symmetry

MDPI

Article

Deep Convolutional Symmetric Encoder—Decoder Neural Networks to Predict Students' Visual Attention

Tomasz Hachaj *, Anna Stolińska, Magdalena Andrzejewska and Piotr Czerski

Institute of Computer Science, Pedagogical University of Krakow, 2 Podchorazych Ave, 30-084 Krakow, Poland; anna.stolinska@up.krakow.pl (A.S.); magdalena.andrzejewska@up.krakow.pl (M.A.); piotr.czerski@up.krakow.pl (P.C.)

* Correspondence: tomekhachaj@o2.pl; Tel.: +48-126-627-845

Abstract: Prediction of visual attention is a new and challenging subject, and to the best of our knowledge, there are not many pieces of research devoted to the anticipation of students' cognition when solving tests. The aim of this paper is to propose, implement, and evaluate a machine learning method that is capable of predicting saliency maps of students who participate in a learning task in the form of quizzes based on quiz questionnaire images. Our proposal utilizes several deep encoder–decoder symmetric schemas which are trained on a large set of saliency maps generated with eye tracking technology. Eye tracking data were acquired from students, who solved various tasks in the sciences and natural sciences (computer science, mathematics, physics, and biology). The proposed deep convolutional encoder–decoder network is capable of producing accurate predictions of students' visual attention when solving quizzes. Our evaluation showed that predictions are moderately positively correlated with actual data with a coefficient of 0.547 ± 0.109. It achieved better results in terms of correlation with real saliency maps than state-of-the-art methods. Visual analyses of the saliency maps obtained also correspond with our experience and expectations in this field. Both source codes and data from our research can be downloaded in order to reproduce our results.

Keywords: cognition; deep learning; convolutional network; encoder–decoder; visual attention; saliency map; solving quizzes

Citation: Hachaj, T.; Stolińska, A.; Andrzejewska, M.; Czerski, P. Deep Convolutional Symmetric Encoder—Decoder Neural Networks to Predict Students' Visual Attention. *Symmetry* **2021**, *13*, 2246. https://doi.org/10.3390/sym13122246

Academic Editors: Pecchinenda Anna and Vilfredo De Pascalis

Received: 10 October 2021
Accepted: 17 November 2021
Published: 25 November 2021

1. Introduction

The latest advances in machine learning have allowed the creation of data-driven deep learning models which train on properly preprocessed eye tracking data and can be applied to simulate and predict a person's cognition when solving particular tasks.

1.1. State of the Art

Eye Tracking Data as Indicators of Cognitive Processes

Eye tracking tools, which are increasingly common in experimental laboratories, provide rich and precise data on the mechanisms and time course of cognitive processing. Eye tracking measurements generate a dataset that provides insights into cognitive processes that cannot be obtained from other behavioral measurements [1].

Analysis of cognitive processing relies primarily on data from eye movements, which are, in some simplistic terms, a composite of short pauses called fixations and rapid eye movements, called saccades, during which suppression of visual information occurs. Although fixations are moments of increased (effective) exploration of the presented visual scene (stimuli), saccades are also important for the analysis of cognitive processing. They provide a measure of information selection important in the process of visual search [2]. Gaze provides the data necessary for a decision-making system, hence the analysis of fixations and saccades provides information about the mechanism by which humans make decisions [3,4]. Eye tracking has led to the conclusion that people, when deciding between different options, are more likely to make their first fixation on the option they will go

on to choose [5]. Analyzing eye tracking data also provides insight into, for example, the strategies taken when performing computations. The authors of [6] have shown the utility of eye movement data to examine the spatial–temporal distribution of attention when solving arithmetic problems using different strategies and age-related differences in these strategies and distributions. Eye tracking indices are also proving to be useful in analyses of cognitive processes such as memorization (e.g., [7]), or those of the highest complexity, such as problem solving [8], cognitive reasoning [9], and the learning process [10]. Another issue, of particular interest to us, has been addressed by [11], who used eye movement trackers to detect where learners' attention is focused, so that the system is able to provide prompts when learners' eye movements suggest that they have encountered difficulties. It has been hypothesized that individuals who are able to solve a problem successfully tend to focus their visual attention on the most salient information needed to solve the problem [12]. This hypothesis was confirmed in the study [13]. Eye tracking scaffolding in the form of introducing an immediate feedback mechanism has also been shown to have a significant impact on promoting students' sense of self-efficacy when learning C programming [11]. The eye tracking technique thus contributes to the understanding of cognitive processes and allows attempts to simulate them, providing valuable input from behavioral measurements for machine learning.

1.2. Eye Tracking-Based Cognition Analysis

Reliable knowledge about cognitive processes can only be gained through a combination of objective data (as we consider eye tracking indicators to be) and parallel survey research. Publication limitations did not allow a detailed description of the research procedure that made it possible to obtain the data, but it is the norm that eye tracking research is accompanied by in-depth interviews and tests (this was also the case in our study). As eye tracking is an unobtrusive method to gain deeper understanding of cognitive processes, such as problem solving and decision making and by measuring eye movements, researchers gain insight into the ongoing mental processes during tasks, write, among others: Bueno et al. [14], Ke et al. [15], Kiefer et al. [16], Semmelmann and Weigelt [17], and Aslin and McMurray [18]. In summary, it can be said that the view is well established that knowing where people look at a given point in time can thus provide crucial insights into perceptual and cognitive development.

There have already been successful attempts at solving various problems related to eye tracking-based cognition analysis using modern approaches with machine learning [19]. Many methods of that type are devoted to classification or approximation. In [20], the authors proposed a machine learning-based method that is capable of automatically detecting task demand using eye movements. The solution utilizes eye tracking-based features, linear and kernel-based support vector machine (SVM), and random forests. The authors of [21] attempt to classify eye tracking data using a convolutional neural network (CNN). With the aid of carefully designed CNN-based features, the extractor authors were able to classify two different web interfaces for browsing news data. Then, in a second experiment, a CNN was used to classify the nationalities of users. Chen et al. [22] propose a method to recognize strabismus by applying a neural network-based approach on eye tracking data. That approach utilizes a so-called gaze deviation (GaDe) image which is then classified by the network pretrained on an ImageNet dataset. Dalrymple et al. [23] use machine learning to perform infants' age recognition using age-related variability in gaze patterns. Lee et al. [24] classify programmer expertise and difficulty of the task using eye tracking signals. The classification was based on SVM.

There is also research that demonstrates the prognostic and generative abilities of deep neural networks (DNNs) in the spatial domain. The application on convolutional neural networks for visual attention prediction has been reported by Louedec et al. [25] and Wang et al. [26]. In [25], a deep neural network is applied in a chess game, while in [26], visual attention is predicted on plain photos.

