

Article

Salient Arithmetic Data Extraction from Brain Activity via an Improved Deep Network

Nastaran Khaleghi ¹, Shaghayegh Hashemi ², Sevda Zafarmandi Ardabili ³, Sobhan Sheykhanvand ⁴  and Sebelan Danishvar ^{5,*} 

¹ Department of Electrical and Computer Engineering, University of Tabriz, Tabriz 51666-16471, Iran; khaleghi.nstr@tabrizu.ac.ir

² Department of Computer Science and Engineering, Shahid Beheshti University, Tehran 19839-69411, Iran; sh.hashemi@alumni.sbu.ac.ir

³ Electrical and Computer Engineering Department, Southern Methodist University, Dallas, TX 75205, USA; szafarmandiardabili@smu.edu

⁴ Department of Biomedical Engineering, University of Bonab, Bonab 55517-61167, Iran; s.sheykhanvand@tabrizu.ac.ir

⁵ College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK

* Correspondence: sebelan.danishvar@brunel.ac.uk

Abstract: Interpretation of neural activity in response to stimulations received from the surrounding environment is necessary to realize automatic brain decoding. Analyzing the brain recordings corresponding to visual stimulation helps to infer the effects of perception occurring by vision on brain activity. In this paper, the impact of arithmetic concepts on vision-related brain records has been considered and an efficient convolutional neural network-based generative adversarial network (CNN-GAN) is proposed to map the electroencephalogram (EEG) to salient parts of the image stimuli. The first part of the proposed network consists of depth-wise one-dimensional convolution layers to classify the brain signals into 10 different categories according to Modified National Institute of Standards and Technology (MNIST) image digits. The output of the CNN part is fed forward to a fine-tuned GAN in the proposed model. The performance of the proposed CNN part is evaluated via the visually provoked 14-channel MindBigData recorded by David Vivancos, corresponding to images of 10 digits. An average accuracy of 95.4% is obtained for the CNN part for classification. The performance of the proposed CNN-GAN is evaluated based on saliency metrics of SSIM and CC equal to 92.9% and 97.28%, respectively. Furthermore, the EEG-based reconstruction of MNIST digits is accomplished by transferring and tuning the improved CNN-GAN's trained weights.

Keywords: arithmetic content; visual perception; electroencephalogram; deep learning; MNIST



Citation: Khaleghi, N.; Hashemi, S.; Ardabili, S.Z.; Sheykhanvand, S.; Danishvar, S. Salient Arithmetic Data Extraction from Brain Activity via an Improved Deep Network. *Sensors* **2023**, *23*, 9351. <https://doi.org/10.3390/s23239351>

Academic Editors: Hyun Jae Baek, Heenam Yoon, Sang Ho Choi and Xi Long

Received: 12 October 2023

Revised: 6 November 2023

Accepted: 14 November 2023

Published: 23 November 2023



Copyright: © 2023 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

Neural activity decoding is of great importance in neurocognitive research. Electroencephalography is a popular technique for recording brain activities, and analyzing EEG results in response to unique stimulation is the objective of brain–computer interface (BCI) applications. An essential part of the human perception of the surrounding environment occurs via vision, and inferring the connection between brain response and vision has been considered in several neuroscience studies. Patterns related to brain signals can be distinguished, corresponding to various classes of visual stimuli [1–3].

Classifying electroencephalogram (EEG) recordings is the central part of neural decoding, and various methods have been presented for this purpose [4–6]. In recent years, the deep learning approach has attracted the attention of researchers [7,8], and deep network applications have been extended to the concept of EEG signal processing in BCI for emotion recognition [9], fatigue and sleep databases [10], epilepsy diagnosis and treatment [11], motor movement/imagination databases [12], and SSVEP datasets [13]. Deep learning

models have also been employed to explore the effect of visual stimulation on brain activity. For example, sequential LSTMs were used by Simone Palazzo et al. in [14] to realize the classification of the EEG-ImageNet signals corresponding to the brain activity of human volunteers when displaying a set of images from the ImageNet dataset. In another study [11], the use of deep network structure has been considered for image reconstruction from brain signals.

