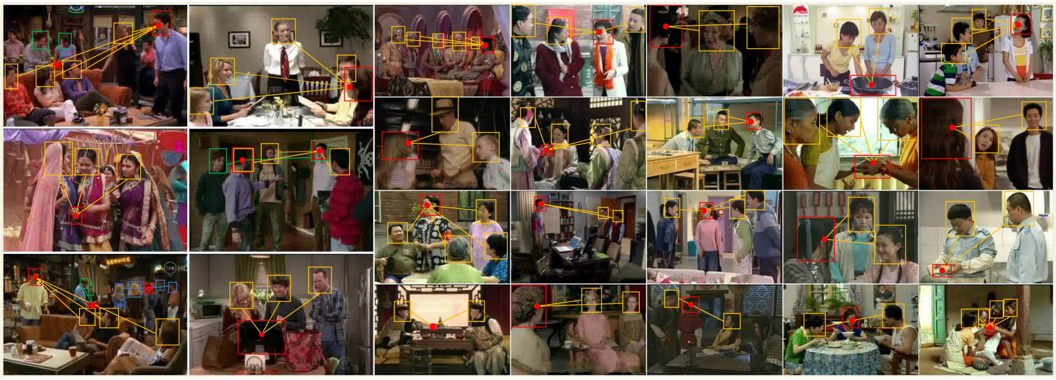

Figure 12: Sample data of the VideoCoAtt dataset [11]

The data was manually annotated with the tool Vatic. The data contains 139.348 frames with one attention bounding box and 3.284 frames with multiple attention bounding boxes. There was a great focus on having a high generality of dataset containing social interactions from great variety of places and cultures. In the dataset 44% of the frames are in an American cultural context, 40% is in a Chinese cultural context and the rest in others. The locations are 30% in a living room, 14% in a kitchen, 7% in a restaurant, 7% in a bedroom and the rest in others [11]. The VideoCoAtt dataset is publicly available online at https://drive.google.com/a/g.ucla.edu/file/d/1Fp79WQjgOxOXlflZGCh2jlPat8cJenzJ/view?usp=sharing.

2.2 Gaze Recognition Systems

2.2.1 GazeFollow

In the development of GazeFollow it was reasoned that people first look at a person's head to estimate the field of view. Then the salient objects in the field of view are being considered. Through combination of these information Recasens et al. argue people reason where other people are looking. This approach was intended to be replicated by a deep learning ANN. In Figure 13 the general structure of the ANN is visualized. It takes three inputs. The full image, the image of a head and the general head location in the full image. With the help of a shifted grid the output information is turned into a gaze heat map at the output. The data is taken from the GazeFollow dataset published alongside with this system.

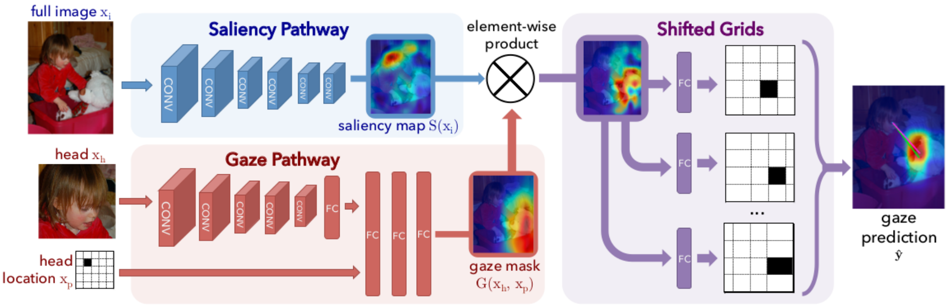

Figure 13: A sketch of the GazeFollow system [10]

Since each pathway of the data cannot solve the full problem of gaze prediction alone, Recasens et al. argue the Saliency Pathway will produce a saliency map while the Gaze Pathway will produce a gaze mask. Then the information of both pathways is multiplied since the object of focus so Recasens et al. is likely to be both inside the field of view and salient. In Figure 14 the result of the Saliency Pathway is visualized imposed on the image. In the b) category the result of the Saliency Pathway is compared to a saliency network. The saliency map is generated by a machine learning based system [49].

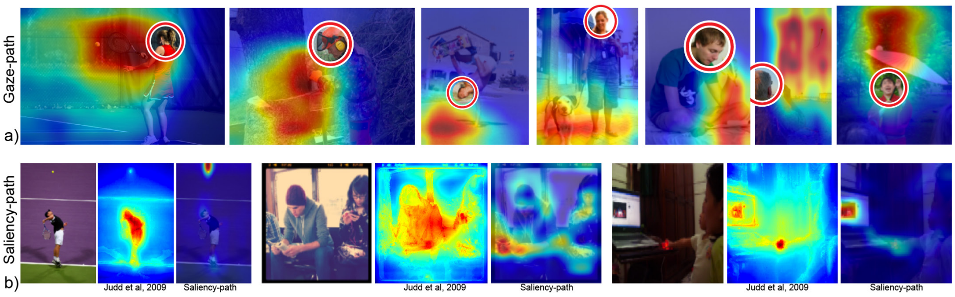

Figure 14: GazeFollow system output Saliency Pathway [10]

To make the classification penalize neighboring cells to the classification target less than cells further away. Recasens et al. proposed the use of a shifted grid. This turns the classification problem into one of overlapping classifications. The convolutional layers of the saliency pathway were initialized with the Places-CNN [48] and the gaze pathway with ImageNet-CNN [50]. This ImageNet-CNN is built according the AlexNet structure [50].