Among many applications of eye tracking and machine learning, the authors of [27] propose to use these approaches to enhance motivation and learning. The proposed method can supply students with gaze-aware feedback by using machine learning for the prediction of student performance from their behavior. Eye tracking together with machine learning (ML) provides technical support for learning path research [28,29], analysis of collaborative problem solving, and cognitive and social–emotional learning skills [30]. In [31,32], the authors show that people can improve their decision and learning skills by examining eye tracking data that visualize what experts paid attention to when solving a certain problem. Another important applications of ML and eye tracking is evaluation of students' engagement [33,34], performance [35,36], and learning style classification using eye tracking data [37]. An additional broad survey on the application of ML and eye tracking in science learning activities can be found in [38,39].

1.3. Motivation of This Paper

As can be seen from the previous section, the prediction of visual attention is a new and challenging subject, and to the best of our knowledge, it has not yet been successfully applied to anticipate students' cognition when solving tests, so our research is among the first of this type. In contrast to other researchers, we do not focus on using machine learning methods to classify eye tracking results, however, our goal is to make predictions of visual attention of students. Due to this fact, the aim of this study is to propose, implement, and evaluate a machine learning method that is capable of predicting saliency maps of students who participate in learning tasks in the form of quizzes based on quiz questionnaire images. Our proposal utilizes several deep encoder–decoder symmetric schemas which are trained on a large set of saliency maps generated with eye tracking technology. We also compare our results with state-of-the-art approaches for visual attention prediction, however, in contrast to previously published methods that are typically trained on eye tracking data collected from viewing a variety of photographs, our eye tracking data have been acquired from students who solved various tasks in the sciences and natural sciences (computer science, mathematics, physics, biology). The basic application of the proposed method is to enable one to predict the cognition process of students when solving particular quiz-based tasks, for example, in order to anticipate which parts of the document will focus the reader's attention and will generate more fixations. This is an important factor that has to be taken into account in the process of correct test design.

2. Material and Methods

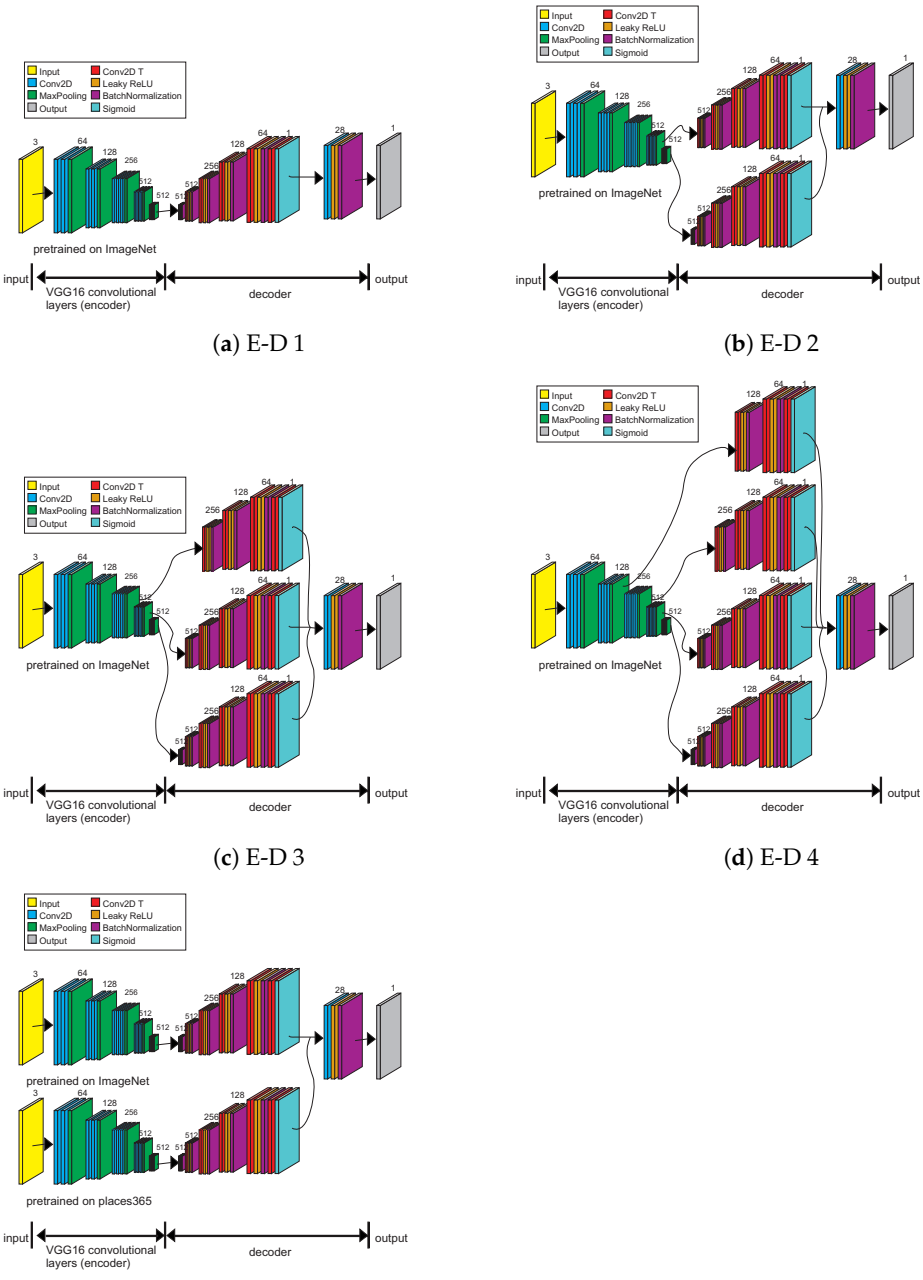
In this section, we will present the methodology that we have developed in order to generate a prediction of students' visual attention and the datasets we have used in the evaluation of the proposed methodology.

2.1. Network Architectures

Encoder–decoder neural network architectures typically have a topology of two vertex-connected pyramids with a symmetric number of neurons on corresponding levels of those pyramids. The first pyramid acts as an encoder whose task is to reduce the dimensionality of the output image and generate image embedding. The second pyramid has the task of converting the image embedding to the resultant image. If the resulting image is to have the same modality as the output image, the network is called an autoencoder. If the output image is of different modality than the output image, the network is called a hetero-encoder–decoder. Decoding is most often implemented by the deconvolution layers of the network, also called transposed deconvolution (Conv2D T) [40].

In the case of an encoder–decoder network modeling a saliency map, we are dealing with a hetero-encoder–decoder because the input to the network is an RBG image, while the saliency map is a grayscale image in which the brightness value of a pixel is proportional to the time spent focusing on a given portion of the image.

Efficient encoder–decoders that utilized the embedding method have been successfully implemented using pretrained convolutional networks, for example, VGG16 [41]. The basic architecture of such a hetero-encoder–decoder using a single deconvolution pyramid (E-D 1) is shown in Figure 1a. In practice, more sophisticated structures are also used that utilize several deconvolution pyramids that take embedding from several convolutional layers and perform deconvolution for each of these embeddings separately. An example of such a multi-stage encoder–decoder is the network presented in the paper [26] and shown in Figure 1c.



