

Article

Visual Saliency and Image Reconstruction from EEG Signals via an Effective Geometric Deep Network-Based Generative Adversarial Network

Nastaran Khaleghi ¹, Tohid Yousefi Rezaii ^{1,*}, Soosan Beheshti ², Saeed Meshgini ¹, Sobhan Sheykhivand ^{1,*} and Sebelan Danishvar ³ 

- ¹ Biomedical Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51666-16471, Iran
- ² Department of Electrical, Computer and Biomedical Engineering, Toronto Metropolitan University, 350 Victoria St., Toronto, ON M5B 2K3, Canada
- ³ College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK
- * Correspondence: yousefi@tabrizu.ac.ir (T.Y.R.); s.sheykhivand@tabrizu.ac.ir (S.S.)

Abstract: Reaching out the function of the brain in perceiving input data from the outside world is one of the great targets of neuroscience. Neural decoding helps us to model the connection between brain activities and the visual stimulation. The reconstruction of images from brain activity can be achieved through this modelling. Recent studies have shown that brain activity is impressed by visual saliency, the important parts of an image stimuli. In this paper, a deep model is proposed to reconstruct the image stimuli from electroencephalogram (EEG) recordings via visual saliency. To this end, the proposed geometric deep network-based generative adversarial network (GDN-GAN) is trained to map the EEG signals to the visual saliency maps corresponding to each image. The first part of the proposed GDN-GAN consists of Chebyshev graph convolutional layers. The input of the GDN part of the proposed network is the functional connectivity-based graph representation of the EEG channels. The output of the GDN is imposed to the GAN part of the proposed network to reconstruct the image saliency. The proposed GDN-GAN is trained using the Google Colaboratory Pro platform. The saliency metrics validate the viability and efficiency of the proposed saliency reconstruction network. The weights of the trained network are used as initial weights to reconstruct the grayscale image stimuli. The proposed network realizes the image reconstruction from EEG signals.

Keywords: visual saliency; electroencephalogram; image reconstruction; geometric deep network; generative adversarial network



Citation: Khaleghi, N.; Rezaii, T.Y.; Beheshti, S.; Meshgini, S.; Sheykhivand, S.; Danishvar, S. Visual Saliency and Image Reconstruction from EEG Signals via an Effective Geometric Deep Network-Based Generative Adversarial Network. *Electronics* **2022**, *11*, 3637. <https://doi.org/10.3390/electronics11213637>

Academic Editor: Jichai Jeong

Received: 11 October 2022

Accepted: 2 November 2022

Published: 7 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Elucidating the function of the brain in perceiving the input data from the outside world is of particular importance to help improve biometric innovations and BCI challenges. Brain recording techniques play an essential role in realizing this concept. The recordings pave the way to modeling the representation of the information in the brain to recognize how it works to perceive the input data. This modeling has been the subject of a number of studies in the fields of brain encoding and decoding [1].

EEG as one of the most popular non-invasive brain recording methods has been used vastly in studies related to understanding the brain activities in various circumstances; for example, experiments concerning attention, memory [2], motor control [3], drowsiness in the brain [4], EEG-based driving safety monitoring [5], emotions [6], driver fatigue [7,8], visual decoding [9,10], brain activities during sleep [11], and movement intention detection [12]. Some studies have focused on the relation between visual input to the brain and EEG recordings. In 2010, Ghebreab et al. [13] investigated the recorded EEG signals in response to natural visual stimulation, and the prediction of visual inputs was realized

using EEG responses. A better accuracy was achieved in comparison to a similar work by Kay et al. in 2008 [14]. These studies have had the potential to reveal the effects of visual features such as color [15], orientation [16], and position [17] on the brain signals of the visual cortex.

The representation of visual stimuli in the brain relates to important points of the picture. To illustrate the priority of a location in a visual image to represent in the brain, and to identify them efficiently, the concept of a saliency map was first proposed by Koch and Ullman in 1985 [18]. Followed by the concept introduced by Koch and Ullman, Itti et al. in 1998 introduced a computational model corresponding to the understanding of the saliency map [19]. Following the work of Itti et al. in 1998 [19], detecting rarity, distinctiveness, and uniqueness in a scene is compulsory for salient object detection. Based on the proposed model by Itti [19], many models have been developed for predicting image saliency.

Realizing how the salient region affects the brain signal is of great importance to understanding how the visual system works. Although some works have been made to explore the relationship between the brain activity through recorded EEG signals and the salient regions of the visual stimuli, the mapping of the EEG signals to image saliency has not been realized. Moreover, the use of dynamic information between the connected EEG channels according to the functional connectivity between different brain regions has not been considered to explore the connection between brain activity and salient regions.

To achieve an efficient mapping of EEG signals to the salient region corresponding to the visual stimuli, a deep network based on the graph representation of EEG records is introduced. The mapping would extract the visual saliency map related to the recorded EEG signals. The proposed network consists of two parts, including the geometric network and the generative adversarial network. The graph representation of the EEG records facilitates to exploit the functional connectivity between different channels in each EEG recordings in the classification procedure of the geometric deep network part of the proposed network. The overall model realizes the visual saliency reconstruction through the EEG records.

The contributions made by this article can be highlighted as:

- (i) It provides an efficient deep network to extract a saliency map of visual stimuli from visually provoked EEG signals.
- (ii) Reconstruction of the visual stimuli is possible through the proposed deep network.
- (iii) It provides a geometric visual decoding network for extracting features from the EEG recordings to identify 40 different patterns of EEG signals corresponding to 40 image categories.
- (iv) A graph representation of the EEG channels is imposed as an input to the proposed GDN-GAN, in which functional connectivity between 128 EEG channels is employed to construct the graph.
- (v) In the proposed method, the time samples of EEG channels are used directly as the graph nodes to remove the feature extraction phase and to decrease the computational burden.
- (vi) For the first time, it presents a model to connect the EEG recordings, visual saliency, and visual stimuli together.
- (vii) For the first time, it proposes a fine-tuning process to realize image reconstruction from EEG signals via visual saliency reconstruction.

The remainder of this paper is arranged as follows. Section 2 reviews the related works. Section 3 provides the details of the EEG-ImageNet database, and reviews the mathematical preliminaries of graph convolution and generative adversarial networks. Section 4 describes the details and the structure of the proposed framework for EEG-based visual saliency detection and visual stimuli reconstruction. Section 5 provides and presents the experimental results, and validates the performance of the proposed framework compared with the state-of-the-art methods, and finally, the conclusions are provided in Section 6.

2. Related Work

In a number of early works on salient region detection [20–22], saliency was considered as being unique, and was frequently calculated as center–surround contrast for every pixel. In 2005, Hu et al. [23] used generalized principal component analysis (GPCA) [24] to compute salient regions. GPCA has been used to estimate the linear subspaces of the mapped image without segmenting the image, and salient regions have been determined by considering the geometric properties and feature contrast of regions. Rosin [25] proposed an approach for salient object detection, which has required very simple operations for each pixel, such as moment preserving binarization, edge detection, and threshold decomposition. Valenti et al. [26] proposed an isophote-based framework where isocenter clustering, color boosting, and curvedness have been used for the estimation of the saliency map. In addition, some supervised learning-based models for saliency detection were proposed, such as support vector machine in 2010 with Zhong et al. [27], regression in 2016 with Zhou et al., and neural networks with Duan in 2016 [28].

Some of the saliency detection methods are based on models developed for simulating the visual attention processes. Visual attention is a selective procedure that occurs for understanding the the visual input to the brain from the surrounding environment. Neisser, in 1967 [29], suggested that bottom-up and top-down processes occur in the brain during the time of the processing objects of a visual scene. Bottom-up is a pre-attentive that considers primitive feature-driven, and top-down is a task-driven attentive model. According to these processes, bottom-up-based models, top-down-based models, and some others, considering both of the processes, have been proposed for visual attention.

An analysis of a bottom-up-based visual attention mechanism has resulted in bottom-up-based saliency detection models. It is a fast process and it uses low-level visual properties such as color, intensity, and orientation. A number of researchers have made efforts to improve the performance of the bottom-up-based saliency models. In 2013, Zhang and Sclaroff measured the contour information of regions using a set of Boolean maps to segment the salient objects from the background, and the efficiency of the model was demonstrated by five sets of eye tracking databases [30]. In 2015, Mauthner et al. proposed an estimation of the joint distribution of motion and color features based on Gestalt theory, in which the local and global foreground saliency likelihoods have been described with an encoding vector, and these individual likelihoods generated the final saliency map [31].

Top-down visual attention process resulted in the top-down-based saliency map detection models. The intention and thoughts have been involved in this process, and it is impressed by the prior knowledge and given task to the brain. To realize the difference of the impact of these two processes on saliency models, consider an image including two kinds of fruits. Two kinds will have the same saliency level in the bottom-up model. However, in the top-down model, the given task will have an impact on the saliency levels of each kind. Top-down saliency-based models, as in the work of Xu et al. in 2014 [32], have been conducted through contextual guidance and pre-defining of the discriminant features and allocating learned weights for different features, as performed by Zhao and Koch in 2011 [33], and in 2017, Yang [34] adapted feature space in a supervised manner to obtain the saliency output.

The third category is an integration of the bottom-up and top-down saliency detection models. The detection of possible salient regions is achieved through the bottom-up process, and the effect of the given task is processed according to the top-down model.

After designating the process model between these three models, the features extracted from every pixel in the input, or the spatial attributes according to the regions are considered to compute the saliency features. Although real-time saliency detection with hand-crafted features has good performance, it does not work well in challenging scenarios to capture salient objects. One of the proposed solutions to these challenges is using neural networks [35,36]. One of the most popular networks in machine learning are convolutional neural networks (CNNs) [35], and they have been implemented to solve a number of vision problems such as edge detection, semantic segmentation [37], and object

recognition [38]. Recently, in the work by Shengfeng He et al. and Ghanbin Li et al. [39,40], the effectiveness of CNNs has been shown when applied to salient object detection. A series of techniques has been proposed to learn saliency representations from large amounts of data by exploiting the different architectures of CNNs. Some of the models proposed for saliency detection via neural networks use multilayer perceptrons (MLPs). In these models, the input image is usually oversegmented into small regions and feature extraction is performed using a CNN. The extracted features are fed to an MLP to determine the saliency value of each small region. The saliency problem in [39] has been solved using the one-dimensional convolution-based methods by He et al. Li and Yu [40] have utilized a pre-trained CNN as a feature extractor, such that the input image has been decomposed into a series of non-overlapping regions and a CNN with three different-scale inputs has been proposed to extract features from the decomposed regions. Advanced features at different scales have been captured using three subnetworks of the proposed CNN, and have been concatenated to feed into a small MLP with only two fully connected layers. These dense layers act as a regressor to output a distribution over binary saliency labels.

Two recently proposed deep learning-based saliency models are salicon [41,42] and salnet [43]. Like other saliency detection methods, the purpose of the salicon is to realize and to predict visual saliency. This model has used the coefficients of pre-trained AlexNet, VGG-16, and GoogleNet. The last layer of the proposed salicon is a convolutional layer that is used to extract the salient points. The initial parameters have been determined using the pre-trained network based on ImageNet dataset, and the back propagation has been used to optimize the evaluation criterion, in spite of previous approaches that used support vector machine. The training process in salnet has been achieved using the Euclidean distance between the mapped predicted salient points and the ground truth pixels. A shallow and a deep network have been presented. The shallow net consists of three convolutional layers and two fully connected layers with trained weights. ReLU is used as the activation function of each layers of shallow net. The deep network consists of 10 layers and 25.8 million parameters.

In recent years, some efforts have been made to understand the connection between the visual saliency content and the brain activity. In 2018, Zhen Liang et al. [44] presented a model to study this connection and extracted sets of efficient features of EEG signals to map to the visual salient related features of the video stimuli. The model has used the work of Tavakoli et al. in 2017 [45]. The reconstruction of the features of the salient visual points based on the features of the EEG signal has been performed with good accuracy in [44], and prediction of the temporal distribution of salient visual points has been done using EEG signals recorded in a real environment. In another study [46], the identification of the objects in images recorded by robots was the purpose of the study, and a method based on P300 wave was applied to identify the objects. The significant challenge for extracting the objects of interest in navigating the robots is how to use a machine to extract the objects of interest for humans. The combination of a P300-based BCI and a Fuzzy color extractor has been applied to identify the region of interest. Humbeck et al. [47] have presented a model for calculating the importance of the salient points for the fixation positions. Brain function related to the extracted model has been studied using the eye-tracker and recording the EEG signal. An evaluation of the connection between the importance of salient points and the amplitude of the EEG signal has been done via this modeling. A multimodal learning of EEG and image modalities has been performed in [48] to achieve a Siamese network for image saliency detection. The idea of the work in [48] is the training of a common space of brain signal and image input stimuli by maximizing a compatibility function between these two embeddings of each modality. The estimation of saliency is achieved by masking the image with different scales of image patch and computing the corresponding variation in compatibility. This process is performed at multiple image scales, and results in a saliency map of the image.

In this article, we propose a novel deep network for mapping the visually provoked EEG signals to image saliency. In the next section, we explain the database settings and the mathematical background of the proposed method.

3. Materials and Methods

The EEG-ImageNet database is used in this paper and is explained in detail in this section. The mathematical background of Chebyshev graph convolution will be explained to know the function of a convolutional layer of the geometric deep network. Furthermore, we have an overview of generative adversarial networks and saliency evaluation metrics.

3.1. Database Settings

In this section, the details of the EEG-ImageNet database is described. This dataset is publicly available in perceive lab [48,49]. The EEG-ImageNet dataset has been recorded using a 128-channel cap (actiCAP 128Ch) [50]. Figure 1 illustrates the EEG placement according to this standard. The EEG-ImageNet includes the EEG signals of six human subjects produced as the result of visual stimulation.

The visual stimulation used in this research contains 40 categories of different images of the ImageNet database [51], comprising ‘sorrel’, ‘Parachute’, ‘Iron’, ‘Anemone’, ‘Espresso maker’, ‘Coffee mug’, ‘Bike’, ‘Revolver’, ‘Panda’, ‘Daisy’, ‘Canoe’, ‘Lycaenid’, ‘Dog’, ‘Running Shoe’, ‘Lantern’, ‘Cellular phone’, ‘Golf ball’, ‘Computer’, ‘Broom’, ‘Pizza’, ‘Missile’, ‘Capuchin’, ‘Pool table’, ‘Mailbag’, ‘Convertible’, ‘Folding chair’, ‘Pajama’, ‘Mitten’, ‘Electric guitar’, ‘Reflex camera’, ‘Piano’, ‘Mountain tent’, ‘Banana’, ‘Bolete’, ‘Watch’, ‘Elephant’, ‘Airliner’, ‘Locomotive’, ‘Telescope’, ‘Egyptian cat’.