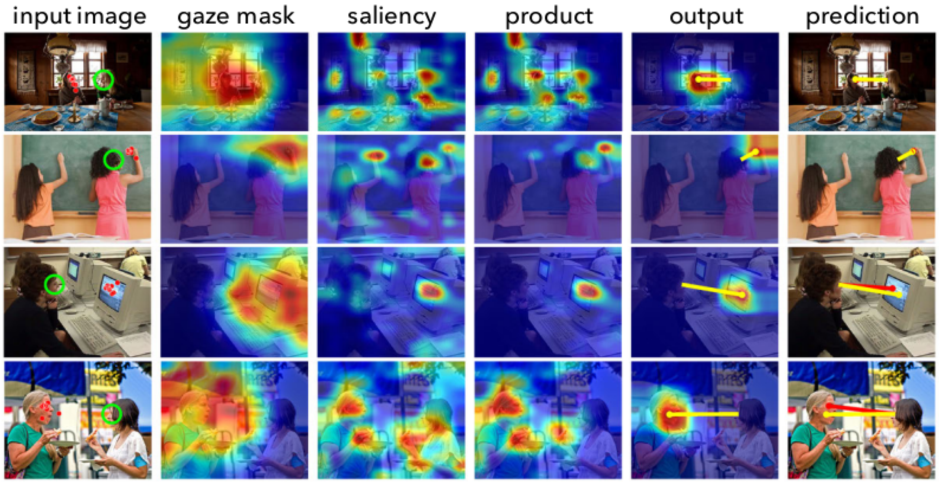

Figure 15: Gazefollow system prediction output and internal states [10]

The GazeFollow system achieved an Area Under Curve (AUC) of 87,8% with a mean angular error of 24° degrees and mean distance to the target location of 19,0% of the image side length [10]. In comparison the saliency map generated by the system of Judd et al. achieves according to Recasens et al. 71,1% accuracy, 54° mean angular error and a mean distance to the target location 33,7% of the image side length. A human so Recasens et al. achieves 92,4% AUC with a mean angular error of 11° and a mean distance to the target location 4,0% of the image side length. A random placement achieves according to Recasens et al. 50,4% AUC with a mean angular error of 69° and a mean distance to the target location 48,4% of the image side length. In Table 3 the results can be seen in tabular form compared to all the others. In Figure 15 the output of the GazeFollow system is visualized. The red line is the ground-truth of the image while the yellow line is the most likely prediction. Further the outputs of the pathways are visualized as well [10]. This system was developed in Caffe and the system can be publicly accessed online at http://gazefollow.csail.mit.edu/.

2.2.2 GazeFollowing

The GazeFollowing system by Lian et al. builds on the GazeFollow system by Recasens et al. aiming to improve the accuracy of the system. Lian et al. reason a two-stage solution of Gaze tracking would be more effective since in human gaze tracking an estimation for the gaze direction is first made and then the gaze point is inferred from both the estimation and the scene. In Figure 16 the structure of the system is visualized. The upper half of the structure is the first stage where the gaze direction estimation is made. The lower half of the structure is the second stage where the gaze prediction is made.

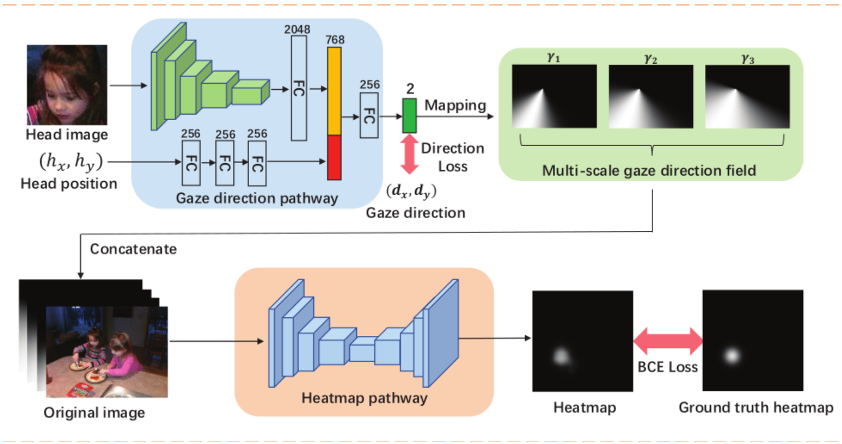

Figure 16: A sketch of the GazeFollowing net [12]

The first stage of the system proposed by Lian et al. generates the gaze direction field. The system produces a probability mapping dependent on the angle difference to a machine learning based estimation. This is done at three different intensities resulting in narrower and wider arcs. In Figure 16 the arcs for three different intensities γ are shown. The intensities $\gamma_1 = 5$, $\gamma_2 = 3$ and $\gamma_3 = 1$. The result is concatenated with the input image for the second stage.

In the second stage the system produces with a feature pyramid network a heat map for gaze prediction. The last layer uses a sigmoid function to guarantee a probability output between 0 and 1 for each target pixel. A heat map prediction was chosen by Lian et al. due to a higher robustness of the output.

This system was trained using the PyTorch framework. The convolutional layers to decode the head are built like the ResNet-50 and initialized with the model pretrained with ImageNet [51]. The training was performed using the GazeFollow dataset with the same test/training data split as the GazeFollow system. The resulting network achieved an Area Under Curve (AUC) of 90,6%, 14,5% mean distance to the target location and 17,6% mean angular error. In Table 3 the results can be seen in tabular form compared to all the others.

2.2.3 Inferring Shared Attention

The Inferring Shared Attention detection system proposed by Fan et al. tries to solve a similar problem to gaze tracking. The system tries to identify whether a group of people share attention or not. For this the proposed system uses four modules as shown in Figure 17.

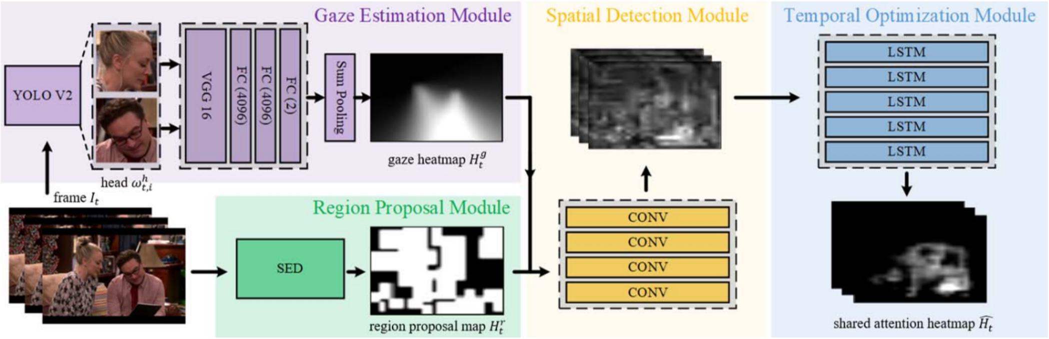

Figure 17: A sketch of the Inferring Shared Attention detection system [11]

