Journal of
Imaging

MDPI

Article

Variational Autoencoder for Image-Based Augmentation of Eye-Tracking Data

Mahmoud Elbattah [1,*], Colm Loughnane [2], Jean-Luc Guérin [1], Romuald Carette [1,3], Federica Cilia [4] and Gilles Dequen [1]

[1] Laboratoire Modélisation, Information, Systèmes (MIS), Université de Picardie Jules Verne, 80080 Amiens, France; jean-luc.guerin@u-picardie.fr (J.-L.G.); r.carette@evolucare.com (R.C.); gilles.dequen@u-picardie.fr (G.D.)
[2] Faculty of Science and Engineering, University of Limerick, V94 T9PX Limerick, Ireland; 9926186@studentmail.ul.ie
[3] Evolucare Technologies, 80800 Villers-Bretonneux, France
[4] Laboratoire CRP-CPO, Université de Picardie Jules Verne, 80000 Amiens, France; federica.cilia@u-picardie.fr
* Correspondence: mahmoud.elbattah@u-picardie.fr

Abstract: Over the past decade, deep learning has achieved unprecedented successes in a diversity of application domains, given large-scale datasets. However, particular domains, such as healthcare, inherently suffer from data paucity and imbalance. Moreover, datasets could be largely inaccessible due to privacy concerns, or lack of data-sharing incentives. Such challenges have attached significance to the application of generative modeling and data augmentation in that domain. In this context, this study explores a machine learning-based approach for generating synthetic eye-tracking data. We explore a novel application of variational autoencoders (VAEs) in this regard. More specifically, a VAE model is trained to generate an image-based representation of the eye-tracking output, so-called scanpaths. Overall, our results validate that the VAE model could generate a plausible output from a limited dataset. Finally, it is empirically demonstrated that such approach could be employed as a mechanism for data augmentation to improve the performance in classification tasks.

Keywords: deep learning; variational autoencoder; data augmentation; eye-tracking

check for updates

Citation: Elbattah, M.; Loughnane, C.; Guérin, J.-L.; Carette, R.; Cilia, F.; Dequen, G. Variational Autoencoder for Image-Based Augmentation of Eye-Tracking Data. *J. Imaging* **2021**, *7*, 83. https://doi.org/10.3390/jimaging7050083

Academic Editor: Marco Porta

Received: 23 March 2021
Accepted: 1 May 2021
Published: 3 May 2021

1. Introduction

Human eyes represent a rich source of information, for communicating emotional and mental conditions, as well as for understanding the functioning of our cognitive system. An eye gaze can serve as an appropriate proxy for learning a user's attention or focus on context [1]. Therefore, eye-tracking technology has been intensively utilized for studying and analyzing many aspects of gaze behavior.

Eye-tracking refers to the process of capturing, tracking, and measuring the absolute point of gaze (POG) and eye movement [2]. Interestingly, the field of eye-tracking has quite a long history, dating back to the 19th century. The French ophthalmologist Louis Javal, from Sorbonne University, started the initial analysis of gaze behavior in 1878. It is largely acknowledged that Javals' studies [3,4] laid out the foundations that initially explored the behavior of human gaze in terms of fixations and saccades. Subsequently, Edmund Huey built a primitive eye-tracking tool for analyzing eye movements [5]. More advanced implementations of eye-tracking were developed by [6,7]. Photographic films were utilized to record eye movements while looking at a variety of paintings. The eye-tracking records included both direction and duration of movements.

With technological advances, the field of eye-tracking has evolved towards the nearly universal adoption of video-based methods. Video-based eye-trackers can be classified into the following: (1) video-based tracking using remote or head-mounted cameras and (2) video-based tracking using infrared pupil-corneal reflection (P-CR) [2]. Furthermore, recent developments have discussed the use of virtual reality-based methods for eye-tracking [8]. Eye-tracking has been widely utilized in a multitude of applications for commercial and research purposes. Examples include marketing [9], psychology studies [10], product design [11], and many other applications.

However, the scarce availability or difficulty of acquiring eye-tracking datasets represents a key challenge, while access to image or time series data, for example, has been largely facilitated thanks to large-scale repositories such as ImageNet [12] or UCR [13]. The eye-tracking literature still lacks such data repositories. In this respect, we explore the use of machine learning (ML) for generating synthetic eye-tracking data in this study. An image-based approach is adopted based on transforming the eye-tracking scanpaths into a visual representation. Using unsupervised learning, a variational autoencoder (VAE) is employed for the generative modeling task. Subsequently, empirical experiments robustly demonstrated that the inclusion of VAE-generated images could improve the performance of models in classification tasks. The primary contribution of this study is claimed to be as exploring a novel application of VAEs in this context. To the best of our knowledge, the proposed approach has not been discussed yet in the literature.

2. Background

In this section, we provide a preliminary background on autoencoders and their applications in general. Initially, in the first section, we review the classical autoencoders, mostly used for tasks related to data compression, feature extraction, or denoising. Subsequently, we discuss the VAE approach and its suitability for generative modeling, which is the focus of the present study.

2.1. Autoencoders

Generally, autoencoders are considered to be a special implementation of artificial neural networks (ANNs). In contrast to typical ANN applications (e.g., regression and classification), autoencoders are fully developed in an unsupervised manner. Using unsupervised learning, autoencoders learn compressed representations of data, the so-called "codings". As such, training an autoencoder does not require any label information. The compression and decompression are automatically inferred from data in contrast to being formulated using mathematical equations or hand-crafted features. Figure 1 illustrates the basic architecture of autoencoders including encoding and decoding.

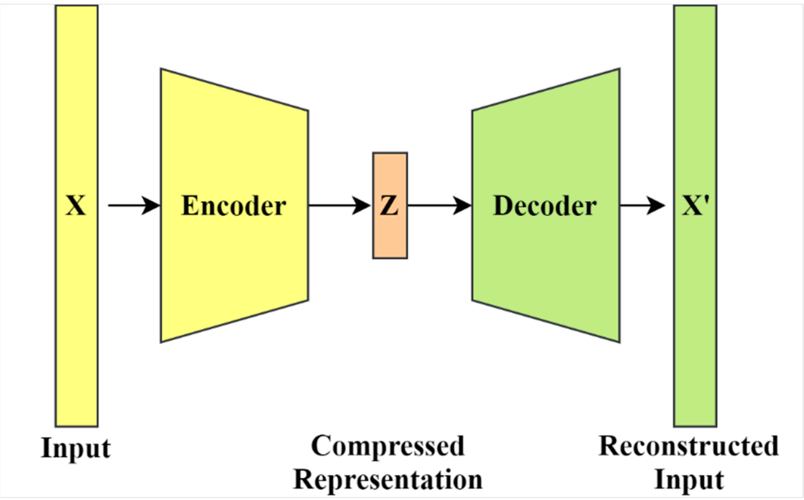

Figure 1. The general architecture of autoencoders.