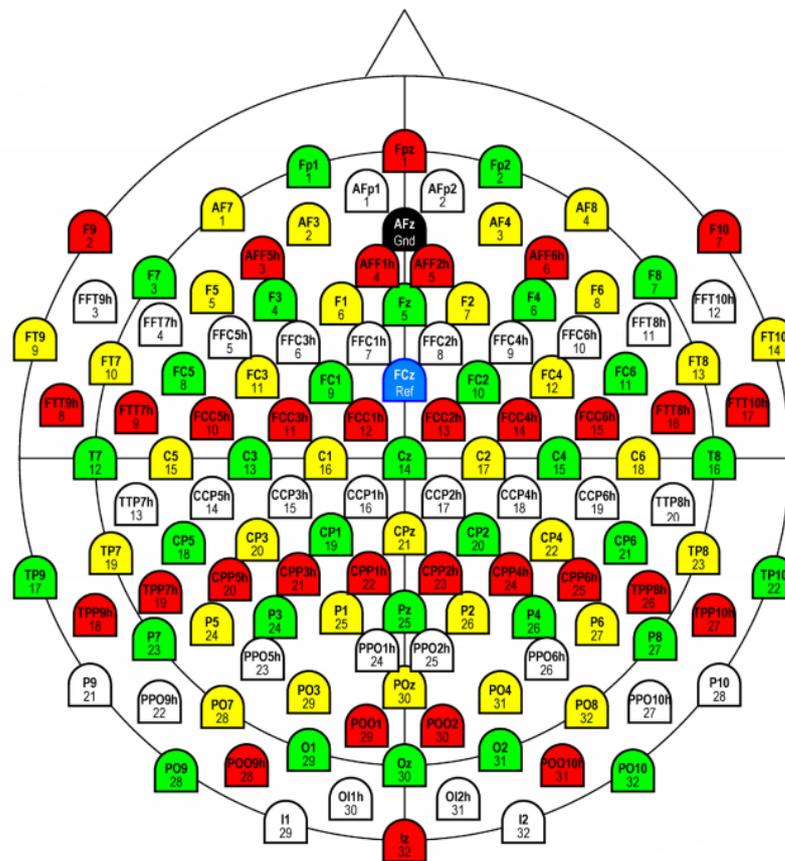


Figure 1. ActiCAP128 standard-2: EEG channel placement; each channel in this figure is identified by a prefix letter referring to brain cortex (Fp: frontal, T: temporal, C: central, P: parietal, O: occipital) and a number indicating the electrode [50].

To record the data as described in [49], each image has been shown on the computer screen for 500 ms, and a set of 50 images of each category has been shown to each of the subjects in 25 s. A total running time of 1400 s has been dedicated to recording the EEG data of each subject. The total number of records used in our experiments is equal to 11,965.

3.2. Chebyshev Graph Convolution

In this section, we make a brief overview on graph convolution. The research of Michaël Defferrard et al. [52] was the cause of the popularization of graph signal processing (GSP). The functions in GSP take into consideration the properties of the graph’s components, and also the structure of the graph. GSP is used to expand convolutions to the graph domain, and this field of research uses signal processing functions such as the Fourier transform and applies them to the graphs. The use of Fourier transform in GSP leads to graph spectral filtering, also called graph convolution [53].

We explain graph convolution, as described in [53]. Let $D \in \mathbb{R}^{(N \times N)}$ and $W \in \mathbb{R}^{(N \times N)}$, respectively; denote the diagonal degree matrix and the adjacency matrix of a graph. The i -th diagonal element of the degree matrix can be calculated by

$$D_{ii} = \sum_j w_{ij} \tag{1}$$

Then, L , the Laplacian matrix of the graph, is expressed as

$$L = D - W \in \mathbb{R}^{(N \times N)} \tag{2}$$

The basis functions in the graph domain are calculated according to the eigenvectors of the graph Laplacian matrix. The eigenvectors of the graph Laplacian matrix denoted by U can be acquired via the singular value decomposition (SVD):

$$L = U \Lambda U^T \tag{3}$$

in which the columns of $U = [u_0, \dots, u_{N-1}] \in \mathbb{R}^{(N \times N)}$ constitute the Fourier basis, and $\Lambda = \text{diag}([\lambda_0, \dots, \lambda_{N-1}])$ is a diagonal matrix. Calculating the eigenvectors of the Laplacian returns the Fourier basis for the graph.

The graph convolution operation is defined as (4). Substituting $f(L)$ in (4) with the Chebyshev polynomial expansion of L , we will have the Chebyshev graph convolution of X .

$$Y = f(L)X = U f(\Lambda) (U^T) X \tag{4}$$

where L can be calculated from W based on (2), and the calculation of Λ can be conducted using (3).

The approximation of the $f(\Lambda)$ is performed via the K -order Chebyshev polynomials. Approximating the $f(\Lambda)$ function is accomplished via the normalized version of Λ . The largest element among the diagonal entries of Λ is defined by λ_{Max} , and the normalized Λ is as follows:

$$\hat{\Lambda} = 2\Lambda / \lambda_{max} - I_N \tag{5}$$

where I_N is the $N \times N$ identity matrix, and the diagonal elements of $\hat{\Lambda}$ lie in the interval of $[-1,1]$. The approximation of $g(\Lambda)$ based on the K -order Chebyshev polynomials framework is as follows:

$$f(\Lambda) = \sum_{k=0}^K (\theta_k) (T_k(\hat{\Lambda})) \tag{6}$$

where θ_k denotes the coefficient of Chebyshev polynomials, and $T_k(\hat{\Lambda})$ can be acquired according to the following formulas:

$$\{T_0(\hat{\Lambda}) = 1, T_1(\hat{\Lambda}) = \hat{\Lambda}, T_k(\hat{\Lambda}) = 2(\hat{\Lambda})(T_{k-1}(\hat{\Lambda})) - T_{k-2}(\hat{\Lambda})\} \tag{7}$$

According to (6), the graph convolution operation defined in (4) can be expressed using (7), as illustrated in (8).

$$\begin{aligned}
 Y &= \mathbf{U}f(\Lambda)(\mathbf{U}^T)\mathbf{X} \\
 &= \sum_{k=0}^K \mathbf{U} \left(\begin{bmatrix} \theta_k T_k(\hat{\lambda}_0) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \theta_k T_k(\hat{\lambda}_{N-1}) \end{bmatrix} \right) (\mathbf{U}^T)\mathbf{X} \\
 &= \sum_{k=0}^K \theta_k T_k(\hat{\mathbf{L}})\mathbf{X}
 \end{aligned} \tag{8}$$

where $\hat{\mathbf{L}} = 2(L/\lambda_{max}) - \mathbf{I}_N$ is the normalized Laplacian matrix.

The expression of Chebyshev graph convolution in (8) shows that it is equivalent to the combination of the convolutional results of x , with each components of the Chebyshev polynomial [53].

3.3. Generative Adversarial Network

Generative deep modeling is considered as an unsupervised learning task that discovers and learns the contents in input data in such a way that the extracted model can be used to generate new examples that could have been extracted plausibly from the original dataset. A spatial case of generative models is the generative adversarial network (GAN) that constructs two sub-networks, including generator and discriminator, to solve the problem.

The generator network is trained to generate new examples, and the classification of these examples as either real or fake is performed through the discriminator sub-network. The two sub-networks are trained in an adversarial way, such that the generator part outputs some examples to the real data, and the discriminator part is fooled and cannot diagnose a difference between the real domain and the generated examples. The generator should learn how to generate data in such a way that detection between fake and real cannot happen by the discriminator.

The two sub-networks are trained simultaneously, such that a generative model G settles random vector y adapted from preceding distribution $P(y)$ into the domain data; additionally, a discriminative model D tries to detect dissimilarity between true examples obtained from the training input data domain P and simulated examples from the generator G .

Such networks are trained inconsistently until none of them can make additional progress against one another. An illustration of the GAN objective function is depicted as follows:

$$\min_G \max_D V(D, G) = \min_G \max_D [E_{X \sim P_{data}(x)} [\log D(x)] + E_{Y \sim p_{y(y)}} [\log(1 - D(G(y)))] \tag{9}$$

In the cost function of GAN as in (9), x is the real data and y signifies the feature vector imposed to the generator; furthermore, $G(y)$ portrays the output of the generator, given a feature vector y . $D(x)$ is the output of the discriminator with real image data, and has to be as close as possible to 1, to perform better. $D(G(y))$ represents the output of the discriminator, considering the generated samples indicated with $G(y)$. The probability density of x and y is represented accordingly, with $P_{data}(x)$ and $p_{y(y)}$ in the cost function of (9).

In the training procedure of a GAN, G is trained in a way that reduces $\log(1 - D(G(y)))$ to mislead discriminator D . Contrarily, D is trained so that it can increase the likelihood that the generated data is analogous to the real data, and the likelihood would be near to 1 and far from 0, which is the likelihood of being fake data.

3.4. Saliency Metrics

In this section, saliency evaluation metrics are described. Ground truth is necessary for calculating these metrics. Another input would be the saliency map. Considering these two inputs and computing these metrics, the degree of the similarity between them would be available.

Similarity (SIM) is a metric for measuring the intersection between distributions [54]. The similarity between two distributions can be measured with this metric. The input maps are normalized, and the sum of the minimum values at each pixel is computed as SIM. Considering a saliency map SM and a continuous fixation map F^M :

$$SIM(SM, P^t) = \sum_j \min(SM_j, F_j^M) \tag{10}$$

where $\sum_j SM_j = \sum_j (F_j^M) = 1$

In (10), iteration is made for discrete pixel locations j . For the same distributions, SIM is equal to one, while if there is no overlap between distributions, SIM would be zero.

Structural similarity (SSIM) is calculated using the different windows of an image [55]. Considering two windows g and h of size $K \times K$, SSIM can be calculated as follows:

$$SSIM(g, h) = \frac{(2 \times \mu_g \mu_h + c_1)(2 \times \sigma_{gh} + c_2)}{(\mu_g^2 + \mu_h^2 + c_1)(\sigma_g^2 + \sigma_h^2 + c_2)} \tag{11}$$

- μ_g is the mean-value of g ;
- μ_h is the mean-value of h ;
- σ_g^2 is the variance of g ;
- σ_h^2 is the variance of h ;
- σ_{gh} is the covariance of g and h ;
- $c_1 = ((k_1)L)^2$;
- $c_2 = ((k_2)L)^2$;
- two variables to stabilize the division with a weak denominator;
- L is the dynamic range of the pixel-values (typically, this is $(2^{(bits\ per\ pixel)} - 1)$);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

Pearson’s correlation coefficient (CC) is a metric for evaluating the linear relationship between distributions [54]. Considering saliency and fixation maps, SM and F^M , CC can be calculated as follows:

$$CC(SM, F^M) = \sigma(SM, F^M) / \sigma(SM) \times \sigma(F^M) \tag{12}$$

In (12), $\sigma(SM, F^M)$ is the covariance of SM and F^M .

CC is invariant to linear transformations. This metric corresponds to a symmetric function, and it is the reason for why it would deal equally with false positives and false negatives. If both the saliency map and ground truth have similar magnitudes, high positive CC values occur.

The normalized scanpath saliency (NSS) is computed as the average normalized saliency at fixated locations. Because the mean-value of saliency is subtracted during computation, NSS is robust against linear transformations [54]. This metric is sensitive to false positives. False positives would contribute to lower the normalized saliency value at each fixation location, and the overall NSS would be reduced. Given a saliency map SM and a binary map of fixation locations F^B :

$$\begin{aligned}
 NSS(SM, F^B) &= (1/K) \left(\sum_i S\bar{M}_i \times (F_i)^B \right) \\
 \text{where } K &= \sum_i (F_i)^B \\
 \text{and } S\bar{M} &= SM - \mu(SM) / \sigma(SM)
 \end{aligned} \tag{13}$$

where i indexes the i th pixel, and K is the total number of fixated pixels.

The shuffled area under the ROC curve (s-AUC) is a metric that uses the receiver operating characteristic (ROC) curve. Considering various thresholds of the saliency map, the ROC curve is obtained by plotting the true positives against the false positives. This metric needs sampling thresholds to obtain the ROC curve [54]. An important issue in computing the s-AUC metric is how to sample thresholds to approximate the ROC curve. Sampling the threshold is performed at a fixed step size (from 0 to 1 by increments of 0.1), and the calculation of this metric would be realized.

4. Proposed Geometric Deep Network-Based Generative Adversarial Network

The details of the proposed geometric deep network-based generative adversarial network (GDN-GAN) for visual saliency and image reconstruction is explained in this section, and the structure of the proposed framework is shown in Figure 2.

4.1. The Proposed Network Architecture

The proposed geometric deep network-based generative adversarial network (GDN-GAN) architecture contains two parts of sequential layers. Each part consists of a number of layers to map the EEG signals to the image saliency and to reconstruct the image stimuli. The GDN part extracts discriminative features of the different categories that the input belongs to. The GAN part maps the extracted feature vector to the image saliency. The trained weight vectors of the network parameters are used as initial weight vectors to train the network to map the EEG signal to the image stimuli and realize the image reconstruction from the brain activity. The detailed schematic of the proposed network architecture is represented in Figure 2. After functional connectivity-based graph embedding of the recorded visually evoked EEG signals, it imposed to the GDN part of the proposed network.

Figure 3 shows the detailed structure of the first GDN part of the network, and as it can be seen, it includes four layers of graph convolution. The Laplacian of the input graph is necessary to estimate the graph convolution of the input in each layer. The estimation is performed via the Chebyshev polynomial expansion of the Laplacian graph. Then, a batch normalization filters the output of each layer. After the fourth graph convolution layer, the extracted feature vector is passed through a dropout layer. Then, the flattened output of the dropout layer is fed to a dense fully connected layer, and a log-softmax function is used for the classification of the output of the fully connected layer.

The weights are trained to classify 40 categories of image stimulation and the flattened vector before the last dense layer is used to impose to the next GAN part of the network. The dimension of the flattened vector is equal to 6400.

Figure 4 illustrates the differences in the dimensions of every layer of the GDN. As each of the recorded EEG signals includes 128 channels, the constructed graph as input to the proposed GDN part in Figure 2 has 128 nodes. Every node in the constructed graph includes 440 samples. The input dimension of the graph convolutional layer independent of the number of graph nodes is considered to be 440, equal to the number of samples in each node. The obtained graph with the first graph convolution has 128 nodes with 440 samples in each vertex. A graph with 128 nodes with 220 samples in each vertex is the output of the second graph convolution, and the output of the third graph convolution operation is a 128-node graph with 110 samples in each of the nodes, and accordingly, the graph output of the fourth layer has 50 samples in each node. The attained 128-node graph with 50 samples in each node outputs a vector with 6400 elements. The flattened vector is

passed through a dense layer and the dimensions of the inputs and outputs of the dense layer are 6400 and 40, respectively.

