Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-101018-96962

Adam Melicher

Prediction of human attention behaviour for unseen scenes

Bachelor's thesis

Supervisor: Ing. Miroslav Laco

May 2022

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-101018-96962

Adam Melicher

Prediction of human attention behaviour for unseen scenes

Bachelor's thesis

Degree course: Information Security

Field of study: Computer Science

Training workplace: Institute of Informatics and Software Engineering, FIIT STU, Bratislava

Supervisor: Ing. Miroslav Laco

May 2022

Annotation

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Information Security

Author: Adam Melicher
Bachelor's Thesis: Prediction of human attention behaviour for unseen scenes
Supervisor: Ing. Miroslav Laco
May 2022

The goal of visual attention modeling is to mimic the human brain's complex psychological functioning and to find the regions or objects that attract human visual attention. This topic has grown in interest in both academia and industry in recent decades, owing to advances in the fields of computer vision and neural networks. Many present approaches to saliency prediction are general and objective, and they ignore the observer. As a result, the attention prediction for a specific person may be incorrect. This is where personalized saliency prediction comes in, when models take into account an individual's personality features. However, only few datasets dedicated to personalized attention modelling exists, because it is very exhausting for observers to view such large amounts of images. Hence the aim of this work is to analyze artificial fixation maps generated by personalized models trained on existing datasets and evaluate them against unseen data.

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Information Security

Autor: Adam Melicher
Bakalárska práca: Simulácia správania sa pozornosti človeka pri nikdy nevidených scénach
Vedúci bakalárskej práce: Ing. Miroslav Laco
Máj 2022

Cieľom modelovania vizuálnej pozornosti je napodobniť komplexné psychologické fungovanie ľudského mozgu a nájsť oblasti alebo objekty, ktoré priťahujú ľudskú vizuálnu pozornosť. Záujem o túto tému v akademickej i priemyselnej sfére vzrástol v posledných desaťročiach vďaka pokroku v oblasti počítačového videnia a neurónových sietí. Mnohé súčasné prístupy k predikcii pozornosti sú všeobecné a objektívne a ignorujú pozorovateľa. V dôsledku toho môže byť predpoveď pozornosti pre konkrétnu osobu nesprávna. Tu prichádza na scénu personalizovaná predikcia pozornosti, keď modely zohľadňujú osobnostné črty jednotlivca. Existuje však len malé množstvo datasetov venovaných modelovaniu personalizovanej pozornosti, pretože pre pozorovateľov je veľmi vyčerpávajúce prezerať si také veľké množstvá obrázkov. Cieľom tejto práce je teda analyzovať umelo vytvorené fixačné mapy generované personalizovanými modelmi trénovanými na existujúcich datasetoch a vyhodnotiť ich oproti nevideným údajom.

Slovenská technická univerzita v Bratislave
Ústav počítačového inžinierstva a aplikovanej informatiky

Fakulta informatiky a informačných technológií
Akademický rok: 2021/2022
Evidenčné číslo: FIIT-101018-96962

ZADANIE BAKALÁRSKEJ PRÁCE

Študent:	**Adam Melicher**
ID študenta:	96962
Študijný program:	informačná bezpečnosť (konverzný)
Študijný odbor:	informatika
Vedúci práce:	Ing. Miroslav Laco
Vedúci pracoviska:	Ing. Katarína Jelemenská, PhD.

Názov práce: **Simulácia správania sa pozornosti človeka pri nikdy nevidených scénach**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Skúmanie a modelovanie vizuálnej pozornosti človeka je dôležitou výskumnou oblasťou vzhľadom k mnohým aplikačným doménam. V súčasnosti už vieme tvoriť tzv. personalizované modely pozornosti, inak povedané odtlačky vizuálnej pozornosti jednotlivca, s využitím umelej inteligencie, hlbokého učenia a dát z experimentov so sledovaním pohľadu. Takéto modely vieme následne použiť pre simuláciu správania sa pozornosti jednotlivca pri scénach, ktoré v skutočnosti nikdy nevidel. Takéto využitie simulácie pozornosti konkrétneho človeka pre generovanie dát o ľudskej pozornosti otvára široké možnosti použitia v doménach ako modelovanie používateľa pri interakcii človeka s počítačom, či medicínska diagnostika. Analyzujte existujúce prístupy v modelovaní vizuálnej pozornosti so zameraním na personalizované modely pozornosti. Na základe analýzy zvoľte a implementujte existujúce riešenie pre tvorbu personalizovaných modelov pozornosti s využitím dostupných datasetov z experimentov so sledovaním pohľadu. Takto vytvorenými modelmi simulujte správanie sa pozornosti človeka na nikdy nevidených obrazových dátach. Vaše riešenie vyhodnoťte a diskutujte s ohľadom na mieru istoty pri simulovaných dátach.

Rozsah práce: 40

Termín odovzdania bakalárskej práce:	16. 05. 2022
Dátum schválenia zadania bakalárskej práce:	23. 11. 2021
Zadanie bakalárskej práce schválil:	doc. Dr. Ing. Michal Ries – garant študijného programu

Declaration of honour

I honestly declare that I have written this thesis independently under the supervision of Ing. Miroslav Laco with the cited bibliography.

.....

Adam Melicher

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual attention modelling aim to imitate the complex psychological functioning of the human brain and to identify the regions or objects that draw human visual attention. This topic has gained popularity in both academic community and industry during the last few decades, mainly due to advancements in the computer vision and neural networks field.

Human visual attention is subjective and biased based on the viewer's personal preferences. However, many existing efforts on saliency prediction are general and objective, and do not take the observer into account. As a result, the attention prediction for a certain person could be inaccurate. Here comes to place personalized saliency prediction, where models consider individual's personality traits.

1.1 Motivation

Personalized attention modelling finds its application in many fields. When developing an user interface, it is important to design a layout that would attract the observer in correct spots. One could use such model to verify which parts of

the scene are the most salient ones. Furthermore medical diagnostics field, where it is important to start the treatment on time. Differences in person's visual attention could identify early stages of various diseases. In personalized attention modelling, it is important to minimize the amount of data required for training. This is because displaying hundreds of images can be difficult for a single viewer, especially for clinical patients with cognitive disorders.

1.2 Goals

Thus the major goal of our work is to analyze existing approaches and utilize a personalized model that can extract a person's unique visual attention characteristics from minimal amount of data. The proposed model could be used to generate artificial saliency maps which would be used as ground truth maps to further expand datasets.

Chapter 2

Vision and visual attention

In this chapter, we briefly describe the anatomy of human eye and principles of human attention. In order to predict an individual's attention, it is necessary to know and understand how his brain processes visual stimuli.

2.1 Human vision

All the visual data contained in the surrounding world are transmitted by the light reaching the eyes. Outer parts of the visual system are transparent to enable undisrupted transmission to the inner neural layer. The innermost layer of the eye is retina. In Fig. 2.1, we can see that anatomically, retina consists of ten layers. From perception of information processing, these ten layers can be divided into three main layers: the photoreceptor layer (as rods and cones in Fig. 2.1), bipolar cell layer and ganglion cell layer. The rods are activated in poor light conditions, whereas cones are responsible for colour vision.[1] Ganglion cell layer projects to the lateral geniculate nucleus among others, which is responsible of sending 90% of information to the primary visual cortex.[2] From there information is handled via

two segregated paths, one for motion processing and other for object processing (see Fig. 2.2). Information processing up to the primary visual cortex is termed as lower level processing, whereas beyond primary visual cortex is termed as higher level processing, mainly because it involves high-order cognitive mechanisms like object recognition.[2]

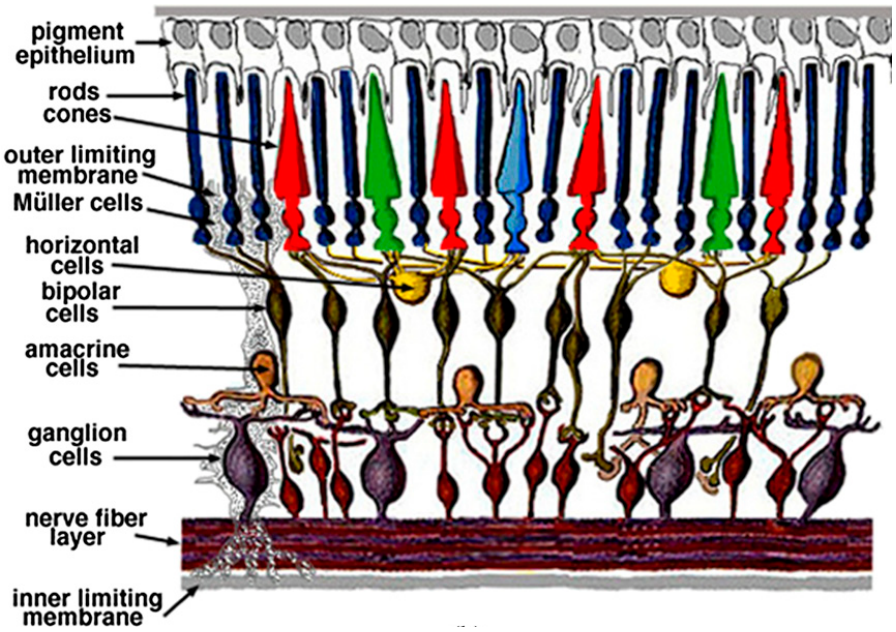

Fig. 2.1: Anatomy of the human eye retina.[3]

2.2 Eye movements

Due to eye movements, the observer can gather and process more information about visual stimuli. When both eyes are changing their direction from one visual stimulus to another, this movement is called a saccade. Saccades are rapid changes in eye positions that occur 3-4 times every second and last about 30 ms.[5] Between those changes, there are intervening microsaccades which form up so-alled fixations. The eye is relatively blind during saccades, so information is acquired

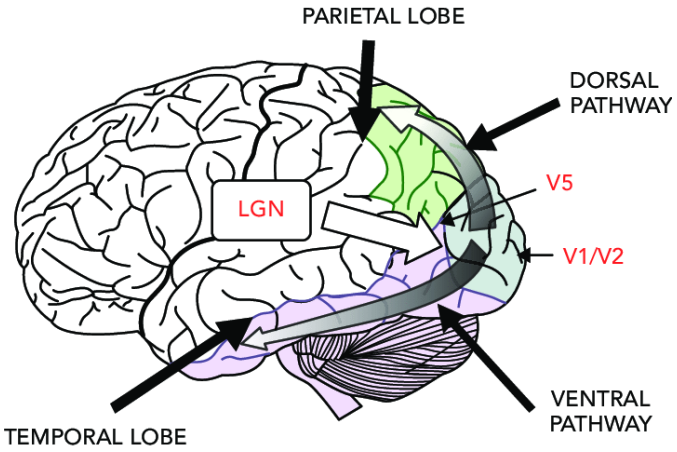

Fig. 2.2: Model of the human visual system. Lateral geniculate nucleus (LGN) directly projecting information to primary visual cortex (V1) and extrastriate visual cortex (V5). Then it is projected to parietal cortex to process motion, and to temporal lobe for object recognition and categorization. [4]

during fixations, which last approximately 250 ms (depends on task or stimulus complexity).[5] Saccades are important during reading and scanning of scenes which require sharp sight. Locations and sequences of saccades are not random, but follow a highly replicable path also called "scan path" (can be seen in Fig. 2.3). Scan paths can be used to generate eye tracking heat maps - a visualization that can effectively reveal the focus of visual attention for dozens or even hundreds of observers at a time.