The idea of autoencoders was originally introduced in the 1980s by the parallel distributed processing (PDP) group including Geoffrey Hinton, at the University of California, San Diego. They were generally motivated by the challenge of training a multi-layered ANN, which could allow for learning any arbitrary mapping of input to output [14].

Their work eventually led to the development of the backpropagation algorithm, which has become the standard approach for training ANNs.

There is a variety of valid applications that could be realized by autoencoders. Fundamentally, autoencoders can be used as an effective means to reduce data dimensionality [15,16], whereas codings represent a latent space of significantly lower dimensionality as compared with the original input. Furthermore, autoencoders provide a potent mechanism for feature extraction. More interestingly, they can perform the functionality of generative modeling. The codings learned can be utilized to randomly generate synthetic samples, similar to the original data.

Data denoising is another well-explored application of autoencoders. Denoising autoencoders were first developed by Vincent et al. [17,18]. The basic idea is that the encoder can consider its input as corrupted data, while the decoder attempts to reconstruct the clean uncorrupted version. Therefore, denoising autoencoders can learn the data distribution without constraints on the dimensions or sparsity of the encoded representation. Several studies have experimentally implemented denoising autoencoders in a variety of important applications. For example, denoising autoencoders were successfully applied for speech enhancement and restoration [19,20]. By the same token, a convolutional denoising autoencoder was utilized for reducing the noise in medical images [21].

2.2. Variational Autoencoders

Kingma and Welling [22] originally introduced the VAE framework in 2014, which has been considered as one of the paramount contributions for generative modeling or representation learning in general. The VAE approach provided a novel method that jointly coupled probabilistic models with deep learning. In contrast to traditional autoencoders, the fundamental distinction of VAEs is that they learn latent variables with continuous distributions, which has proven to be a particularly useful property while approaching tasks of generative modeling.

VAE encoding has been cleverly designed to return a distribution over the latent space rather than discrete values. More specifically, the encoder produces a set of two vectors including a vector of means (μ), and another vector of standard deviations (σ). As such, the VAE attempts to learn the distributions of latent variables based on the mean values and their variances, instead of learning a deterministic mapping, as in traditional autoencoders. Figure 2 shows a sketch of the VAE architecture and it can be observed that the latent dimensional space is stochastic based on the samples of μ and σ values. A comprehensive presentation of the VAE approach goes beyond the scope of this study, however, we recommend the tutorial by Kingma and Welling [23] in this regard.

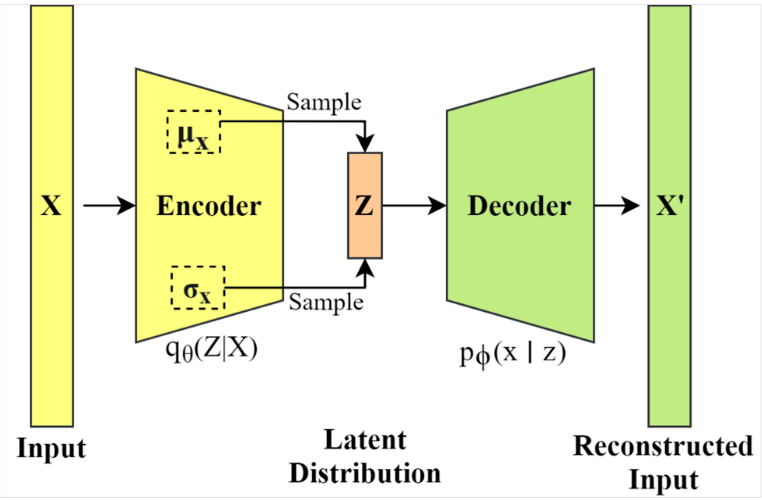

Figure 2. The variational autoencoder (VAE) architecture. X represents the input to the encoder model and Z is the latent representation along with weights and biases (θ).

Since its inception, the VAE approach has been increasingly adopted in a diversity of generative modeling tasks. For example, an RNN-based VAE architecture was implemented for text generation [24]. Likewise, a study [25] developed a hybrid architecture of convolutional neural networks (CNN) and recurrent neural networks (RNN) for text generation as well, while other studies explored the VAE potentials for generating natural images [26,27]. It is also worth mentioning that the generative adversarial network (GAN) by Goodfellow et al. [28] is another popular approach for generative modeling, however, it is not the focus of the present study.

3. Related Work

The literature review is divided into two sections as follows: Initially, the first section includes representative studies that implemented VAE-based applications for the purpose of data augmentation or generative modeling in general. The second section reviews contributions that attempted to synthetically generate or simulate eye-tracking output. In this respect, we aim to review approaches that have developed algorithmic models, as well as ML-based methods. The review is selective rather than exhaustive, therefore, it basically aims to highlight representative approaches in this context.

3.1. Variational Autoencoder (VAE)-Based Methods for Data Augmentation

The VAE approach has been intensively applied for synthetic data generation, or representation learning in a broader sense. The literature already includes a diversity of studies that made use of VAE-based implementations as a mechanism for data augmentation.

For instance, a study by [29] explored the beneficial use of VAEs in the case of imbalanced datasets. To this end, they extracted an imbalanced subset of the popular MNIST dataset. The dataset was augmented with synthetic samples generated by a VAE model. Their empirical results demonstrated that the inclusion of VAE-generated samples had a positive impact on the classification accuracy in general. Similarly, a more recent study analyzed the impact of using different augmentation techniques on the model accuracy in supervised ML problems [30]. Their experiments focused on smaller datasets, where the number of samples per class were lower than 1000. The experiments were based on a set of 19 benchmark datasets selected from the University of California Irvine (UCI) data repository [31]. Using VAE and GAN models, their results demonstrated that data augmentation could boost the prediction accuracy by approximately 3%.

From a practical standpoint, the literature includes a broad variety of applications using the VAE approach for augmentation. One recent study used a VAE model to generate traffic data pertaining to crash events [32]. Their work demonstrated how the VAE latent space could be used to generate millions of synthetic crash samples. The use of data augmentation had a significant effect on the model performance since the original dataset was extremely imbalanced. In another application related to acoustic modeling, a VAE-based framework was developed to perform data augmentation and feature extraction [33]. The dataset size of speech corpus could be doubled using the latent variables extracted by the VAE model. Similarly, their results demonstrated that augmentation could improve the performance of speech recognition.

In the context of electroencephalography (EEG), a study used augmentation techniques including VAE [34]. They applied a VAE model to generate realistic features of EEG records, which were used to augment the training data. The empirical results reported a significant improvement in the accuracy of the emotion recognition models. More specifically, the models could achieve up to 10% improvement. Similarly, recent efforts [35] have explored VAE-based methods to augment EEG datasets.