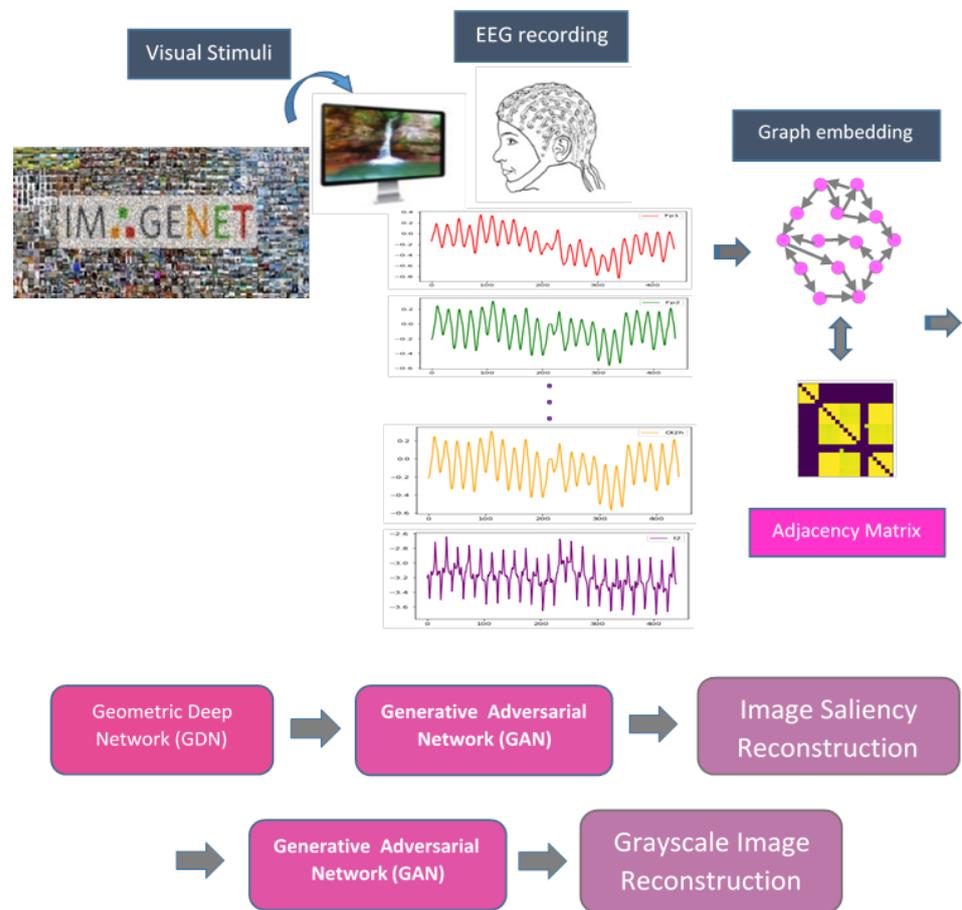


Figure 2. The schematic overview of the proposed GDN-GAN.

Table 1 shows the dimensions of weight tensors for different layers of GDN part of the proposed GDN-GAN. Moreover, it shows the total number of parameters of graph convolutional layers according to the order of the Chebyshev polynomial expansion considered for each layer.

Figure 5 illustrates different layers of the GAN part of the proposed network. Tables 2 and 3 give information about the details of the generator and discriminator parts of the proposed network, respectively. The generator part of the GAN consists of two dense layers, followed by four sequential transposed two-dimensional (2D) convolution layers, and one 2D convolution layer and leaky rectified linear unit is used as the activation function of all layers except for the first dense layer. The output of the GDN is imposed to the generator, the input dimension of the generator is equal to 6400, and the output dimension of the first layer is 100. The output dimension of the second dense layer is equal to 20,000. The reshape layer converts the shape of the 20,000-dimensional vector to a three-dimensional output to impose to a 2D convolutional layer. Eight two-dimensional output with (50, 50) dimensions are imposed to the first transposed two-dimensional convolution layer. The kernel size in each of the transposed convolutional layers is equal to 4×4 , and the number of filters in each of them is equal to eight. The size of the strides in the first transposed convolution layer is equal to 2×2 , in the second transposed convolutional layer, it is equal to 3×3 , and in the next two transposed layers, it is equal to 1×1 . The output of the fourth transposed convolution 2D is imposed to the 2D convolution layer. The kernel size of the 2D convolution layer is considered as being equal to 2×2 , and the

size of the strides in this layer is equal to 2×2 . The output dimension of this layer is equal to $(299, 299)$, and is imposed to the last reshape layer. The output of the generator is a 299×299 -dimensional image. The schematic view of the outputs of each layer and the differences in the dimensions of the generator part of the proposed GDN-GAN are illustrated in Figure 6.

Table 1. Number of training parameters of the GDN part of the proposed GDN-GAN.

Layer	Layer Type	Shape of Weight Array	Shape of Bias	Number of Parameters
1	graph convolution	[1, 440, 440]	[440]	194,040
2	batch normalization	[440]	[440]	880
3	graph convolution	[1, 440, 220]	[220]	97,020
4	batch normalization	[220]	[220]	440
5	graph convolution	[1, 220, 110]	[110]	24,310
6	batch normalization	[110]	[110]	220
7	graph convolution	[1, 110, 50]	[50]	5550
8	batch normalization	[110]	[110]	220
9	dense layer	[6400, 40]	[40]	256,040

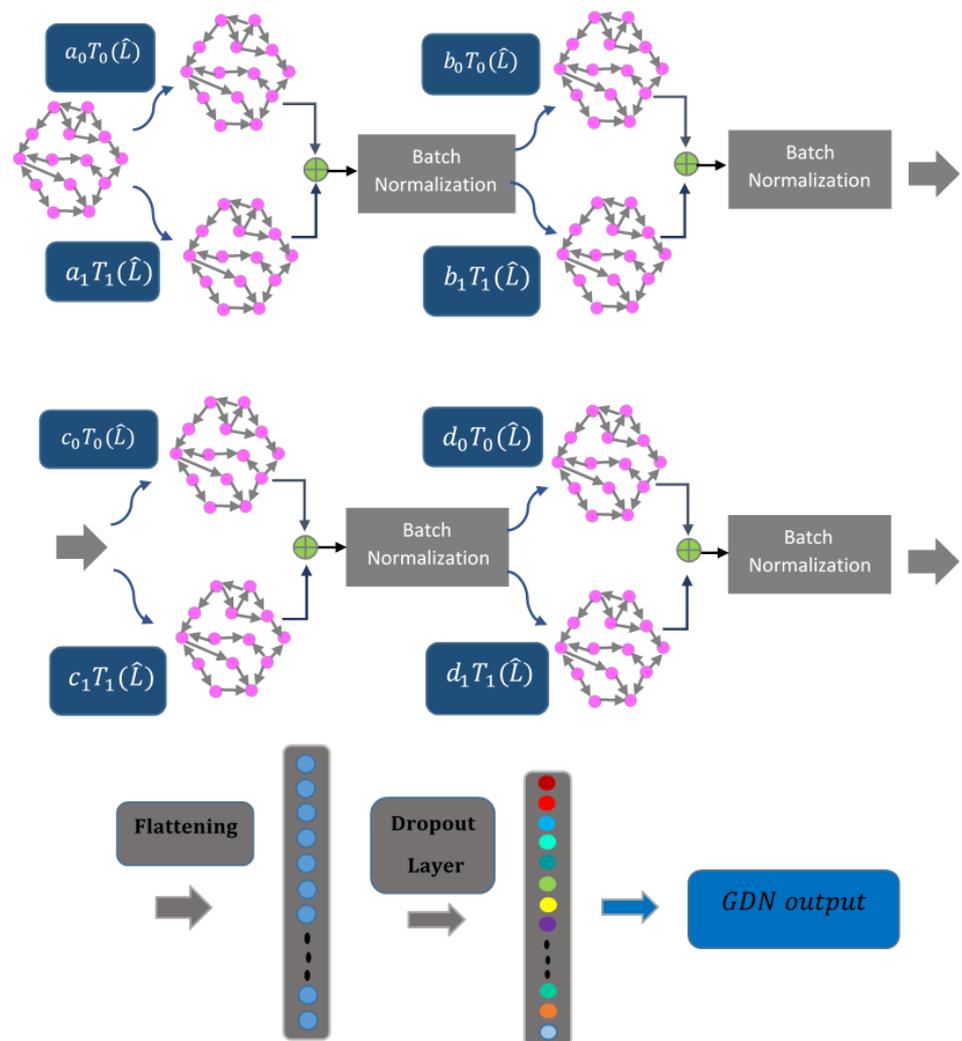


Figure 3. The network architecture of the GDN part of the proposed GDN-GAN.

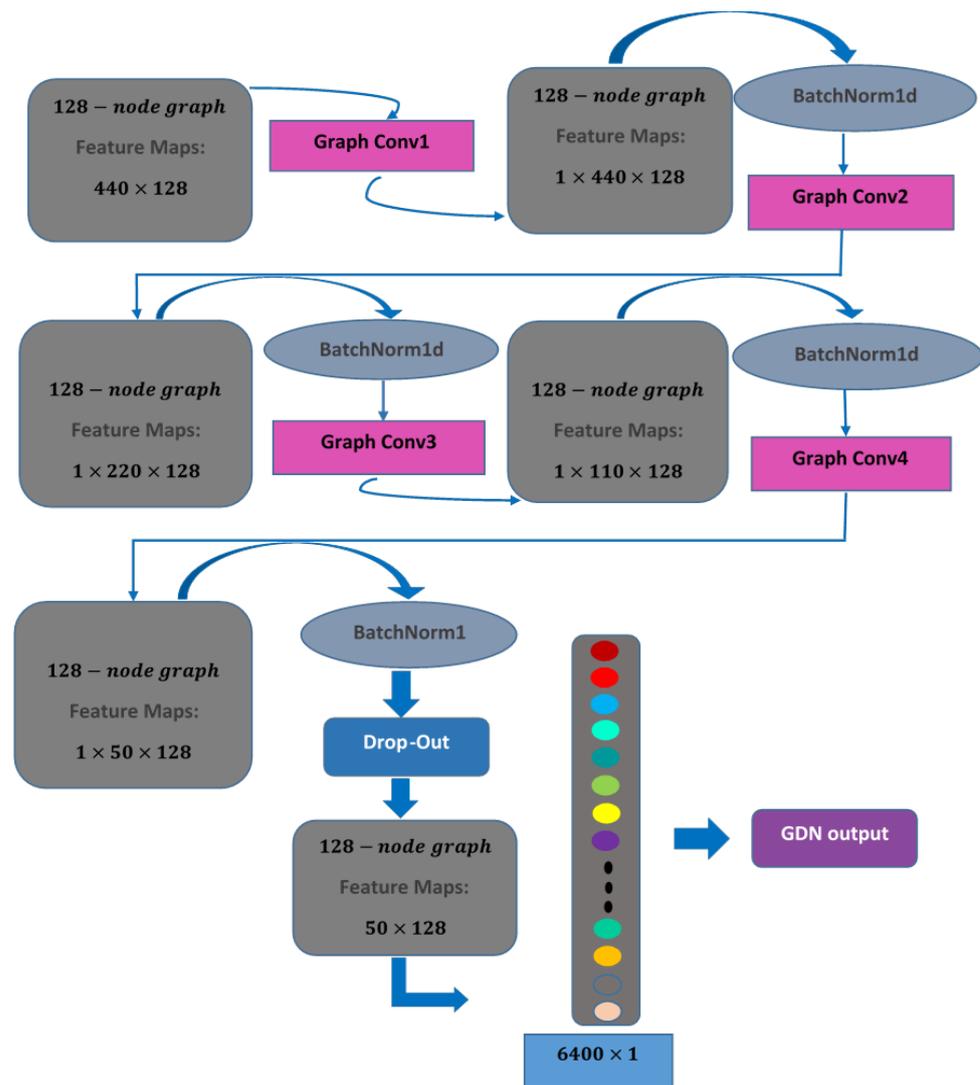


Figure 4. Dimensions of different layers of the GDN part of the proposed GDN-GAN.

Table 2. Details of the generator part of the proposed GDN-GAN architecture.

Layer	Layer Type	Activation Function	Output Shape	Size of Kernel	Strides	Number of Filters	Padding	Number of Parameters
1	Dense Layer		(None, 100)					640,000
2	Dense Layer	LeakyReLU (alpha = 0.2)	(None, 20,000)					2,000,000
3	Reshape		(None, 50, 50, 8)					0
4	Transposed Convolution 2D	LeakyReLU (alpha = 0.2)	(None, 100, 100, 8)	4 × 4	2 × 2	8	yes/same	1024
5	Transposed Convolution 2D	LeakyReLU (alpha = 0.2)	(None, 300, 300, 8)	4 × 4	3 × 3	8	yes/same	1024
6	Transposed Convolution 2D	LeakyReLU (alpha = 0.2)	(None, 300, 300, 8)	4 × 4	1 × 1	8	yes/same	1024
7	Transposed Convolution 2D	LeakyReLU (alpha = 0.2)	(None, 300, 300, 8)	4 × 4	1 × 1	8	yes/same	1024
8	Convolution 2D	LeakyReLU (alpha = 0.2)	(None, 299, 299, 1)	2 × 2	2 × 2	1	no/valid	33
9	Reshape		(None, 299, 299, 1)					0

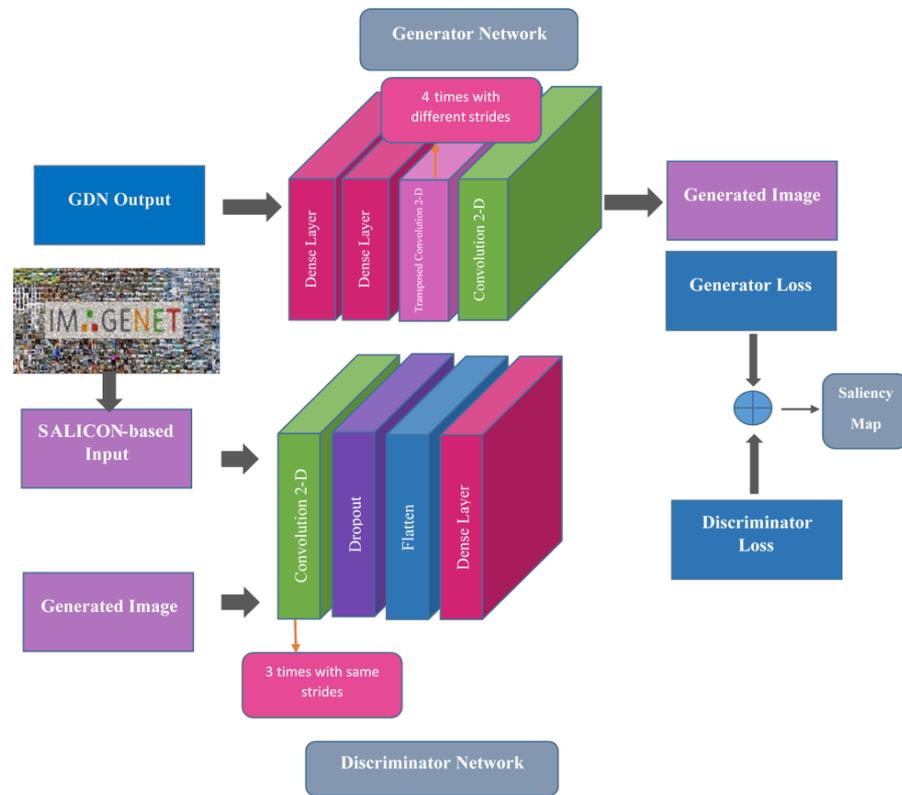


Figure 5. The block diagram of the GAN part of the proposed GDN-GAN for saliency reconstruction.