2.3 Visual attention

Usually, the scenes we view are composed with many different objects. However, as the capacity of the visual system to process information about multiple objects at any given moment in time is limited, multiple objects visible at the same time compete for neural processing.[7] The competition can be divided into two different types of processes: bottom-up and top-down.

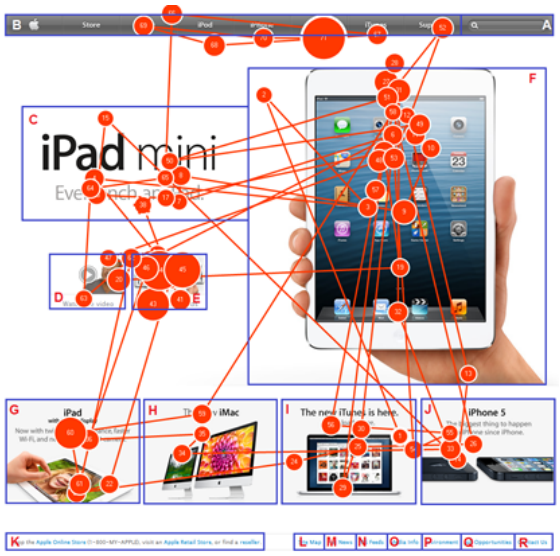

Fig. 2.3: Scan path of a particular observer. Saccades are represented with lines, fixations as circles. The bigger the circles are, the more time eyes of the observer fixated in that location.[6]

Bottom-up process (also called stimulus-driven) operates on raw sensory input, shifting attention to salient visual features of potential importance.[8] This is because stimulus salience depends on various object properties, such as orientation, color, shape or dissimilarity between stimulus and nearby stimuli. Bottom-up attention acts almost instantly because salience effects based on simple properties could be implemented at early visual processing stages.

Unlike bottom-up, top-down process implements long-term cognitive strategies.[8] Attention bias can be shifted by directing attention to a particular stimulus location, using a pointer or a circle as Fig. 2.4 shows. Natural attention bias shift occurs when we are experiencing emotions like hunger or fear, for example focusing on colored spots which could mean fruits when we are hungry. Top-down attention takes act approximately 100 milliseconds after bottom-up.[8] It is also called task-driven attention, because given tasks or previous experiences affect visual attention.

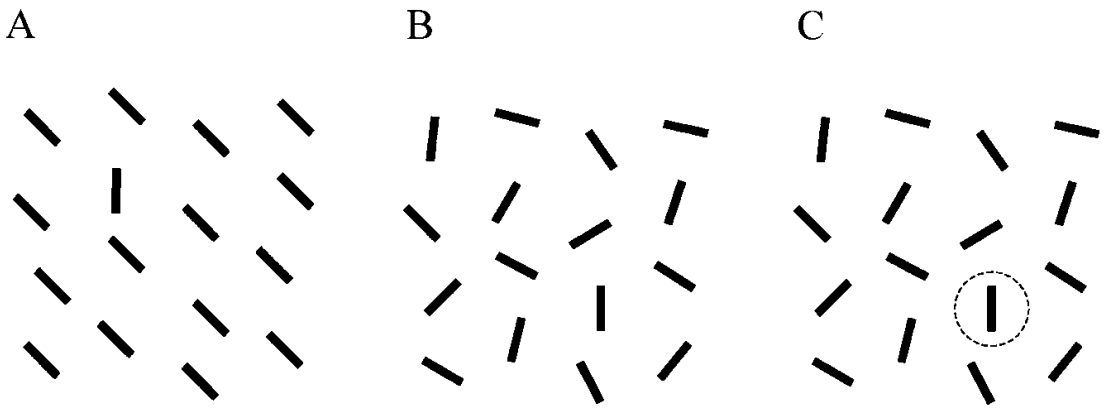

Fig. 2.4: In (A), thanks to bottom-up process attention is directly shifted to vertical line as a consequence of its high saliency. When there is no salience stimulus (B), it can be biased by using dashed circle (C) due to top-down process.[7]

Significant picture changes are essentially imperceptible under natural view circumstances, yet viewers have no trouble detecting them if they are guided to them.[9] Overt and covert attention controls access to these features and ensures that only relevant part of the scene is focused. While overt attention shift happens with previously mentioned eye movements, covert shift happens without changing the direction of the gaze. [10]

2.4 Capturing visual perception

As technology in visual field advanced, there have been developed many methods to capture visual perception of observer.[11] However, in this thesis we will focus on remote eye-tracking technologies, because they are the most utilized technology for creating visual attention datasets.[12, 13, 14, 15, 16, 17] Remote trackers have high tolerance for head movement, and are frequently used when studying people with certain medical conditions and infants who are unable to control their movements.

2.4.1 Eye tracking

An remote eye tracker uses high definition cameras and near-infrared light projectors to record the direction how it's reflected off the cornea. The camera is set up with a view of the eyes from a distance, and therefore the systems can automatically alter the camera field of view to catch up on head movements. To calculate the position of the attention and determine exactly where it's focused, various advanced algorithms come to use (more in next section). This makes it possible to measure and study visual behavior and fine eye movements, because the position of the attention are often mapped multiple times a second. How quickly an eye tracker is able to capture these images is understood as its frequency.

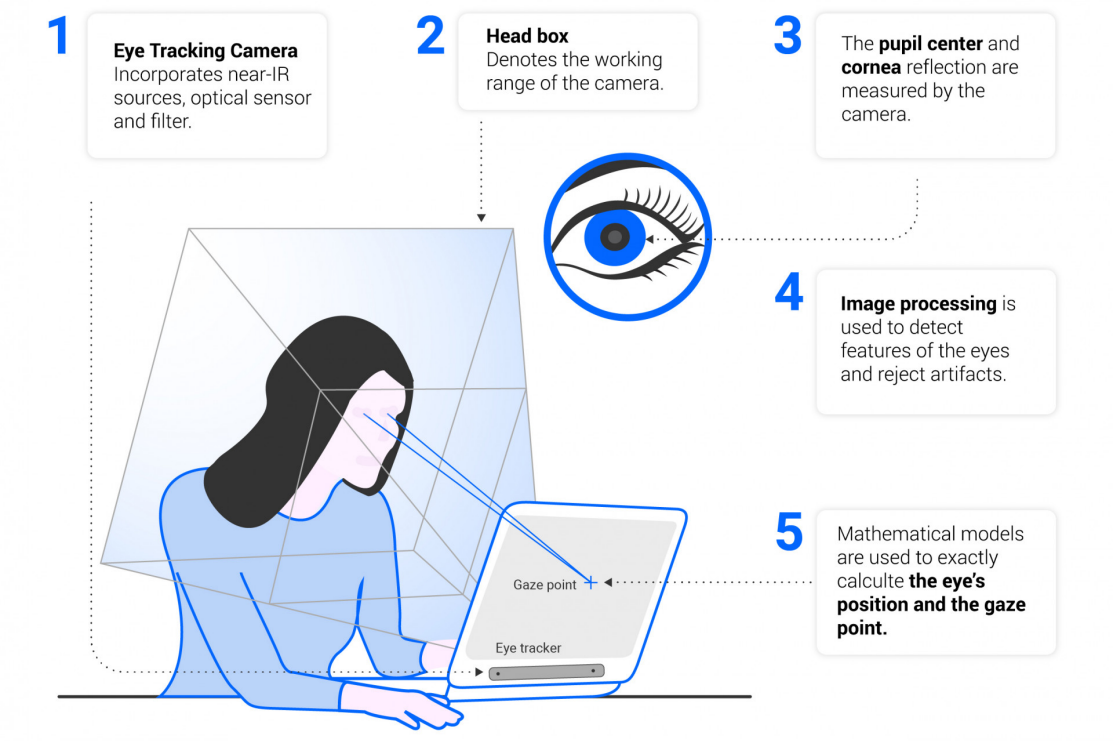

Fig. 2.5: Visualisation of eye tracking mechanism.[18]

2.4.2 Gaze estimation

Machine learning algorithms are used to deliver highly accurate tracking of the observer's gaze. Based on approach used, these algorithms can be classified into several categories:

- **Support Vector Machine**: Mainly based on regression, this method was able to detect repetitive patterns in eye movements with 76.1% accuracy.[19] However, this type of algorithm requires significant size of the dataset.
- **Naive Bayes**: Faster training than other algorithms, and worked better with smaller sample size. Despite the fact that this method cannot incorporate feature interactions, achieved accuracy was 90.22%.[20]
- **Convolutional Neural Networks**: Achieved high accuracy even with small amount of training data. Learning phase is crucial when utilising such model, if not appropriately controlled it may affect accuracy. [21]
- **Deep learning**: Further improved accuracy and performance. Needs large dataset for training, yet high-quality images showed to be more efficient.[21]

Other algorithms such as Random Forest or Hidden Markov Model are not used as widely as the ones listed above, and their usefulness is subject of further research.[21]

Chapter 3

Visual attention modeling

First attempts to model visual attention trace back to 1985, when Koch and Ullman introduced neural network to select the most salient locations.[22] Since then, interest in the field has been increasing. Numerous approaches have been proposed and have been evaluated against different datasets. Almost all of them were inspired by human visual system. In this chapter, we will cover most influential models and datasets.

3.1 Datasets

Numerous datasets were created to evaluate and compare performance of individual models. First widely used dataset was MIT300, which contained 300 images with tracking data of 39 observers aged between 18 and 50 years and viewing time of each picture was 3 seconds.[12] They motivated observers to pay attention by telling them that it is a memory test and after viewing the images, they will be asked if they seen particular pictures before. Another popular dataset was CAT2000. A larger scale benchmark, containing 4000 images equally divided into

20 categories. Each of 120 observers viewed 800 images split into 4 sessions and viewing time for each image was 5 seconds.[13] These two datasets are used in popular MIT/Tuebingen Saliency Benchmark.[14] Different approach was used during creation of SALICON dataset, where they used a general purpose mouse instead of eye trackers, which was proven highly similar to eye tracking approach.[15] This enabled large-scale collection of 10000 images each viewed by 60 observers.

Personalized saliency dataset is to our knowledge the largest dataset dedicated to personalized attention modelling. It consists of 1600 images with eye tracking data of 30 observers. Images were viewed for 3 seconds, 4 times each producing 4 saliency maps.[16] These maps were then combined to obtain the final personalized saliency map for the observer (see Fig. 3.1).