Furthermore, numerous applications have been experimentally studied in the field of medical imaging. For instance, a convolutional VAE model was developed to generate realistic samples of left ventricular segmentations for data augmentation [36]. Another study demonstrated the effectiveness of VAEs for generating synthetic images of clinical datasets including ultrasound spine images and Magnetic Resonance Imaging (MRI) brain images [37]. More complex tasks were approached using VAE-based architectures as well. For example, a VAE-based approach was adopted for the three-dimensional (3D) reconstruction of the fetal skull from two-dimensional (2D) ultrasound planes acquired during the screening process [38]. They developed a VAE model that could integrate ultrasound planes into conditional variables to generate a consolidated latent space. Likewise, a VAE architecture was implemented for the reconstruction of 3D high-resolution cardiac segmentation [39].

3.2. Generative Modeling of Eye-Tracking Data

The literature is rife with methods applied for synthesizing or simulating human eye movements, typically captured by eye trackers. The methods can be broadly classified into two schools of thoughts. On the one hand, the early efforts aimed to craft algorithmic models based on characteristics driven from the eye-tracking research. On the other hand, recent studies have been more inclined towards ML-based approaches.

For instance, a study proposed to synthesize the eye gaze behavior from an input of head-motion sequences [40]. Their method was mainly based on the statistical modeling of the natural conjugation of head and gaze movements. Similarly, another study developed a stochastic model of gaze behavior [41]. The synthetic output could be parameterized based on a set of variables such as sampling rate, micro-saccadic jitter, and simulated measurement error.

In a similar vein, there have been plentiful contributions for developing gaze models that can generate realistic eye movements in animations or virtual environments. To name a few, one study implemented statistical models of eye-tracking output based on the analysis of eye-tracking videos [42]. The models were aimed at reflecting the dynamic characteristics of natural eye movements (e.g., saccade amplitude and velocity). Another framework was proposed to automate the generation of realistic eye and head movements [43]. It was basically aimed at separately learning inter-related statistical models for each component of movement based on pre-recorded facial motion data. The framework also considered the subtle eyelid movement and blinks.

Recent experimental studies have been purely ML-based approaches for generating synthetic eye-tracking data. Eye-tracking instruments produce an abundant amount of data including a variety of eye-gaze information. A few minutes of operating time can typically output thousands of records describing gaze positions and eye movements. Hence, ML could be viewed as an ideal path to also develop predictive and generative models. In addition, the emergence of deep learning has played a key role in this regard. Deep learning provides a potent mechanism for learning complex mappings from raw data automatically, avoiding the need for developing hand-crafted features. Implementations of CNNs [44,45] and RNNs [46] have been successfully applied to tackle complex tasks such as computer vision and machine translation.

In this respect, a CNN-based architecture was developed for the semantic segmentation of eye-tracking data [47]. A CNN-based architecture was utilized for the reconstruction and generation of eye movement data. Another study proposed a convolutional-recurrent architecture, named "PathGAN" [48]. On the basis of adversarial learning, the PathGAN framework presented an end-to-end model for predicting the visual scanpath. In another application, a real-time system for gaze animation was developed using RNNs [49]. Motion and video data were both used to train the RNN model, which could predict the motion of body and eyes. The data were captured by a head-mounted camera.

Moreover, long short-term memory (LSTM) architectures have been developed to generate synthetic eye-tracking data, for instance, a sequence-to-sequence LSTM-based architecture was developed to this end [50]. More recently, another recent study proposed a text-based approach using an LSTM implementation [51]. The key idea was to represent eye-tracking records as textual strings, which described the sequences of fixations and saccades. As such, they could apply methods from the natural language processing

(NLP) domain to transform and model eye-tracking sequences, while an LSTM model was employed for the generative modeling task.

4. Data Description

The dataset under consideration was collected as part of our earlier work related to the detection of autism using eye-tracking [52]. Abnormalities of eye gaze have been largely identified as the hallmark of autism spectrum disorder (ASD) [53]. As such, eye-tracking methods are widely utilized in this context.

The dataset was originally constructed as follows: A group of 59 children participated in a set of eye-tracking experiments. The age of participants ranged from 3 to 12 years old. The participants were grouped into two cohorts as follows: (i) typically developing (TD) and (ii) ASD. The participants engaged in watching a set of photographs and videos, which included social cognition scenarios according to their age, to stimulate the viewer's gaze. The average period of time of each eye-tracking experiment was about 5 min.

The experiments were conducted using an eye-tracker by SensoMotoric Instruments (SMI) (Teltow, Germany) with 60 Hz sampling rate. The eye-tracking device captured three categories of eye movements including fixations, saccades, and blinks. A fixation describes a brief period of gaze focus on an object, which allows the brain to perform the process of perception. The average timespan of fixations is estimated to be around 330 ms [54]. Saccades include rapid and short eye movements that perform constant scanning and consist of quick ballistic jumps of 2° or longer, with an average duration of about 30–120 ms each [55]. The output of a sequence of fixations and saccades is defined as a scanpath.

A set of 25 eye-tracking experiments was conducted to produce the output dataset. The dataset was stored in multiple CSV files, which collectively included more than 2M records. For the purpose of demonstration, Table 1 provides a few eye-tracking records as captured by the eye-tracking device which describe the category of movements and the POG coordinates over the experiment runtime. Specifically, each row represents a point in the experiment timeline, where the eye-tracking timing was approximately 20 ms. Due to limited space, many other variables had to be excluded from the table (e.g., pupil position and pupil size).

Table 1. Samples of the of eye-tracking dataset.

Recording Time [ms]	Category of Movement	Coordinates of POG (X,Y) [px]	Diameter of Pupil (Right, Left) [mm]
8,075,764.426	Fixation	784.4646, 707.7583	4.7591, 4.6711
8,075,808.431	Saccade	784.6325, 707.6475	4.642, 4.6457
8,075,852.429	Fixation	784.4073, 707.5976	4.7215, 4.6723
8,075,896.554	Saccade	784.5244, 708.0931	4.7478, 4.6683
8,075,940.549	Saccade	784.2977, 708.3432	4.6815, 4.6917

5. Data Transformation

Data transformation was of paramount importance since the eye-tracking output was obviously high-dimensional. Therefore, the aim was to transform the eye-tracking data into a representation more amenable for ML. The basic idea of our approach was to produce a compact image-based format of eye-tracking scanpaths. This section elaborates on the data transformation procedures.