(**a**) E-D 1 (**b**) E-D 2

(**c**) E-D 3 (**d**) E-D 4

(**e**) 2xE-D 1

Figure 1. Various DNN encoder–decoder architectures that use VGG16 convolutional layers. Numbers above the input and output layer indicate the dimensionality of images. Numbers above Conv2D and Conv2D T indicate the number of filters. Conv2D is a two-dimensional convolutional layer and Conv2D T is a transposed two-dimensional convolutional layer.

There are some details that are not presented in that paper, however, they were proposed in its implementation by Wenguan Wang and Jianbing Shen CAFFE and later in

Suraj Maniyar's Keras/Tensorflow 1.X implementation of [26], namely thresholding the grayscale value levels of the original saliency map.

Our proposition of encoder–decoder networks for students' visual attention prediction is based on previous work of Wang and Shen [26] which is an efficient and effective state-of-the-art network. The main idea in [26] was an application of a deep encoder–decoder network with a VGG16-based encoder to generate prediction of visual attention. That encoder utilizes VGG16 convolution layers pretrained on the ImageNet dataset [42]. Pretrained layers have a role as multi-scale feature extractors.

The model uses three separate decoders that are basically stacks of transposed convolutional layers that recover the original resolution of the input image. The first encoder takes data from the last, fifth layer of VGG16, the second from the fourth, and the third from the third layer of VGG16. After this step, each output of the decoder is concatenated into a single tensor, which is then filtered by a single convolution layer. The loss function is defined as the averaged value of binary cross entropy loss between the predicted and ground truth value:

$$loss(G,P) = \frac{BCE(G,P_{d1}) + BCE(G,P_{d2}) + BCE(G,P_{d3}) + BCE(G,P_{out})}{4} \quad (1)$$

where $BCE(G,P_{d1})$ is binary cross entropy calculated between ground truth G and value P_{d1} predicted by decoder $d1$. $d1$ is the "deepest" decoder, that is connected to the last convolutional layer. P_{out} is a value returned by the output layer of the whole network. The last layer has a sigmoid activation function that produces a saliency map with the same size as the input image.

Summarizing, each of three decoders produces a separate prediction of visual attention using a different number of convolutional filters taken from encoder layers that are optimized together with the final output of the network.

The main idea behind our research was to examine if various adjustments of architecture [26] will improve the quality of visual attention prediction of eye tracking-based saliency maps. We anticipated that an appropriate combination of multi-level image convolution filtering might result in improvements in image feature selection. Those features might be then used by multi-scale deconvolution layers to reconstruct the desired saliency map.

For our research, we have adjusted the architecture of the network presented in Figure 1c. Each stack of decoders is composed of three layers instead of only a transposed convolutional layer; they are: transposed convolutional layer, leaky ReLu layer, and batch normalization. Application of batch normalization as part of the multi-scale model allows us to use much higher learning rates and to be less careful about initialization [43]. Application of leaky ReLu should limit the so-called dying ReLU problem [44]:

$$f(x) = \begin{cases} \alpha \cdot x; & if \quad x < 0 \\ x; & otherwise \end{cases} \quad (2)$$

where $f(x)$ is an activation function; we have set $\alpha = 0.2$.

Based on the above architecture, we proposed three more encoder–decoder architectures that can be used for generating saliency maps. The key aspect of successful network training is the proper formulation of the loss function. In case of a complex network structure, which has several parallel operating structures, it is necessary to ensure that the parameters of each of them are evenly trained. Therefore, it is advisable to build the loss function in such a way that it minimizes the error in the output layers of each of these parallel structures. This is the approach we used in our work: our loss functions minimize not only the binary cross entropy between ground truth and network output but also the binary cross entropy between ground truth and the output of each of the parallel subdecoder networks. The network in Figure 1b is a poorer version of the network in Figure 1c, which has two decoder pyramids instead of three. The loss function is defined

as the averaged value of binary cross entropy loss between the predicted and ground truth value:

$$loss(G, P) = \frac{BCE(G, P_{d1}) + BCE(G, P_{d2}) + BCE(G, P_{out})}{3} \quad (3)$$

The network in Figure 1d has an additional decoder pyramid that decodes images from earlier encoder layers. The loss function is defined as the averaged value of binary cross entropy loss between the predicted and ground truth value:

$$loss(G, P) = \frac{BCE(G, P_{d1}) + BCE(G, P_{d2}) + BCE(G, P_{d3}) + BCE(G, P_{d4}) + BCE(G, P_{out})}{5} \quad (4)$$

Our last proposed architecture uses two encoders, one pretrained on the ImageNet set and the other pretrained on the Places365 [45] set. This architecture uses two single pyramid decoders analogous to the network presented in Figure 1a. The loss function is defined as the averaged value of binary cross entropy loss between the predicted and ground truth value:

$$loss(G, P) = \frac{BCE(G, P_{d1}) + BCE(G, P_{d1places}) + BCE(G, P_{out})}{3} \quad (5)$$

where value $P_{d1places}$ is predicted by decoder $d1$ from VGG16 pretrained on the Places365 dataset.

We set VGG parameters to be fixed (not optimized during training). We summarize the architectures of all the networks described in this section in Table 1.

Table 1. Summary of the architectures of all the networks described in Section 2.1.

	E-D 1	**E-D 2**	**E-D 3**	**E-D 4**	**2xE-D 1**
Number of encoders	1	1	1	1	2×1
Number of decoders	1	2	3	4	2×1
Total params	20,989,519	24,902,489	25,863,779	26,086,509	41,979,033
Trainable params	6,271,885	10,182,935	11,143,329	11,365,675	12,543,767
Non-trainable params	14,717,634	14,719,554	14,720,450	14,720,834	29,435,266
Loss function	$BCE(G, P_{out})$	(3)	(1)	(4)	(5)

2.2. Dataset

In this research, we have used two datasets:

- our (primary) dataset, that contains stimuli and saliency maps acquired from students that were solving quizzes. Our goal in this research was to maximize the effectiveness of our method on this dataset.
- a second, third-party SALICON dataset [46] that was used in order to extend the training dataset and to improve the final method's performance. Stimuli from the second dataset, and the saliency maps that correspond to them, are not directly connected to education and no data from second dataset were used during the validation step (only during training).

The primary dataset we used in this research consisted of 487 saliency maps generated from eye tracking data. An eye tracker from SensoMotoric Instruments, iViewX™ Hi-Rate500/1250, which recorded a data stream at a 500 Hz time resolution, was used in the investigation. A comfortable interface construction in this system holds the subject's head stable and motionless without limiting the field of vision of the subject. During the experiment, the images were presented on a diagonal LCD screen with full HD resolution of 1920×1080. Before each test, a 9-point calibration with validation was performed. All individuals were subjected to similar ambient conditions such as temperature, lighting, and acoustic insulation. The experiment was performed with the SMI Experiment Suite™ 360 software. The raw eye tracking data were generated using SMI BeGaze™ 2.4.