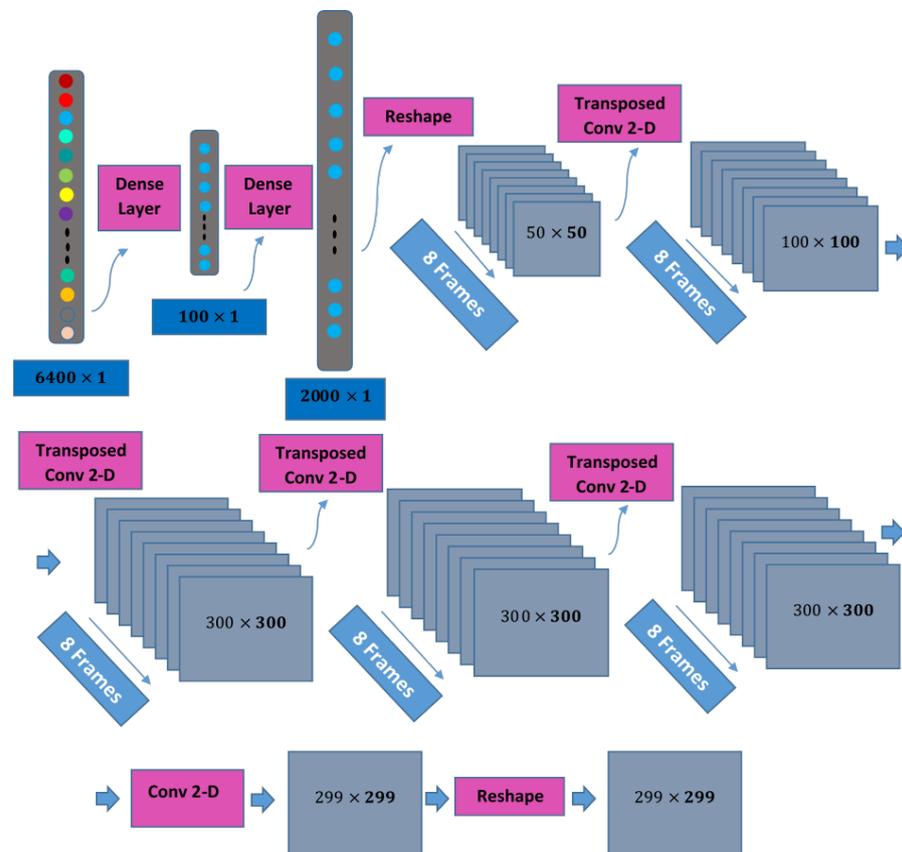


Figure 6. Dimensions of different layers of generator part of the proposed GDN-GAN.

The adversarial part of the proposed GAN has three 2D convolution layers with the rectified linear unit as the activation function. The size of the kernel for each of these convolutional layer is considered equal to 4×4 , the size of the strides is equal to 2×2 , and the number of filters for each of them is equal to four. The output of the third 2D convolution is flattened and imposed to a dense layer with an output dimension that is equal to one, to discriminate between fake or real images generated by the generator part of the GAN. Figure 7 illustrates schematic view of dimensions of different layers and it presents a tangible view of the outputs in each phase of the network.

Table 3. Details of the discriminator part of the proposed GDN-GAN architecture.

Layer	Layer Type	Activation Function	Output Shape	Size of Kernel	Strides	Number of Filters	Padding	Number of Parameters
1	Convolution 2D	LeakyReLU (alpha = 0.2)	(1, 150, 150, 4)	4×4	2×2	4	yes/same	68
2	Dropout (0.3)		(1, 150, 150, 4)					0
3	Convolution 2D	LeakyReLU (alpha = 0.2)	(1, 75, 75, 4)	4×4	2×2	4	yes/same	260
4	Dropout (0.3)		(1, 75, 75, 4)					0
5	Convolution 2D	LeakyReLU (alpha = 0.2)	(1, 38, 38, 4)	4×4	2×2	4	yes/same	260
6	Dropout (0.3)		(1, 38, 38, 4)					0
7	Flatten		(1, 5776)					0
8	Dense		(1, 1)					5777

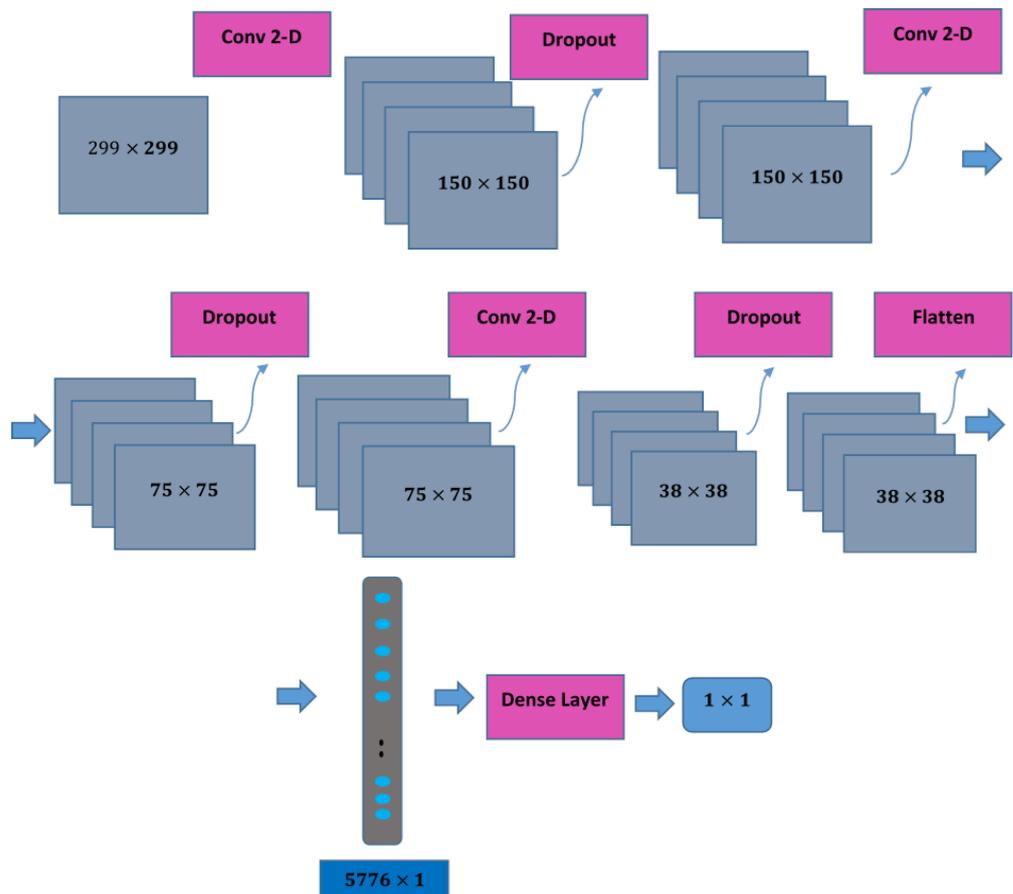


Figure 7. Dimensions of different layers of discriminator part of the proposed GDN-GAN.

Figure 8 presents an overview of the proposed method for image reconstruction using the trained network for realizing the saliency map reconstruction. As it can be seen in

this figure, the weights of the network are initialized with the pre-trained weights of the saliency map reconstruction scenario. Fine-tuning the transferred weights realizes the image stimuli reconstruction.

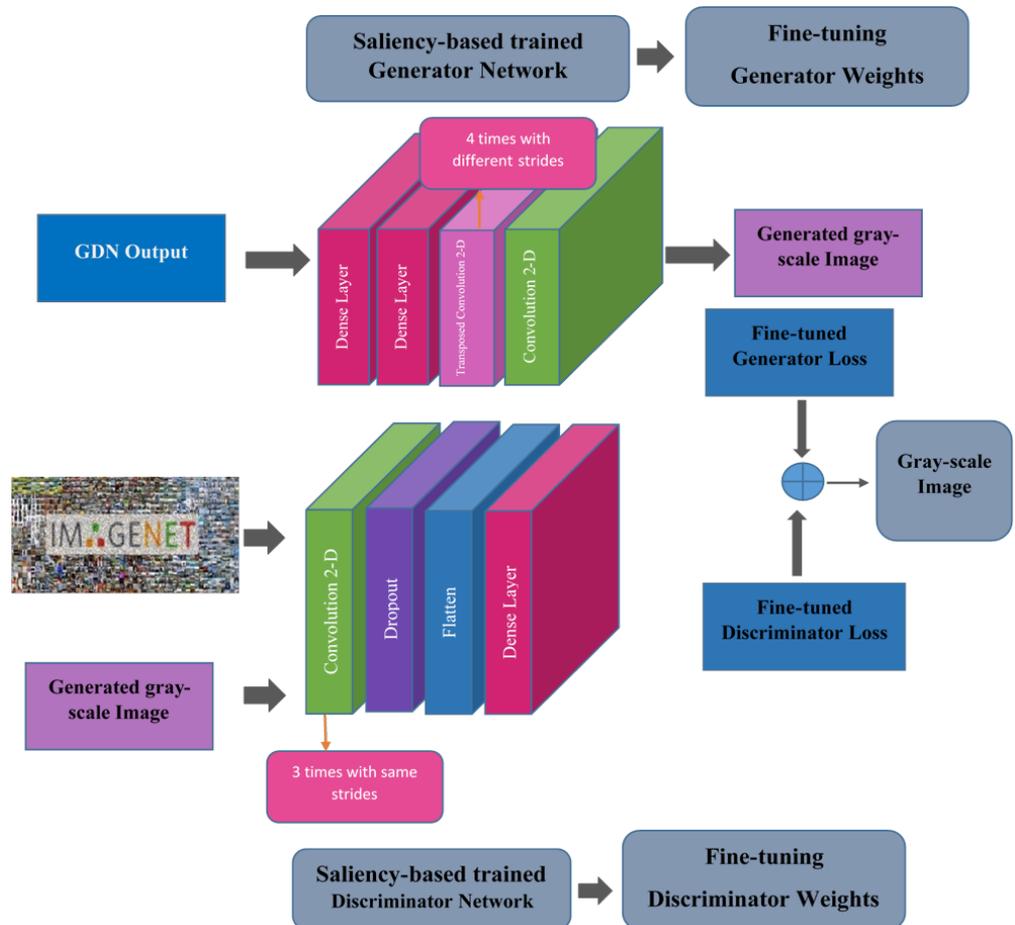


Figure 8. The block diagram of the GAN part of the proposed GDN-GAN for image reconstruction.

4.2. Training and Evaluation

In order to fit the proposed GDN part of the proposed GDN-GAN to the EEG-ImageNet dataset, a training procedure is implemented, and the parameter weights of the network are optimized. A 10-fold cross-validation strategy is used to train and evaluate the proposed network. A standard gradient descent (SGD) is used to optimize the proposed GDN in each iteration, and the optimum parameters of the GDN are determined with the convergence of the train and test accuracy. The trained weights of the GDN are transferred to the GDN-GAN to train the reconstruction part of the network.

Binary cross-entropy is used as a loss function for the GAN part of the GDN-GAN. Discriminator loss is considered as the sum of the loss of the original image and the loss of the generated image. For the loss of the discriminator output of the original image, instead of the ones vector as reference for calculating the cross-entropy between the reference and the original image, 0.9 is used as the coefficient of the ones vector. For the loss of discriminator output of the generated image, the cross-entropy is calculated between the generated image and the zeros vector with dimensions equal to the generated image. Generator loss is considered as the cross-entropy between the generated image and the ones vector with dimensions equal to the generated image. An Adam optimizer with a learning rate equal to 0.0001 is used to train both the generator and discriminator networks.

The tuning of different parameters of the proposed GDN-GAN is achieved through a trial-error procedure. Training is performed with the use of different parameters available

in Table 4 as a search space. The optimal values for training with good convergence are represented in this table.

Table 4. Search space for selecting the optimal parameters for the proposed GDN-GAN.

Parameters	Search Space	Optimal Value
Optimizer of GDN part	Adam, SGD	SGD
Sparsity level for graph embedding for GDN part	0.1, 0.2, 0.5, 0.8, 0.9, 0.98	0.9
Number of graph convolution layers	2, 3, 4, 5, 6, 8	4
Size of output sample in graph convolution layers	25, 50, 100, 200, 400	400, 200, 100, 50
Learning rate of GDN part	0.1, 0.01, 0.001	0.001
Weight decay of SGD optimizer of GDN part	4×10^{-3} , 4×10^{-5}	4×10^{-5}
Dropout rate of GDN part	0.1, 0.2	0.2
Optimizer of GAN part	Adam, SGD	Adam
Learning rate of GAN part	0.01, 0.001, 0.0001, 0.00001	0.0001
Number of transposed 2D convolution layers of generator of GAN part	2, 3, 4	4
Number of 2D convolution layers of discriminator of GAN part	2, 3, 4	3
pre-trained weights in GAN part	Inception—V3	none

5. Results and Discussion

In this section, the simulation results of the proposed GDN-GAN are presented. Our framework is implemented on a laptop with a 2.8 GHz Core i7 CPU, 16 GB RAM, and a GeForce GTX 1050 GPU using the EEG-ImageNet database described in Section 3.1, available in perceive lab [48,49]. The proposed network is trained using the Google Colaboratory Pro platform.