Initially, it is important to clearly define a scanpath, which is the building block of data. A scanpath represents a sequence of consecutive fixations and saccades as a trace through time and space that may overlap itself [56]. The term was first brought into use by Noton and Stark in 1971 [57]. Scanpaths are commonly utilized in eye-tracking applications as a practical means to depict gaze behavior in a visual manner. Figure 3 represents an example of a basic scanpath, which includes a small number of fixations and saccades. The fixations are shown as circles, while the saccades represent the lines connecting those fixations. The diameter of fixations indicates the duration, and the lengths of lines represent the continuation of saccades.

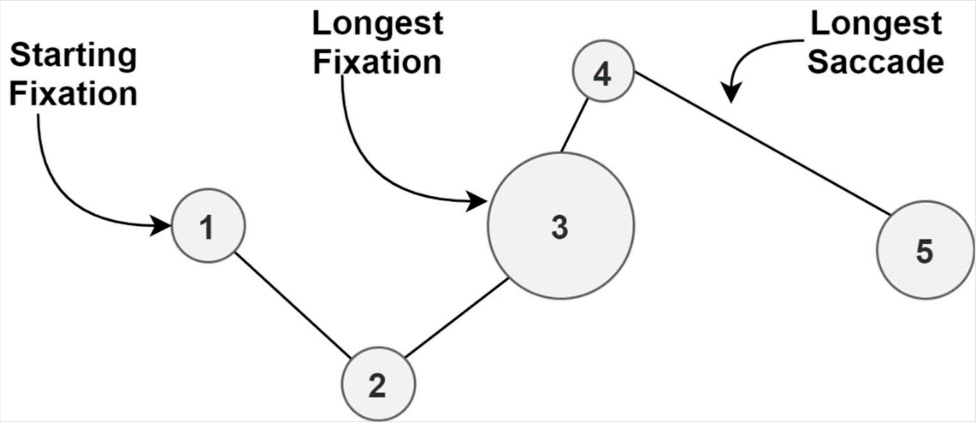

Figure 3. Eye-tracking scanpath [56].

As we previously mentioned, our approach was based on transforming eye-tracking output (i.e., scanpaths) into an image-based format. Our representation of scanpaths follows on the core idea of visualizing fixations and saccades. Moreover, we aimed to visually encode the dynamics of gaze using color gradients. Given the coordinates/time information, we were able to calculate the velocity of gaze movement. Using the grayscale spectrum, the color values were tuned based on the magnitude of velocity with respect to time. The visualizations were produced using Matplotlib library [58]. A comprehensive presentation of that part is elaborated in our earlier work [52].

The outcome of the transformation process was an image dataset containing more than 500 images. Specifically, 328 images related to the TD participants, and another 219 images for the ASD-diagnosed. The default image dimensions were set as 640 × 480. The dataset along with its metadata files have been made publicly available on the Figshare repository [59]. Figure 4 presents two examples from the dataset.

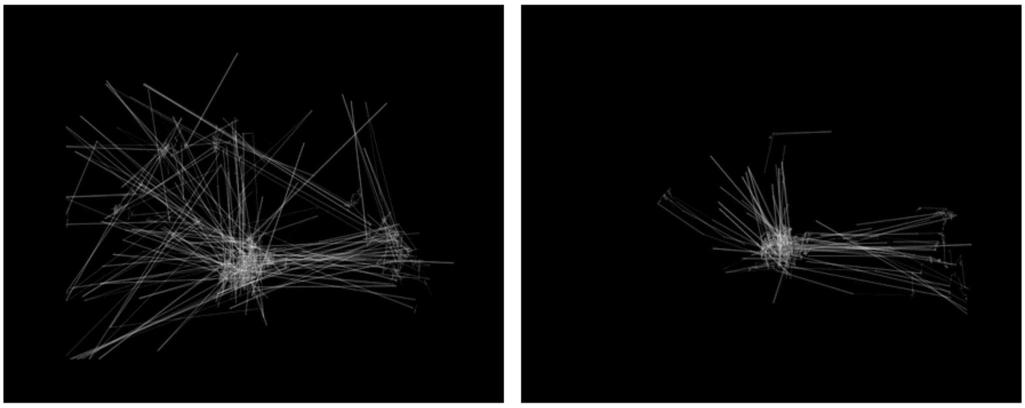

Figure 4. Visualization of eye-tracking scanpaths [52]. The **left**-sided image represents an autism spectrum disorder (ASD) sample, while the **right**-sided image represents the typically developing (TD).

6. Experiments

The empirical ML experiments consisted of two stages. The initial experiments included the generative modeling of eye-tracking scanpaths. This included the design and implementation of the VAE model. Subsequently, the other stage of our experiments included the development of a classification model to predict ASD based on the scanpath images. The original dataset was augmented using the VAE-generated images produced earlier. The experiments basically aimed to explore the impact of data augmentation on the model performance.

6.1. Preprocessing

Initially, a set of preprocessing procedures was applied to simplify the representation of scanpath images. First, the images were cropped in order to remove the blank background. The cropping was based on finding the contour area around the scanpath, which would minimize the background. The cropping was facilitated by using functions from the OpenCV 4.5 library [60].

Second, the images were scaled down to dimensions of 100 × 100. Resizing the images generally heled to reduce the data dimensionality by decreasing the number of features under consideration. Furthermore, it was clear that high-resolution images were not necessary in our case at all, whereas the scanpaths basically represented geometric visualizations rather than natural images.

6.2. VAE Experiments

A convolutional VAE was implemented to investigate the latent representation of scanpath images. The VAE model was designed based on a simple symmetric design, where both the encoder and decoder were composed of two convolutional layers, followed by a single fully connected layer. The input images of (100 × 100) dimensions were encoded into a set of (128 × 1) latent variables, which followed a continuous distribution. The mean and variance of distributions were also estimated by the encoder model.

The decoder model was a "flipped" version of the encoder. Inversely, a fully connected layer followed by two deconvolutional layers were stacked in the decoder model. The decoder's output is a reconstructed scanpath image. Figure 5 shows a sketch of the VAE model architecture.