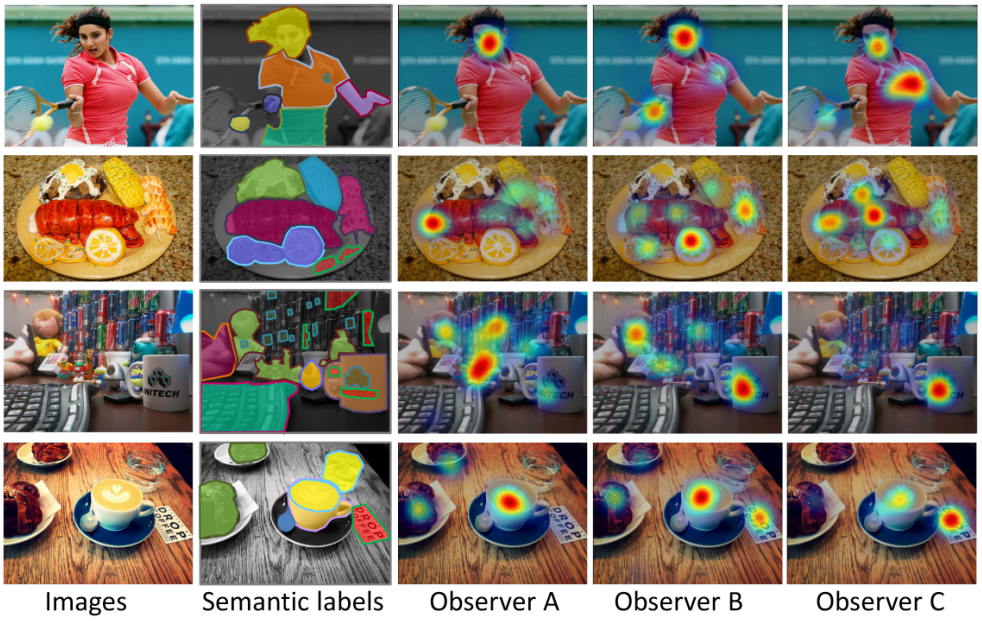

Fig. 3.1: Example image from the Personalized saliency dataset. Individuality in observer visual attention can be seen in obtained saliency maps, where different observers looked at different spots.[16]

Personalized datasets from medical diagnostics field are very rare and hard to

obtain. This is because patients suffering from cognitive diseases might have problems viewing large number of images. However, we were able to obtain dataset proposed by J. Chung et al. [17], which contains eye tracking data of 104 observers. Each observer viewed 106 slides. It is important to note that 48 of the observers were diagnosed with Alzheimer's disease, 16 of them was also diagnosed with apathy.

3.2 Pre-deep-learning era in visual attention modelling

This section describes visual attention models in terms of mechanisms for achieving saliency, mainly based on the work of Borji et al.[23]. Many models were directly or indirectly influenced by the human psychological or neuropsychological findings, but Cognitive models derived from those the most. [23] They utilise decomposition of input image into separate channels, followed by other transformations to produce saliency map. A Cognitive model worth mentioning was proposed by Itti et al.[24], where they used three feature channels for color, orientation and intensity. This model was used as basis for later models and often used for comparison. [23] Bayesian models fuse top-down and bottom-up maps based on statistical methodology to obtain final saliency map. This allows them to learn from data and unify many factors in a equitable manor. Next category of attention models use graph to describe the conditional independence between random variables, thus the name Graphical models. Thanks to their complexity, they are able to predict more sophisticated attention mechanisms.

Decision theoretic models can be used to anticipate fixation or classification tasks. The overall point is that visual attention should be guided by optimality with

respect to the end task. These models are not directly derived from biology, they can be implemented as Cognitive models as a network with a layer of simple and complex cells. On the other hand, Information theoretic models select the most informative parts of scene while discard the rest. They simulate human saccadic scan paths, by choosing locations with highest residual perceptual information. Spectral analysis models maintain high success rate and are simple to implement. Models in this category derive the saliency of an image in the frequency domain rather than in the spatial domain.

3.3 Deep-learning-based models of visual attention

Convolutional neural networks (CNNs) have been widely used in many computer vision fields. [25] Their multi-level and multi-scale features allow them to predict most salient regions even when shades or reflections exists. According to Borji et al.[25], they are divided into two categories: classic convolutional networks (CCN), and fully convolutional networks (FCN).

3.3.1 Classic convolutional networks

To detect saliency, they use multilayer perceptrons and oversegmentation of the input image into small regions. It is a classification problem where each segment is marked with a certain degree of significance. However, after this process the spatial information cannot be preserved.

In [26], Wang designed model utilising two subnetworks, one each for global search and local estimation. When estimating local saliency, it predicts the saliency score of each pixel by learning local patch features. Another deep neural network is using global features to to determine saliency value of each object region. The weighted sum of local and global salient object regions produces the final saliency

map.

Another interesting approach was proposed by Lee, where they used predefined features to further improve the performance of the VGG-net model.[27] By comparing low level features with other parts of the input image they obtain a distance map that is encoded by CNN and later unified with high level features computed by the VGG-net. The unification is done using a two-layer multilayer perceptron to to obtain the saliency of each region. Li and Yu pushed this idea even further, when they utilized a pre-trained CNN to extract multilevel features. [28]

3.3.2 Fully convolutional networks

Recent models adopt FCN architecture and became superior in the field of saliency prediction.[29] Their biggest advantages are in pixelwise prediction corresponding to input image and relatively fast saliency prediction. Most FCN networks work in single-stream process, as illustrated in Fig. 3.2. A good example of such network is UCF model, where was introduced a reformulated dropout in encoder FCN to learn deep uncertain convolutional features.[30] To avoid checkboard artifacts, an effective hybrid upsampling method was used in the decoder FCN. This led to more detailed object boundary representation.

There are other types of architectures, like multi-stream, side-fusion, branched or capsule networks. Multi-stream networks are composed of multiple network streams to learn multi-level saliency features from multiple inputs. Utilizing two streams, one for extracting regional saliency features, other for producing pixel-wise saliency map, considerable advance was made in the saliency prediction task.[32] TSPOANet network was implemented by combining multi-stream and capsule approaches, and was demonstrated to be superior over the other state-of-the-art methods.[33]

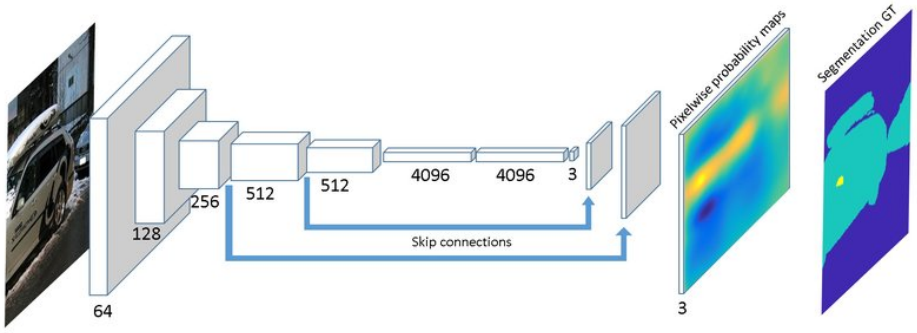

Fig. 3.2: Single-stream FCN network architecture with VGG-16 classification network and three deconvolution blocks.[31]

3.4 Personalized attention modelling

Human visual attention is individual and might be affected by observer's gender, age, preferences or other factors.[16] Previously discussed models were focused on generalizing attention of the observers as one universal observer. While this approach might be sufficient to predict salient regions common to all observers, it fails to predict individuality in visual attention.[34]

Research shows that high-level features such as human faces make bigger impact on individuality of human attention than low-level features.[34] Also complex high-level features are better extracted by deeper layers of CNN models. Outliers in predictions can direct us to further analysis of individuality in terms of what exactly is behind their unique behaviour. This uniqueness might also be a consequence of some sort of cognitive disorder. [17]

3.5 Performance evaluation

There are several different metrics for evaluating the ability of models to predict saliency maps. It is not yet defined which metric is the best, so the choice of

metric depends on what the model is trying to achieve. Metrics can express how the fixation maps are represented, how the false-positives (FP) and false-negatives (FN) are penalized, and others.[35] Due to this, research evaluations are done on multiple metrics simultaneously.[16, 33, 34, 32, 27, 28] The following list describes the most relevant ones.

- **Area Under Curve (AUC-Judd)**: Derived from signal detection theory. The saliency map is treated as a binary classifier, and the ratio of TP and FP is plotted on graph. [35, 36] Final score is area under the curve it creates, see Fig.3.3 for more details.

- **Shuffled AUC (sAUC)**: This metric is a modification of AUC-Judd and penalizes models for center bias. If there was a center biased dataset, a model would incorporate center bias into predictions and get a high score on the AUC-Judd metric. [35, 36]

- **Information Gain (IG)**: Calculates information gain of predicted saliency over certain ground-truth in terms of probability, so it does not specifically penalize models for center bias or other factors. Given a saliency map P, baseline map B and a binary map of fixations Q^B, information gain is computed as:

$$IG(P, Q^B) = \frac{1}{N} \sum_i Q_i^B [log_2(\epsilon + P_i) - log_2(\epsilon + B_i)] \tag{3.1}$$

  where i is the index of i^{th} pixel, N is the total number of fixated pixels, and ϵ is for regularization. Any score above zero indicates that the prediction is better than baseline.[35, 36]

- **Pearson's Correlation Coefficient (CC)**: Statistical method used to calculate dependency of two variables. In this case P is the generated saliency

map and Q^D is the fixation map:

$$CC(P, Q^D) = \frac{\sigma(P, Q^D)}{\sigma(P) \times \sigma(Q^D)} \tag{3.2}$$

where $\sigma(P, Q^D)$ is the covariance of P and Q^D. Treats FP and FN equally, so if the goal is to determine which is having more impact on model performance, other metrics would be preferable. [35, 36]

- **Similarity or histogram intersection (SIM)**: Measuring the similarity between the distributions of predicted saliency map and ground-truth map. It is computed as the minimal sum of values at each pixel, given a continuous fixation map Q^D and a saliency map P:

$$SIM(P, Q^D) = \sum_i min(P_i, Q_i^D) \text{ where } \sum_i P_i = \sum_i Q_i^D = 1 \tag{3.3}$$

iterating over discrete pixel locations i. It is very sensitive to FN values and blur value of the training set. [35, 36]

- **Kullback-Leibler divergence (KL-div)**: Contradictory to the SIM metric, it represents the difference between the predicted saliency map and ground-truth distributions. It is calculated as:

$$KL(P, Q^D) = \sum_i Q_i^D log(\epsilon + \frac{Q_i^D}{\epsilon + P_i}) \tag{3.4}$$

where P is predicted saliency map, Q^D fixation map, and ϵ is regularization constant. It penalizes FN even more than the SIM metric. Zero value indicates perfect prediction. [35, 36]

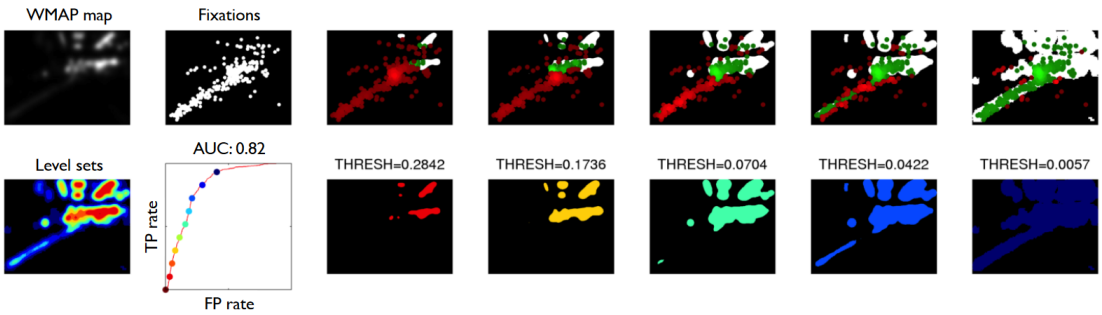

Fig. 3.3: To plot the AUC graph, generated saliency map is thresholded at various values (THRESH). This produces the level sets in the bottom row, where on each there are plotted fixations (top row). The proportion of level set's pixels covered by fixations is TP rate, and vice versa. [35]

Chapter 4

Related Work

In this chapter, we discuss 4 close related works. In all of them, there are different solutions proposed for personalized attention modelling.