At first, in order to illustrate the effect of different visual stimuli on brain activity during the visual process in the brain, we consider the average of the time-domain samples of each EEG channel among all the recordings, in accordance with the particular category of visual stimulation. A representational similarity analysis of signals is represented in Figure 9. This representation shows the similarity between brain activity according to different categories. This is a good evidence that EEG signals contain visually related information in order to lead a person to the recognition of the surrounding environment.

The functional connectivity estimation of EEG channels is the first step of the GDN part of the proposed GDN-GAN. Approximating the connectivity matrix according to the specific sparsity level is achieved, and the number of nonzero elements of the corresponding adjacency matrix would decrease to avoid computational complexity. The adjacency matrix would be the sparsely approximated connectivity matrix. Figure 10 illustrates the circular connectivity, considering the threshold level for sparsifying the connectivity matrix with the best training convergence result. The circular connectivities for the green ($Ch_1 - Ch_{32}$), yellow ($Ch_{33} - Ch_{64}$), red ($Ch_{65} - Ch_{96}$), and white ($Ch_{97} - Ch_{128}$) electrodes according to Figure 1 are shown separately.

Figure 11a shows the training/test accuracy of GDN part of the proposed GDN-GAN, and Figure 11b shows the training/test loss function variations with respect to the number of iterations in this network for the classification of 40 different categories of visual stimuli.

Figure 12 shows the receiver operating characteristic (ROC) plot for the GDN part of the proposed GDN-GAN and other state-of-the-art methods for classification of the EEG-ImageNet dataset, including region-level stacked bi-directional LSTMs [56], stacked LSTMs [57], and Siamese network [48]. The superiority of the GDN in terms of the area under the ROC can be seen in this figure compared to the other existing methods.

Furthermore, the performance of the GDN against the above-mentioned state-of-the-art methods in terms of precision, F1-score, and recall metrics is shown in Table 5.

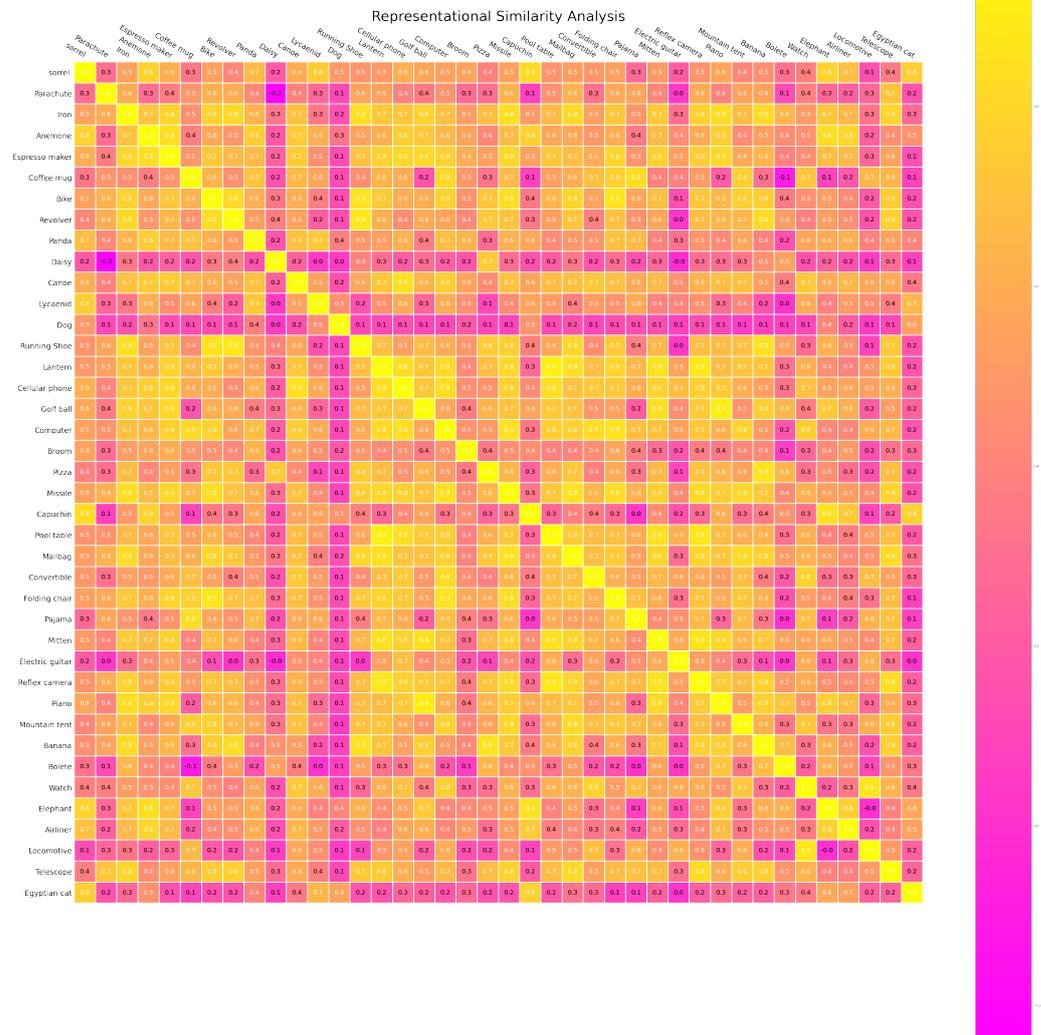


Figure 9. Representational similarity analysis.

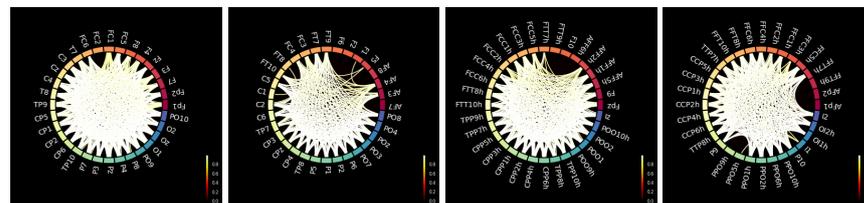
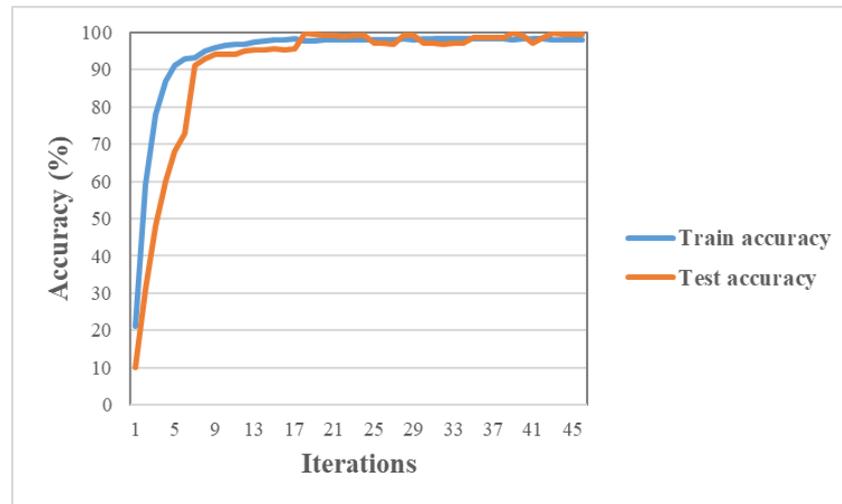


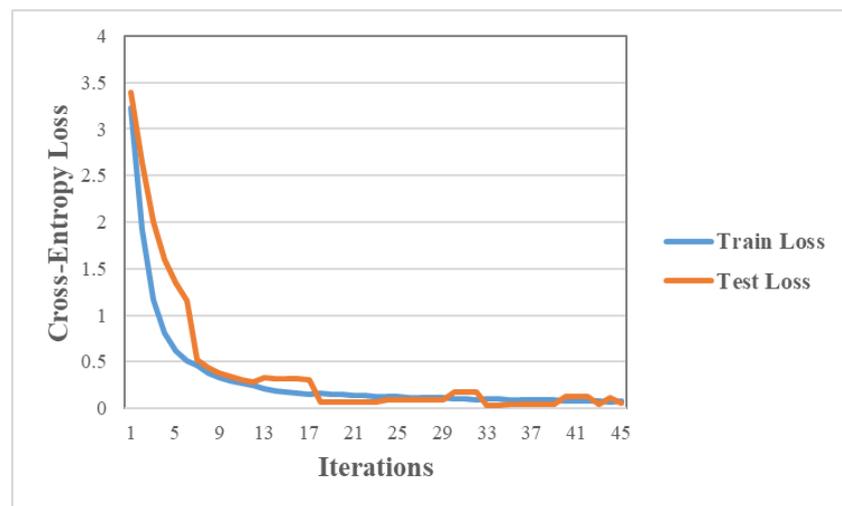
Figure 10. Circular connectivity of the approximated connectivity matrix; from (left) to (right): green ($Ch_1 - Ch_{32}$), yellow ($Ch_{33} - Ch_{64}$), red ($Ch_{65} - Ch_{96}$), white ($Ch_{97} - Ch_{128}$) labeled electrodes according to actiCAP128 standard-2, described in Figure 1.

To demonstrate the efficiency of the GDN, we compare the performance of our method with traditional feature-based CNN and MLP. For this purpose, three hidden layers for MLP and CNN with a learning rate of 0.001 have been considered. Maximum, skewness, variance, minimum, mean, and kurtosis have been used as feature vectors for every single channel. According to Figure 13, feature-based traditional deep networks such as MLP, CNN, and feature-based GDN have a poor performance in the case of classification of the EEG-ImageNet dataset with 40 different categories. This figure shows the obtained accuracy of feature-based MLP, CNN, and GDN in 50 iterations. This figure illustrates

that feature-based GDN, MLP, and CNN have shown relatively similar performances, and the training times per epoch for these methods are very high for these traditional methods. This figure confirms the good efficiency of the proposed GDN against the traditional feature-based deep networks. According to this figure, the proposed network has high classification accuracy compared to the other networks.



(a)



(b)

Figure 11. Model performance tracking of GDN part of the proposed GDN-GAN. (a) Accuracy for training and test phases. (b) Loss for training and test phases.

Table 5. Performance evaluation in terms of precision, F1-score, and recall for different networks.

Metric	GDN	Region-Level Stacked BiLSTMs	Stacked LSTMs	Siamese Network
Precision	98.56%	97.5%	91.1%	87.3%
F1-score	98.56%	97.5%	91.1%	87.3%
Recall	98.56%	97.5%	91.1%	87.3%

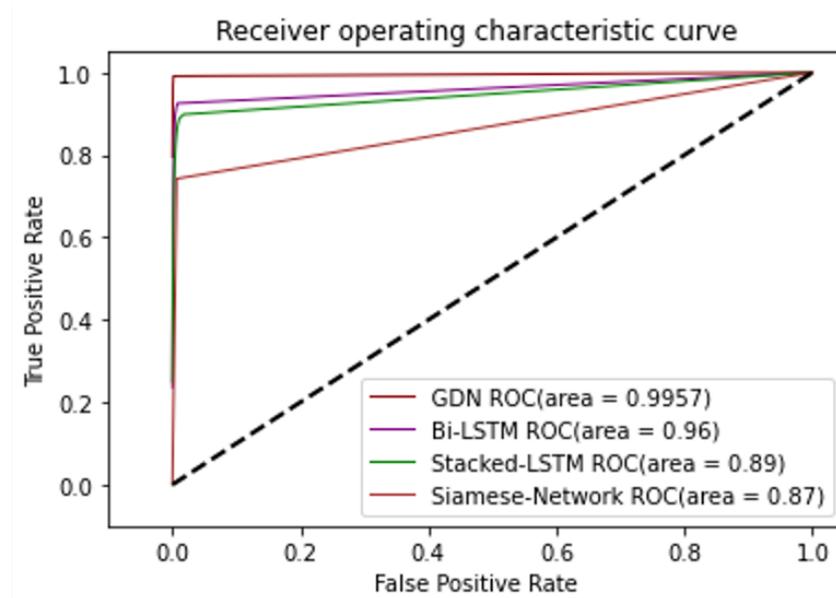


Figure 12. ROC in 10-fold cross-validation technique for different networks.

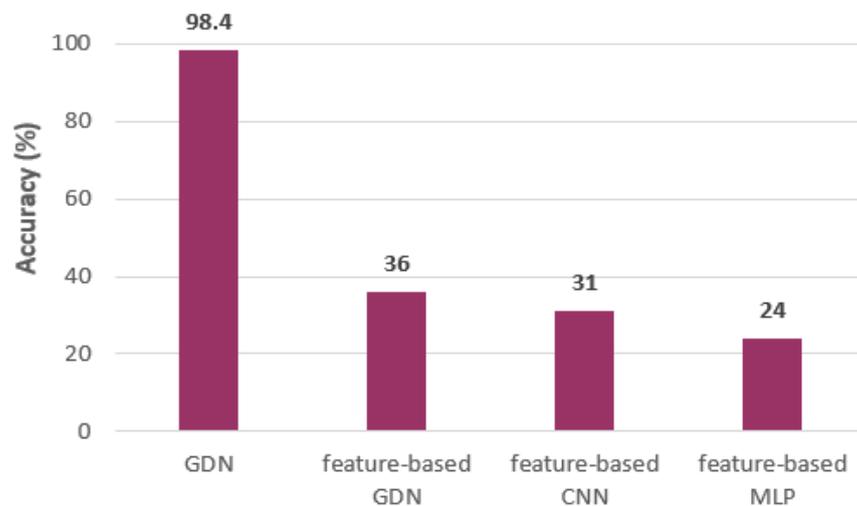


Figure 13. Comparison between feature-based classical methods.

A good confirmation to the performance of the proposed method is the confusion matrix shown in Figure 14. The confusion matrix is an appropriate illustration of the performance of a network on test splits in the case of multi-class classification. Figure 14 shows the confusion matrix of the GDN part of the proposed method. This figure confirms the good performance of the classification part of the GDN-GAN.

For the reconstruction phase of the proposed GDN-GAN, the training and evaluation of the proposed GDN-GAN are conducted according to the 10-fold cross-validation. The ground truth data are obtained using open-Salicon. The open-Salicon has been implemented using compiled the Python-compatible Caffe environment. The saliency evaluation metrics of the proposed GDN-GAN for different categories of visual stimuli are reported in detail in Table 6.