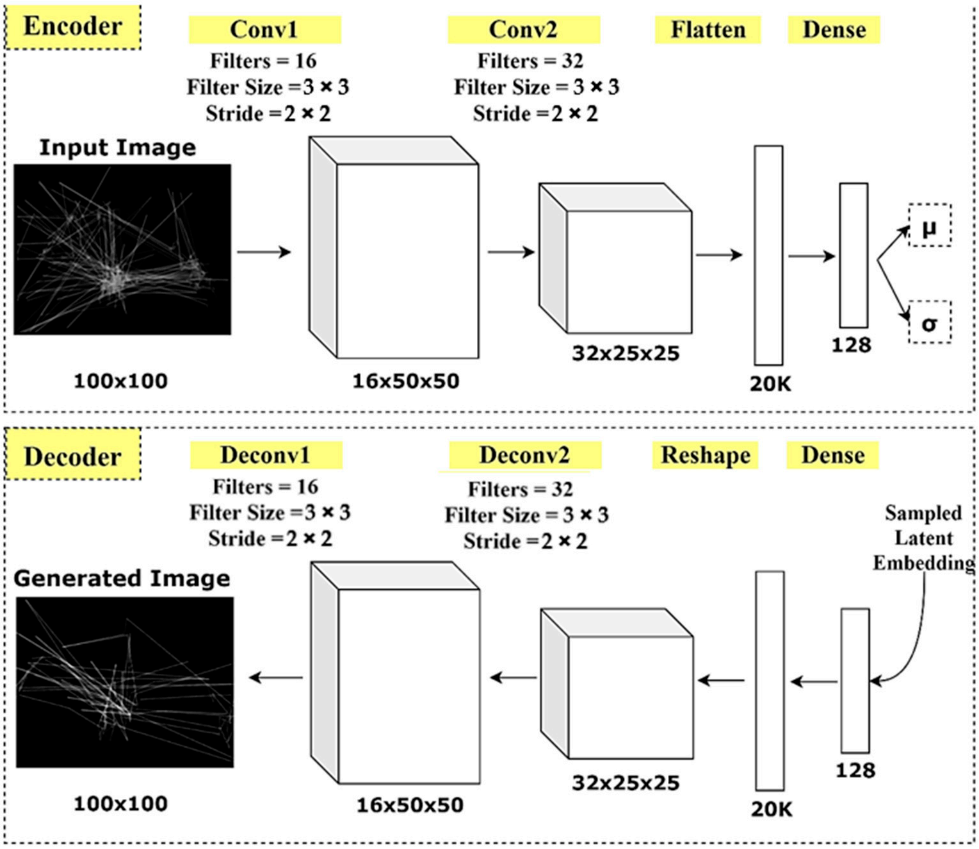

Figure 5. The architecture of the VAE model.

Specifically, two versions of the VAE model were trained using the ASD and TD samples separately. As such, the dataset was initially split into two partitions, where each partition included exclusively a single category of samples. Each VAE model was trained over 20 epochs, and 30% of the dataset was used for validation. Figures 6 and 7 plot the model loss in the training and validation sets for the positive and negative datasets, respectively. It can be observed that the VAE models both largely converged after 10 epochs.

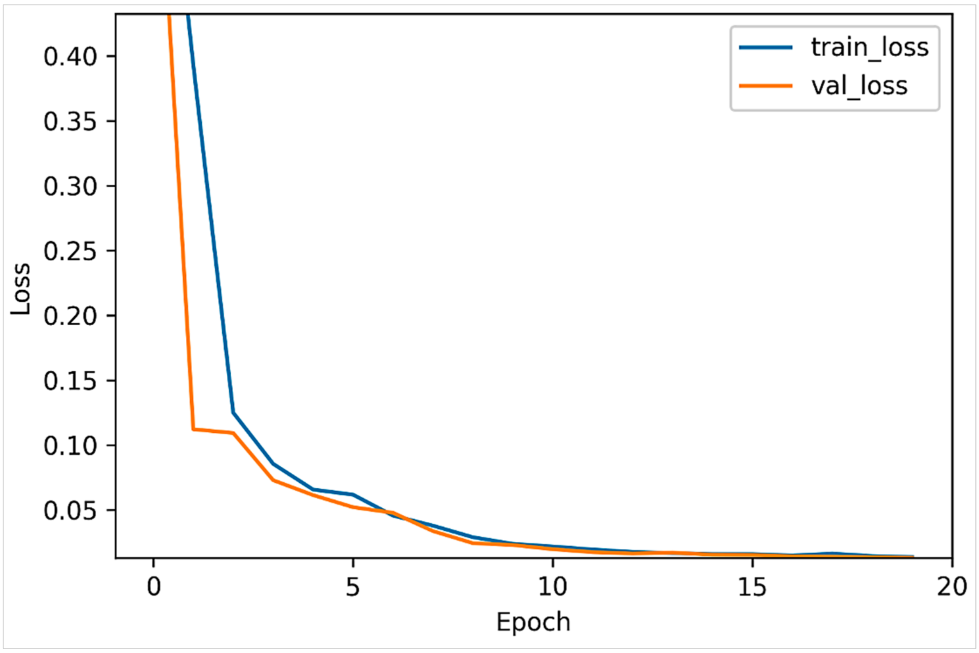

Figure 6. VAE loss in training and validation sets (ASD-diagnosed set).

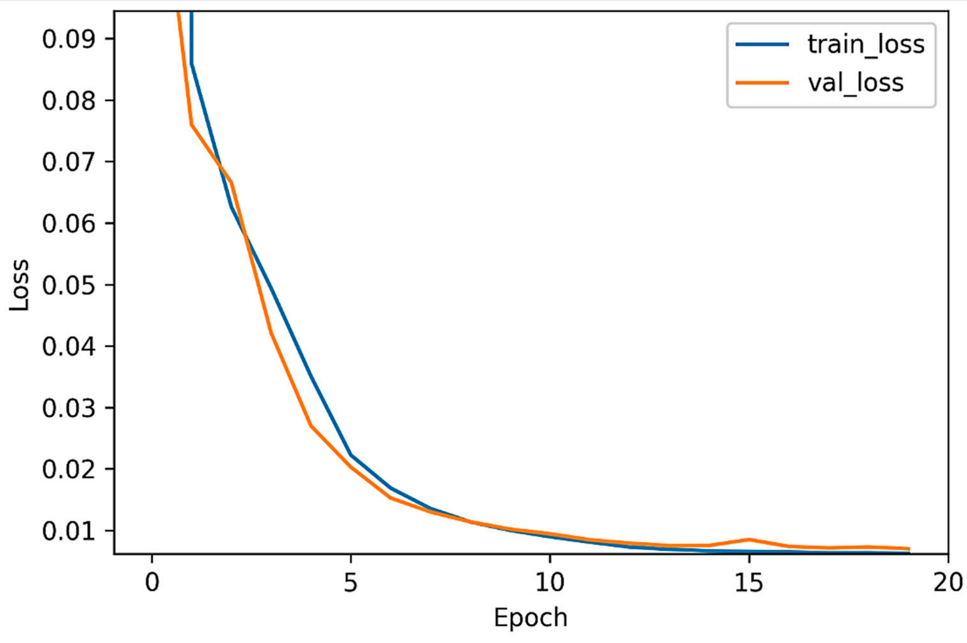

Figure 7. VAE loss in training and validation sets (TD set).