The data were acquired from 52 students. The group was composed of 25 girls and 27 boys. The mean age of the participants was 16 years. Because of technical issues, data of 3 persons were removed and further analysis was performed on the remaining 49. Tasks that were solved by participants were from the field of natural sciences: mathematics, computer science, biology, and physics. Each test was performed individually and consisted of 14 tasks (stimuli). Each task required an analysis of the content of the instruction and an appropriate illustration. The time for task solving was not limited. The students evaluated the level of difficulty of the task being solved by answering a questionnaire, in which they gave an answer to the following question: Evaluate on a scale from 0 to 10 (where 0—very easy, 10—very difficult) to what extent the tasks you solved were easy/difficult. The task's difficulty level was measured using Likert's scale. The correlation between the mean evaluation of the difficulty level of the whole task set and the percentage of the correct answers provided by the students was moderately strong, $r = -0.40$, and statistically significant ($p < 0.05$). The lowest percentage of correct answers was provided for the task in mathematics, 25%, and a very close value was provided for the algorithmic task, 27%. In the group of difficult tasks, there was also a second mathematical task, 38%. The highest percentage of correct answers was found for the biology task, 76%, and the second algorithmic task, 63%. Saliency maps have been grouped by stimulus.

Due to output data quality issues and the quiz procedure (some stimuli were presented to only half of the participants), the number of maps for each stimulus differs.

The third-party SALICON dataset was composed of 10,000 images. In this dataset, eye fixation annotations were simulated by the mouse movements. It has been showed by Jiang et al. [46] that this type of simulation strongly correlates with actual eye tracker evaluation. The same dataset has been also used by Wenguan and Shen in paper [26]. We used the SALICON dataset to augment our primary dataset with additional examples of saliency maps to increase the stability of the weight optimization process during network training. The SALICON dataset was used only during training. Each time, we mixed it with subsets of our primary dataset that were used in training. Elements of the SALICON dataset were not involved in the validation process.

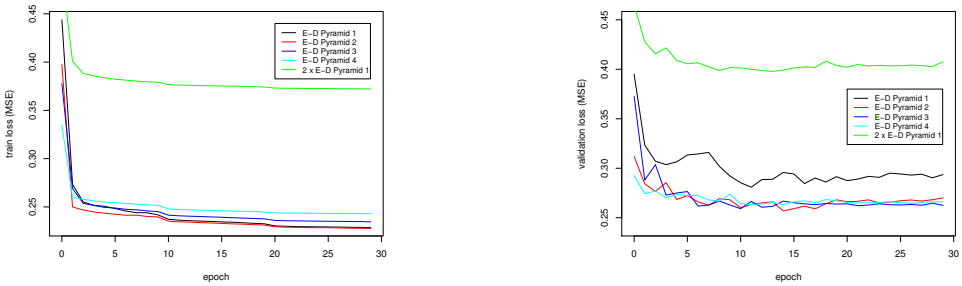
We created a training dataset that contained all but one of the stimuli and the saliency maps and validation dataset corresponding to those from the remaining data. We enlarged the training dataset by 10,000 stimuli and the saliency maps take data from the SALICON dataset. Then, we performed a 14-fold cross validation in which each validation was performed on a different stimulus from our primary dataset.

3. Results

The goal of our experiment was to test which of the network architectures presented in Section 2.1 can be trained to effectively predict saliency maps on the primary dataset from Section 2.2. We also performed a comparison of the results obtained from each network to select the best architecture to predict students' visual attention. Our implementation and dataset can be downloaded from GitHub (https://github.com/browarsoftware/DeepVisualAttentionPrediction/ accessed on 10 October 2021) and our results can be reproduced.

We implemented all networks described in Section 2.1 in Python 3.6. Among the most important libraries, we used Keras 2.4.3 with Tensorflow 2.3.0 for deep neural network modeling and calculation and opencv-python 4.2.0.32 for general purpose image processing. A pretrained model of VGG16 was downloaded with the Keras-Applications 1.0.8 package. Network training and evaluation was carried out on a PC equipped with Intel i7-9700 3 GHz, 64 GB RAM, and an NVIDIA GeForce RTX 2060 GPU on Windows 10 OS. For the optimization of network weights, we used the stochastic gradient descent Adam optimizer [47]. The learning rate was set to 0.01. Plots of train loss and validation loss for a single evaluation are presented in Figure 2a,b.

In Tables 2 and 3, we present detailed values of train loss (MSE) and validation loss (MSE) obtained after 10 epochs of DNN training.



(**a**) Train loss (**b**) Validation loss

Figure 2. Plots of the train loss and validation loss for each evaluated encoder–decoder.

Table 2. Values of train loss (MSE) obtained after 10 epochs of DNN training. The last row contains averaged values of all subvalidations.

Stimulus	E-D 1	E-D 2	E-D 3 [26]	E-D 4	2xE-D 1
S1	0.242	0.24	0.245	0.252	0.379
S2	0.243	0.239	0.246	0.253	0.379
S3	0.242	0.239	0.245	0.252	0.379
S4	0.32	0.24	0.246	0.252	0.379
S5	0.32	0.24	0.246	0.252	0.379
S6	0.242	0.239	0.245	0.252	0.379
S7	0.248	0.238	0.246	0.251	0.379
S8	0.243	0.242	0.245	0.252	0.378
S9	0.245	0.238	0.245	0.252	0.379
S10	0.242	0.239	0.245	0.253	0.378
S11	0.32	0.24	0.246	0.253	0.379
S12	0.32	0.239	0.246	0.252	0.378
S13	0.243	0.24	0.244	0.253	0.379
S14	0.243	0.239	0.245	0.252	0.379
Average	0.265 ± 0.036	0.239 ± 0.001	0.245 ± 0.001	0.252 ± 0.001	0.379 ± 0.000

Table 3. Values of validation loss (MSE) obtained after 10 epochs of DNN training. The last row contains averaged values of all subvalidations.

Stimulus	E-D 1	E-D 2	E-D 3 [26]	E-D 4	2xE-D 1
S1	0.292	0.268	0.263	0.274	0.402
S2	0.255	0.224	0.235	0.231	0.38
S3	0.323	0.3	0.3	0.298	0.42
S4	0.426	0.314	0.305	0.318	0.435
S5	0.464	0.328	0.328	0.328	0.443
S6	0.325	0.297	0.307	0.302	0.423
S7	0.359	0.337	0.341	0.362	0.464
S8	0.203	0.213	0.212	0.22	0.358
S9	0.271	0.232	0.246	0.237	0.375
S10	0.13	0.148	0.147	0.156	0.322
S11	0.386	0.292	0.291	0.306	0.416
S12	0.316	0.253	0.25	0.255	0.41
S13	0.277	0.275	0.246	0.241	0.397
S14	0.344	0.32	0.315	0.316	0.454
Average	0.312 ± 0.086	0.272 ± 0.053	0.270 ± 0.053	0.275 ± 0.054	0.407 ± 0.039

Figure 2a,b present train loss and validation loss for a single evaluation of 14-fold cross validation. In Tables 2 and 3, we present detailed values of train loss (MSE) and validation loss (MSE) obtained after 10 epochs of DNN training for each 14-fold cross validation. We

use graphs to show how values of train and validation loss change during the training while the tables are used to show our final result.