The modules are the Gaze Estimation Module, the Region Proposal Module, the Spatial Detection Module and the Temporal Optimization Module. The Gaze Estimation Module produces a probability mapping of gaze for each person which is combined through sum pooling to a single gaze map. The Region Proposal Module utilizes a Structured Edge Detector (SED) to find bounding boxes inside the image. All regions enclosed by bounding boxes are assigned the binary value 1, while all others are assigned the value 0. The Spatial Detection Module combines the produced gaze map and region map to a frame based spatial map. The Temporal Optimization Module utilizes a Long Short-Term Memory (convLSTM) to consider previous frames for the final evaluation of shared attention. In Figure 18 the combination of the gaze heat maps is visualized.

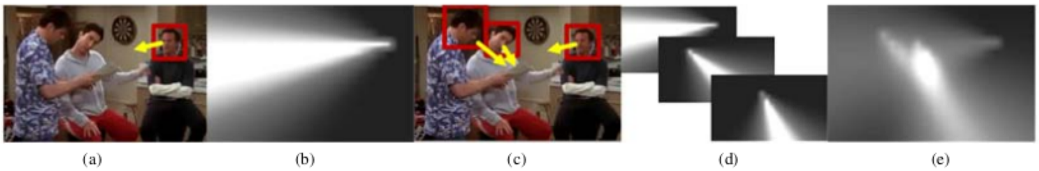

Figure 18: Inferring Shared Attention gaze heat map combination [11]

In Figure 19 the output of the various modules is visualized. In the last column the ground truth is shown in green. The prediction is shown in red. The green arrows start at the person partaking in the shared attention and end at the ground truth.

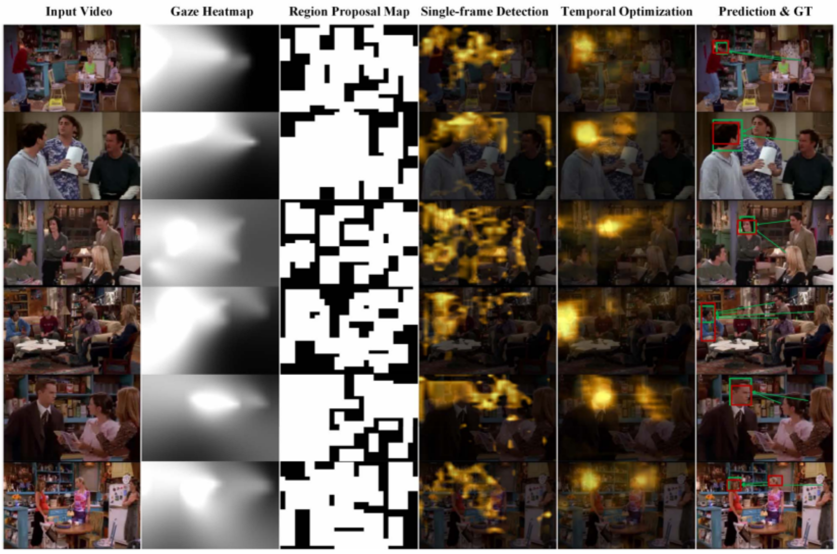

Figure 19: Inferring Shared Attention processing [11]

The system uses the VideoCoAtt dataset published alongside the system. The trained models were however not made publicly available. However, Fan et al. state the head detector is a fine tuned YOLOv2 Darknet [52]. For gaze direction network the VGG16 with the last layer replaced output layer was used. The 1000 neuron wide fully connected layer was replaced with a 2 neuron wide fully connected layer with a tanh output function. The gaze cone is created by assuming a gaussian standard distribution with a standard deviation of $\sigma = 0,5$. The SED was done using the Structured Edge Detection Toolbox [53]. The system was trained in the Keras framework with Tensorflow.

A prediction is considered accurate when the correct attention bounding box was found. The L^2 distance is the Euclidean distance between the predicted bounding box and the ground truth as measured in pixels. In Table 2 the results of Fan et al. are displayed.

Model	Prediction Acc.	L^2 Dist.
Raw Img. [11]	52,3 %	188
Only Gaze [11]	64,0 %	108
Only RP [11]	58,0 %	110
Gaze+RP [11]	68,5 %	74
Gaze+RP+Img. [11]	54,0 %	72
Fixed Bias [11]	52,4 %	122
Random [11]	50,8 %	286
Gaze Follow [10]	58,7 %	102
Gaze+Saliency [54]	59,4 %	83
Gaze+Saliency [54] + LSTM	66,2 %	71
Fan et al. (Gaze+RP+LSTM) [11]	71,4 %	62

Table 2: Results of Fan et al.

3 The Proposed System

3.1 Training and test data

For the creation of the proposed system the GazeFollow dataset was used. It was chosen since many other systems were built with and compared to the dataset. Furthermore, the dataset is available in an easy to use format. The dataset has a decent size of 130.339 rows of data making overfitting less likely. The 130.339 rows contain 4.782 rows for testing.
A machine learning system can only be as good as training data it was provided with. The better the training data reflects the possible situations the better. Therefore, more data tends to improve convergence towards the application of the system. However, the amount of recorded training data is always limited and hence does not contain every possible situation. Because of this data augmentation is commonly employed to increase the amount of available data. The amount of augmented data can help only to a certain amount since it is still based on the original data. For image augmentation several methods are commonly used. Popular methods are shifting and cropping, rotations and flipping as well as color space modifications.