The model was implemented using Keras [61] with the TensorFlow backend [62]. Eventually, the VAE models were used to generate synthetic scanpath images. Around 300 images were generated for each category. Figure 8 demonstrates two sample images generated by the VAE model.

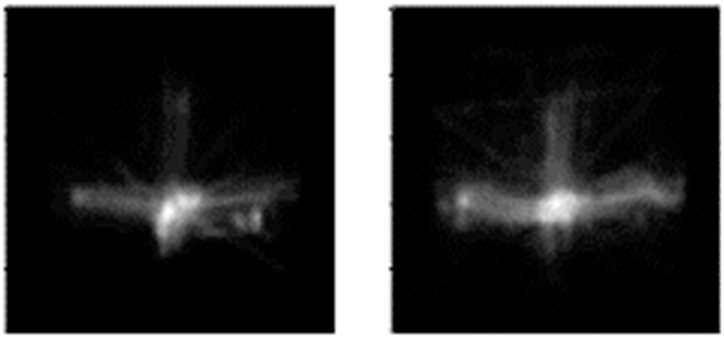

Figure 8. Examples of VAE-generated images.

6.3. Classification Experiments

This part aims to investigate the impact of data augmentation on the performance of classification models. Specifically, we compared the model performance before and after the inclusion of the VAE-generated images as part of the training set.

A CNN model was implemented for the classification experiments. The model was composed of four convolutional layers. Each convolutional layer was followed by a max-pooling operation. Eventually, the model included two fully connected layers. A Rectified Linear Unit (ReLU) was used as the activation function in all layers. The dataset was partitioned into training and test sets based on a three-fold cross-validation. The experiments included two scenarios. On the one hand, the model was trained without including the synthetic images. On the other hand, the model was re-trained after the inclusion of the VAE-generated images in the training set. However, the test set always included samples from the original dataset in both scenarios.

The classification accuracy was analyzed based on the receiver operating characteristics (ROC) curve. The ROC curve plots the relationship between the true positive rate and the false positive rate across a full range of possible thresholds. Figure 9 plots the ROC curve in the baseline case (i.e., without augmentation), while Figure 10 plots the ROC curve in case of applying the VAE-based data augmentation, as previously explained. The figures give the approximate value of the area under the curve and its standard deviation over the three-fold cross-validation. The AUC-ROC values demonstrate that the model performance consistently improved after augmenting the dataset with the synthetic images. Table 2 elaborates further on the model performance in terms of accuracy and AUC-RCO as well. The results demonstrated that the overall classification accuracy was improved by approximately 3%.

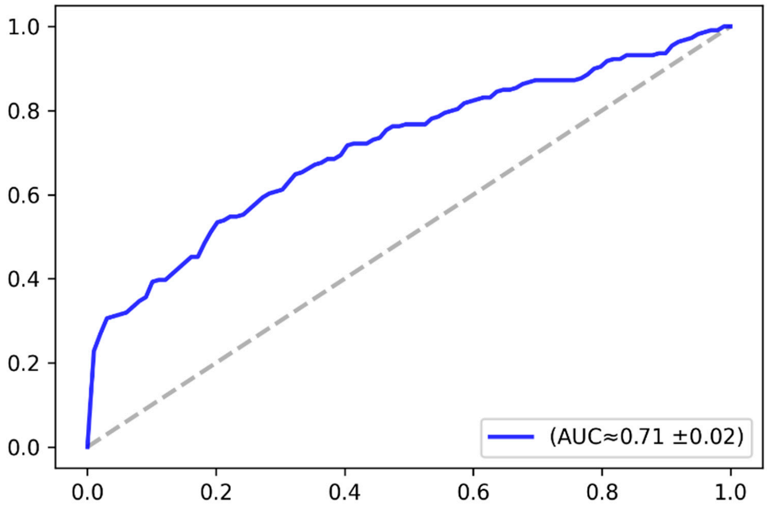

Figure 9. Receiver operating characteristics (ROC) curve-baseline model (no data augmentation).

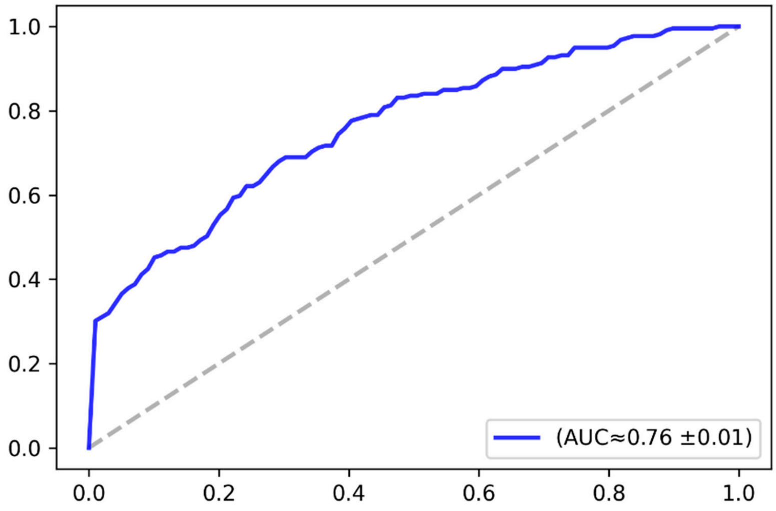

Figure 10. ROC curve after applying VAE-based data augmentation.

Table 2. Model performance after data augmentation.

Model	AUC-ROC	Avg. Accuracy
Baseline case (no augmentation)	0.71	67%
With VAE-based augmentation	0.76	70%

The training process was completed over 10 epochs using an Adam optimizer [63] with its default parameters. The dropout technique [64] was applied, which helped to minimize the possibility of overfitting. The classification models were implemented using Keras [61] with the TensorFlow backend [62]. Other libraries were certainly useful including Scikit-Learn [65] and NumPy [66]. All experiments were run on the Google Cloud platform using a VM containing a single P-100 Nvidia GPU, and 25 GB RAM.

7. Conclusions

The application of data augmentation has been recognized to generally improve the prediction accuracy of image classification tasks [67]. Earlier studies [68,69] sought to generate synthetic images by applying various transformations. Examples included geometric transformations such as random translation, zooming, rotation, flipping, or other manipulations such as noise injection. More recent studies have aimed to utilize the state-of-the-art approaches for generative modeling. In this respect, VAE-based and GAN-based implementations are being increasingly adopted for data augmentation tasks.

In this regard, the results of the present study support the potential of VAE models to perform as an effective mechanism for data augmentation. We demonstrated how a VAE-based approach could be used to generate synthetic eye-tracking data. The mainstay of our approach is the visual representation of eye-tracking data, which allowed for an amenable representation for training the VAE model.