4.1 Personalized Attention Network

The goal of the work of Xu et al.[16] was to incorporate observer's preferences into attention modelling by proposed two-stream FCN architecture model called Personalized Attention Network (PANet) (Fig. 4.1). First part of the network is VGG-16 without the classification layers. Then it's divided - one stream is for predicting generalized saliency map, and other generates preference map based on object recognition and observer's preferences. Final personalized saliency map is obtained by combining the saliency and preference maps.

To collect preferences, up to 4000 images were chosen from the SALICON and MS COCO datasets such that each image contained at least one object from the observer's preferences. Based on collected preferences, artificial ground truth data was generated from the SALICON dataset. The model was trained in three phases,

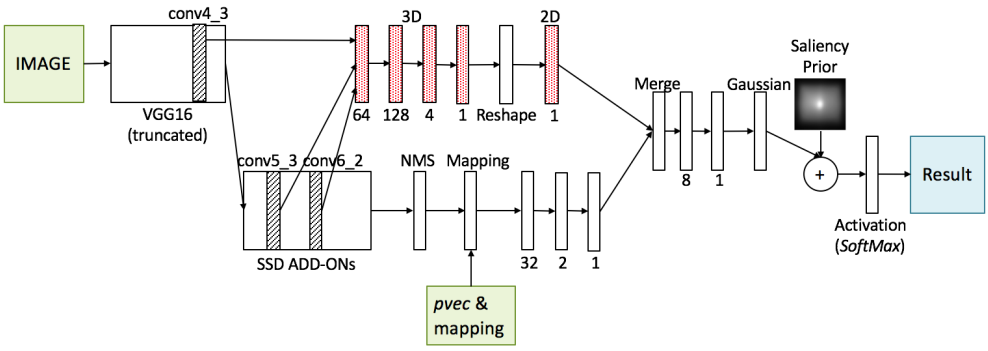

Fig. 4.1: Architecture of the PANet network. [16]

where in the first one the SSD model was pre-trained on the MS COCO dataset and the pre-trained weights were further fed to the second phase. There they used the SALICON dataset to pre-train the saliency layers. In the last phase, the model as a whole was trained, based on the mentioned artificial data.

During evaluation, they found out that observer's preferences had the biggest impact on the KL-Divergence metric. The more precise and biased were the preferences defined, the more the performance of the model improved. Compared to generalized models, PANet predicted saliency maps better.

4.2 Personalized Saliency Prediction based on Adaptive Image Selection

In the work of Moroto et al.[37], a novel method for reducing the size of training dataset was proposed, called Adaptive Image Selection (AIS). A multi-task CNN model was proposed for saliency map prediction. The first part of the model was composed of one encoding stream, and then divided into P decoding streams, where P represents the number of people in the dataset. Each stream predicted a personalized saliency map for that person. In parallel with this process, the AIS

selects the images that the target person must view for model to fit their attention. The final personalized saliency map is then generated based on the personalized saliency maps of similar people. This whole process is visualized in Fig. 4.2.

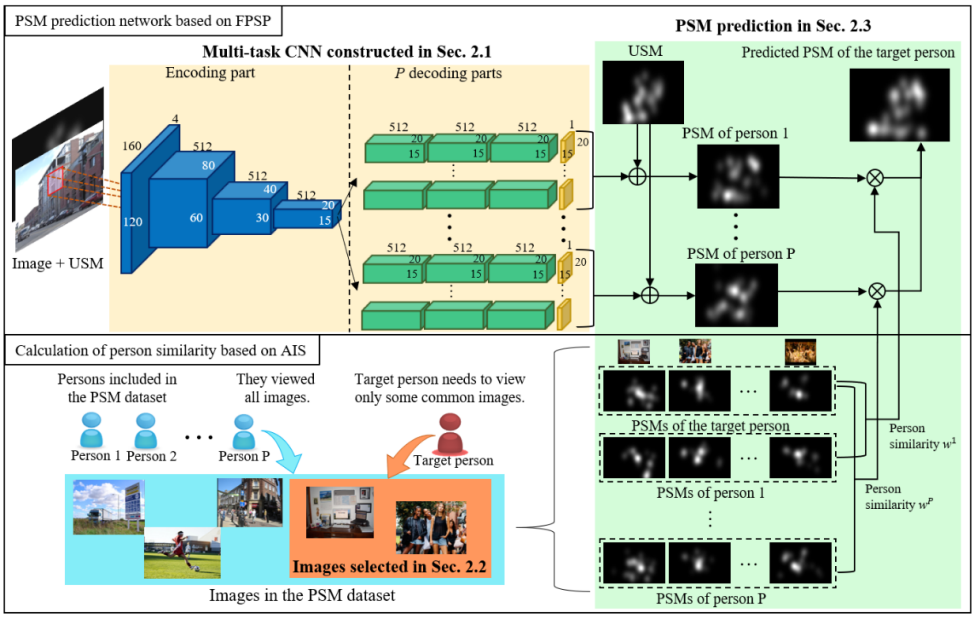

Fig. 4.2: Visualization of the PSM prediction process. [37]

They used images from the PSM dataset (3.1) to train and test the network. Saliency maps collected from observations by 20 people were used for training, which was 1,100 images for each person. During training, AIS selected various sized subsets ranging from 10 to 100 images. The remaining 500 images viewed by 10 people were used as a test set.

They compared the effectiveness of the proposed solution against four USM-based models from the MIT saliency benchmark, and two PSM-based models. It is important to note that all models were trained by images selected by AIS. They used the metrics CC, KL-div and Sim (3.5), where on each metric they achieved the best score. Thus, the evaluation showed the robustness of the proposed AIS solution.

4.3 Personalized Saliency Prediction based on Collaborative Multi-Output Gaussian Process Regression

Research work of Moroto et al.[38] is a continuation of the work described in the previous section, where a Multi-task CNN with AIS was proposed. Hovewer, now Collaborative Multi-Output Gaussian Process Regression (CoMOGP) was introduced as the final prediction method. CoMOGP is a probabilistic method, thus it can avoid overfitting even on small amounts of training data. It uses the PSM of similar people generated by the CNN model to generate the saliency map, but also takes into account the visual features from the target image.

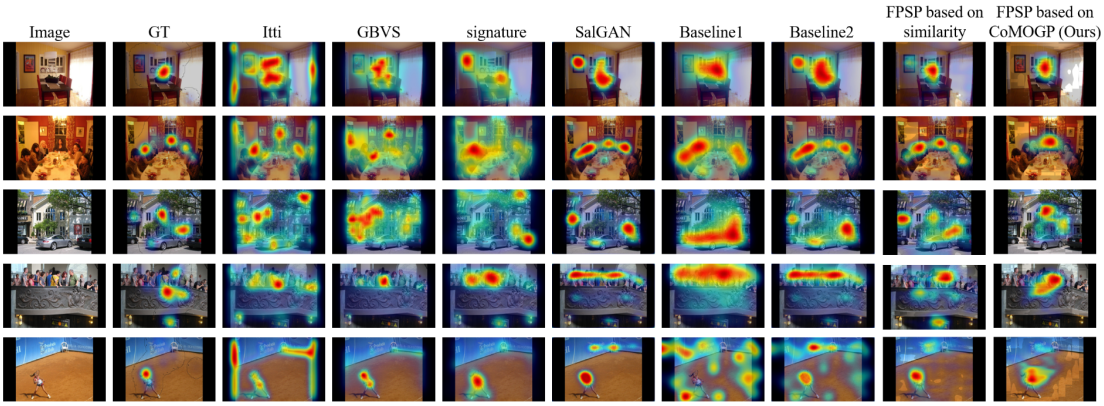

Fig. 4.3: Predicted saliency maps by evaluated models. First column from left represents target image. Second is ground truth map. Then, there are four columns corresponding to four USM-based models. Next, three PSM-based models. In the last column there are saliency maps predicted by the proposed CoMOGP model. [38]

Model was trained on images from the PSM dataset (3.1) at the same conditions as described in previous section. Proposed model was evaluated against four USM-based and three PSM-based models, and outperformed all of them. Also it can be seen in Fig. 4.4, that saliency map generated by proposed model is most similar

to ground truth map.

4.4 Impact of individual human characteristics on visual attention

Model proposed in the work of Hoffer et al.[39] utilizes lightweight generalized model as a base to capture traits common to all observers. Then fine-tuning of the model for each observer separately is used (also called transfer learning), therefore learning individual traits. In this work, they also used various classifiers to identify people suffering from Alzheimer's disease.

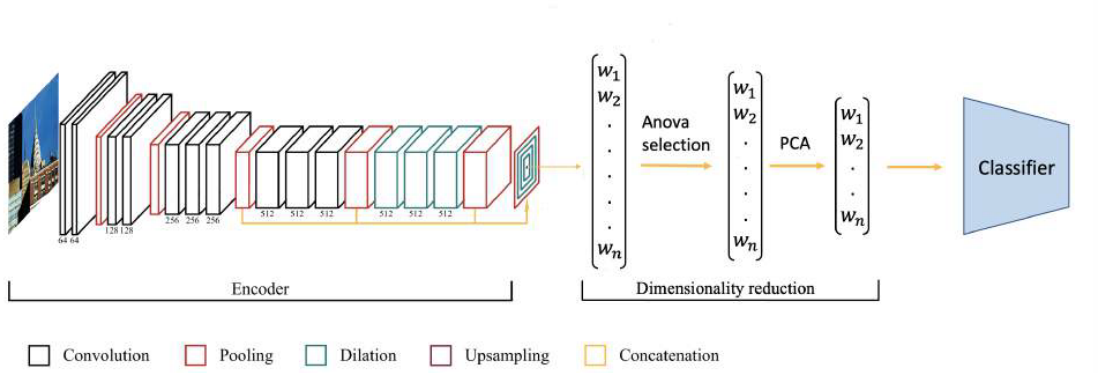

Fig. 4.4: Architecture of the proposed model including classification and transfer learning. [39]

Generalized model was trained on the SALICON [15] dataset and stored as initial state for further training. Then N models were trained from this baseline, where N is the number of observers. This personalized training was done on the PSD dataset (more in 3.1). In evaluation, they compared the proposed model with generalized models, as well as with current state-of-the-art personalized models. They showed that their model produced an average improvement across all evaluated metrics.

Chapter 5

Solution proposal

It is currently possible to generate personalized ground truth maps using neural network models. However, it is necessary to verify how representative this artificial ground truth maps would be and what information we would lose by such approach. Therefore, we will evaluate the generated maps against real fixation data. Evaluation metrics would be used as a confidence-factor to quantify how similar the generated data is to human observers. To our knowledge, no such research was published so far.