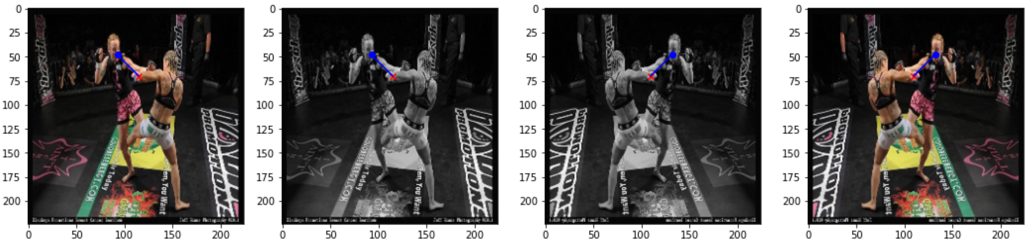

Figure 20: All data augmentations on one image

For the GazeFollow dataset only some augmentations are sensible. Two augmentations to increase the training data were chosen. The transformation of the dataset into grey scale data and the flipping of the data by the y-axis. The flipping by the y-axis is possible since the direction of gaze would just flip accordingly without any further difficulty for a human trying to assess the target of the gaze. The conversion into grey scale is possible since the gaze prediction of a human is mostly based on shapes and contours. Therefore, it is mostly independent by the presence of color. Both of these transformations can be used simultaneously. The augmentations therefore turn the 125.557 rows of training data into 502.228 rows of data. The test data was not augmented. In Figure 20 one training data element in all its forms after augmentation is shown. The blue circle marks the position of the eyes, while the red cross marks the target of the gaze. They are connected with a blue line.

3.2 Prototypes

3.2.1 Two-path network without head position

The prototype is built in Caffe with a two-path system in mind. Similar to Recasens et al. the first path is intended to be the saliency pathway and the second path is the gaze pathway. Different than Recasens et al. was however, the concatenation of the pathways rather than the multiplication for the result. This was done to allow the system to assign importance to the output of the two pathways. Lian et al. showed the concatenation of the gaze maps with the image had better results than the multiplication thereof. Further the head position is not given to the system. It was presumed the system would be able to find out where the head image is placed. Further the output map as proposed by Lian et al. is chosen to be the output datatype. This is done since it seems to be a convenient and robust solution. In Figure 22 the network is visualized.

The training of the system was performed with several different values for the hyperparameter. The system did however not converge. The system is running towards a fixed input independent assertion of the position of gaze. In Figure 21 two training outputs are visualized. Each of the output contains a matrix of three rows with four columns. Each row in the figure is one data row. The first column is the full image. The second column is the head image. The third column is the ground truth. The fourth column is the system output.

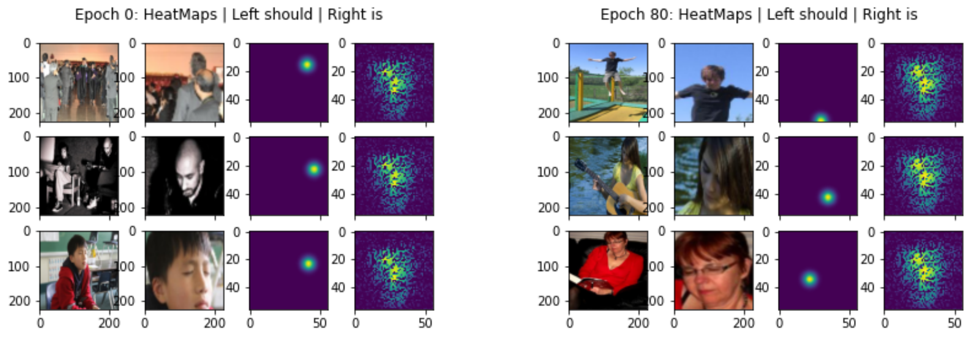

Figure 21: Training heat maps of prototype 1

It was concluded the system is incapable of getting a decent result of the gaze path way, since the difference mostly existed there. Therefore, the gaze pathway had to be reimagined. It was concluded that for the gaze prediction environmental aspects are relevant. They cause occlusions rendering lines of view spatially impossible, since light can freely through all materials. This line of reasoning lead to prototype nr. 2, see Chapter 3.2.2.

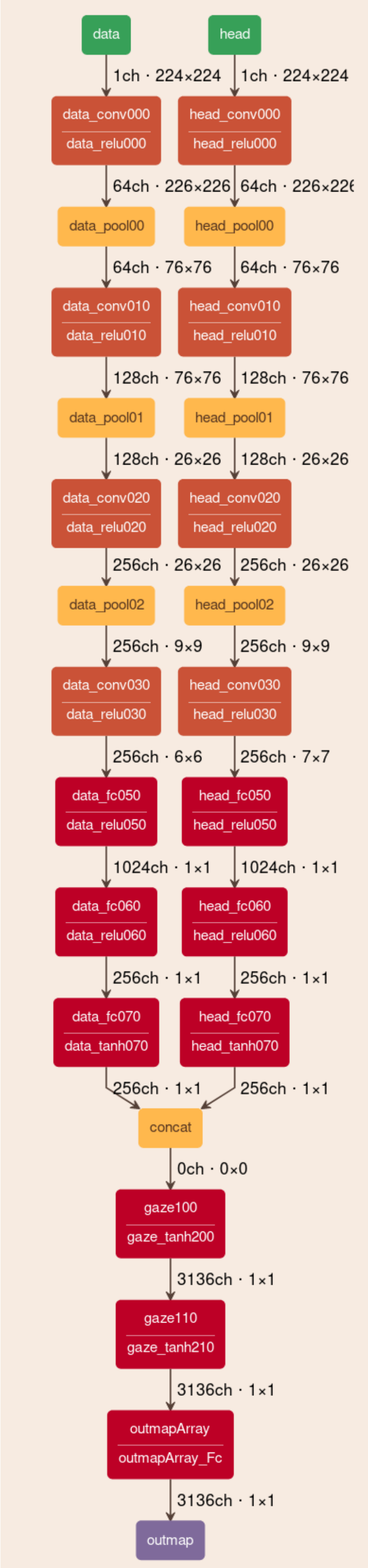

Figure 22: The network architecture of prototype 1

3.2.2 Three-path network without head position

The second prototype is built in Caffe considering the failing of the first prototype. Like the first prototype there are two major pathways. However, to allow the gaze path way to deal with spatial conditions the gaze pathway is now made of two minor pathways. The first minor gaze pathway utilizes the head position of the person. The second minor gaze pathway uses the full image. The gaze pathway is supposed to conclude the gaze considering the spatial properties of the scene. The saliency pathway is intended to provide the saliency map. The second minor gaze pathway and the saliency pathway use different convolution systems, since the required information to estimate occlusions and to estimate saliency are different. Like the first prototype the system is planned with an output map like the first prototype. In Figure 24 the network is visualized.