The hemispheric lateralization concept of the brain attracted the attention of Fares et al. [15], who used the information of hemispheric regions to construct a two-directional deep neural network for classifying the EEG-ImageNet signals. Nicolae Cudlenco et al. in [16] considered the Gabor filtering of EEG records to extract hidden features related to visual concepts in EEG signals. The projection of the entire spectrum on the space of Gabor wavelets was considered across a relatively large range of frequency bands to extract discriminative features, and they compared the performance of classification using ridge regression and deep network approaches. The processing of changes in the shape and color of objects was explored in a study by Nalin Mathur [17]. EEG data were collected using a feature-binding experiment that required subjects to detect changes in color and shape binding after 100 ms and after 1500 ms. The experiment was implemented to predict two stages of color and shape processing of visual stimuli in brain EEG data recordings. Familiar and unfamiliar face detection was considered by Lidia Ghosh et al. [18]. A deep model has been designed by them to quantify the ability of face perception in human subjects by classifying the EEG responses to facial images.

The brain response to natural visual stimulation was investigated by Ghebreab et al. [19] and was predicted using EEG responses. Similar research was carried out by Kay et al. in 2008 [20], but a better accuracy was achieved in 2010 [19]. Visual stimulation effects of the orientation of images [21], position of objects [22], and color of things [23] on brain activity have been explored in other studies.

The salient points in a picture have an essential effect on the visual interpretation occurring in the brain. Ulman and Koch [24], in 1985, proposed the concept of the optical salient region in an image to identify a region's dominance in the brain's visual information processing. Another model designed to improve understanding of the salient map was introduced by Itti et al. [25] in 1998. According to this work, a scene's uniqueness, distinctiveness, and rarity are essential parts of saliency detection. Many visual saliency detection models have been developed following the model proposed by Itti [25].

Understanding how the salient region influences brain signals is vital in eliciting the visual system's function. Although some works have been dedicated to the connection between the saliency of visual stimulation and brain activity, the relationship between salient arithmetic content and brain activity has yet to be studied. Furthermore, the use of time-related information of EEG signals has yet to be considered to extract the connection between visually evoked brain activity and salient arithmetic content.

The classification step is the critical step to achieve the salient arithmetic content corresponding to the visual stimuli. This work introduces a deep convolutional network to recognize the arithmetic category of EEG records. The proposed network consists of a convolutional network and a generative adversarial network to extract the arithmetic visual saliency map related to the recorded EEG signals. The overall model realizes the extraction of salient arithmetic data through visually evoked EEG records.

The achievements in this article can be summarized as follows:

- (i) Presentation of an effective deep network to acquire salient arithmetic content of visual stimulation from EEG recordings.
- (ii) Extraction of the arithmetic data is possible using the proposed architecture.
- (iii) Presentation of a convolutional deep network for extracting EEG features to recognize 10 patterns of EEG recordings corresponding to 10-digit categories.
- (iv) A 14-channel time sample of the EEG dataset is imposed directly as an input signal to the proposed CNN-GAN. The removal of feature vector extraction step results in decreasing the computational load.

- (v) It paves the way to connect three modalities: image data, visual salient data, and EEG signals.

The remaining parts of this article are arranged as follows. Related works to the saliency models have been explained in Section 2. The details of the MindBigData dataset and the mathematical precursors related to convolution, GAN, and saliency metrics are explained in Section 3. Section 4 describes the architectural and structural details of the proposed EEG-related arithmetic data recognition framework. The investigational results and validation against other newest methods are provided in Section 5. The conclusions are in Section 6.

2. Related Work

In early studies about salient content recognition [26,27], the saliency calculation was performed with pixel contrast. Hu et al. [28], in 2005, considered the geometric features of different regions and used principal component analysis to estimate the salient subspaces. A saliency detection method has been proposed with some simple operations of binarization, threshold decomposing, and edge detection for every pixel by Rosin et al. [29]. Estimating the salient content has been considered with isocenter clustering and curvedness by Valenti et al. [30].