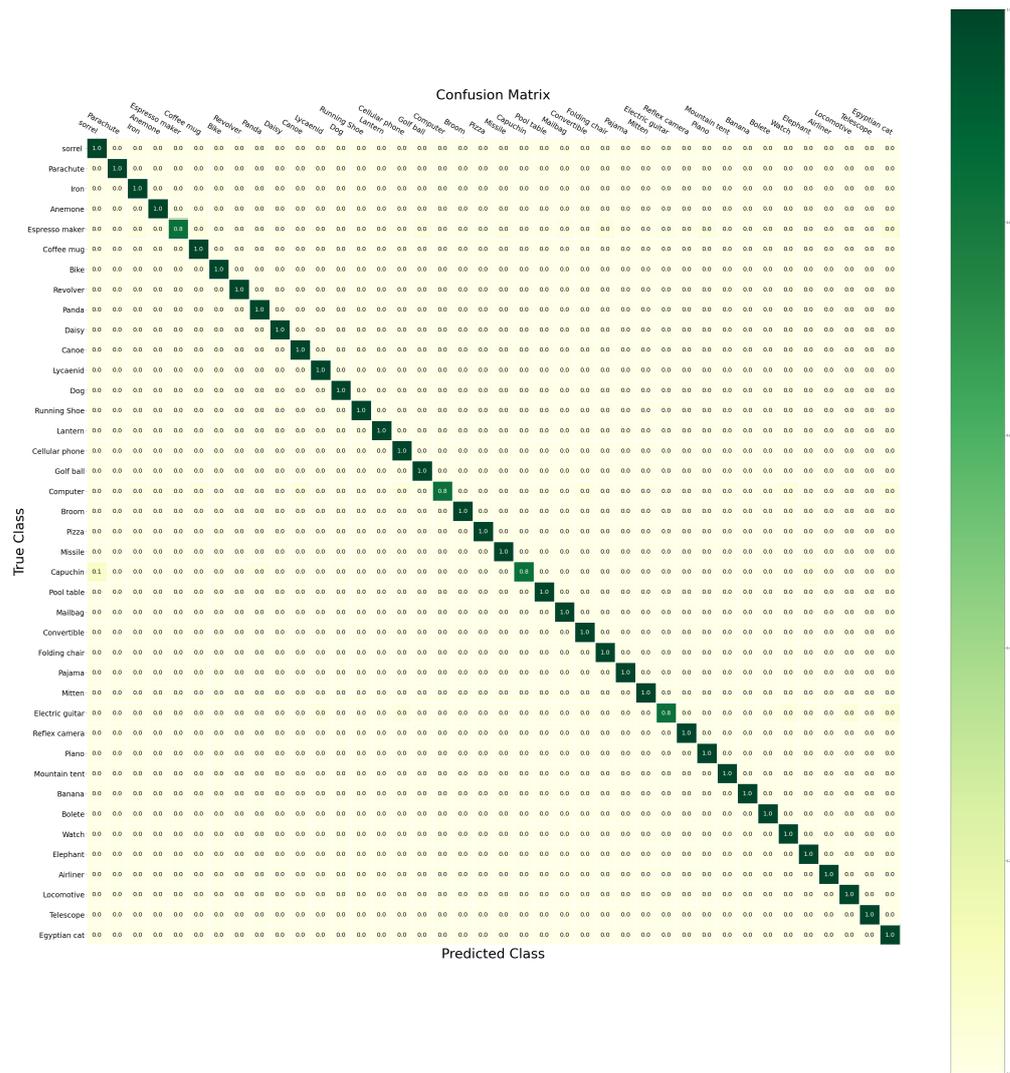


Figure 14. Confusion matrix for classification of test splits in 10-fold cross-validation technique with GDN part of the proposed GDN-GAN.

Table 6. SSIM, CC, NSS, and s-AUC of the proposed GDN-GAN for each category.

Category Number	Visual Category	SSIM	CC	NSS	s-AUC
1	'sorrel'	96.02%	99.61%	99.83%	96.01%
2	'Parachute'	97.40%	99.52%	99.73%	97.2%
3	'Iron'	96.02%	99.73%	99.82%	96.02%
4	'Anemone'	96.03%	99.25%	99.35%	96.01%
5	'Espresso maker'	95.55%	99.41%	99.52%	95.4%
6	'Coffee mug'	97.37%	99.69%	99.82%	97.2%
7	'Bike'	93.79%	99.50%	99.62%	93.44%
8	'Revolver'	67.03%	99.73%	99.82%	66.92%
9	'Panda'	94.04%	99.19%	99.42%	94.01%
10	'Daisy'	95.35%	99.46%	99.54%	95.2%
11	'Canoe'	96.98%	99.64%	99.82%	96.88%

Table 6. Cont.

Category Number	Visual Category	SSIM	CC	NSS	s-AUC
12	'Lycaenid'	94.04%	98.54%	98.74%	94%
13	'Dog'	96.72%	99.52%	99.64%	96.44%
14	'Running Shoe'	54.17%	99.45%	99.63%	58%
15	'Lantern'	25.28%	99.85%	99.9%	56%
16	'Cellular phone'	65.82%	98.98%	99.9%	68%
17	'Golf ball'	23.70%	99.81%	99.9%	64.42%
18	'Computer'	95.41%	99.54%	99.8%	96.54%
19	'Broom'	96.72%	99.43%	99.54%	95.52%
20	'Pizza'	92.98%	99.83%	99.9%	91.99%
21	'Missile'	94.28%	99.39%	99.63%	93.34%
22	'Capuchin'	98.20%	99.71%	99.82%	97.7%
23	'Pool table'	95.59%	99.43%	99.66%	94.87%
24	'Mailbag'	91.97%	99.22%	99.64%	90.03%
25	'Convertible'	91.57%	98.79%	98.96%	90.14%
26	'Folding chair'	94.51%	98.11%	98.78%	91.12%
27	'Pajama'	96.51%	99.70%	99.80%	95%
28	'Mitten'	95.04%	99.57%	99.68%	94%
29	'Electric guitar'	93.38%	98.52%	98.89%	90.09%
30	'Reflex camera'	97.17%	99.06%	99.42%	95.68%
31	'Piano'	94.43%	99.40%	99.55%	93.50%
32	'Mountain tent'	90.71%	99.11%	99.44%	89.09%
33	'Banana'	94.21%	99.77%	99.82%	90.09%
34	'Bolete'	93.93%	98.70%	98.4%	91.88%
35	'Watch'	97.41%	99.52%	99.02%	96.03%
36	'Elephant'	95.23%	99.47%	99.1%	95.01%
37	'Airliner'	97.79%	99.70%	99.83%	95.65%
38	'Locomotive'	96.60%	99.03%	99.54%	94.73%
39	'Telescope'	97.21%	99.63%	99.82%	96.43%
40	'Egyptian cat'	98.84%	99.88%	99.94%	96.54%
-	Overall Average	89.46%	99.39%	99.55%	90.51%

This table illustrates the saliency evaluation metrics according to the proposed method. The EEG signals are categorized in first part of the GDN-GAN. According to the extracted label, image stimuli is determined, and this image with the extracted feature of the first phase of the proposed method is imposed to the GAN part of the network to map the EEG signal to the saliency map of the image stimuli. After training, to test the GAN part, the EEG signals are imposed to the GDN-GAN, and the extracted images are compared to the original ground truth data through different saliency evaluation metrics, and the average of these metrics are reported in this table according to each category. Furthermore, the overall SSIM, CC, NSS, and s-AUC are represented through computing of the average of the saliency evaluation metrics among all categories.

According to this table, the proposed category-level performance of the visual saliency reconstruction method is over 90% except for six categories including Revolver, Running shoe, Lantern, Cellular phone, Golf ball and Mountain tent, in terms of SSIM and s-AUC. SSIM interprets the structural similarity index using the mean and standard deviation of pixels of a selected window with fixed size in reconstructed image and the ground truth data, and it would bring a reliable measure of similarity. The s-AUC uses true positives and false positives according to the pixels of the reconstructed image in the locations of fixations in ground truth data, and is a confident metric of similarity between the two images. Considering these details, SSIM and s-AUC illustrate the limitations of the proposed GDN-GAN. However, considering the detailed values of the four saliency metrics, this table shows that the proposed GDN-GAN is a reliable and efficient method to map the EEG signals to the saliency map of the visual stimuli.

The trained GDN-GAN for saliency reconstruction is fine-tuned for image construction issues. The loss plots in the result of training the generator and discriminator networks for visual saliency and image reconstruction are represented in Figure 15 for three number of categories. In addition, the SSIM and CC plots of both visual saliency and image reconstruction per epoch for these categories can be seen in this figure.

The loss plots corresponding to both the saliency reconstruction and image reconstruction illustrate that the variations in the generator and discriminator loss plots tend to oscillate around one, as saliency evaluation metrics, including SSIM and CC, start to converge. These are the behaviors of GANs, and these plots are confirmation of the effectiveness of the proposed reconstruction of the GDN-GAN.

The results of visual saliency and image reconstruction for all of the 40 categories of image stimuli are illustrated in Figures 16–19. In addition, the ground truth data and the gray-scale versions of the original input image stimuli are shown in these figures. The visual evaluations of these figures besides the saliency evaluation metrics confirm the efficiency of the proposed GDN-GAN.

A comparison of the proposed GDN-GAN with state-of-the-art methods for image saliency extraction is conducted, and the performance metrics are reported in Table 7. The results of SalNet [43], SALICON [42], visual classifier-driven detector, ref. [48] and neural-driven detector [48] are demonstrated in this table.

SALICON and SalNet are valuable approaches, considering the image data for saliency map extraction according to the eye-fixation points of the eye tracking process while a subject looking at an image. Another valuable approach, the visual classifier-driven detector and the visual neural-driven detector by Pallazo et al. [48], merges two modalities of EEG signals and image data to extract the image saliency map efficiently. Our proposed GDN-GAN is the first method that maps the EEG signals to the corresponding saliency map of the visual stimuli and reconstructs the saliency map and image stimuli. Considering the metrics according to these state-of-the-art methods concerning saliency map extraction in Table 7, this confirms the efficiency of the the proposed GDN-GAN.

Table 7. Saliency performance comparison between the proposed GDN-GAN and state-of-the-art methods.

Method	CC	NSS	s-AUC
SalNet	27.10%	61.80%	63.70%
SALICON	34.80%	72.80%	67.80%
visual classifier-driven detector	17.30%	49.50%	53.20%
neural-driven detector	35.70%	94.20%	64.30%
Proposed GDN-GAN	99.39%	99.55%	90.51%

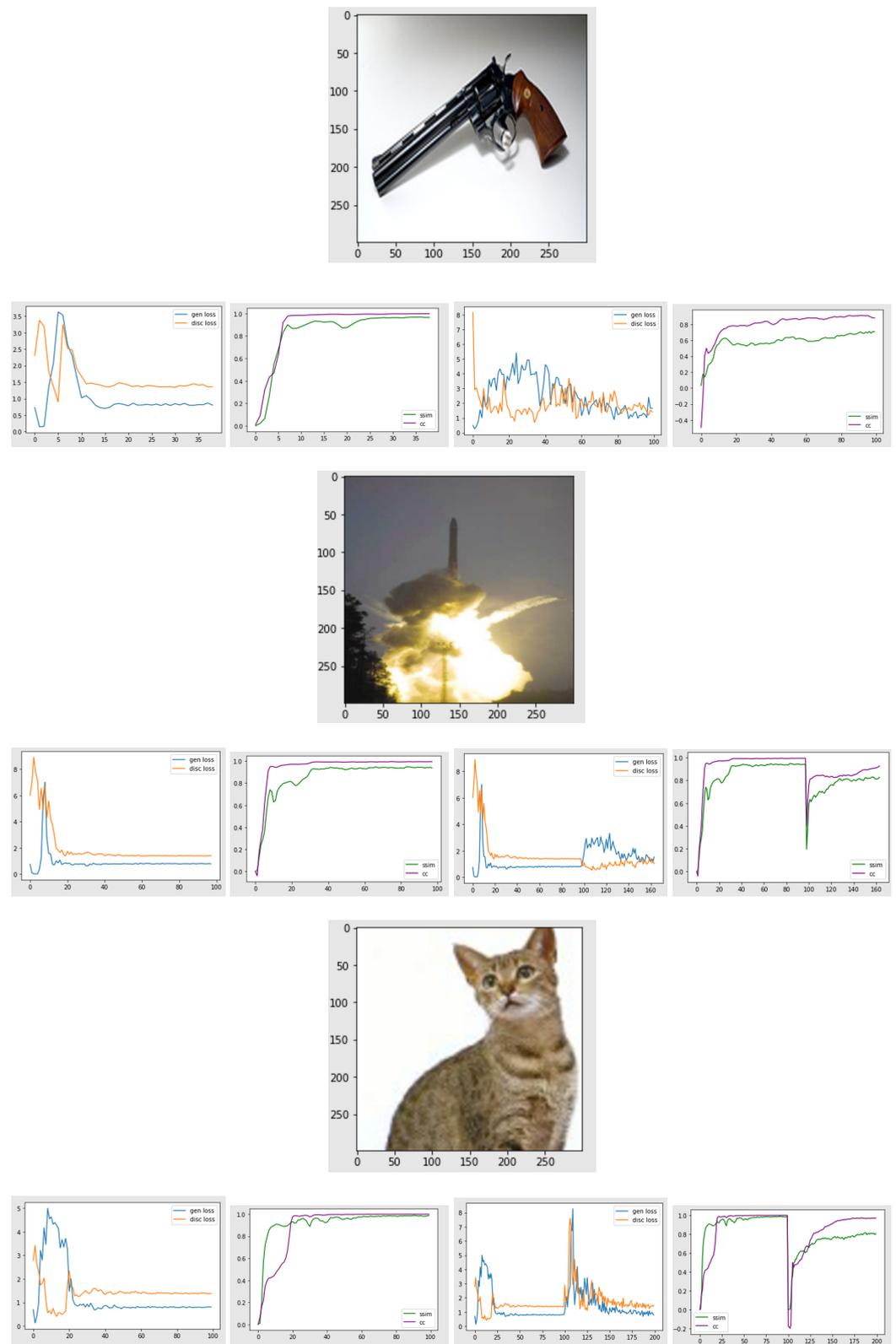


Figure 15. Training loss and accuracy plots; from left to right: Image used as stimulation, generator, and discriminator loss for image saliency reconstruction, SSIM, and CC for image saliency reconstruction, generator, and discriminator loss for image reconstruction, SSIM, and CC for image reconstruction for three categories, including 8, 21, and 40.