5.1 Model training and personalization

We propose to utilize the model discussed in Section 4.4, but without the classification part. We choose this model because of its good prediction performance according to the MIT/Tuebingen benchmark [14], lightweight architecture, and the whole environment is dockerized which makes it easy to implement. As stated in Section 3.4, personalized models are more capable of predicting individuality in visual attention. Also, a custom evaluator is included which makes the evalua-

tion process faster by utilizing all available CPU cores and GPU resources. This custom evaluator has implemented all the metrics discussed in 3.5. We will try to further improve prediction performance of the model by fine-tuning the configuration parameters.

The model would be trained on data from the SALICON and PSD datasets. SALICON would be used to train the generalized base of the model, because it is the largest generalized dataset to our knowledge. Then, this salicon base would be fine-tuned on generalized saliency maps from the PSD dataset. Personalization of the model would be done by training the pre-trained base on the PSD dataset. For each observer, there would be separate training process and same configuration as in previous trainings. This is visualized in Fig. 5.1.

5.2 Evaluation

Evaluation of the model performance would be done on all evaluation metrics discussed in Section 3.5, except Infogain metric because we won't have saliency maps generated by other models to compare. We will use the personalized models to generate artificial ground truth maps from the training images available in the CAT2000 dataset, which would not be previously seen by the model. Generating the artificial maps would be the same as in test phase. Since we will produce artificial ground-truth for each personalized model, we will merge them in a single generalized ground-truth by making a weighted sum for each pixel of each image. Then we will evaluate whole dataset against the fixation data. We will also evaluate each of the CAT2000 dataset's categories, and analyse the results to check if there are any similar stimuli that the model can not predict well.

5.3 Data preparation

To ensure everything will run correctly during training and evaluation steps, we need to preprocess the individual datasets in the first place. Our model will be trained on the SALICON dataset, but no preprocessing is needed there. Then, to train it on generalized data from the PSD dataset, we must produce the required saliency maps. This is done by combining the fixations for each observer. Then, combined fixation data are converted to structured format, and black images with white pixels in place of fixations are generated. These are also reffered to as binary fixation maps, and used later in evaluations. To personalize the model we use original fixation maps provided by Xu et al. [16]. CAT2000 preprocessing is the same. Fixation data are converted to structured format, and binary fixation maps are generated once again.

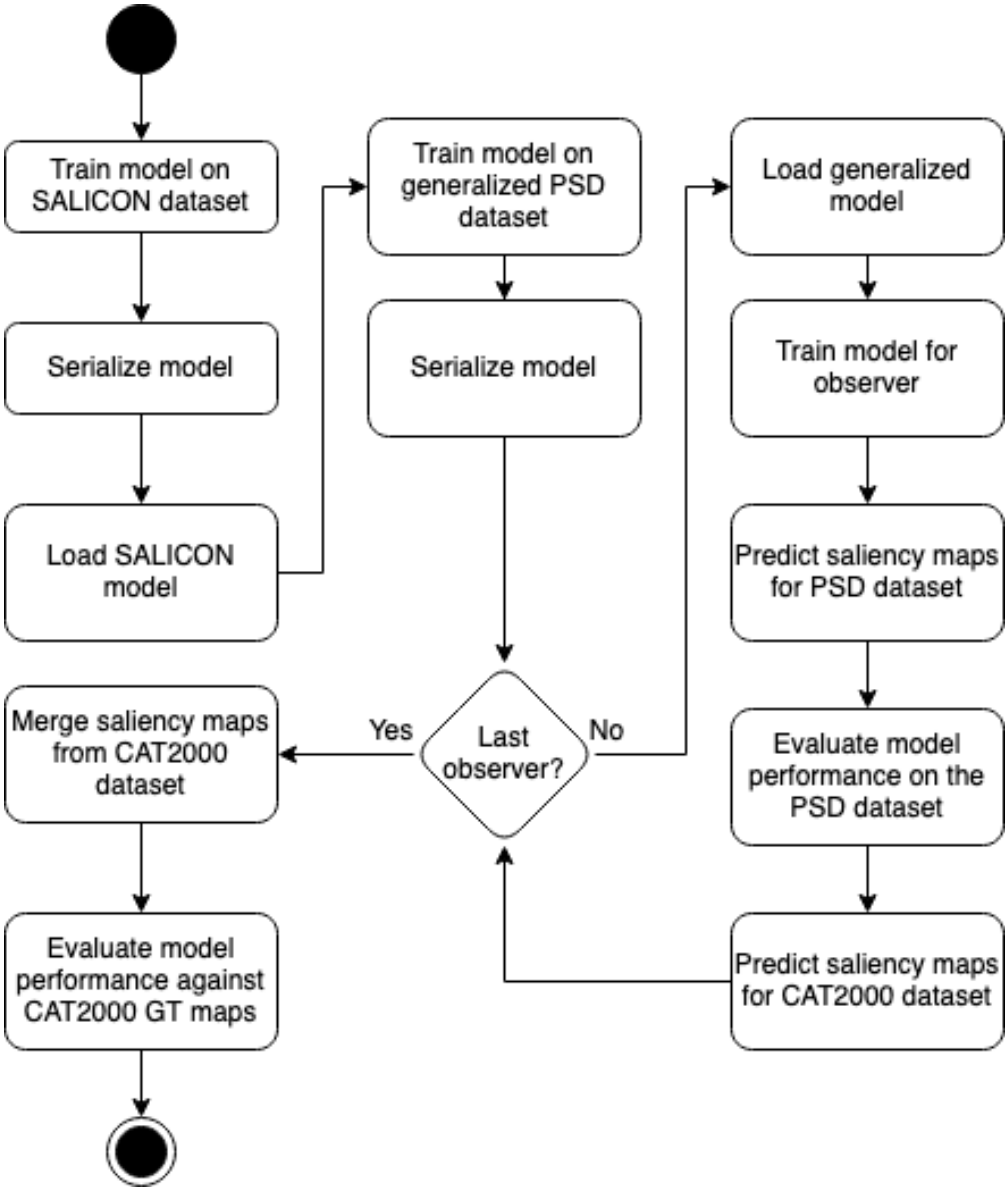

Fig. 5.1: Activity diagram describing steps of our proposed solution.

Chapter 6

Implementation

In this chapter, we will describe environment in which the experiments were conducted. The implementation is run in Docker so we highly recommend using it, but the individual steps can also be done without it. Also to make training and evaluation process faster, all available GPU resources are utilized.

6.1 Docker environments

Three separate docker images are used by our experiments. Datasets preprocessing is done with the python3 image. It has all the base requirements needed for generating saliency maps from fixations, like the python library OpenCV which is widely used for computer vision problems.

Model training and testing is done in the python3-tensorflow image. Tensorflow is a python library used for machine learning and artificial intelligence development. Its implementation is allowing scripts to fully utilize available GPU resources, which is useful in computation-heavy tasks.

The evaluation is done by older python implementation that uses version 2.7. This is where the Docker comes in really useful, because we can just run a script by a different image and have everything nicely separated.

6.2 Model training and evaluation

We have found that the initial parameters for model training described in are optimal, so we used them for our experiments:

- Number of epochs: 150
- Batch size: 1
- Learning rate: 0.000001
- Validation split size: 0.1
- Failing epochs: 1

Number of epochs is set to a large number, because we use early-stopping of training to prevent model overfitting. This is set with parameter failing epochs, when there is no improvement on validation loss anymore. An improvement on some metrics was achieved by using original fixation maps provided by Xu et al. [16], not the ones generated by scripts. More detailed analysis of results is in Section 7.1.

For PSD evaluation we have used the code provided by Hoffer [39] with slight adjustments to how it is run. We have tried to evaluate the CAT2000 dataset with implementation from the MIT/Tuebingen saliency benchmark [14], but due to fact that the current code is uncompillable and needs to be revised by the authors, we have avoided the implementation for now. Thus we switched back to the same evaluation used for PSD dataset.

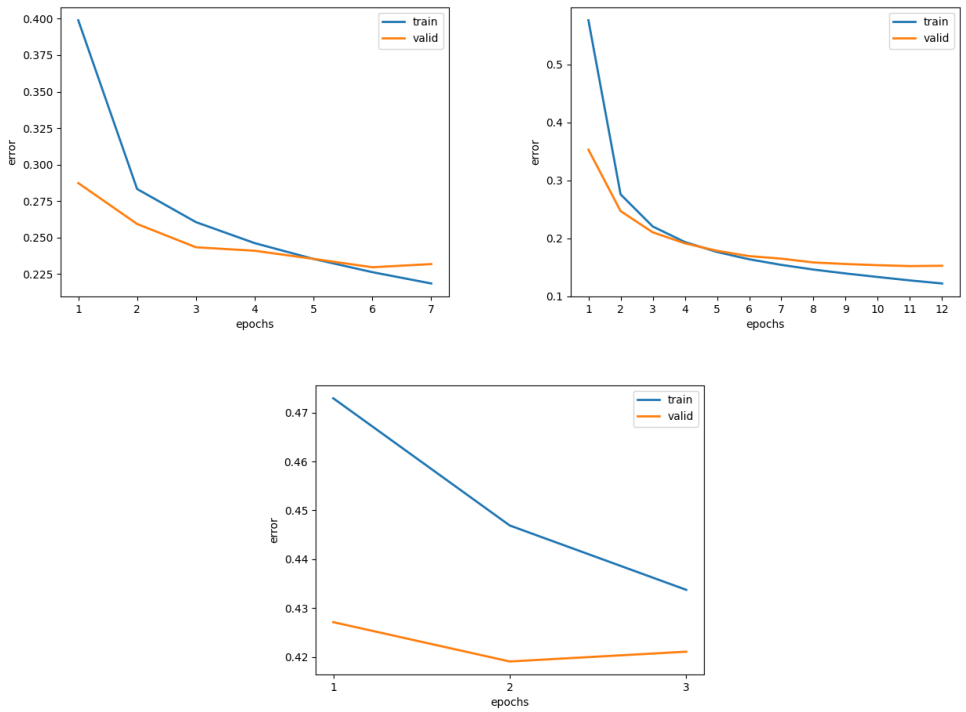

Fig. 6.1: First image shows the learning curve of training on the SALICON dataset. Next is training on the generalized data from the PSD dataset. Last image shows learning curve of training the personalized model for Subject 1 from the PSD dataset.

6.3 Command-line interface

While doing our research, we found the existing implementation unnecessarily complicated. All of the steps were done as bash commands with poorly documented parameters. Also preprocessing datasets to their initial state was exhausting, because there was no hints at which scripts should be run or how should the initial state look like.

To make all the mentioned processes simpler, we have wrapped them in a command-line interface. It comes with commands for all the steps needed to reproduce our results, with proper help texts. Each command supports multiple options. Also

automatic preprocessing of datasets is implemented, so user just needs to set path where each files should be stored. It is possible to serialize and then load multiple models. This allows us to make transfer learning possible and also revert models to their original state. Testing and training can also be done on any serialized model.

Chapter 7

Evaluation

In this chapter we evaluate and discuss the results of our solution. After reaching the final state of the personalized models, we test them against the train part of the CAT2000 dataset. The resulting saliency maps were combined into a generalized one, and we evaluated them against the fixation maps from the CAT2000 dataset.