Visual attention procedure has been used for modeling the saliency recognition. This procedure is a selective phenomenon for training the brain to understand the surrounding environment. Bottom-up pre-attentive feature-based and attentive task-based top-down processes have been suggested by Neisser [31] in 1967.

Bottom-up models use visual features of orientation, color, and intensity. Zhang et al. in 2013 [32] and Mauthner et al. in 2015 have proposed some saliency likelihood calculations for this type of modeling. Top-down saliency models consider the primary knowledge about features and the intentions corresponding to the brain task. The 2011 model by Zhao et al. [33], the work by Xu et al. [34] in 2014, and the model by Yang in 2017 [35] are examples of a top-down saliency process. Another category is based on the combination of top-down and bottom-up approaches. In these models, the top-down approach helps implement the given task, and the bottom-up technique detects the salient points.

Many neural networks have been proposed to implement models corresponding to saliency detection processes. Shengfeng He et al. [36], in their recent work, have assessed the convolutional neural networks to detect the salient object. The efficiency of these neural networks in salient object recognition has been analyzed by Guanbin et al. [37]. A one-dimensional convolutional neural network has been proposed by He et al. [36] to solve the saliency problem. Also, features of decomposed regions corresponding to different input scales have been imposed to MLP with two dense layers in a work by Yu and Li [37] in order to obtain a binary label for salient region detection.

Salicon [38] is one of the recent methods based on deep neural networks proposed for saliency detection. Transfer learning of GoogleNet, VGG-16, and AlexNet has been used in this model such that the weights of this model have been initialized with transfer learning. Another recently proposed deep network for saliency modeling is SalNet [39]. This network consists of two shallow and deep networks. The shallow network contains five convolutional and dense layers. A total of 10 layers have been employed in the deep part of the SalNet. More than 25 million parameters are trained during the deep learning process of the network.

Some recent studies have been dedicated to realize the connection between the salient content and brain activity. A model has been presented by Humbeeck et al. [40] to evaluate the effect of visual saliency on the amplitude of the EEG recordings. Fixation positions of the pupil in the eye via the eye-tracker and the brain activity using the EEG device have been considered in the modeling. The study by Zhen Liang et al. [41] in 2018 presented an approach based on the model by Tavakoli et al. [42] that shows this interaction with the use of video stimuli. A good accuracy has been reported in [41] in reconstructing and

predicting the temporal distributions of the features corresponding to salient regions using the extracted features of EEG recordings. In robot navigation and object identification in recorded images by robots by Mao et al. [43], P300 waves of the volunteers have been considered in identifying the objects of interest. A siamese network has been proposed in [44] through multimodal learning of image and EEG modalities for visual saliency detection. A cost function has been defined in the modeling, and its maximization has been considered to realize the connectedness of the modalities to the salient regions. Different scales of image masks have been considered for calculating the cost function and estimating the visual saliency. The mapping between EEG patterns and the salient picture of the visual stimulation has been realized in [45]. A deep network using a graph-embedded representation of EEG recordings has been introduced to acquire this map. Functional connectivity in different brain regions has been considered in exploring the interaction between visual saliency and brain activity [46].

This article proposes a novel approach for extracting the salient arithmetic content from visually evoked 14 EEG signals. In the next section, we explain the details of the database setting and the mathematical preliminaries to implement the proposed method.

3. Materials and Methods

The MindBigData [47] is used in this article, and recording details are explained in this section. The mathematical precursors of convolution 1-d, dropout, dense, and batch normalization layers will be presented. Furthermore, the details of generative adversarial networks, in addition to saliency metrics, are described.

3.1. Database Settings

The MindBigData has been recorded by David Vivancos [47] with the Emotiv EPOC device. This dataset has been gathered using a 14-channel cap. The placement of the EEG channel electrodes is shown in Figure 1, and the channel names considered in the recording procedure are A.F.3, F.7, F.3, FC.5, T.7, P7, O.1 in the left hemisphere and A.F.4, F.8, F.4, F.C.6, T.8, P8, O.2 in the right hemisphere as shown in dark blue in this figure. The channel names with details are described in Table 1.