The empirical results clearly confirmed the positive impact of data augmentation on the model's performance. The classification accuracy could be improved by augmenting the training set with the VAE-generated images. It is proposed that the lack of open access eye-tracking datasets could make our approach attractive for further investigation. For instance, VAE models can serve as an alternative method for data generation in a wide range of eye-tracking applications.

Author Contributions: Conceptualization, M.E.; R.C.; G.D. and J.-L.G.; methodology, M.E.; C.L. and R.C.; software, M.E.; C.L. and R.C.; validation, M.E.; G.D. and J.-L.G.; data curation, F.C. and R.C.; writing, M.E.; C.L.; R.C. and F.C.; visualization, R.C. and M.E.; supervision, G.D. and J.-L.G.; project administration, G.D. and J.-L.G.; funding acquisition, G.D. and J.-L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Université de Picardie Jules Verne in France.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from the parents of children who took part in the eye-tracking experiments.

Data Availability Statement: The image dataset is publicly available on the Figshare data repository, https://figshare.com/s/5d4f93395cc49d01e2bd (accessed on 2 May 2021).

Acknowledgments: We acknowledge the support received from Evolucare Technologies company. Run by an enthusiastic team of medical IT professionals, Evolucare aspires to bring a broad range of products and services to the whole healthcare community (www.evolucare.com) (accessed on 2 May 2021).

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

References

1. Zhai, S. What's in the eyes for attentive input. *Commun. ACM* **2003**, *46*, 34–39. [CrossRef]
2. Majaranta, P.; Bulling, A. Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing*; Human–Computer Interaction Series; Fairclough, S., Gilleade, K., Eds.; Springer: London, UK, 2014.
3. Javal, L. Essai sur la physiologie de la lecture. *Ann. d'Oculistique* **1878**, *80*, 240–274.
4. Javal, L. Essai sur la physiologie de la lecture. *Ann. d'Oculistique* **1879**, *82*, 242–253.
5. Huey, E.B. *The Psychology and Pedagogy of Reading*; The Macmillan Company: New York, NY, USA, 1908.
6. Buswell, G.T. *Fundamental Reading Habits: A Study of Their Development*; American Psychological Association: Worcester, MA, USA, 1922.
7. Buswell, G.T. *How People Look at Pictures: A Study of the Psychology and Perception in Art*; University of Chicago Press: Chicago, IL, USA, 1935.
8. Meißner, M.; Pfeiffer, J.; Pfeiffer, T.; Oppewal, H. Combining virtual reality and mobile eye tracking to provide a naturalistic experimental environment for shopper research. *J. Bus. Res.* **2019**, *100*, 445–458. [CrossRef]
9. Meißner, M.; Musalem, A.; Huber, J. Eye tracking reveals processes that enable conjoint choices to become increasingly efficient with practice. *J. Mark. Res.* **2016**, *53*, 1–17. [CrossRef]
10. Cilia, F.; Aubry, A.; Le Driant, B.; Bourdin, B.; Vandromme, L. Visual exploration of dynamic or static joint attention bids in children with autism syndrome disorder. *Front. Psychol.* **2019**, *10*, 2187. [CrossRef]
11. Guo, F.; Ding, Y.; Liu, W.; Liu, C.; Zhang, X. Can eye-tracking data be measured to assess product design: Visual attention mechanism should be considered. *Int. J. Ind. Ergon.* **2016**, *53*, 229–235. [CrossRef]
12. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
13. Dau, H.A.; Bagnall, A.; Kamgar, K.; Yeh, C.C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E. The UCR time series archive. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 1293–1305. [CrossRef]
14. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning internal representations by error propagation. In *Parallel Distributed Processing. Vol 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986.
15. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [CrossRef]
16. Petscharnig, S.; Lux, M.; Chatzichristofis, S. Dimensionality reduction for image features using deep learning and autoencoders. In Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, Florence, Italy, 19–21 June 2017; pp. 1–6.
17. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning (ICML), Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
18. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A.; Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res. (JMLR)* **2010**, *11*, 3371–3408.
19. Lu, X.; Tsao, Y.; Matsuda, S.; Hori, C. Speech enhancement based on deep denoising autoencoder. In Proceedings of the 14th Annual Conference of the International Speech Communication Association, Lyon, France, 25–29 August 2013; pp. 436–440, (INTERSPEECH).
20. Lu, X.; Tsao, Y.; Matsuda, S.; Hori, C. Ensemble modeling of denoising autoencoder for speech spectrum restoration. In Proceedings of the 15th Annual Conference of the International Speech Communication Association, Singapore, 10–20 June 2014; (INTERSPEECH).
21. Gondara, L. Medical image denoising using convolutional denoising autoencoders. In Proceedings of the IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016; pp. 241–246.
22. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.