7.1 Results validation

First we needed to achieve state of our model where all of the relevant metrics for model evaluation are comparable to the ones of Hoffer [39]. As the table Table 7.1 shows, we have achieved that. We've even seen improvements in some metrics. SIM, CC and KL-Div metrics mainly penalize FN predictions. The improvement on these metrics is because Hoffer used his generated fixation maps smoothed by Gaussian convolution. In the analysis of his solution, we noticed shortcomings in the generation of the mentioned fixation maps, so we used the original fixation maps provided by Xu et al. [16] for training and evaluation.

Metric	Hoffer's personalized model	Our personalized model
AUC-Judd	**0.9156**	0.9101
AUC-Borji	**0.8576**	0.8382
sAUC	0.8232	**0.8575**
NSS	2.410	**2.4389**
SIM	0.4428	**0.6760**
CC	0.5679	**0.8092**
KL-div	1.1104	**0.4333**

Table 7.1: Comparison of our and Hoffer's solution prediction performance. We have seen a deterioration in the Auc-Judd and AUC-Borji metrics, but it is only worse by two tenths, which is acceptable. In addition, we noticed an improvement in the metrics SIM, CC, KL-div and NSS by several tenths. We skipped Infogain metric, because we didn't have available saliency maps generated by Hoffer, so we didn't have anything to compare.

After combining the saliency maps generated by our personalized models from the CAT2000 dataset, we noticed that very indistinct regions often appear there, caused by discussed individuality in the visual attention (see Section 3.4). Therefore, we tried to augment the data using thresholding to remove a certain amount of indistinct regions, followed by smoothing of edges with Gaussian filter. We tried various combinations of input parameters. However, as can be seen in the Table7.2, the metrics show that there is no significant improvement. We continued to work only with non-augmented data.

Since we use the whole dataset to evaluate, and our models has never seen the data from the CAT2000 dataset, there was no need to do cross validation. Instead, we divided the dataset into categories, and did the evaluation separately. More in the following sections.

Metric	DeepGaze II	Merged CAT2000	- with augmentations
AUC-Judd	0.8640	0.8578	0.8462
AUC-Borji	-	0.8351	0.8278
sAUC	0.6498	0.8680	0.8612
NSS	1.9619	1.8558	1.8372
SIM	0.7564	0.6421	0.6314
CC	0.5137	0.7114	0.7132
KL-div	0.6392	0.8704	1.876

Table 7.2: In the left column is the best model from the MIT/Tuebingen benchmark DeepGaze II. Middle column are results of our model. The column on the far right contains the best results that we managed to achieve by data augmentation.

7.2 Quantitative evaluation

For performance comparison, we chose the DeepGaze II [40] model, which achieved the best score on the AUC-Judd metric in the MIT/Tuebingen benchmark [14]. However, the DeepGaze model was evaluated on a CAT2000 test set and our model on the training set. However, this should not be a problem, as the test data should be representative of the dataset to maintain its consistency. We achieved similarly good results on AUC-Judd and CC metrics. As far as sAUC metric is concerned, we achieved comparable results to DeepGaze II with even worse performing model base using our proposed solution.

Figure 7.1 shows a comparison of how well the model predicted individual categories from the CAT2000 dataset. We see that in terms of most metrics, the worst predictions are in the Satellite category, and the best in Sketch. This can be due to the fact that satellite images can contain many different unclear objects in one stimuli, or none if it is a photo of a grassy area for example. Also such images were not included in training of the models, they are very specific and the model is not well prepared to handle them. On the other hand, there is always some significantly salient part in comparison to the background in the Sketch category,

so observers focus straight on it. In the Affective and Action categories, the KL-div metric achieved the worst values. NSS metric values were variable across all dataset categories. The other metrics achieved average results.

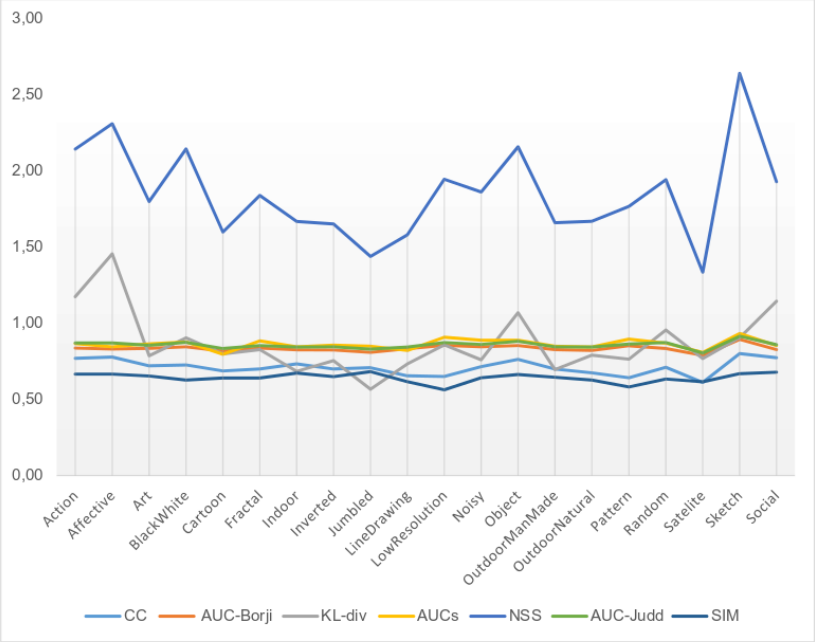

Fig. 7.1: Evaluation of metrics by CAT2000 dataset categories. The left axis describes values in each metrics. The lower axis describes the individual categories. The metrics are marked according to the legend at the bottom of the chart.

7.3 Qualitative evaluation

After quantitative analysis showed us which categories to focus on further, we can take a closer look on individual predictions of images. In the Fig. 7.2 we see how the individual predictions for the categories mentioned in 7.2 looked like. The predictions from the Satellite category were justified as the worst of the evaluated categories. We see that the images either contain few stimuli or, contrariwise, too

many stimuli, and the model then cannot determine where the observer could look. On the other hand, in the Sketch category, the predictions were the best, because they mainly contained clearly salient stimuli.

The last three pictures show the predictions of the models in different categories. These pictures are more like real scenes, as seen by the greater distribution of fixations, because different people looked at different areas. Personalized models have also been able to predict this diversity in human attention. However, overall, it can be seen that the model cannot predict attention exactly, but the predictions are very similar to human observers. In several cases, the model predicts wider areas than those on fixations, but the positions of these fixations are mostly correct.

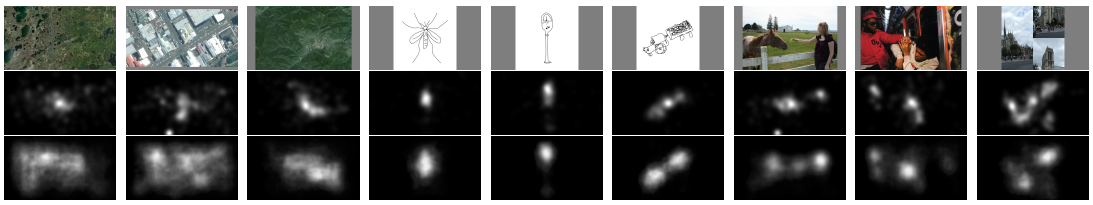

Fig. 7.2: Display of individual predictions against fixations and stimuli images. The first row shows stimuli images. The second row contains ground-truth fixation maps obtained from real observers. The third row contains combined predictions of 30 personalized models. The first three images are from the Satellite category. The following three from the Sketch category. The last three are each from a different category - Action, Affective and Jumbled.

Chapter 8

Conclusion

In the first part of this thesis, we introduced the topic and set the goals of our work. Then we analyzed the internal processes behind human visual attention and techniques for visual attention data collection. Next we covered datasets relevant to our work, a brief history of how neural network models in visual attention field evolved over the years, and their performance evaluation metrics. We have proposed a solution based on discussed closely related work and successfully implemented it. After all of the personalized models training was done, we evaluated them against real observers data. This was done by combining all the generated saliency maps into a generalized ground truth. We tried multiple augmentations while we combined these maps, but with unsatisfactory results. This way, we mimicked the approach of the creation of CAT2000 ground-truth fixation maps obtained from real observers, and got similar results in many categories. To our knowledge, no such research was published so far.

We can conclude that generating artificial ground truth fixation maps can be done with personalized models. In our experiments, we have shown that the models predict saliency with good accuracy even on unseen data. Our solution can be used

to generate generalized or personalized datasets from any available stimuli.

Future work could be aimed at further improving model prediction performance. Also deeper analysis can be made on how to combine personalized saliency maps into generalized one, if making a generalized dataset.

References

1. HILDEBRAND, Göran Darius; FIELDER, Alistair R. Anatomy and Physiology of the Retina. In: *Pediatric Retina*. Ed. by REYNOLDS, James; OLITSKY, Scott. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 39–65. ISBN 978-3-642-12041-1. Available from DOI: 10.1007/978-3-642-12041-1_2.
2. RAMAMURTHY, Mahalakshmi; LAKSHMINARAYANAN, Vasudevan. Human Vision and Perception. In: *Handbook of Advanced Lighting Technology*. Springer International Publishing, 2015, pp. 1–23. Available from DOI: 10.1007/978-3-319-00295-8_46-1.
3. MACGILLIVRAY, Tom; TRUCCO, Emanuele; CAMERON, James; DHILLON, Baljean; HOUSTON, John; BEEK, Edwin. Retinal Imaging as a Source of Biomarkers for Diagnosis, Characterisation and Prognosis of Chronic Illness or Long-Term Conditions. *The British journal of radiology*. 2014, vol. 87, pp. 20130832. Available from DOI: 10.1259/bjr.20130832.
4. CREWTHER, SHEILA GILLARD; GOHARPEY, NAHAL; BANNISTER, LOUISE; LAMP, GEMMA. 14| GOAL-DRIVEN ATTENTION AND WORKING MEMORY. 2012, pp. 191–208.
5. FISCHER, Burkhart; WEBER, Heike. Express saccades and visual attention. *Behavioral and Brain Sciences*. 1993, vol. 16, no. 3, pp. 553–567.

6. ERASLAN, Sukru; YESILADA, Yeliz; YANEVA, Victoria; HARPER, Simon. Eye-tracking scanpath trend analysis for autism detection. 2020, no. 128, pp. 1–8. Available from DOI: 10.1145/3441497.3441498.
7. UNGERLEIDER, Sabine Kastner; G., Leslie. Mechanisms of Visual Attention in the Human Cortex. *Annual Review of Neuroscience*. 2000, vol. 23, no. 1, pp. 315–341. Available from DOI: 10.1146/annurev.neuro.23.1.315. PMID: 10845067.
8. CONNOR, Charles E.; EGETH, Howard E.; YANTIS, Steven. Visual Attention: Bottom-Up Versus Top-Down. *Current Biology*. 2004, vol. 14, no. 19, pp. R850–R852. Available from DOI: 10.1016/j.cub.2004.09.041.
9. LEVIN, Daniel T; SIMONS, Daniel J. Failure to detect changes to attended objects in motion pictures. *Psychonomic Bulletin & Review*. 1997, vol. 4, no. 4, pp. 501–506.
10. ITTI, Laurent; KOCH, Christof. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*. 2000, vol. 40, no. 10, pp. 1489–1506. ISSN 0042-6989. Available from DOI: https://doi.org/10.1016/S0042-6989(99)00163-7.
11. MELE, Maria Laura; FEDERICI, Stefano. Gaze and eye-tracking solutions for psychological research. 2012, vol. 13, no. S1, pp. 261–265. Available from DOI: 10.1007/s10339-012-0499-z.
12. JUDD, Tilke; DURAND, Frédo; TORRALBA, Antonio. A Benchmark of Computational Models of Saliency to Predict Human Fixations. In: *MIT Technical Report*. 2012.
13. BORJI, Ali; ITTI, Laurent. CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. *CoRR*. 2015, vol. abs/1505.03581. Available from arXiv: 1505.03581.