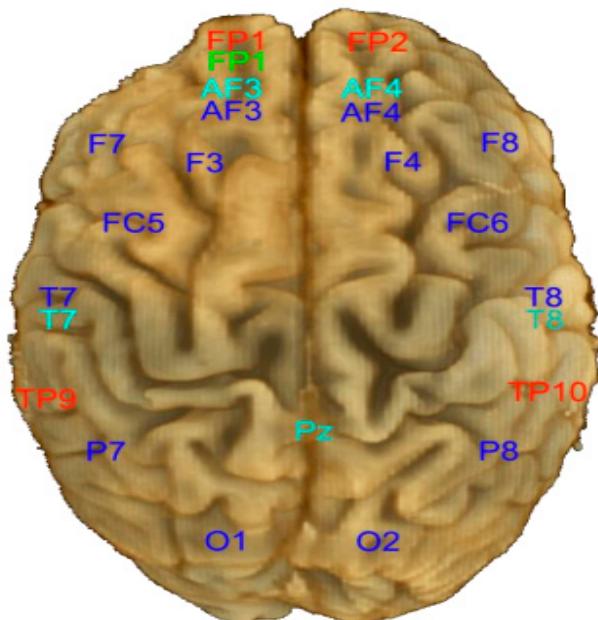
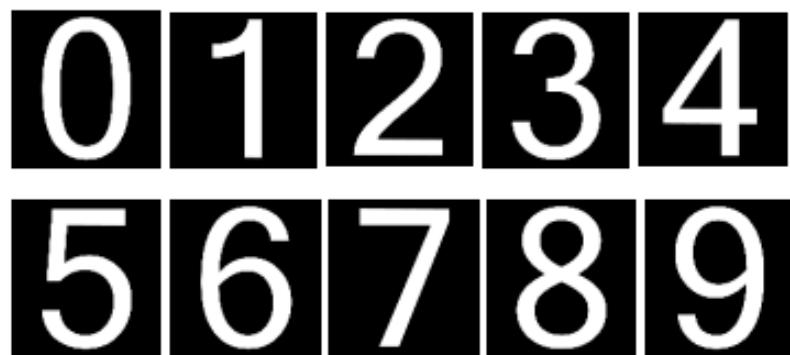


Figure 1. The placement of EEG electrodes of Emotiv EPOC.

Table 1. Channel names of Emotiv EPOC.

Channel Name in Left Hemisphere	Channel Name in Right Hemisphere	Channel Full Name
O1	O2	Occipital
P7	P8	Parietal
T7	T8	Temporal
FC5	FC6	FrontoCentral
F7	F8	Frontal
F3	F4	Frontal
AF3	AF4	Between Prefrontal and Frontal
FP1	FP2	PreFrontal

A total of 9120 brain signals of 2 s captured at a theoretical sampling rate of about 128 samples per second or 128 Hz are selected as MindBig dataset used in this paper. Total number of samples in each channel used for processing is 250. The brain signals were captured while a single digit from 0 to 9 corresponding to digits of the Modified National Institute of Standards and Technology (MNIST) dataset in Figure 2 has been shown for 2 s. The numbers have been represented on a 65-inch TV screen in a white font over a total black background. The appearance of digits was random, with a black screen between them. The number of 9120 EEG records corresponding to 912 signals of each category are considered for classification and salient arithmetic data extraction.

**Figure 2.** The digits of the MNIST dataset.

3.2. The Layers of Convolutional Neural Networks

The work of Hubel and Wiesel [48] can be considered the most important neural network in machine vision. The origin of this network goes back to the biological experiment conducted by them in 1962. The research conducted by these two individuals led to the discovery of simple and compound cells of vision. The identification of vision patterns was possible based on this study using these two types of cells. Simple cells in the vision system are responsible for detecting edges and columns in an image's specific direction and location. In contrast, the detection capability of composite cells is not limited to a particular area of the image, and this is possible at any point of the picture. This composite cell capability is obtained by collecting information from several simple cells. The first convolutional neural network was proposed based on the concepts of simple and compound vision cells and designed in 1979 by Kunihiko Fukushima [49]. The first project using these networks was identifying handwritten figures of digits, which was carried out by Yan LeCun et al. [50,51], and satisfactory results were obtained in the study.