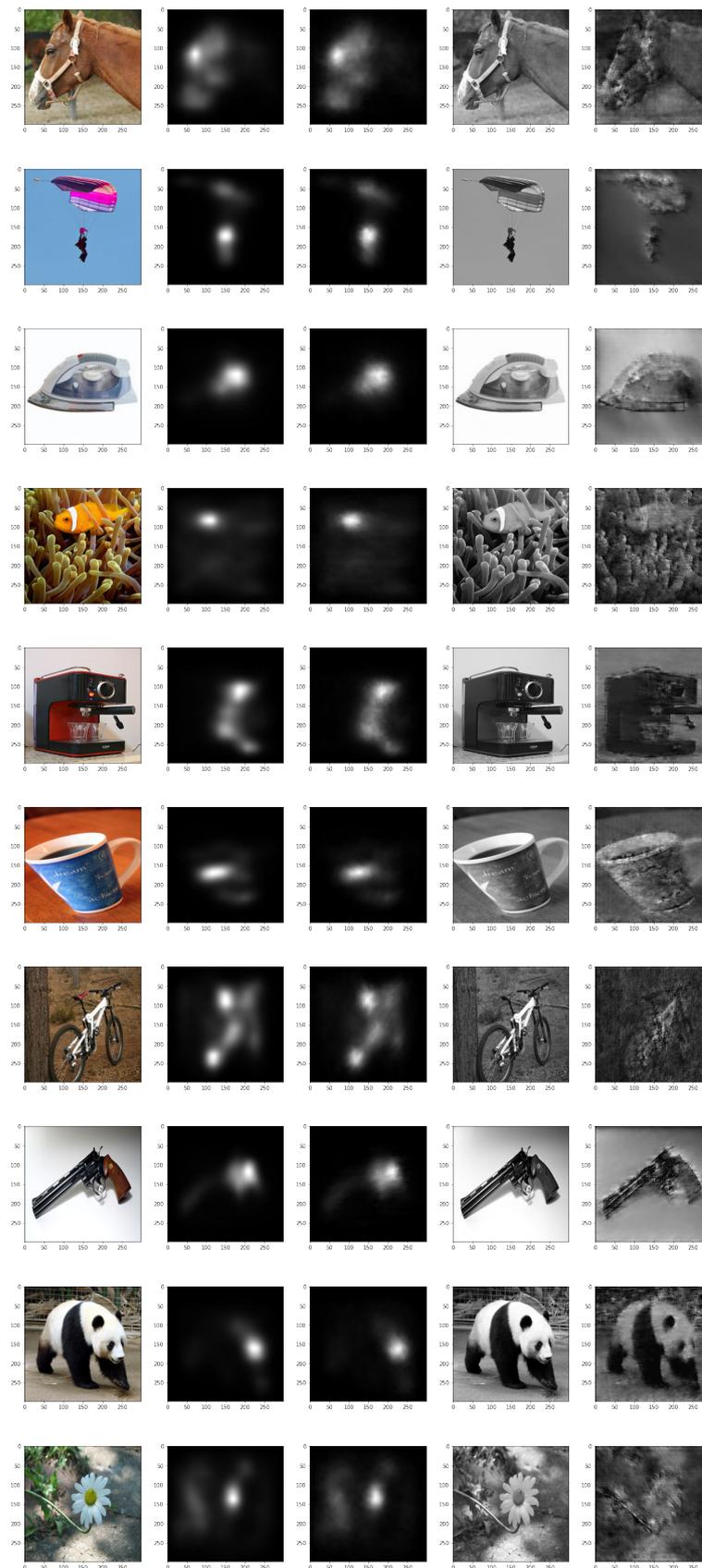


Figure 16. Reconstructed saliency and gray-scale images for categories 1–10; from (left) to (right): Image used as stimulation, fixation map, EEG-based reconstructed saliency map, gray-scale image, EEG-based reconstructed gray-scale image.

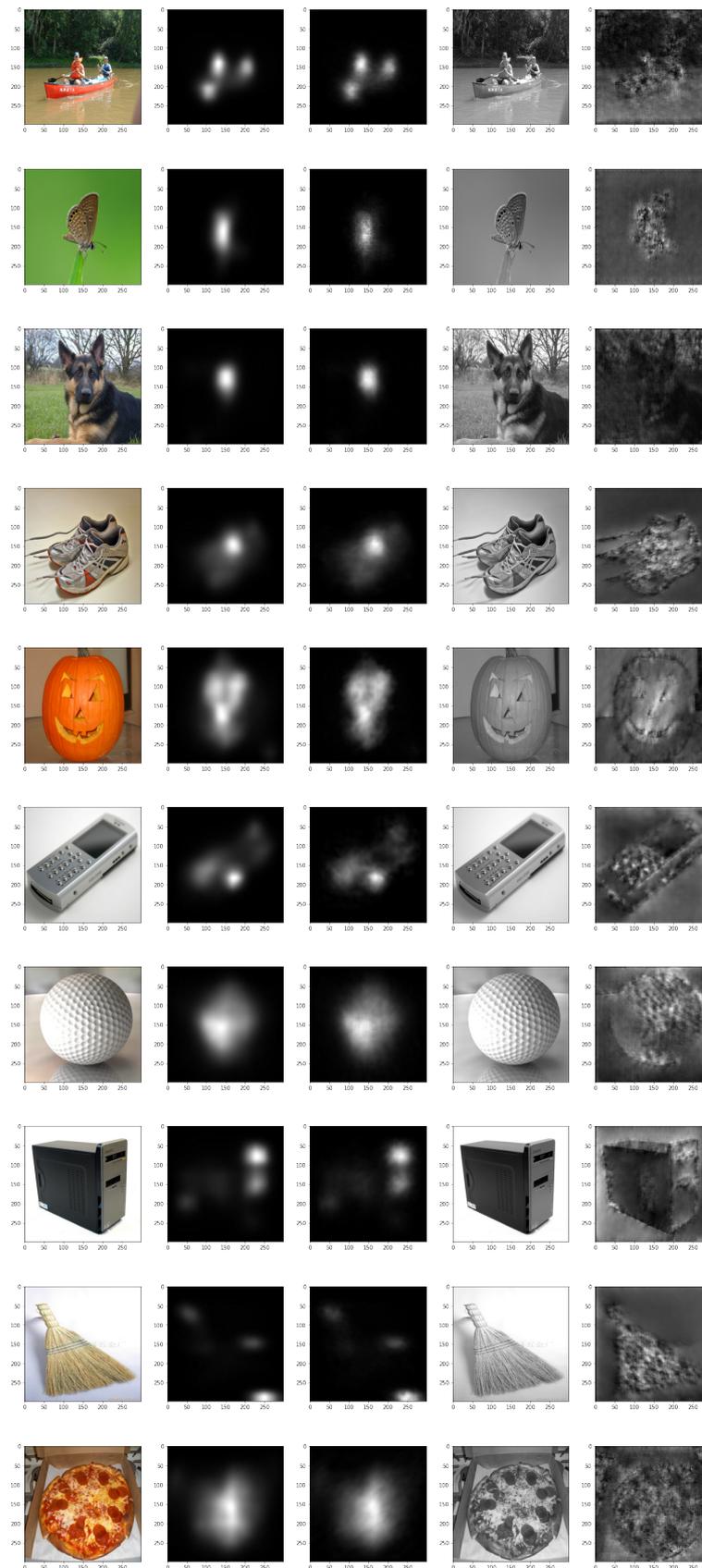


Figure 17. Reconstructed saliency and gray-scale images for categories 11–20; from (left) to (right): image used as stimulation, fixation map, EEG-based reconstructed saliency map, gray-scale image, EEG-based reconstructed gray-scale image.

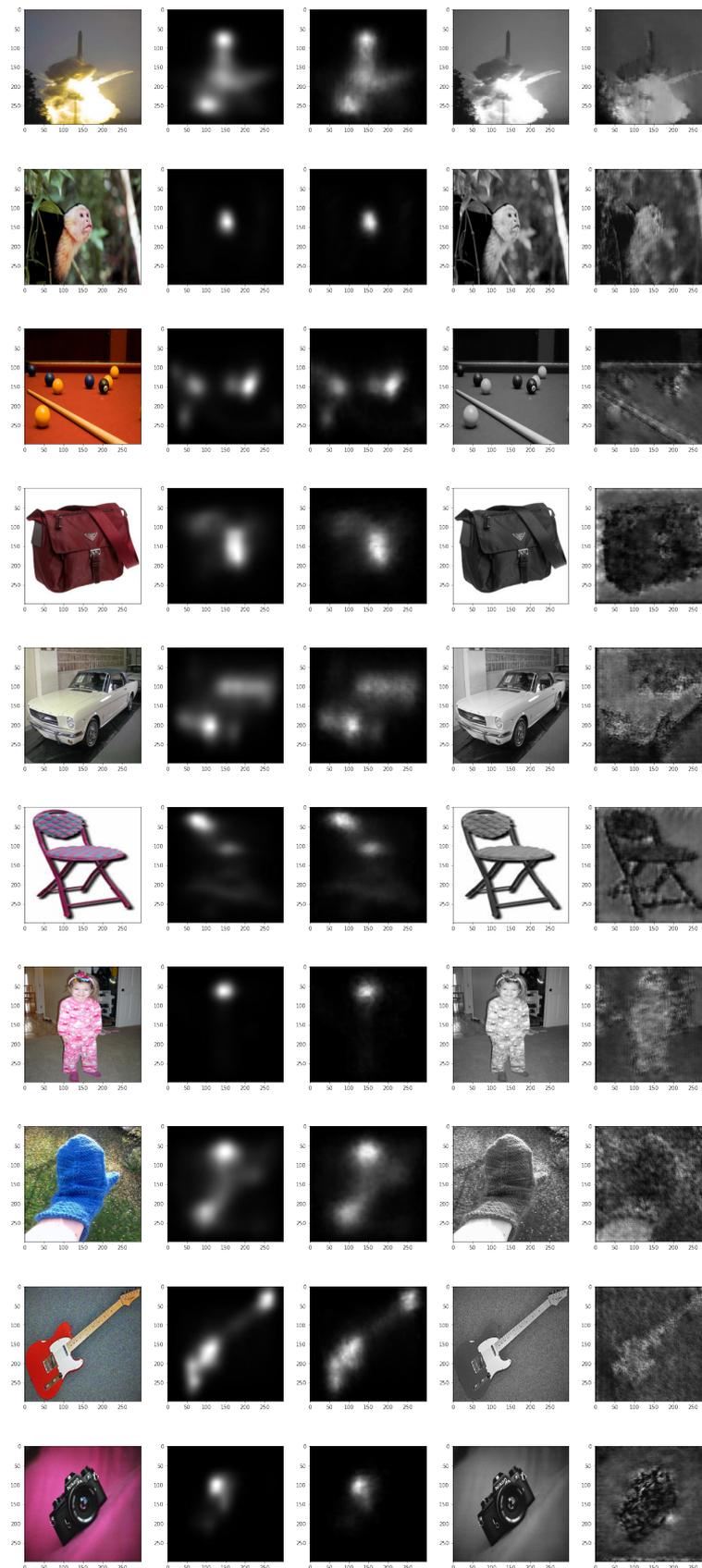


Figure 18. Reconstructed saliency and gray-scale images for categories 21–30; from (left) to (right): image used as stimulation, fixation map, EEG-based reconstructed saliency map, gray-scale image, EEG-based reconstructed gray-scale image.

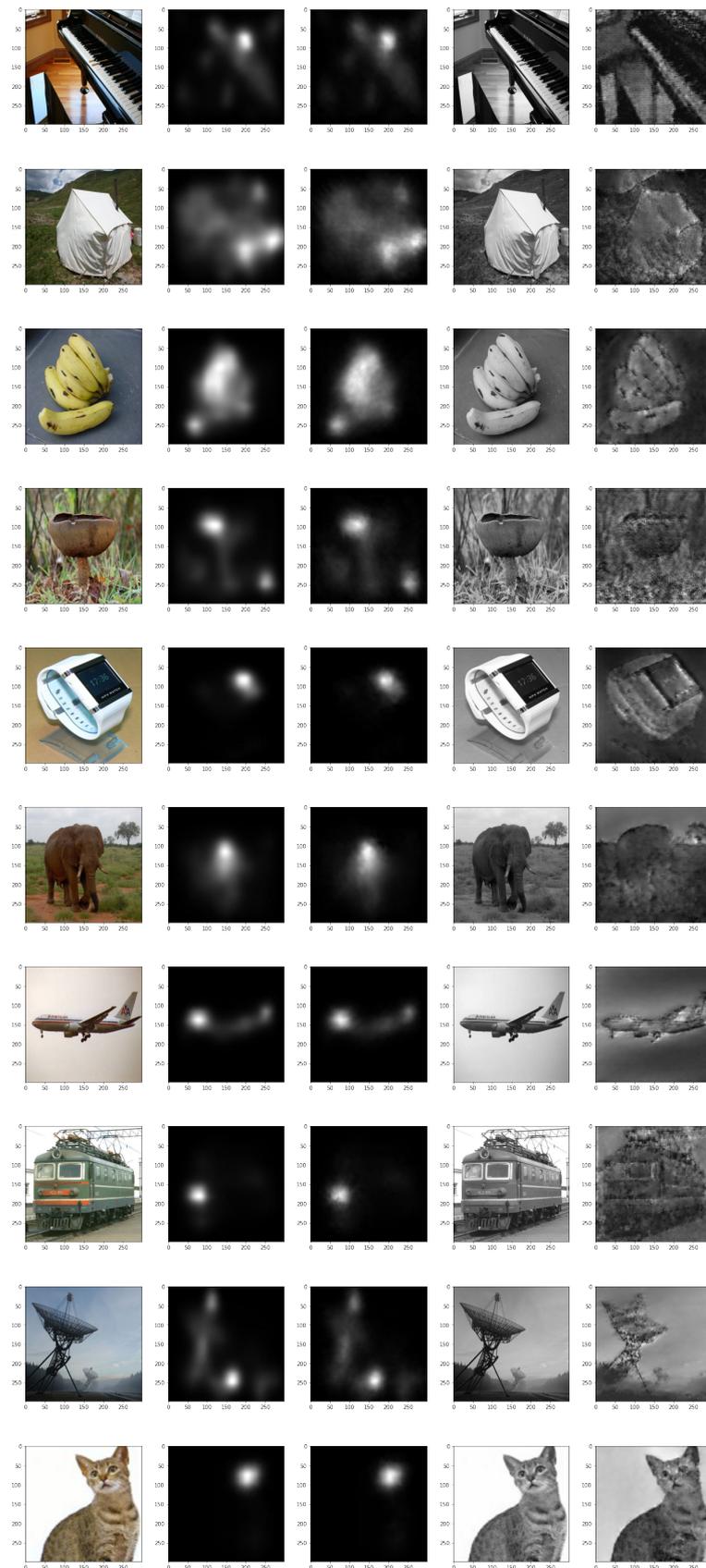


Figure 19. Reconstructed saliency and gray-scale images for categories 31–40; from (left) to (right): image used as stimulation, fixation map, EEG-based reconstructed saliency map, gray-scale image, EEG-based reconstructed gray-scale image.