14. KÜMMERER, Matthias; BYLINSKII, Zoya; JUDD, Tilke; BORJI, Ali; ITTI, Laurent; DURAND, Frédo; OLIVA, Aude; TORRALBA, Antonio. *MIT/Tübingen Saliency Benchmark* [https://saliency.tuebingen.ai/].

15. JIANG; MING; HUANG; SHENGSHENG; DUAN; JUANYONG; ZHAO; QI. SALICON: Saliency in Context. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

16. XU, Yanyu; GAO, Shenghua; WU, Junru; LI, Nianyi; YU, Jingyi. Personalized Saliency and Its Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2019, vol. 41, no. 12, pp. 2975–2989. Available from DOI: 10.1109/TPAMI.2018.2866563.

17. CHUNG, Jonathan; CHAU, Sarah A.; HERRMANN, Nathan; LANCTÔT, Krista L.; EIZENMAN, Moshe. Detection of Apathy in Alzheimer Patients by Analysing Visual Scanning Behaviour with RNNs. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, United Kingdom: Association for Computing Machinery, 2018, 149–157. KDD '18. ISBN 9781450355520. Available from DOI: 10.1145/3219819.3219908.

18. MENTO, A. Mark. *Different kinds of eye tracking devices*. 2021. Available also from: https://www.bitbrain.com/blog/eye-tracking-devices.

19. BULLING, Andreas; WARD, Jamie A.; GELLERSEN, Hans; TRÖSTER, Gerhard. Eye Movement Analysis for Activity Recognition Using Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011, vol. 33, no. 4, pp. 741–753. Available from DOI: 10.1109/TPAMI.2010.86.

20. BHATTARAI, Rasa; PHOTHISONOTHAI, Montri. Eye-Tracking Based Visualizations and Metrics Analysis for Individual Eye Movement Patterns. In: *2019 16th International Joint Conference on Computer Science and Software*

Engineering (JCSSE). 2019, pp. 381–384. Available from DOI: 10.1109/JCSSE.2019.8864156.

21. KLAIB, Ahmad F.; ALSREHIN, Nawaf O.; MELHEM, Wasen Y.; BASHTAWI, Haneen O.; MAGABLEH, Aws A. Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies. *Expert Systems with Applications*. 2021, vol. 166, pp. 114037. ISSN 0957-4174. Available from DOI: https://doi.org/10.1016/j.eswa.2020.114037.
22. KOCH, C; ULLMAN, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum. Neurobiol.* 1985, vol. 4, no. 4, pp. 219–227.
23. BORJI, Ali; ITTI, Laurent. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*. 2012, vol. 35, no. 1, pp. 185–207.
24. ITTI, L.; KOCH, C.; NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998, vol. 20, no. 11, pp. 1254–1259. Available from DOI: 10.1109/34.730558.
25. BORJI, Ali; CHENG, Ming-Ming; HOU, Qibin; JIANG, Huaizu; LI, Jia. Salient object detection: A survey. 2019, vol. 5, no. 2, pp. 117–150. Available from DOI: 10.1007/s41095-019-0149-9.
26. WANG, Lijun; LU, Huchuan; RUAN, Xiang; YANG, Ming-Hsuan. Deep networks for saliency detection via local estimation and global search. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3183–3192. Available from DOI: 10.1109/CVPR.2015.7298938.
27. GAYOUNG, Lee; YU-WING, Tai; JUNMO, Kim. Deep Saliency with Encoded Low level Distance Map and High Level Features. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

28. LI, Guanbin; YU, Yizhou. Visual Saliency Based on Multiscale Deep Features. *CoRR*. 2015, vol. abs/1503.08663. Available from arXiv: 1503.08663.

29. WANG, Wenguan; LAI, Qiuxia; FU, Huazhu; SHEN, Jianbing; LING, Haibin; YANG, Ruigang. Salient Object Detection in the Deep Learning Era: An In-Depth Survey. *TPAMI*. 2021.

30. ZHANG, Pingping; WANG, Dong; LU, Huchuan; WANG, Hongyu; YIN, Baocai. *Learning Uncertain Convolutional Features for Accurate Saliency Detection*. 2017. Available from arXiv: 1708.02031 [cs.CV].

31. GORODISSKY, Hadar; HARARI, Daniel; ULLMAN, Shimon. Large Field and High Resolution: Detecting Needle in Haystack. *Journal of Vision*. 2018, vol. 18. Available from DOI: 10.1167/18.10.517.

32. LI, Guanbin; YU, Yizhou. *Deep Contrast Learning for Salient Object Detection*. 2016. Available from arXiv: 1603.01976 [cs.CV].

33. LIU, Yi; ZHANG, Qiang; ZHANG, Dingwen; HAN, Jungong. Employing Deep Part-Object Relationships for Salient Object Detection. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 1232–1241. Available from DOI: 10.1109/ICCV.2019.00132.

34. LI, Aoqi; CHEN, Zhenzhong. Personalized Visual Saliency: Individuality Affects Image Perception. *IEEE Access*. 2018, vol. 6, pp. 16099–16109. Available from DOI: 10.1109/ACCESS.2018.2800294.

35. BYLINSKII, Zoya; JUDD, Tilke; OLIVA, Aude; TORRALBA, Antonio; DURAND, Frédo. What do different evaluation metrics tell us about saliency models? *CoRR*. 2016, vol. abs/1604.03605. Available from arXiv: 1604.03605.

36. RICHE, Nicolas; DUVINAGE, Matthieu; MANCAS, Matei; GOSSELIN, Bernard; DUTOIT, Thierry. Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 1153–1160. Available from DOI: 10.1109/ICCV.2013.147.

37. MOROTO, Yuya; MAEDA, Keisuke; OGAWA, Takahiro; HASEYAMA, Miki. Few-Shot Personalized Saliency Prediction Based on Adaptive Image Selection Considering Object and Visual Attention. *Sensors*. 2020, vol. 20, no. 8. ISSN 1424-8220. Available from DOI: 10.3390/s20082170.

38. MOROTO, Yuya; MAEDA, Keisuke; OGAWA, Takahiro; HASEYAMA, Miki. Few-Shot Personalized Saliency Prediction using Person Similarity based on Collaborative Multi-Output Gaussian Process Regression. In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 1469–1473. Available from DOI: 10.1109/ICIP42928.2021.9506583.

39. HOFFER, Tomáš. *Impact of individual human characteristics on visual attention*. 2021. Master's Thesis. Institute of Informatics and Software Engineering, FIIT STU, Bratislava. Under the supervision of Miroslav Laco.

40. KÜMMERER, Matthias; WALLIS, Thomas S. A.; BETHGE, Matthias. *DeepGaze II: Reading fixations from deep features trained on object recognition*. arXiv, 2016. Available from DOI: 10.48550/ARXIV.1610.01563.

Appendix A

Resumé

A.1 Úvod

Modelovanie vizuálnej pozornosti má za cieľ napodobniť komplexné psychologické fungovanie ľudského mozgu a identifikovať oblasti alebo objekty, ktoré pútaju ľudský zrak a vizuálnu pozornosť. Ľudská vizuálna pozornosť je subjektívna a zaujatá na základe osobnosti diváka a jeho preferencií.

A.1.1 Motivácia

Pri modelovaní personalizovanej vizuálnej pozornosti je dôležité minimalizovať množstvo údajov potrebných na trénovanie. Je to preto, že zobrazenie stoviek obrázkov môže byť pre jedného diváka náročné, najmä pre klinických pacientov s kognitívnymi poruchami.

A.1.2 Ciele

Navrhovaný model by sa dal použiť na generovanie umelých fixačných máp, ktoré by sa použili ako fixačné mapy na trénovanie iných modelov a rozširovanie datasetov.

A.2 Zrak a pozornosť

V tejto kapitole sme stručne popísali anatómiu ľudského oka a princípy ľudskej pozornosti. Aby bolo možné predvídať pozornosť jednotlivca, je potrebné vedieť a pochopiť, ako jeho mozog spracováva vizuálne podnety.

A.2.1 Ľudský zrak

Spracovanie informácií až po primárny vizuálny kortex sa nazýva ako nižšia úroveň spracovania, zatiaľ čo za primárnou vizuálnou kôrou sa nazýva vyššia úrovni spracovania, hlavne preto, že zahŕňa kognitívne mechanizmy vysokého rádu, ako napríklad rozpoznávanie objektov.

A.2.2 Pohyby očí

Keď obe oči menia svoj smer z jedného vizuálneho podnetu k inému, tento pohyb sa nazýva sakáda. Vizuálny systém počas sakád nespracovává informácie. To sa deje počas fixácií, ktoré trvajú oproti sakádam niekoľkokrát dlhšie.

A.2.3 Vizuálna pozornosť

Pri spracovávaní informácií, objekty v zornom poli súťažia o našu pozornosť. Výber objektov na spracovanie je na základe dvoch typov spracovania: zdola nahor, a

zhora nadol. Spracovanie zdola nahor sa automaticky zameriava na najvýraznejšie objekty, no spracovanie zdola nahor je ovplyvniteľné.

A.2.4 Zachytenie vizuálnej pozornosti

V tejto práci sa zameriame na technológie sledovania očí na diaľku, pretože ide o najpoužívanejšiu technológiu na vytváranie datasetov o vizuálnej pozornosti.

A.2.4.1 Sledovanie očí

Vzdialený sledovač očí využíva kamery s vysokým rozlíšením a projektory blízkeho infračerveného svetla na zaznamenanie smeru, akým sa odráža od rohovky.

A.2.4.2 Určenie pozície pohľadu

Algoritmy strojového učenia sa používajú na poskytovanie vysoko presného sledovania pohľadu pozorovateľa.

A.3 Modelovanie vizuálnej pozornosti

Takmer všetky boli prístupy na modelovanie pozornosti boli inšpirované ľudským vizuálnym systémom. V tejto kapitole sme sa zaoberali najvplyvnejšími modelmi a datasetmi.

A.3.1 Datasety

MIT300 a CAT2000 sú jedny z najpopulárnejších datasetov pre modelovanie vizuálnej pozornosti. Personalized saliency dataset je podľa našich vedomostí najväčším datasetom venovaným modelovaniu personalizovanej pozornosti.

A.3.2 Obdobie pred hlbokým učením v modelovaní vizuálnej pozornosti

V tejto sekcií sme opísali modely vizuálnej pozornosti podľa mechanizmov akým sa učia. Väčšinou využívajú rozklad vstupného obrazu do samostatných kanálov, po ktorých nasledujú ďalšie transformácie na vytvorenie mapy význačnosti.