Each deep convolutional network is composed of several layers. The pooling layer is an essential layer in CNN, which minimizes the spatial size of feature vector maps obtained

from the convolutional layer. This layer has no training parameter and performs a simple sampling. The most famous pooling layers are called average pooling and max pooling. For example, for maximum integration, a predefined window is considered that moves over the image to select the maximum value and ignore the rest of the numbers. The size of the filter and the size of the stride step in this layer are considered proportional to the optimal size for mapping the obtained feature of each layer.

The fully connected layer forms the final layer of CNN networks, which is used to classify the extracted feature maps. This layer is similarly present in multilayer perceptron (MLP) networks. After displaying the feature vectors obtained from convolutional layers, weight vector coefficients are assigned in this layer. The output corresponding to the number of classes available for classification can be achieved.

The following describes some other commonly used layers in CNN networks, including the random elimination layer and the batch normalizer layer. The use of the dropout layer in CNN networks strongly prevents the phenomenon of data overfitting in the training process. The function of this layer is to omit some neurons during training randomly. To optimize the coefficients, these randomly selected neurons are not considered during the learning process. Mathematically, neurons are discarded with probability $(p-1)$, and other neurons are retained with probability (p) .

The normalization layer is used to normalize the data inside the network. By performing various calculations on data, the distribution of data will change. The batch normalizer layer increases the training speed of the network by reducing the internal covariance of data distribution and accelerates the convergence process. The performance of this layer will be based on the calculation of the average and variance of data according to (1).

$$\begin{aligned}\mu_B &= \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i^l \\ \sigma_B^2 &= \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i^l - \mu_B)^2 \\ \hat{\mathbf{y}}^l &= \frac{\mathbf{y}^l - \mu_B}{\sqrt{(\sigma_B^2 + \epsilon)}}\end{aligned}\quad (1)$$

3.3. Generative Adversarial Networks

The generative adversarial networks (GANs) [7] include generator and discriminator networks. The generative model G consists of some layers to fit a random vector y with probability distribution $P(y)$ into a desired data distribution. The discriminative part, named D , compares data from the expected distribution and data obtained from the generator part. These two networks are trained simultaneously, and the training will continue to see no improvement in network optimization. The cost function of GAN can be described as follows:

$$\min_{\text{Gen Disc}} \max_{\text{Disc}} \left[E_{X \sim P_{\text{data}}(x)} [\log \text{Disc}(x)] + E_{Y \sim p_y(y)} [\log (1 - \text{Disc}(\text{Gen}(y)))] \right] \quad (2)$$

In this loss function corresponding to GAN, the desired data are depicted as x , and the random feature vector employed as input to the generator is represented with y . The generator's output is represented by $\text{Gen}(y)$, and the work of the discriminator is shown with $\text{Disc}(x)$. The convergence will occur when the loss function according to each network is as close as possible to 1. The output of the discriminator is represented by $\text{Disc}(\text{Gen}(y))$. The probability distribution of generated data and the desired data is represented respectively, with $P_{\text{data}}(x)$ and $p_y(y)$.

3.4. Evaluation Metrics for Classification and Salient Image Extraction

This section provides a brief description of classification and saliency evaluation metrics. The saliency feature map and the ground truth feature vector are two necessary inputs for calculating the saliency evaluation metrics. The level of similarity can be represented by considering these metrics.

The most used metrics for classification are described in (3) as sensitivity, accuracy, precision, and recall based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

$$\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{precision} = \frac{\text{TP}}{\text{FP} + \text{TP}}$$

$$\text{F1-score} = \frac{\text{precision} + \text{recall}}{2} \quad (3)$$

Cohen's Kappa coefficient is another metric for classification, as described in (4).

$$\text{kappa} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP}) \times (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) \times (\text{FN} + \text{TN})} \quad (4)$$

Three important metrics for evaluating saliency are described as follows.