The training of the system was performed with several different values for the hyperparameter. The system did however not converge. The system is running towards a fixed input independent assertion of the position of gaze. In Figure 23 two training outputs are visualized. Each of the output contains a matrix of three rows with four columns. Each row in the figure is one data row. The first column is the full image. The second column is the head image. The third column is the ground truth. The fourth column is the system output.

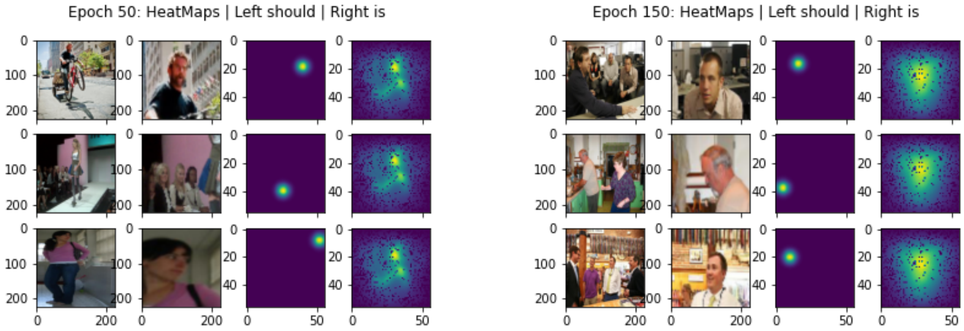

Figure 23: Training heat maps of prototype 2

It was therefore concluded the position of the head in the image is a necessary information for the system, since both Recasens et al. and Lian et al. have it as an input in one form or another. Recasens et al. give the head position as a 13x13 grid into the network, see Chapter 2.2.1. Lian et al. on the other hand gives the network the information as an axis wise normalized 2D vector, see Chapter 2.2.2.

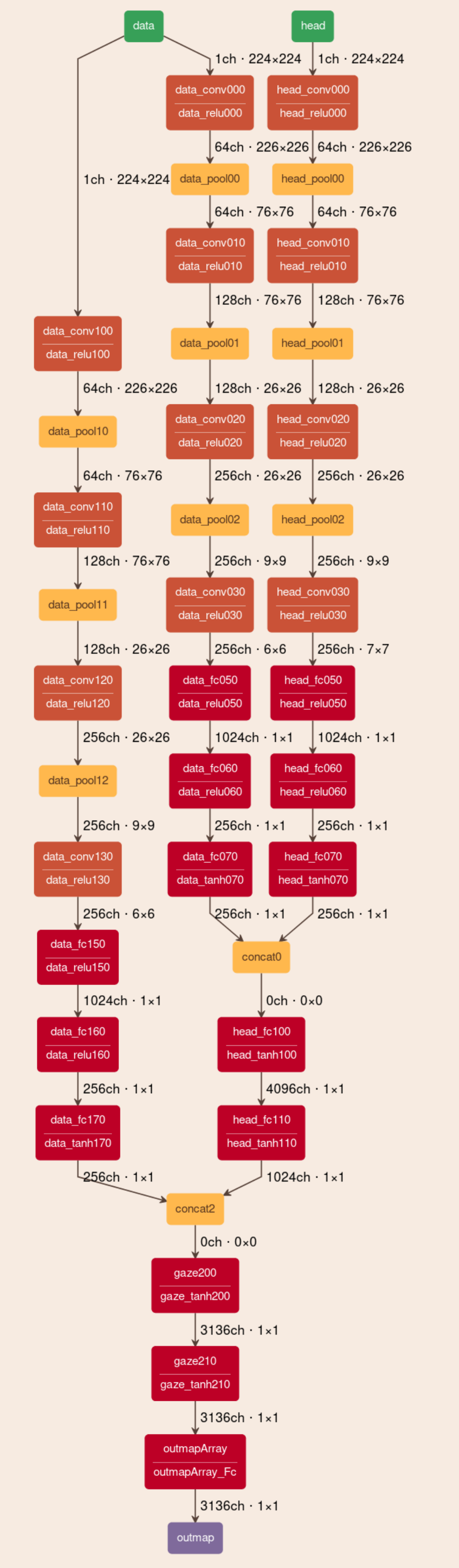

Figure 24: The network architecture of prototype 2

3.3 Final System

The final system is built in Caffe improving on the second design. There is the same pathway structure as in the second design. However, the gaze direction pathway now gets the location of the head as an additional input. This is done as proposed by Lian et al. as an axis wise normalized 2D vector. The saliency pathway uses the image of the head to build a saliency map for the image. The gaze pathway now has two pathways and a vectorized input. The first minor gaze pathway utilizes the head position of the person. The second minor gaze pathway uses the full image. In Figure 25 two training outputs are visualized. Each of the output contains a matrix of three rows with four columns. Each row in the figure is one data row. The first column is the full image. The second column is the head image. The third column is the ground truth. The fourth column is the system output.

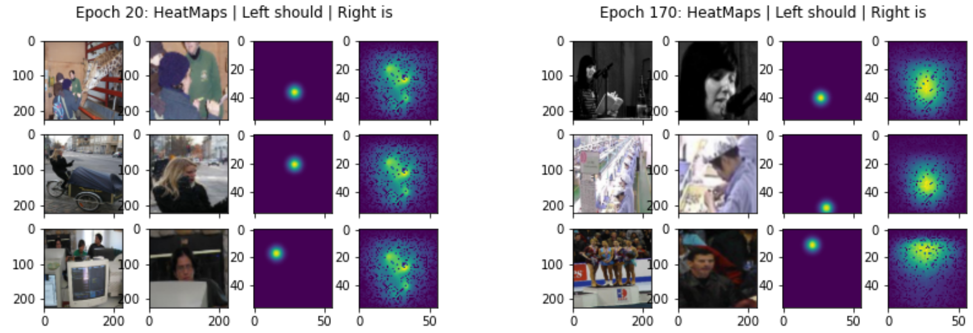

Figure 25: Training heat maps of the final system

The system structure showed more promising results than the previous prototypes. Having an input dependent peak on the target heat map. However, it still had problems to converge. For this purpose, a second loss function was introduced. As can be seen in Figure 26 the system has an additional secondary output, which is trained with a normalized direction vector. A direction vector was chosen since an angle would inherently add the nonlinear jump from an angle close to a full circle of almost 2π to an angle 0. A vector on the unit circle on the other hand is completely continuous without any sudden jumps making it easier to train. Unfortunately, this did not resolve the convergence problem either.