A.3.3 Modely vizuálnej pozornosti založené na hlbokom učení

Konvolučné neurónové siete boli široko používané v mnohých oblastiach počítačového videnia. Delia sa na dve podkategórie, Klasické konvolučné siete a Plne konvolučné siete.

A.3.3.1 Klasické konvolučné siete

Na detekciu význačnosti využívajú viacvrstvové perceptróny a segmentáciu vstupného obrazu do malých oblastí. Po tomto procese však nemožno pôvodné priestorové informácie zachovať.

A.3.3.2 Plne konvolučné siete

Najnovšie modely implementovali architektúru plne konvolučných sietí a stali sa lepšími v oblasti predikcie význačnosti. Existuje mnoho typov architektúr, ako sú single-stream, multi-stream, side-fusion, rozvetvené alebo kapsulové siete.

A.3.4 Personalizované modelovanie pozornosti

Ľudská vizuálna pozornosť je individuálna a môže byť ovplyvnená pohlavím pozorovateľa, vekom, preferenciami alebo inými faktormi. Výskum ukazuje, že rysy na vysokej úrovni, ako sú ľudské tváre, majú väčší vplyv na individualitu ľudskej pozornosti ako rysy na nízkej úrovni.

A.3.5 Hodnotenie výkonu modelov

Existuje niekoľko rôznych metrík na hodnotenie schopnosti modelov predpovedať mapy významnosti. Zatiaľ nie je definované, ktorá metrika je najlepšia, takže výber metriky závisí od toho, čo sa model snaží dosiahnuť.

A.4 Príbuzné práce

Cieľom práce Xu a spol.[34] bolo začleniť preferencie pozorovateľa do modelovania pozornosti navrhovaným dvojprúdovým modelom architektúry FCN s názvom Personalized Attention Network (PANet). Počas evaluácie zistili, že preferencie pozorovateľa mali najväčší vplyv na metriku KL-Divergence.

V práci Moroto a spol.[37] bola navrhnutá nová metóda na zmenšenie veľkosti súboru trénovacích údajov s názvom Adaptive Image Selection (AIS). Porovnali účinnosť navrhovaného riešenia so štyrmi modelmi založenými na USM z benchmarku významnosti MIT a dosiahli najlepšie výsledky v metrikách CC, SIM a KL-div. Vo svojej ďalšej práci zaviedli Collaborative Multi-Output Gaussian Process Regression (CoMOGP) ako konečnú metódu predikcie, čo umožnilo ich modelu vyhnúť sa pretrénovaniu aj na malom množstve údajov.

Model navrhnutý v práci Hoffera[39] využíva generalizovaný model ako základ na zachytenie vlastností spoločných pre všetkých pozorovateľov. Potom trénoval N personalizovaných modelov, jeden pre každého pozorovateľa z datasetu PSD. Ukázal, že jeho model priniesol zlepšenie vo všetkých hodnotených metrikách.

A.5 Návrh riešenia

It is currently possible to generate personalized ground truth maps using neural network models. However, it is necessary to verify how representative this artificial ground truth maps would be and what information we would lose by such approach. We propose to utilize the model narhnutý Hofferom, but without the classification part. The model would be trained on data from the SALICON and PSD datasets. Evaluation of the model would be done on all the metrics discussed in 3.5, except for infogain metric.

V súčasnosti je možné vytvárať personalizované fixačné mapy pomocou modelov neurónových sietí. Treba si však overiť, nakoľko reprezentatívne by tieto umelé fixačné mapy boli a o aké informácie by sme takýmto prístupom prišli. Navrhujeme použiť model navrhnutý Hofferom [39], avšak bez klasifikačnej časti. Model by bol trénovaný na údajoch z datasetov SALICON a PSD. Vyhodnotenie modelu by sa vykonalo na datasete CAT2000 a pomocou všetkých metrík diskutovaných v 3.5, okrem metriky infogain.

A.6 Implementácia

Implementácia beží v Dockeri, preto ho vrelo odporúčame používať, no jednotlivé kroky sa dajú zvládnuť aj bez neho. Na urýchlenie procesu trénovania a evaluácie sa využívajú všetky dostupné zdroje GPU. Poskytli sme rozhranie príkazového riadka na uľahčenie školenia a vyhodnocovania modelov.

A.7 Evaluácia

Po dosiahnutí konečného stavu personalizovaných modelov ich otestujeme oproti trénovacej časti datasetu CAT2000. Výsledné umelé fixačné mapy sme spojili do

generalizovanej a vyhodnotili sme ich oproti fixačným mapám zo súboru údajov CAT2000. Tieto výsledky sme podrobili kvalitatívnej a kvantitatívnej analýze.

A.8 Zhodnotenie

Môžeme dospieť k záveru, že generovanie umelých fixačných máp možno vykonať pomocou personalizovaných modelov. V našich experimentoch sme ukázali, že modely predpovedajú významnosť s dobrou presnosťou aj na nikdy nevidených údajoch. Naše riešenie možno použiť na generovanie generalizovaných alebo personalizovaných datasetov z akýchkoľvek dostupných vstupných obrázkov. Ďalšia práca by mohla byť zameraná na ďalšie zlepšenie výkonnosti predikcie modelu.

Appendix B

User manual

We provide a set of steps to reproduce our results. At first, it is important to have appropriate graphics card drivers installed with CUDA support. Then, all datasets have to be downloaded and file paths in config.py file have to be configured correctly. This is not required since we already provide downloaded datasets. If needed, commands have built-in help texts (see Figure B.1).

```
Usage: run.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  evaluate
  evaluate-categories
  merge-maps
  preprocess-dataset
  show-results
  test
  train
```

Figure B.1: Example help texts in our script

```
# Build docker images
docker build --pull -t python2 -f Dockerfiles\python2 \.
docker build --pull -t python3 -f Dockerfiles\python3 \.
docker build --pull -t python3-tensorflow -f Dockerfiles\python3-tensorflow
\.

# Prepare both datasets
python run.py preprocess-dataset PSD
python run.py preprocess-dataset CAT2000

# Train on SALICON dataset
python run.py train SALICON --save-name salicon --model-type generalized

# Train generalized base on PSD generalized fixations
python run.py train PSD --save-name generalized --model-type generalized
--load-name salicon

# Train personalized models on PSD dataset
python run.py train PSD --save-name personalized --model-type personalized
--load-name generalized

# Test and evaluate personalized models against PSD dataset
python run.py test PSD --model-type personalized --load-name personalized

python run.py evaluate PSD --load-name personalized

# Show overall results
```

```
python run.py show-results PSD --load-name personalized

# Test models against CAT2000 dataset
python run.py test CAT2000 --model-type personalized --load-name personalized

# Merge result images
python run.py merge-maps --load-name personalized

# Evaluate the merged images against CAT2000 fixations
python run.py evaluate CAT2000 --load-name personalized --merged
```

Appendix C

Technical documentation

To correctly run our scripts it is needed to have docker installed and running. We tested our work in a Windows 10 environment with WSL2 integration enabled and Docker Desktop. Also python has to be installed on the host machine.

We created a command-line wrapper around helper scripts, model training, testing, and evaluation to eliminate the need for typing exhaustive non-documented commands. The heart of the interface lies within piece of code that runs all of the formatted commands in docker (see Figure C.1). It utilises a python library called Docker SDK for Python to work with the docker api, and library click to parse input commands and generate proper help texts. Each command is run on all available GPU resources by default, this can only be overriden by changing the source code.

All scripts are tailored to run only on PSD and CAT2000 datasets. Model training can also be done on the SALICON dataset, but only training. Each dataset class needs to have configured hard-coded paths to appropriate files. Example configuration can be seen in Figure C.2, where all observer names and file paths are

```python
def _run_in_docker(image: str, command: str, args: str):
    instance = DOCKER_CLIENT.containers.run(image,
                                command=command + args,
                                volumes=[DOCKER_VOLUME],
                                environment=[DOCKER_ENV],
                                remove=True,
                                detach=True,
                                runtime="nvidia",
                                device_requests=[docker.types.DeviceRequest(count=-1, capabilities=[['gpu']])],
                                )

    output = instance.attach(stdout=True, stream=True)

    for line in output:
        print(line.decode("utf-8"), end="", flush=True)
```

Figure C.1: Piece of code that runs commands in docker images.

defined. It is important to note, that all paths must be relative to the directory where the run and config scripts are located.

```python
class PSD(DATASET):
    name = "psd"
    observers = ["Sub_1", "Sub_2", "Sub_3", "Sub_4", "Sub_5", "Sub_6", "Sub_7", "Sub_8", "Sub_9", "Sub_10",
                 "Sub_11", "Sub_12", "Sub_13", "Sub_14", "Sub_15", "Sub_16", "Sub_17", "Sub_18", "Sub_19", "Sub_20",
                 "Sub_21", "Sub_22", "Sub_23", "Sub_24", "Sub_25", "Sub_26", "Sub_27", "Sub_28", "Sub_29", "Sub_30"]

    def __init__(self):
        super().__init__()

        self.fixations = Path("data/psd/fixations")
        self.raw_fixations = Path("data/psd/raw")
        self.binary_fixations = Path("data/psd/binary")
        self.generalized_fixations = Path("data/psd/generalized")
        self.stimuli = Path("data/psd/images")
        self.test_set = Path("data/psd/test")

        self.ensureconfig()
```

Figure C.2: Example configuration of a dataset class in our config.py file.

It should be possible to add a new dataset class if there was a need to work with other datasets, but there would need to be work done even in run.py script.

Appendix D

Work plan

Week No.	Activity
1-2	Design of work content structure
3	Write the Introduction chapter
4-6	Write the Human Vision chapter
7-9	Write the Visual attention modelling chapter
9-11	Write the Related Work chapter
11-13	Solution proposal and Conclusion, also editing of the first pages, including assignment and annotation

Table D.1: Work done first semester.

Week No.	Activity
1-2	Implement the proposed solution solution
3	Try to improve performance of the model, write Implementation chapter
4-6	Generate artificial dataset, try augmentations
7-9	Evaluate model performance, write evaluation
9-11	Create command line wrapper
11-13	Write the Conclusion chapter and finish up the Thesis

Table D.2: Work done second semester.

Appendix E

CD-ROM

All files were uploaded to google classroom. In the following list, we have listed only files and folders that might be of interest, and ones that are not generated.

- `data/` — folder for datasets
  - `dataset_name/` — each structure for a dataset is the same except for small details
    - `binary/` — generated binary fixation maps
    - `fixations/` — fixation maps
    - `images/` — stimuli images
    - `raw/` — unprocessed fixation files
  - `cat2000/` — cat2000 dataset
    - `trainSet/` — directory from the original cat2000 dataset
  - `PSD/` — personalized saliency dataset
    - `generalized/` — generated generalized fixation maps

        * `test/` — directory containing test image set

- `Dockerfiles/` — dockerfiles for docker images building
- `encoder-decoder-model/` — folder containing model
  - `results/` — serialized models
  - `weights/` — pretrained weights
- `src/` — helper scripts
- `test-results/` — directory for generated images and evaluation files
- `venv/` — python virtual environment with installed libraries
- `.gitignore` — github ignore file
- `config.py` — configuration file, where dataset classes are stored
- `README.md` — github readme file
- `requirements.txt` — docker python2 requirements
- `requirements-python3.txt` — docker python3 requirements
- `run.py` — CLI wrapper, main script