23. Kingma, D.P.; Welling, M. An introduction to variational autoencoders. *arXiv* **2019**, arXiv:1906.02691. Available online: https://arxiv.org/abs/1906.02691 (accessed on 2 May 2021). [CrossRef]
24. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.M.; Jozefowicz, R.; Bengio, S. Generating sentences from a continuous space. *arXiv* **2015**, arXiv:1511.06349. Available online: https://arxiv.org/abs/1511.06349 (accessed on 2 May 2021).
25. Semeniuta, S.; Severyn, A.; Barth, E. A hybrid convolutional variational autoencoder for text generation. *arXiv* **2017**, arXiv:1702.02390. Available online: https://arxiv.org/abs/1702.02390 (accessed on 2 May 2021).
26. Bachman, P. An architecture for deep, hierarchical generative models. *arXiv* **2016**, arXiv:1612.04739. Available online: https://arxiv.org/abs/1612.04739 (accessed on 2 May 2021).
27. Gulrajani, I.; Kumar, K.; Ahmed, F.; Taiga, A.A.; Visin, F.; Vazquez, D.; Courville, A. Pixelvae: A latent variable model for natural images. *arXiv* **2016**, arXiv:1611.05013.
28. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *arXiv* **2014**, arXiv:1406.2661. Available online: https://arxiv.org/abs/1406.2661 (accessed on 2 May 2021).
29. Wan, Z.; Zhang, Y.; He, H. Variational autoencoder based synthetic data generation for imbalanced learning. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–7.
30. Moreno-Barea, F.J.; Jerez, J.M.; Franco, L. Improving classification accuracy using data augmentation on small data sets. *Expert Syst. Appl.* **2020**, *161*, 113696. [CrossRef]
31. Asuncion, A.; Newman, D. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu (accessed on 2 May 2021).
32. Islam, Z.; Abdel-Aty, M.; Cai, Q.; Yuan, J. Crash data augmentation using variational autoencoder. *Accid. Anal. Prev.* **2021**, *151*, 105950. [CrossRef]
33. Nishizaki, H. Data augmentation and feature extraction using variational autoencoder for acoustic modeling. In Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 12–15 December 2017; pp. 1222–1227.
34. Luo, Y.; Zhu, L.Z.; Wan, Z.Y.; Lu, B.L. Data augmentation for enhancing EEG-based emotion recognition with deep generative models. *J. Neural Eng.* **2020**, *17*, 056021. [CrossRef]
35. Ozdenizci, O.; Erdogmus, D. On the use of generative deep neural networks to synthesize artificial multichannel EEG signals. *arXiv* **2021**, arXiv:2102.08061. Available online: https://arxiv.org/abs/2102.08061 (accessed on 2 May 2021).
36. Biffi, C.; Oktay, O.; Tarroni, G.; Bai, W.; De Marvao, A.; Doumou, G.; Rajchl, M.; Bedair, R.; Prasad, S.; Cook, S.; et al. Learning interpretable anatomical features through deep generative models: Application to cardiac remodeling. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, 16–20 September 2018; pp. 464–471.
37. Pesteie, M.; Abolmaesumi, P.; Rohling, R.N. Adaptive augmentation of medical data using independently conditional variational auto-encoders. *IEEE Trans. Med Imaging* **2019**, *38*, 2807–2820. [CrossRef] [PubMed]
38. Cerrolaza, J.J.; Li, Y.; Biffi, C.; Gomez, A.; Sinclair, M.; Matthew, J.; Knight, C.; Kainz, B.; Rueckert, D. 3d fetal skull reconstruction from 2dus via deep conditional generative networks. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, 16–20 September 2018; pp. 464–471.
39. Biffi, C.; Cerrolaza, J.J.; Tarroni, G.; de Marvao, A.; Cook, S.A.; O'Regan, D.P.; Rueckert, D. 3D high-resolution cardiac segmentation reconstruction from 2D views using conditional variational autoencoders. In Proceedings of the IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 1643–1646.
40. Ma, X.; Deng, Z. Natural eye motion synthesis by modeling gaze-head coupling. In Proceedings of the IEEE Virtual Reality Conference, Lafayette, LA, USA, 14–18 March 2009; pp. 143–150.
41. Duchowski, A.T.; Jörg, S.; Allen, T.N.; Giannopoulos, I.; Krejtz, K. Eye movement synthesis. In Proceedings of the 9th Biennial ACM Symposium on Eye Tracking Research & Applications, Charleston, SC, USA, 14–17 March 2016; pp. 147–154.
42. Lee, S.P.; Badlr, J.B.; Badler, N.I. Eyes alive. In Proceedings of the 29th annual Conference on Computer Graphics and Interactive Techniques, San Antonio, TX, USA, 21–26 July 2002; pp. 637–644.
43. Le, B.H.; Ma, X.; Deng, Z. Live speech driven head-and-eye motion generators. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1902–1914. [CrossRef] [PubMed]
44. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Denver, CO, USA, 27–30 November 1989; pp. 396–404.
45. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
46. Pearlmutter, B.A. Learning state space trajectories in recurrent neural networks. *Neural Comput.* **1989**, *1*, 263–269. [CrossRef]
47. Fuhl, W. Fully Convolutional Neural Networks for Raw Eye Tracking Data Segmentation, Generation, and Reconstruction. *arXiv* **2020**, arXiv:2002.10905. Available online: https://arxiv.org/abs/2002.10905 (accessed on 2 May 2021).
48. Assens, M.; Giro-i-Nieto, X.; McGuinness, K.; O'Connor, N.E. PathGAN: Visual scanpath prediction with generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 406–422.

49. Klein, A.; Yumak, Z.; Beij, A.; van der Stappen, A.F. Data-driven gaze animation using recurrent neural networks. In Proceedings of the ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG), Newcastle Upon Tyne, UK, 28–30 October 2019; pp. 1–11.
50. Zemblys, R.; Niehorster, D.C.; Holmqvist, K. GazeNet: End-to-end eye-movement event detection with deep neural networks. *Behav. Res. Methods* **2019**, *51*, 840–864. [CrossRef]
51. Elbattah, M.; Guérin, J.; Carette, R.; Cilia, F.; Dequen, G. Generative modeling of synthetic eye-tracking data: NLP-based approach with recurrent neural networks. In Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI), Budapest, Hungary, 2–4 November 2020; Volume 1, pp. 479–484.
52. Carette, R.; Elbattah, M.; Dequen, G.; Guérin, J.; Cilia, F. Visualization of eye-tracking patterns in autism spectrum disorder: Method and dataset. In Proceedings of the 13th International Conference on Digital Information Management, Berlin, Germany, 24–26 September 2018.
53. Guillon, Q.; Hadjikhani, N.; Baduel, S.; Rogé, B. Visual social attention in autism spectrum disorder: Insights from eye tracking studies. *Neurosci. Biobehav. Rev.* **2014**, *42*, 279–297. [CrossRef]
54. Henderson, J.M. Human gaze control during real-world scene perception. *Trends Cogn. Sci.* **2003**, *7*, 498–504. [CrossRef] [PubMed]
55. Jacob, R.J. Eye tracking in advanced interface design. In *Virtual Environments and Advanced Interface Design*; Barfield, W., Furness, T.A., Eds.; Oxford University Press: New York, NY, USA, 1995; pp. 258–288.
56. Goldberg, J.H.; Helfman, J.I. Visual scanpath representation. In Proceedings of the 2010 Symposium on Eye-Tracking Research Applications, Austin, TX, USA, 22–24 March 2010; pp. 203–210.
57. Noton, D.; Stark, L. Scanpaths in eye movements during pattern perception. *Science* **1971**, *171*, 308–311. [CrossRef]
58. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
59. Visualization of Eye-Tracking Scanpaths in Autism Spectrum Disorder: Image Dataset. 2019. Available online: https://figshare.com/s/5d4f93395cc49d01e2bd (accessed on 2 May 2021).
60. Bradski, G. The OpenCV library. *Dr. Dobb's J. Softw. Tools* **2000**, *25*, 120–125.
61. Chollet, F.K. GitHub Repository. 2015. Available online: https://github.com/fchollet/keras (accessed on 2 May 2021).
62. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th (USENIX) Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
63. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
64. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res. (JMLR)* **2014**, *15*, 1929–1958.
65. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res. (JMLR)* **2011**, *12*, 2825–2830.
66. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
67. Wang, J.; Perez, L. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621. Available online: https://arxiv.org/abs/1712.04621 (accessed on 2 May 2021).
68. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
69. Taylor, L.; Nitschke, G. Improving deep learning using generic data augmentation. *arXiv* **2017**, arXiv:1708.06020. Available online: https://arxiv.org/abs/1708.06020 (accessed on 2 May 2021).