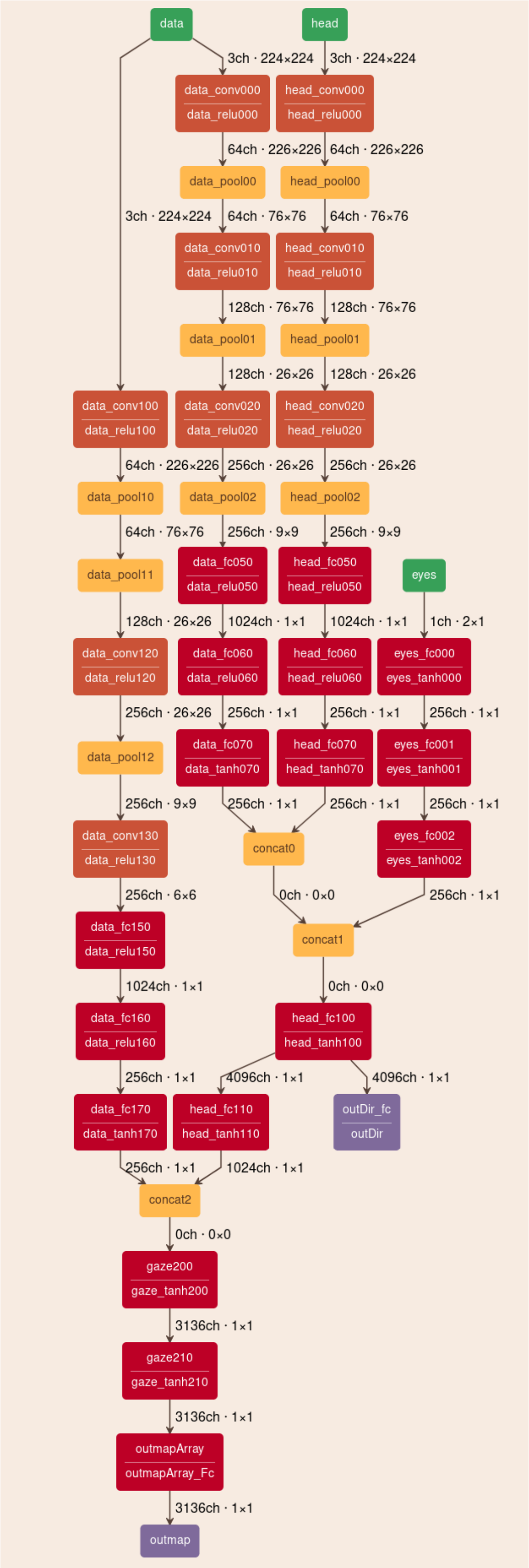

Figure 26: The network architecture of the final system

4 Discussion

4.1 System comparison

4.1.1 Evaluation metrics

AUC

The area under Receiver Operating Characteristic (ROC) curve, which is generated according to Judd et al. The output heat map is treated as a binary classifier were the threshold is varied. The percentage of correctly classified pixel therefore produces a curve. The AUC always lies between 0,5 and 1,0 since an AUC of less than 0,5 could be viewed with inverted interpretation [49].

Dist

The mean Euclidean distance between predicted gaze points and the main corresponding ground truth annotation. The image size is normalized to 1x1.

MDist

The mean minimum Euclidean distance between predicted gaze points and all corresponding ground truth annotations. The image size is normalized to 1x1.

Ang

The mean angular error between predicted gaze directions and the corresponding direction according to the main corresponding ground truth annotation.

MAng

The mean minimum angular error between predicted gaze directions and all corresponding directions according to the main corresponding ground truth annotations.

4.1.2 Evaluation

The analysis of gaze from images is a complex problem. Hence the creation of a system which can effectively predict it is not an easy feat. In this thesis no functional system could be created. There are however a number of different proposed systems. In Table 3 the qualitative performance of these various systems is listed. Recasens et al.* is the system by Recasens et al. with the head detector used by Lian et al. This was done to make their systems more comparable.

Methods	Year	AUC	Dist	MDist	Ang	MAng
Random [10]	2015	0,504	0,484	0,391	69,0°	-
Center [10]	2015	0,633	0,313	0,230	49,0°	-
Fixed bias [10]	2015	0,674	0,306	0,219	48,0°	-
SVM + one grid [10]	2015	0,758	0,276	0,193	43,0°	-
SVM + shift grid [10]	2015	0,788	0,268	0,186	40,0°	-
Judd et al. [49]	2009	0,711	0,337	0,250	54,0°	-
SalGAN [55]	2017	0,848	0,238	0,192	36,7°	22,4°
SalGAN for heatmap [12]	2019	0,890	0,181	0,107	19,6°	9,9°
Recasens et al. [10]	2015	0,878	0,190	0,113	24,0°	-
Recasens et al.* [10, 12]	2019	0,881	0,175	0,101	22,5°	11,6°
Lian et al. (one-scale) [12]	2019	0,903	0,156	0,088	18,2°	9,2°
Lian et al. (multi-scale) [12]	2019	0,906	0,145	0,081	17,6°	8,8°
One human [10]	2015	0,924	0,096	0,040	11,0°	-

Table 3: Gaze tracking accuracy comparison

As can be seen in the table the qualitative performance of the systems is improving over time. In 2015 the best system as proposed by Recasens et al. achieved an AUC of 87,8% with a mean angular error of 24,0°. In 2019 the AUC was improved by the new system by Lian et al. to an AUC of 90,6% with a mean angular error of 17,6°. These systems are yet not as good as the human capability of gaze recognition. The testee in the setup by Recasens et al. achieved an AUC of 92,4% with a mean angular error of 11,0°.