After the training of each of the five DNN encoder–decoders, a stimulus from the validation dataset was used to generate a prediction of the saliency map. This map was compared with data on the real saliency maps from our dataset using the correlation coefficient. The average value plus/minus standard deviation for each stimulus is presented in Table 4. The exact numbers of saliency maps for each stimulus are presented in the #Validation column.

Table 4. Values of correlation coefficients between network predictions and real salicency maps obtained after 10 epochs of DNN training. The last row contains averaged values of all subvalidations.

Stimulus	E-D 1	E-D 2	E-D 3 [26]	E-D 4	2xE-D 1	#Validation
S1	0.569 ± 0.067	0.663 ± 0.072	0.656 ± 0.075	0.688 ± 0.072	0.601 ± 0.071	24
S2	0.556 ± 0.109	0.661 ± 0.118	0.637 ± 0.123	0.664 ± 0.116	0.592 ± 0.104	26
S3	0.497 ± 0.072	0.596 ± 0.079	0.609 ± 0.084	0.61 ± 0.081	0.571 ± 0.076	24
S4	-0.051 ± 0.01	0.115 ± 0.032	0.601 ± 0.073	0.608 ± 0.071	0.528 ± 0.059	25
S5	-0.047 ± 0.011	0.152 ± 0.046	0.565 ± 0.136	0.571 ± 0.136	0.537 ± 0.086	47
S6	0.382 ± 0.103	0.107 ± 0.046	0.472 ± 0.137	0.502 ± 0.142	0.394 ± 0.12	49
S7	0.246 ± 0.087	0.498 ± 0.121	0.427 ± 0.111	0.432 ± 0.112	0.315 ± 0.091	49
S8	0.653 ± 0.124	0.662 ± 0.116	0.663 ± 0.125	0.653 ± 0.117	0.66 ± 0.123	18
S9	-0.03 ± 0.032	0.684 ± 0.126	0.652 ± 0.129	0.66 ± 0.12	0.682 ± 0.131	24
S10	0.233 ± 0.153	0.181 ± 0.141	0.234 ± 0.131	0.325 ± 0.146	0.305 ± 0.117	13
S11	-0.047 ± 0.012	0.54 ± 0.112	0.53 ± 0.107	0.529 ± 0.11	0.476 ± 0.101	49
S12	-0.034 ± 0.013	0.538 ± 0.088	0.537 ± 0.086	0.546 ± 0.091	0.467 ± 0.076	49
S13	0.387 ± 0.067	0.483 ± 0.085	0.481 ± 0.082	0.49 ± 0.08	0.372 ± 0.071	41
S14	0.244 ± 0.107	0.355 ± 0.129	0.37 ± 0.131	0.372 ± 0.133	0.305 ± 0.117	49
Average	0.254 ± 0.251	0.445 ± 0.212	0.531 ± 0.120	0.547 ± 0.109	0.486 ± 0.126	NA

In order to perform additional quantitative analysis of the eye tracking data between real and artificial gaze results, we calculated the structural similar index measure (SSIM) between them. This measure compares local patterns of pixel intensities that have been normalized for luminance and contrast [48]. The SSIM value varies from -1 to 1, where 1 means a perfect structural similarity. The SSIM is designed to measure image quality degradation that might be caused, for example, by compression or data transmission losses. The average value of the SSIM plus/minus standard deviation for each stimulus is presented in Table 5.

Table 5. Values of structural similarity index (SSIM) between network predictions and real salicency maps obtained after 10 epochs of DNN training.

Stimulus	E-D 1	E-D 2	E-D 3 [26]	E-D 4	2xE-D 1
S1	0.337 ± 0.015	0.396 ± 0.019	0.376 ± 0.018	0.368 ± 0.017	0.372 ± 0.015
S2	0.130 ± 0.018	0.324 ± 0.026	0.199 ± 0.024	0.266 ± 0.028	0.191 ± 0.015
S3	0.148 ± 0.010	0.164 ± 0.015	0.369 ± 0.024	0.239 ± 0.015	0.107 ± 0.011
S4	0.041 ± 0.005	0.278 ± 0.019	0.267 ± 0.019	0.310 ± 0.01	0.293 ± 0.012
S5	0.041 ± 0.006	0.183 ± 0.025	0.290 ± 0.024	0.231 ± 0.025	0.257 ± 0.018
S6	0.197 ± 0.013	0.290 ± 0.026	0.192 ± 0.021	0.200 ± 0.016	0.215 ± 0.014
S7	0.127 ± 0.009	0.231 ± 0.017	0.266 ± 0.016	0.334 ± 0.018	0.274 ± 0.011
S8	0.310 ± 0.016	0.346 ± 0.022	0.150 ± 0.016	0.220 ± 0.018	0.429 ± 0.025
S9	0.089 ± 0.014	0.301 ± 0.025	0.256 ± 0.021	0.442 ± 0.034	0.380 ± 0.026
S10	0.396 ± 0.008	0.579 ± 0.008	0.612 ± 0.007	0.415 ± 0.006	0.249 ± 0.007
S11	0.034 ± 0.005	0.311 ± 0.016	0.287 ± 0.014	0.323 ± 0.010	0.250 ± 0.010
S12	0.038 ± 0.008	0.154 ± 0.018	0.410 ± 0.022	0.457 ± 0.019	0.093 ± 0.011
S13	0.224 ± 0.009	0.303 ± 0.017	0.289 ± 0.013	0.351 ± 0.014	0.226 ± 0.009
S14	0.053 ± 0.011	0.224 ± 0.017	0.267 ± 0.019	0.284 ± 0.019	0.241 ± 0.014
Average	0.155 ± 0.117	0.292 ± 0.105	0.302 ± 0.110	0.313 ± 0.082	0.256 ± 0.091

Figure 3 presents a comparison of predictions generated by all networks from Section 2.1 on an example input image compared with an example real data saliency map. Figure 4 presents a stimulus (in a blue frame), four examples of real data saliency maps, and a saliency map predicted by our approach (in a green frame).