Similarity metric (SIM) is used to measure similarity between distributions [52]. Normalization of the input signal vectors is performed, and the sum of minimum values at each pixel results as S.I.M. The saliency map is shown with S.M and the fixation map is represented with F.M:

$$\text{SIM}(\text{SM}, \text{FM}) = \sum_j \min(\text{SM}_j, \text{FM}_j) \quad \text{where} \quad \sum_j \text{SM}_j = \sum_j (\text{FM}_j) = 1 \quad (5)$$

In (5), pixel locations are represented with j . The value of SIM is equal to one for inputs with identical distributions, while this metric would be zero if there is no similarity and overlap between distributions.

Another saliency evaluation metric is structural similarity (SS.I.M), calculated using different windows of an image [53]. The SSIM is computed considering two sampling windows, m and n , with size $L \times L$:

$$\text{SSIM}(m, n) = \frac{(2 \times \mu_m \mu_n + k_1)(2 \times \sigma_{mn} + k_2)}{(\mu_m^2 + \mu_n^2 + k_1)(\sigma_m^2 + \sigma_n^2 + k_2)} \quad (6)$$

μ_m : the average of m ; σ_m^2 : the variance for m ;

μ_n : the average of n ; σ_n^2 : the variance for n ;

σ_{mn} : the covariance between m and n ;

$k_1 = ((d_1)K)^2$; $k_2 = ((d_2)K)^2$;

K : the variation range of the pixel-intensities ($2^{(\text{bits per pixel})} - 1$);

$d_1 = 0.01$ and $d_2 = 0.03$.

A metric for assessing the affine connectedness of distributions is Pearson's correlation coefficient (CC) [52]. CC can be computed as in Equation (7).

$$\text{CC}(\text{FM}, \text{SM}) = \sigma(\text{FM}, \text{SM}) / \sigma(\text{FM}) \times \sigma(\text{SM}) \quad (7)$$

The covariance between FM and SM in (7) is represented with $\sigma(\text{SM}, \text{FM})$. This metric is unaffected by linear transformations. This evaluation metric would treat false negatives and false positives equally and, therefore, is a symmetric function. The similar magnitudes of the saliency map and the reference ground truth would result in high positive values of CC.

4. Proposed Convolutional Neural Network-Based Generative Adversarial Network

The details of the proposed convolutional neural network (CNN) for automatic visual arithmetic content identification from EEG signals and the proposed convolutional neural network-based generative adversarial network (CNN-GAN) for arithmetic data extraction from brain activity are elucidated in this section.

4.1. The Proposed Network Architecture

Figure 3 represents the schematic of the CNN fragment to classify the input EEG signal into the correct category of the MNIST dataset. The visual stimulation related to the MNIST dataset appears on an LCD to a human volunteer, corresponding to the considered timing of occurrence for each image and time-lapse between sequential images. The EEG signals are recorded during the experiment. Normalizing the EEG time samples is performed considering each EEG channel's mean and standard deviation. The pre-processed EEG signals are applied as input to the proposed one-dimensional CNN network.