There are however not only qualitative performance characteristics, but also resource performance characteristics. In Table 4 the resource requirements of the systems by Lian et al. and Recasens et al. are compared. The chosen resources for the comparison are time in milliseconds and GPU Memory in megabytes. This test was done on a computer in a windows 10 environment. The GPU is a GTX 1080Ti and the CPU is a i7-3770. The system by Lian et al. was run as published in the PyTorch framework. The system by Recasens et al. was run as published in the Caffe framework. The measured time is the average runtime of a frame on the GazeFollow test data. The measured time contains the preprocessing and postprocessing of the raw data but not the load time.

Methods	GPU Memory Usage (MB)	Time (ms)
Recasens et al. [10]	520	15,6
Lian et al. [12]	1.450	24,2

Table 4: Gaze tracking resource usage comparison

Lian et al. compared the time needed to run their system compared to SalGAN and the system proposed by Recasens et al. Both systems took roughly longer for a frame compared to the test of Lian et al. Lian et al. tested with a NVIDIA Titan X GPU which has less computational power than a GTX 1080Ti [56]. This might be through the inclusion of the preprocessing and postprocessing necessary for the execution of the systems. The system of Recasens et al. takes roughly 4,3 ms without preprocessing and postprocessing. Roughly half of the value of 10,4 ms as measured by Lian et al. and in line with the increased computational capacity. The two-stage solution of Lian et al. makes it difficult to distinguish between preprocessing and execution since the first stage could be argued to be preprocessing. It could however be argued the preprocessing and postprocessing should be part of the total execution time, since they are necessary for deployment.

Both, Lian et al. and Recasens et al. initialized their input convolution layers with the parameters of dedicated image analyses systems. In the approach tried here, training was tried from scratch. The finding of fitting hyperparameter for learning from scratch is a difficult to undertaking. Here no fitting hyperparameter could be found.

4.2 Future scope

The structure of the analyzed systems is similar to the already existent systems by Recasens et al. and Lian et al. Since no fitting hyperparameter for a full training from scratch could found a different approach has to be taken. One approach would be initializing the input layers with those of a dedicated image analysis system like ResNet-50 with the pretrained model of ImageNet. Another possibility would be training in multiple phases. When training in multiple phases the saliency pathway and gaze pathway would first have to be trained separately to create intermediate results. After they converge the system as a whole would be trained till it converges. Both of these approaches might yield a working system.

Despite colors having significance for gaze tracking, the gaze tracking is mostly dependent on alignment of head and eyes. Therefore, it could be worth investigating how a system, which is not merely assisted by training with grey scale data, but processes grey scale images would perform. It is likely going to have reduced qualitative performance, but is likely going to have less resource consumption when deployed. This topic is completely unexplored.

When building systems for gaze tracking depth information would be useful. A pixel wise large difference close to the point of view of the camera is less significant in metric distance to a point far away from the camera would be in metric distance. Further a 3-dimensional understanding of the environment in the image would allow a system to be able to judge occlusions better. This would likely improve the qualitative performance.

When a running system is developed it is likely to be still lacking considerably in accuracy when compared to methods like EOG and scleral search coil. It is possible the accuracy of these systems cannot be reached with this method. In accuracy it is probably not going to catch up to current table-mounted and head-mounted cameras for quite some time. It is however clear this method can still be improved since humans can achieve better results [10] than any system thus far developed. The problem will be to find out how a system like this will look like. It should further be noted, given the current datasets an improving beyond human capability is impossible since all available datasets are annotated by humans looking at the images. A dataset created by more reliable means like the person seen in the image annotating where they did look. Another way of creating a dataset would be building it based on a more reliable gate tracking method like EOG or the scleral search coil.

Once the accuracy of these systems will be sufficiently reliable, it will be easier for machines to analyze human focus. This knowledge would allow a machine to judge a human's intent with a higher degree of certainty. This can be used in medical application for diagnosis of ocular and psychological health. Further it can be used in criminology to analyze motive of human action.

5 Summary and Conclusions

In this thesis the development of a new machine learning based approach for the recognition of gaze direction was tried. It was however found out a machine learning based approach with learning from scratch is a futile path. Three different prototypes, where the subsequent prototype increased in complexity to the one prior, all failed to converge. It is therefore a better approach to build a system relying on prior developed image analyses systems. Due to this method no working machine learning based system was realized.

At the time of writing the best system for gaze tracking is the system by Lian et al. It utilizes a two-stage process for tracking human gaze. The first makes an estimate building cones of gaze of varying intensity. From this the system generates a prediction with a mean angular error of 17,6°. In Chapter 4.1 the proposed systems are compared in greater detail. The method of remote gaze tracking utilizing a free camera setup is less reliable than other methods in accuracy. It is however far less limited in application once a decent accuracy can be achieved. At this point the difference between these methods is a difference of more than 10,0° in mean angular error.

Given the current accuracy of gaze tracking there seems to be potential for further improvement. A human analyzing an image achieves a mean angular error of 11,0°. An AI achieves at this point a mean angular error of 17,6°. This is a difference of 6,6° given the same input data. It is likely at least this difference can be overcome by better systems. It is however to be considered all current datasets are assigned by humans. Therefore, achieving a higher given the current datasets is meaningless since the error in the dataset should be comparable.

References

[1] M. F. Bear, B. W. Conners, and M. A. Paradiso, *Neuroscience: Exploring the Brain*. Baltimore: Williams & Wilkins, 1996.

[2] Andrew T. Duchowski, *Eye Tracking Methodology: Theory and Practice*. Springer Publishing Company, Incorporated, 2017.

[3] T. Imai, K. Sekine, K. Hattori, N. Takeda, I. Koizuka, K. Nakamae, K. Miura, H. Fujioka, and T. Kubo, "Comparing the accuracy of video-oculography and the scleral search coil system in human eye movement analysis," *Auris Nasus Larynx*, vol. 32, no. 1, pp. 3 – 9, 2005.