In spite of the fact that the proposed GDN-GAN have a good performance in the reconstruction process, the limitations of the approach cannot be ignored. The first is that the ground truth data are generated using the pre-trained Open-Salicon using the image samples corresponding to the EEG-ImageNet database. This point should be considered in future works, and the solution is to use a good eye-tracker device and to record the eye fixation maps at the same time as the EEG recordings. These recorded eye fixation maps should be used as the ground truth data in future works.

Another limitation of the proposed GDN-GAN is the two-phase process of saliency reconstruction and three-phase of image reconstruction, considering the functional connectivity-based graph representation of the EEG signals imposed as the input to the network. An end-to-end process should be considered as the target deep network to decrease the training phases, eventually reducing the computational complexity, and hence increasing the speed of the network.

6. Conclusions

In this paper, an innovative graph convolutional generative adversarial network is proposed to realize the visual stimulation reconstruction using the EEG signals recorded from human subjects while they are looking at images from 40 different categories of the ImageNet database. The graph representation of the EEG records is imposed to the proposed network, and the network is trained to reconstruct the image saliency maps. The effectiveness of the proposed method is demonstrated with different saliency performance metrics. The trained weights are used as the initial weights of the proposed network to reconstruct the gray-scale versions of images used as visual stimulation. The results demonstrate the viability of the proposed GDN-GAN for image reconstruction from brain activity.

This research would be applicable to BCI projects for helping disabled people to communicate with their surrounding world. Neural decoding of the visually provoked EEG signals in BCI will interpret the brain activity of the subject and realize the automatic detection of the stimuli. It will pave the way toward mind reading and writing via EEG recordings, and is a preliminary step to help blind people with producing a module to realize vision through the generation of EEG signals corresponding to the visual surrounding environment.

The limitation concerning the ground truth data would be considered in future works to have a deep network that acts more similarly to real-world circumstances. The ground truth data in the proposed GDN-GAN are generated using the Open-Salicon pre-trained weights. These data should be recorded using a good eye tracker device at the same time as the EEG recordings. Considering the eye fixation maps of the subjects as the ground truth data would increase the efficiency of the proposed GDN-GAN in BCI applications.

Author Contributions: Conceptualization, N.K., T.Y.R. and S.S.; methodology, N.K. and S.M.; software, N.K., S.S. and S.D.; validation, S.S. and S.B.; writing—original draft preparation, N.K. and T.Y.R.; writing-review and editing, S.D., S.M. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: In this research, experimental data was not recorded.

Data Availability Statement: The EEG-ImageNet dataset used in this study is publicly available in this address: <https://tinyurl.com/eeg-visual-classification> (accessed on 10 October 2022).

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

References

1. Naselaris, T.; Kay, K.N.; Nishimoto, S.; Gallant, J.L. Encoding and decoding in fMRI. *Neuroimage* **2011**, *56*, 400–410. [[CrossRef](#)] [[PubMed](#)]
2. Han, J.; Chen, C.; Shao, L.; Hu, X.; Han, J.; Liu, T. Learning computational models of video memorability from fMRI brain imaging. *IEEE Trans. Cybern.* **2014**, *45*, 1692–1703. [[CrossRef](#)] [[PubMed](#)]

3. Heimann, K.; Umiltà, M.A.; Guerra, M.; Gallese, V. Moving mirrors: A high-density EEG study investigating the effect of camera movements on motor cortex activation during action observation. *J. Cogn. Neurosci.* **2014**, *26*, 2087–2101. [[CrossRef](#)] [[PubMed](#)]
4. Allam, J.P.; Samantray, S.; Behara, C.; Kurkute, K.K.; Sinha, V.K. Customized deep learning algorithm for drowsiness detection using single-channel EEG signal. In *Artificial Intelligence-Based Brain-Computer Interface*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 189–201. [[CrossRef](#)]
5. Rundo, F.; Leotta, R.; Battiato, S. Real-Time Deep Neuro-Vision Embedded Processing System for Saliency-based Car Driving Safety Monitoring. In Proceedings of the 2021 4th International Conference on Circuits, Systems and Simulation (ICCS), Kuala Lumpur, Malaysia, 26–28 May 2021; pp. 218–224. [[CrossRef](#)]
6. Alarcao, S.M.; Fonseca, M.J. Emotions recognition using EEG signals: A survey. *IEEE Trans. Affect. Comput.* **2017**, *10*, 374–393. [[CrossRef](#)]
7. Sheykhivand, S.; Rezaei, T.Y.; Meshgini, S.; Makoui, S.; Farzamnia, A. Developing a Deep Neural Network for Driver Fatigue Detection Using EEG Signals Based on Compressed Sensing. *Sustainability* **2022**, *14*, 2941. [[CrossRef](#)]
8. Sheykhivand, S.; Rezaei, T.Y.; Mousavi, Z.; Meshgini, S.; Makouei, S.; Farzamnia, A.; Danishvar, S.; Teo Tze Kin, K. Automatic Detection of Driver Fatigue Based on EEG Signals Using a Developed Deep Neural Network. *Electronics* **2022**, *11*, 2169. [[CrossRef](#)]
9. Khaleghi, N.; Rezaei, T.Y.; Beheshti, S.; Meshgini, S. Developing an efficient functional connectivity-based geometric deep network for automatic EEG-based visual decoding. *Biomed. Signal Process. Control* **2023**, *80*, 104221. [[CrossRef](#)]
10. Sheykhivand, S.; Rezaei, T.Y.; Saatlo, A.N.; Romooz, N. Comparison between different methods of feature extraction in BCI systems based on SSVEP. *Int. J. Ind. Math.* **2017**, *9*, 341–347.
11. Sheykhivand, S.; Yousefi Rezaei, T.; Mousavi, Z.; Meshini, S. Automatic stage scoring of single-channel sleep EEG using CEEMD of genetic algorithm and neural network. *Comput. Intell. Electr. Eng.* **2018**, *9*, 15–28. [[CrossRef](#)]
12. Shahini, N.; Bahrami, Z.; Sheykhivand, S.; Marandi, S.; Danishvar, M.; Danishvar, S.; Roosta, Y. Automatically Identified EEG Signals of Movement Intention Based on CNN Network (End-To-End). *Electronics* **2022**, *11*, 3297. [[CrossRef](#)]
13. Ghebreab, S.; Scholte, S.; Lamme, V.; Smeulders, A. Rapid natural image identification based on EEG data and Global Scene Statistics. *J. Vis.* **2010**, *10*, 1394–1394. [[CrossRef](#)]
14. Kay, K.N.; Naselaris, T.; Prenger, R.J.; Gallant, J.L. Identifying natural images from human brain activity. *Nature* **2008**, *452*, 352–355. [[CrossRef](#)] [[PubMed](#)]
15. Brouwer, G.J.; Heeger, D.J. Decoding and reconstructing color from responses in human visual cortex. *J. Neurosci.* **2009**, *29*, 13992–14003. [[CrossRef](#)] [[PubMed](#)]
16. Haynes, J.D.; Rees, G. Predicting the orientation of invisible stimuli from activity in human primary visual cortex. *Nat. Neurosci.* **2005**, *8*, 686–691. [[CrossRef](#)] [[PubMed](#)]
17. Thirion, B.; Duchesnay, E.; Hubbard, E.; Dubois, J.; Poline, J.B.; Lebihan, D.; Dehaene, S. Inverse retinotopy: Inferring the visual content of images from brain activation patterns. *Neuroimage* **2006**, *33*, 1104–1116. [[CrossRef](#)]
18. Ray, W.J.; Cole, H.W. EEG alpha activity reflects attentional demands, and beta activity reflects emotional and cognitive processes. *Science* **1985**, *228*, 750–752. [[CrossRef](#)]
19. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [[CrossRef](#)]
20. Achanta, R.; Estrada, F.; Wils, P.; Süsstrunk, S. Salient region detection and segmentation. In Proceedings of the International Conference on Computer Vision Systems, Santorini, Greece, 12–15 May 2008; Springer: Berlin, Germany, 2008; pp. 66–75. [[CrossRef](#)]
21. Ma, Y.F.; Zhang, H.J. Contrast-based image attention analysis by using fuzzy growing. In Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, 2–8 November 2003; pp. 374–381. [[CrossRef](#)]
22. Liu, F.; Gleicher, M. Region enhanced scale-invariant saliency detection. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 1477–1480. [[CrossRef](#)]
23. Hu, Y.; Rajan, D.; Chia, L.T. Robust subspace analysis for detecting visual attention regions in images. In Proceedings of the 13th annual ACM international conference on Multimedia, Singapore, 6–11 November 2005; pp. 716–724. [[CrossRef](#)]
24. Vidal, R.; Ma, Y.; Sastry, S. Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1945–1959. [[CrossRef](#)]
25. Rosin, P.L. A simple method for detecting salient regions. *Pattern Recognit.* **2009**, *42*, 2363–2371. [[CrossRef](#)]
26. Valenti, R.; Sebe, N.; Gevers, T. Image saliency by isocentric curvedness and color. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2185–2192. [[CrossRef](#)]
27. Zhong, S.H.; Liu, Y.; Liu, Y.; Chung, F.L. A semantic no-reference image sharpness metric based on top-down and bottom-up saliency map modeling. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, 26–29 September 2010; pp. 1553–1556. [[CrossRef](#)]
28. Duan, P.; Hu, B.; Sun, H.; Duan, Q. Saliency detection based on BP-neural Network. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 551–555. [[CrossRef](#)]
29. Neisser, U. *Cognitive Psychology Appleton-Century-Crofts*; Psychology Press: New York, NY, USA, 1967; p. 351.
30. Zhang, J.; Sclaroff, S. Saliency detection: A boolean map approach. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 153–160. [[CrossRef](#)]

31. Mauthner, T.; Possegger, H.; Waltner, G.; Bischof, H. Encoding based saliency detection for videos and images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2494–2502. [CrossRef]
32. Xu, J.; Jiang, M.; Wang, S.; Kankanhalli, M.S.; Zhao, Q. Predicting human gaze beyond pixels. *J. Vis.* **2014**, *14*, 28. [CrossRef]
33. Zhao, Q.; Koch, C. Learning a saliency map using fixated locations in natural scenes. *J. Vis.* **2011**, *11*, 9. [CrossRef] [PubMed]
34. Yang, J.; Yang, M.H. Top-down visual saliency via joint CRF and dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 576–588. [CrossRef]
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
36. Sabahi, K.; Sheykhivand, S.; Mousavi, Z.; Rajabioun, M. Recognition Covid-19 cases using deep type-2 fuzzy neural networks based on chest X-ray image. *Comput. Intell. Electr. Eng.* **2022**. [CrossRef]
37. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
39. He, S.; Lau, R.W.; Liu, W.; Huang, Z.; Yang, Q. Supercnn: A superpixelwise convolutional neural network for salient object detection. *Int. J. Comput. Vis.* **2015**, *115*, 330–344. [CrossRef]
40. Li, G.; Yu, Y. Visual saliency based on multiscale deep features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5455–5463. [CrossRef]
41. Huang, X.; Shen, C.; Boix, X.; Zhao, Q. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 262–270. [CrossRef]
42. Thomas, C. Opensalicon: An open source implementation of the salicon saliency model. *arXiv* **2016**, arXiv:1606.00110.
43. Pan, J.; Sayrol, E.; Giro-i Nieto, X.; McGuinness, K.; O'Connor, N.E. Shallow and deep convolutional networks for saliency prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 598–606. [CrossRef]
44. Liang, Z.; Hamada, Y.; Oba, S.; Ishii, S. Characterization of electroencephalography signals for estimating saliency features in videos. *Neural Netw.* **2018**, *105*, 52–64. [CrossRef]
45. Tavakoli, H.R.; Laaksonen, J. Bottom-up fixation prediction using unsupervised hierarchical models. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 287–302. [CrossRef]
46. Mao, X.; Li, W.; He, H.; Xian, B.; Zeng, M.; Zhou, H.; Niu, L.; Chen, G. Object extraction in cluttered environments via a P300-based IFCE. *Comput. Intell. Neurosci.* **2017**, *2017*, 5468208:1–5468208:12. [CrossRef]
47. Van Humbeeck, N.; Meghanathan, R.N.; Wagemans, J.; van Leeuwen, C.; Nikolaev, A.R. Presaccadic EEG activity predicts visual saliency in free-viewing contour integration. *Psychophysiology* **2018**, *55*, e13267. [CrossRef]
48. Palazzo, S.; Spampinato, C.; Kavasidis, I.; Giordano, D.; Schmidt, J.; Shah, M. Decoding brain representations by multimodal learning of neural activity and visual features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
49. Spampinato, C.; Palazzo, S.; Kavasidis, I.; Giordano, D.; Souly, N.; Shah, M. Deep learning human mind for automated visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6809–6817. [CrossRef]
50. Available online: <https://www.brainproducts.com> (accessed on 10 October 2022).
51. Available online: <https://image-net.org/> (accessed on 10 October 2022).
52. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852. Available online: <http://papers.nips.cc/paper/6081-sampling-for-bayesian-program-learning.pdf> (accessed on 10 October 2022).
53. Song, T.; Zheng, W.; Song, P.; Cui, Z. EEG emotion recognition using dynamical graph convolutional neural networks. *IEEE Trans. Affect. Comput.* **2018**, *11*, 532–541. [CrossRef]
54. Bylinskii, Z.; Judd, T.; Oliva, A.; Torralba, A.; Durand, F. What do different evaluation metrics tell us about saliency models? *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 740–757. [CrossRef] [PubMed]
55. Gu, K.; Zhai, G.; Yang, X.; Zhang, W.; Liu, M. Structural similarity weighting for image quality assessment. In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), San Jose, CA, USA, 15–19 July 2013; pp. 1–6. [CrossRef]
56. Fares, A.; Zhong, S.H.; Jiang, J. EEG-based image classification via a region-level stacked bi-directional deep learning framework. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 268. [CrossRef] [PubMed]
57. Kavasidis, I.; Palazzo, S.; Spampinato, C.; Giordano, D.; Shah, M. Brain2image: Converting brain signals into images. In Proceedings of the 25th ACM international conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1809–1817. [CrossRef]