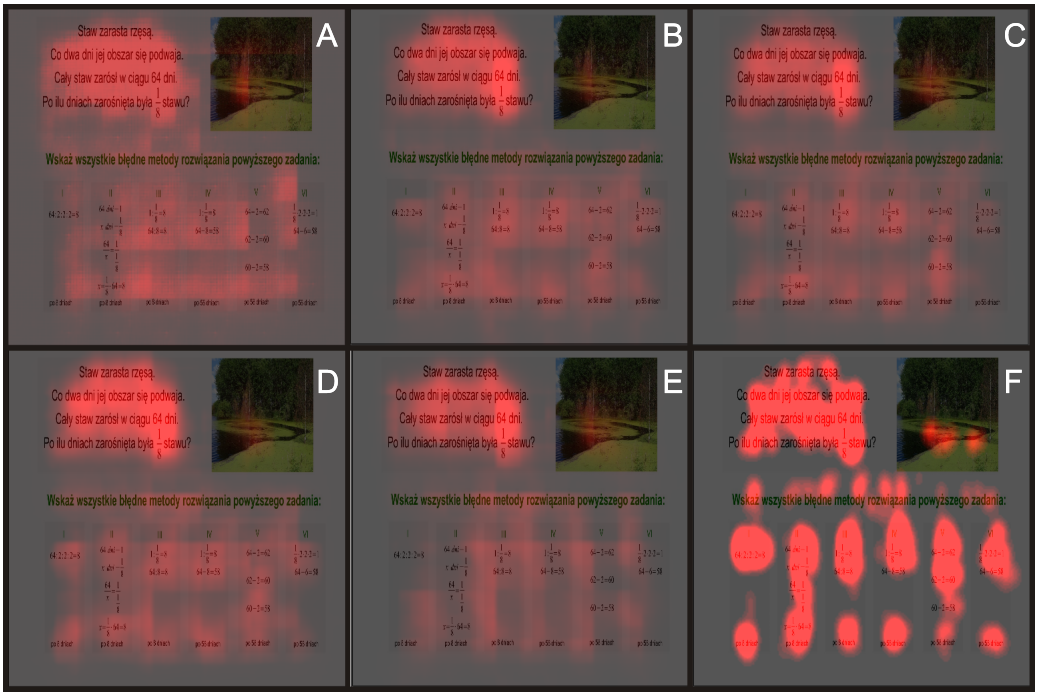


Figure 3. Predictions from (**A**) E-D 1; (**B**) E-D 2; (**C**) E-D 3; (**D**) E-D 4; (**E**) 2xE-D 1; (**F**) examples of real data saliency map.

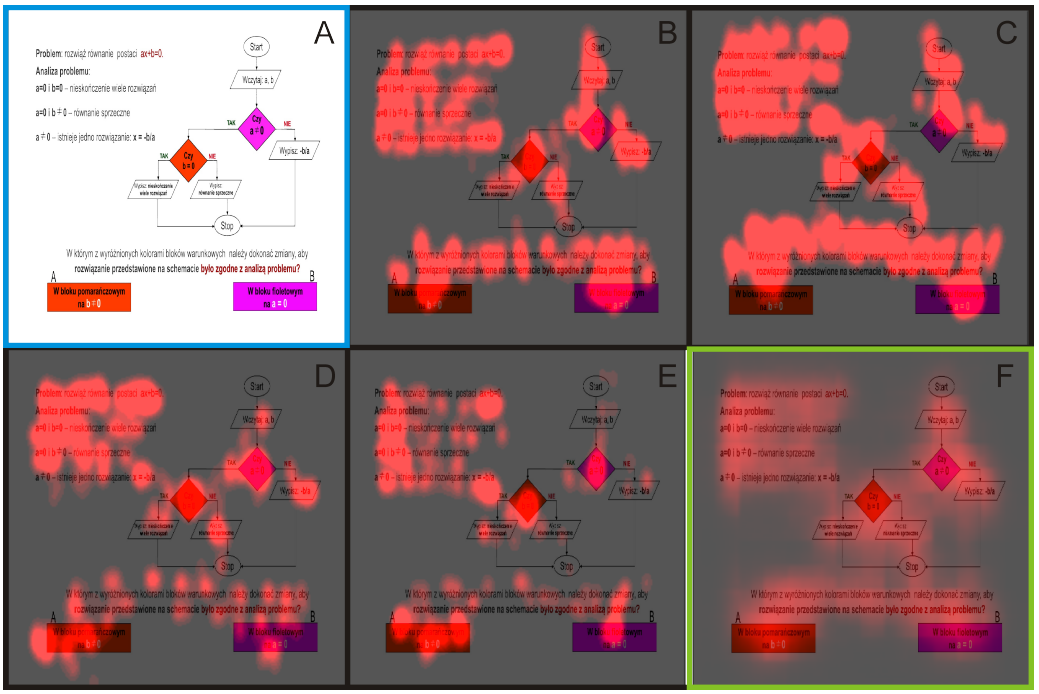


Figure 4. (**A**) a stimulus, (**B**–**E**) examples of real data saliency maps, (**F**) saliency map predicted by E-D 4.

4. Discussion

As can be seen in Figure 2a, neural networks that use a single encoder minimize the loss function to a similar level in the range [0.23, 0.25]. In the case of the 2xE-D 1 network, which has significantly more parameters than the other architectures (see Table 1), the train loss most often reaches 0.379.

In the case of the validation loss (see Table 3), the average results achieved by the E-D 2, E-D 3 [26], and E-D 4 networks are similar and are 0.272 ± 0.053, 0.270 ± 0.053, and 0.275 ± 0.054, respectively. The $E - D1$ network achieves an average validation loss of 0.312 ± 0.086, which is probably due to the fact that the single encoder–decoder pyramid has too little generalization ability to correctly model data on which it has not been trained.

In the case of the 2xE-D 1 network, the average loss is even larger at 0.407 ± 0.039. This could be both an effect of too little generalization power of the decoder, which works by taking embeddings at only one scale (from the last layer of the encoder), but it could also be an effect of overfitting the network, whose numerous parameters fit the training data too closely, losing the generalization ability.

We also made a comparison of the predictions made by each network with the saliency maps (see Figure 3). In this figure, we can observe that the predictions of the E-D 1 and 2xE-D 1 networks contain inaccuracies visible to the naked eye when it comes to gaze fixations on areas containing text. Encoder–decoders E-D 2, E-D 3 [26], and E-D 4 return similar results, with E-D 4 having the highest pixel intensity in areas that correlate with real saliency maps. This observation is also true for the other stimuli in the primary dataset. This observation is confirmed by results presented in Table 4, in which the averaged correlation coefficient between the prediction and the images from the primary dataset is the largest and equals 0.547 ± 0.109. That is higher than in state-of-the-art architecture [26] where the correlation coefficient equals 0.531 ± 0.120. As can be seen, predictions of all but E-D 1 DNNs are moderately positively correlated with validation data.

The structural similarity analysis presented in Table 5 also assures us that E-D 4 seems to be the best of all considered methods. For this network architecture, the averaged SSIM between the prediction and the images from the primary dataset is the largest and equals 0.313 ± 0.082. That is higher than in state-of-the-art architecture [26] where the SSIM equals 0.302 ± 0.110.