[4] A. Larson, J. Herrera, K. George, and A. Matthews, "Electrooculography based electronic communication device for individuals with ALS," in *2017 IEEE Sensors Applications Symposium (SAS)*, pp. 1–5, 2017.

[5] Tobii Technology, "Just One Look: Tobii EyeAsteroids™ Lets Gamers Save the World With a Glance, One Asteroid at a Time," November 2011.

[6] Ergoneers, "Dikablis-Professional-Flyer," December 2015.

[7] M. Wibrow, "Drawing neural network with tikz," January 2014. Available at https://tex.stackexchange.com/questions/153957/drawing-neural-network-with-tikz.

[8] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," December 2018. Available at https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[9] Rafael Alencar, "Dealing with very small datasets," 2020.

[10] A. Recasens*, A. Khosla*, C. Vondrick, and A. Torralba, "Where are they looking?," in *Advances in Neural Information Processing Systems (NIPS)*, 2015. * indicates equal contribution.

[11] L. Fan*, Y. Chen*, P. Wei, W. Wang, and S.-C. Zhu, "Inferring shared attention in social scene videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. * indicates equal contribution.

[12] D. Lian*, Z. Yu*, and S. Gao, "Believe it or not, we know what you are looking at!," 2019. * indicates equal contribution.

[13] M. Roddy and N. Harte, "Detecting conversational gaze aversion using unsupervised learning," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 76–80, 2017.

[14] Z. Ye, Y. Li, Y. Liu, C. Bridges, A. Rozga, and J. M. Rehg, "Detecting bids for eye contact using a wearable camera," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, pp. 1–8, 2015.

[15] F. Shic, B. Scassellati, D. Lin, and K. Chawarska, "Measuring context: The gaze patterns of children with autism evaluated from the bottom-up," in *2007 IEEE 6th International Conference on Development and Learning*, pp. 70–75, 2007.

[16] N. J. Wade, "Pioneers of eye movement research," *Iperception*, vol. 1, pp. 33–68, November 2010.

[17] D. A. Robinson, "The oculomotor control system: A review," *Proceedings of the IEEE*, vol. 56, no. 6, pp. 1032–1049, 1968.

[18] A. E. Kaufman, A. Bandopadhay, and B. D. Shaviv, "An eye tracking computer user interface," in *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*, pp. 120–121, 1993.

[19] Y. Lu and Y. Huang, "A method of personal computer operation using Electrooculography signal," in *2019 IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, pp. 76–78, 2019.

[20] D. A. Robinson, "A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field," *IEEE Transactions on Bio-medical Electronics*, vol. 10, no. 4, pp. 137–145, 1963.

[21] Juergen Werner, *Comprehensive Biomedical Physics*, vol. 5. Elsevier B.V., 2014.

[22] P. J. Murphy, A. L. Duncan, A. J. Glennie, and P. C. Knox, "The effect of scleral search coil lens wear on the eye," *British Journal of Ophthalmology*, vol. 85, no. 3, pp. 332–335, 2001.

[23] Raymond Dodge and Thomas Sparks Cline, "The Angle Velocity of Eye Movements," *Psychological Review*, vol. 8, no. 2, pp. 145–157, 1901.

[24] SR Research, "EyeLink 1000 Plus - The Most Flexible Eye Tracker - SR Research," 2020.

[25] M. Cognolato, M. Atzori, and H. Müller, "Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 5, 06 2018.

[26] Ergoneers, "Ergoneers FAQ-Database - Is it possible to turn off the Dikablis camera(s) IR?," 2020.

[27] Tobii Technology, "How do Tobii Eye Trackers work?," 2020.

[28] J. Steil, *Mobile eye tracking for everyone.* PhD thesis, Universität des Saarlandes, November 2019.

[29] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression," in *2011 International Conference on Computer Vision*, pp. 153–160, 2011.

[30] A. Kar and P. Corcoran, "A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," 2014. Available at https://dl.acm.org/doi/abs/10.1145/2647868.2654889.

[32] Y. Jia, "Caffe — Installation," 2014. Available at http://caffe.berkeleyvision.org/installation.html.

[33] Mathworks, *Supported and Compatible Compilers — Release 2020a*, 2020.

[34] F. Chollet *et al.*, "Keras." https://github.com/fchollet/keras, 2015.

[35] Keras Team, "Keras documentation: Why choose Keras?," 2020.

[36] M. Görner, "Keras and modern convnets, on TPUs," 2020.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8026–8037, Curran Associates, Inc., 2019.

[39] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, "Structured Learning of Human Interactions in TV Shows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, p. 24412453, Dec. 2012.

[40] A. Recasens, C. Vondrick, A. Khosla, and A. Torralba, "Following Gaze in Video," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1444–1452, 2017.

[41] Y. Li, M. Liu, and J. M. Rehg, "In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[42] R. Kothari, Z. Yang, C. Kanan, R. Bailey, J. Pelz, and G. Diaz, "Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities," *arXiv preprint arXiv:1905.13146*, 2019.

[43] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

[44] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[45] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human Action Recognition by Learning Bases of Action Attributes and Parts," in *Proceedings of the 2011 International Conference on Computer Vision*, ICCV 11, (USA), p. 13311338, IEEE Computer Society, 2011.

[46] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, p. 303338, June 2010.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vision*, vol. 115, p. 211252, Dec. 2015.

[48] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 487–495, Curran Associates, Inc., 2014.

[49] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 2106–2113.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[51] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[52] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.

[53] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 391–405, Springer International Publishing, 2014.

[54] J. Pan, E. Sayrol, X. Giró-i-Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and Deep Convolutional Networks for Saliency Prediction," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[55] J. Pan, C. Canton-Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giró-i-Nieto, "SalGAN: Visual Saliency Prediction with Generative Adversarial Networks," *CoRR*, vol. abs/1701.01081, 2017.

[56] userbenchmark.com, "UserBenchmark: Nvidia GTX 1080-Ti vs Titan X," 2020.