Figure 4 shows a comparison of the prediction results of the E-D 4 network (subplot (F)) with the actual saliency maps. As can be seen in Figure 4, visualizations of the predictions obtained are reasonable and in most important aspects they correlate with real-world data. For example, as in real-world data, the network does not pay a lot of attention to the start symbol of the block algorithm that does not bring much information. In addition, a network is not a simple edged detector: it rather predicts that most time is spent reading text and information inside the diagram blocks and does not pay attention to lines that indicate program flow. Additionally, distribution of the predicted fixations inside the text is uneven, as in real data: the largest fixations are generated at the beginning of the lines and in places that contain enhanced text regions where a decision is made while solving the quiz. In Figure 4, we can clearly see that there is a high variance within the real-world saliency maps in the training dataset. That fact makes the generation of accurate predictions of visual attention a very challenging task.

5. Conclusions

Summarizing, as we discussed in the previous section, the deep convolutional encoder–decoder networks E-D 2 and E-D 4 proposed by us are capable of producing accurate predictions of students' visual attention when solving quizzes. Our evaluation showed that predictions are moderately positively correlated with actual data. Our proposed E-D 4 architecture achieved better results in terms of correlation and the SSIM with real saliency maps from our primary dataset than the state-of-the-art E-D 3 architecture [26]. Visual analyses of the obtained saliency maps also correspond with our experience in the field of eye tracking data analysis. We can conclude that the proposed DNNs might be

useful tools in the task of predicting the visual cognition of students while solving various types of technical quizzes, tests, and questionnaires and this application is worth further in-depth research. The main limitation of the proposed algorithm is the resolution of the pretrained VGG16 network, which is 224 × 244. This means that for high-resolution stimuli, some detail will be lost when scaled to the input resolution of the network. The reason for the good results obtained by the E-D 4 architecture is the use of multi-level decoder pyramids that allow the extraction and generation of maps at multiple levels of detail. The source code and data of our implementation might be a valuable state-of-the-art point and reference for other researchers. It is equally important that the proposed DNNs can not only be applied to the prediction of visual attention, they are a general purpose methods which can be used in all scenarios where a multi-scaled encoder–decoder can be applied [49–51].

Author Contributions: Conceptualization: T.H., A.S., M.A.; methodology: T.H.; software: T.H.; validation: T.H., A.S., M.A., P.C.; formal analysis: T.H., A.S., M.A.; investigation: T.H., A.S., M.A.; resources: T.H., A.S., M.A.; data curation: T.H., A.S., M.A.; writing—original draft preparation: T.H., A.S., M.A.; writing—review and editing: T.H., A.S., M.A., P.C.; visualization: T.H.; funding acquisition, T.H., A.S., M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded under the Pedagogical University of Krakow statutory research grant, which was funded by subsidies for science granted by the Polish Ministry of Science and Higher Education.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Source codes and datasets can be downloaded from https://github.com/browarsoftware/DeepVisualAttentionPrediction/ (accessed on 10 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fletcher-Watson, S.; Hampton, S. The potential of eye-tracking as a sensitive measure of behavioural change in response to intervention. *Sci. Rep.* **2018**, *8*, 14715. [CrossRef]
2. Beesley, T.; Pearson, D.; Pelley, M.L. Eye Tracking as a Tool for Examining Cognitive Processes. In *Biophysical Measurement in Experimental Social Science Research*; Academic Press: Cambridge, MA, USA, 2019; pp. 1–30. [CrossRef]
3. Gidlöf, K.; Wallin, A.; Dewhurst, R.; Holmqvist, K. Using Eye Tracking to Trace a Cognitive Process: Gaze Behaviour During Decision Making in a Natural Environment. *J. Eye Mov. Res.* **2013**, *6*. [CrossRef]
4. Chen, X.; Starke, S.D.; Baber, C.; Howes, A. A Cognitive Model of How People Make Decisions Through Interaction with Visual Displays. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017. [CrossRef]
5. Glaholt, M.G.; Reingold, E.M. Eye movement monitoring as a process tracing methodology in decision making research. *J. Neurosci. Psychol. Econ.* **2011**, *4*, 125–146. [CrossRef]
6. Green, H.J.; Lemaire, P.; Dufau, S. Eye movement correlates of younger and older adults' strategies for complex addition. *Acta Psychol.* **2007**, *125*, 257–278. [CrossRef] [PubMed]
7. Hannula, D.E. Worth a glance: Using eye movements to investigate the cognitive neuroscience of memory. *Front. Hum. Neurosci.* **2010**, *4*, 166. [CrossRef]
8. Andrzejewska, M.; Stolińska, A. The eye tracking technique in the analysis of mechanisms for solving algorithmic problems. *e-Mentor* **2018**, *2*, 10–18. [CrossRef]
9. Hao, Q.; Sbert, M.; Ma, L. Gaze Information Channel in Cognitive Comprehension of Poster Reading. *Entropy* **2019**, *21*, 444. [CrossRef] [PubMed]
10. Lai, M.L.; Tsai, M.J.; Yang, F.Y.; Hsu, C.Y.; Liu, T.C.; Lee, S.W.Y.; Lee, M.H.; Chiou, G.L.; Liang, J.C.; Tsai, C.C.; et al. A review of using eye-tracking technology in exploring learning from 2000 to 2012. *Educ. Res. Rev.* **2013**, *10*, 90–115. [CrossRef]
11. Sun, J.C.Y.; Hsu, K.Y.C. A smart eye-tracking feedback scaffolding approach to improving students learning self-efficacy and performance in a C programming course. *Comput. Hum. Behav.* **2019**, *95*, 66–72. [CrossRef]
12. Knoblich, G.; Ohlsson, S.; Raney, G.E. An eye movement study of insight problem solving. *Mem. Cogn.* **2001**, *29*, 1000–1009. [CrossRef]
13. Stolińska, A.; Andrzejewska, M. Analysis of the Strategy Used to Solve Algorithmic Problem: A Case Study Based on Eye Tracking Research. In *Trends in Mathematics New Trends in Analysis and Interdisciplinary Applications*; Birkhäuser: Cham, Switzerland, 2017; pp. 77–86. [CrossRef]

14. Bueno, A.; Sato, J.; Hornberger, M. Eye tracking–The overlooked method to measure cognition in neurodegeneration? *Neuropsychologia* **2019**, *133*, 107191. [CrossRef] [PubMed]
15. Ke, F.; Ruohan, L.; Sokolikj, Z.; Dahlstrom-Hakki, I.; Israel, M. Using Eye Tracking for Research on Learning and Computational Thinking. In *Lecture Notes in Computer Science, Proceedings of the HCI in Games: Serious and Immersive Games, Third International Conference, HCI-Games 2021, Virtual Event, 24–29 July 2021*; Part II; Springer: Cham, Switzerland, 2021; pp. 216–228. [CrossRef]
16. Kiefer, P.; Giannopoulos, I.; Martin, R.; Duchowski, A. Eye tracking for spatial research: Cognition, computation, challenges. *Spat. Cogn. Comput.* **2017**, *17*, 1–19. [CrossRef]
17. Semmelmann, K.; Weigelt, S. Online webcam-based eye tracking in cognitive science: A first look. *Behav. Res. Methods* **2017**, *50*, 451–465. [CrossRef]
18. Aslin, R.; Mcmurray, B. Automated Corneal-Reflection Eye Tracking in Infancy: Methodological Developments and Applications to Cognition. *Infancy* **2004**, *6*, 155–163. [CrossRef]
19. Klaib, A.F.; Alsrehin, N.O.; Melhem, W.Y.; Bashtawi, H.O.; Magableh, A.A. Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies. *Expert Syst. Appl.* **2021**, *166*, 114037. [CrossRef]
20. Shojaeizadeh, M.; Djamasbi, S.; Paffenroth, R.C.; Trapp, A.C. Detecting task demand via an eye tracking machine learning system. *Decis. Support Syst.* **2019**, *116*, 91–101. [CrossRef]
21. Yin, Y.; Juan, C.; Chakraborty, J.; McGuire, M.P. Classification of Eye Tracking Data Using a Convolutional Neural Network. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 530–535. [CrossRef]
22. Chen, Z.; Fu, H.; Lo, W.; Chi, Z. Strabismus Recognition Using Eye-Tracking Data and Convolutional Neural Networks. *J. Healthc. Eng.* **2018**, *2018*, 7692198. [CrossRef] [PubMed]
23. Dalrymple, K.; Jiang, M.; Zhao, Q.; Elison, J. Machine learning accurately classifies age of toddlers based on eye tracking. *Sci. Rep.* **2019**, *9*, 6255. [CrossRef] [PubMed]
24. Lee, S.; Hooshyar, D.; Ji, H.; Nam, K.; Lim, H. Mining biometric data to predict programmer expertise and task difficulty. *Clust. Comput.* **2018**, *21*, 1097–1107. [CrossRef]
25. Louedec, J.L.; Guntz, T.; Crowley, J.L.; Vaufreydaz, D. Deep Learning Investigation for Chess Player Attention Prediction Using Eye-Tracking and Game Data. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA '19, Denver, CO, USA, 25–28 June 2019; Association for Computing Machinery: New York, NY, USA, 2019. [CrossRef]
26. Wang, W.; Shen, J. Deep Visual Attention Prediction. *Trans. Image Proc.* **2018**, *27*, 2368–2378. [CrossRef]
27. Sharma, K.; Giannakos, M.; Dillenbourg, P. Eye-tracking and artificial intelligence to enhance motivation and learning. *Smart Learn. Environ.* **2020**, *7*, 1–19. [CrossRef]
28. Mu, L.; Cui, M.; Qiao, J.; Hu, X. *Visual Analysis Method of Online Learning Path Based on Eye Tracking Data*; Springer: Singapore, 2019; pp. 179–195. [CrossRef]
29. Mu, L.; Cui, M.; Wang, X.; Qiao, J.; Tang, D. Learners' attention preferences of information in online learning: An empirical study based on eye-tracking. *Interact. Technol. Smart Educ.* **2019**, *16*, 186–203. [CrossRef]
30. Chopade, P.; Edwards, D.; Khan, S.M.; Andrade, A.; Pu, S. CPSX: Using AI-Machine Learning for Mapping Human-Human Interaction and Measurement of CPS Teamwork Skills. In Proceedings of the 2019 IEEE International Symposium on Technologies for Homeland Security (HST), Woburn, MA, USA, 5–6 November 2019; pp. 1–6. [CrossRef]
31. Król, M.; Król, M. Learning From Peers' Eye Movements in the Absence of Expert Guidance: A Proof of Concept Using Laboratory Stock Trading, Eye Tracking, and Machine Learning. *Cogn. Sci.* **2019**, *43*, e12716. [CrossRef] [PubMed]
32. Jung, Y.J.; Zimmerman, H.; Perez-Edgar, K. *Mobile Eye-Tracking for Research in Diverse Educational Settings*; Taylor & Francis Group: London, UK; 2020. [CrossRef]
33. Fwa, H.L. Modeling engagement of programming students using unsupervised machine learning technique. In Proceedings of the Computer Science Education: Innovation and Technology, Bologna, Italy, 3–5 July 2017.
34. Emerson, A.; Henderson, N.; Rowe, J.; Min, W.; Lee, S.; Minogue, J.; Lester, J. Early Prediction of Visitor Engagement in Science Museums with Multimodal Learning Analytics. In Proceedings of the ICMI '20: 2020 International Conference on Multimodal Interaction, Virtual Event, 25–29 October 2020; pp. 107–116. [CrossRef]
35. Kuechemann, S.; Klein, P.; Becker, S.; Kumari, N.; Kuhn, J. Classification of Students' Conceptual Understanding in STEM Education using Their Visual Attention Distributions: A Comparison of Three Machine-Learning Approaches. In Proceedings of the 12th International Conference on Computer Supported Education, Prague, Czech Republic, 2–4 May 2020; pp. 36–46. [CrossRef]
36. Dzsotjan, D.; Ludwig-Petsch, K.; Mukhametov, S.; Ishimaru, S.; Kuechemann, S.; Kuhn, J. The Predictive Power of Eye-Tracking Data in an Interactive AR Learning Environment. In Proceedings of the UbiComp '21: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers, Virtual, 21–26 September 2021; pp. 467–471. [CrossRef]
37. Pritalia, G.L.; Wibirama, S.; Adji, T.B.; Kusrohmaniah, S. Classification of Learning Styles in Multimedia Learning Using Eye-Tracking and Machine Learning. In Proceedings of the 2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE), Bandung, Indonesia, 23–24 September 2020; pp. 145–150. [CrossRef]

38. Zhai, X.; Yin, Y.; Pellegrino, J.W.; Haudek, K.C.; Shi, L. Applying machine learning in science assessment: A systematic review. *Stud. Sci. Educ.* **2020**, *56*, 111–151. doi: 10.1080/03057267.2020.1735757. [CrossRef]
39. Rappa, N.A.; Ledger, S.; Teo, T.; Wong, K.W.; Power, B.; Hilliard, B. The use of eye tracking technology to explore learning and performance within virtual reality and mixed reality settings: A scoping review. *Interact. Learn. Environ.* **2019**, 1–13. [CrossRef]
40. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535. [CrossRef]
41. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
42. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2014**, *115*, 211–252. [CrossRef]
43. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
44. Maas, A.L.; Hannun, A.Y.H.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013.
45. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef]
46. Jiang, M.; Huang, S.; Duan, J.; Zhao, Q. SALICON: Saliency in Context. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1072–1080. [CrossRef]
47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
48. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image Quality Assessment: From Error Visibility to Structural Similarity. *Image Process. IEEE Trans.* **2004**, *13*, 600–612. [CrossRef]
49. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
50. Shvets, A.A.; Iglovikov, V.; Rakhlin, A.; Kalinin, A. Angiodysplasia Detection and Localization Using Deep Convolutional Neural Networks. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 612–617.
51. Zhao, X.; Li, H.; Wang, R.; Zheng, C.; Shi, S. Street-view Change Detection via Siamese Encoder-decoder Structured Convolutional Neural Networks. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019), Prague, Czech Republic, 25–27 February 2019.