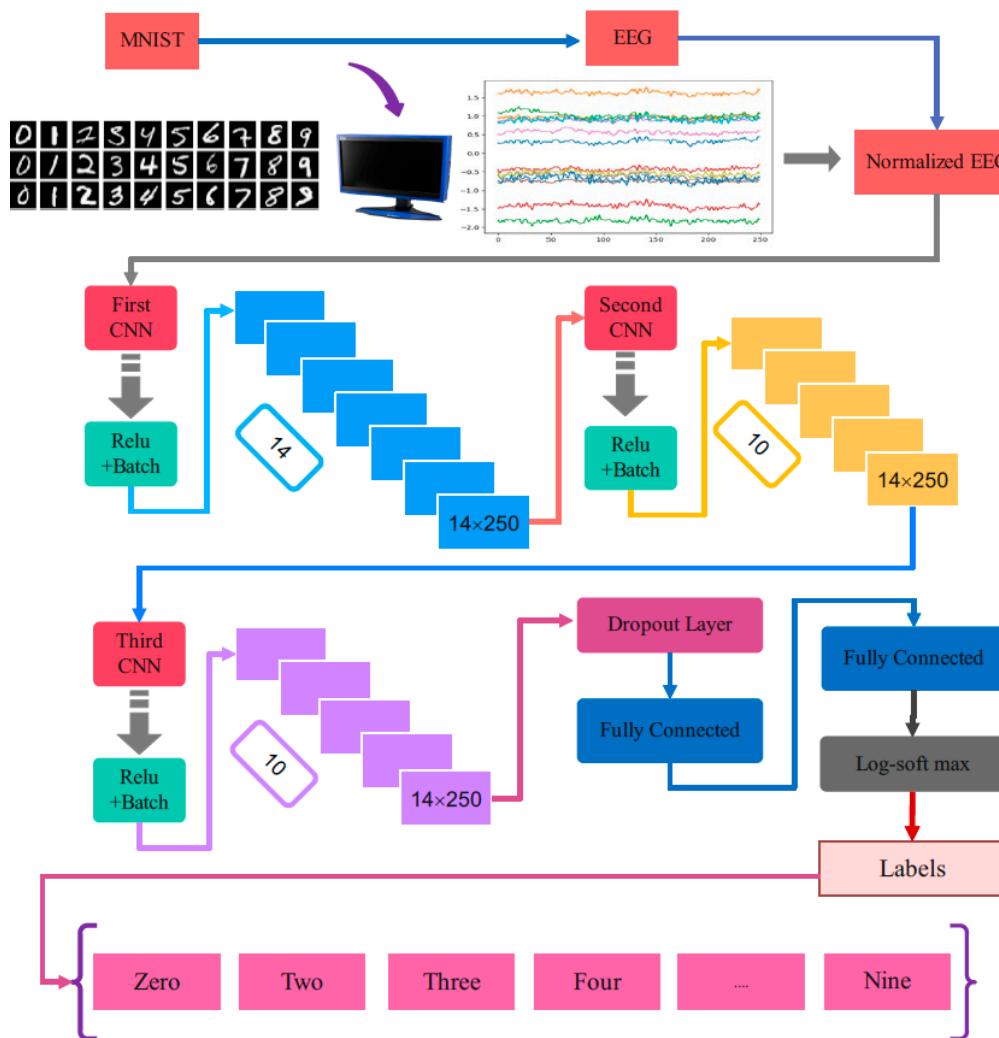


Figure 3. The block diagram of the layers in proposed CNN.

The structure of the CNN fragment of the network consists of three convolutional layers, as illustrated in Figure 3. The rectified linear unit is selected for the activation function in each layer. After each convolutional layer, a dropout and batch normalization are considered to prevent overfitting. Flattening of the output of the third convolutional layer is achieved. It is imposed to a linear layer, and the output vector of a dense, full-connected layer is passed through a log_softmax classifier layer to classify the input EEG

signal. The output of the linear layer is a vector with 2500 elements, and it is the vector applied as input to the next deep network of the proposed CNN-GAN to extract and obtain the MNIST images used as the stimulation. The details of the CNN network of the proposed architecture are explained in Table 2.

Table 2. Details of the CNN network of the proposed architecture.

Layer Number	Layer Name	Activation Function	Size of Kernel	Strides	Total Number of Weights	Output Size
1	Conv-layer	Leaky ReLU (alpha = 0.1)	5×1	1×1	14	(1, 14, 14, 250)
2	Normalization					(1, 14, 14, 250)
3	Conv-layer	Leaky ReLU (alpha = 0.1)	5×1	1×1	10	(1, 10, 14, 250)
4	Normalization					(1, 10, 14, 250)
5	Conv-layer	Leaky ReLU (alpha = 0.1)	5×1	1×1	10	(1, 10, 14, 250)
6	Normalization					(1, 10, 14, 250)
7	Full-connected					(1, 3500)
8	Full-connected					(1, 2500)

The proposed CNN-GAN consists of two fragments of sequential layers, as in Figure 4. The one-dimensional convolutional layers in the first part of the proposed network classify the EEG signals related to 10 different numbers of MNIST images with arithmetic content. EEG signal classification happens in this stage, and the signal can be classified according to the extracted features for 10 categories. After the preprocessing stage of the recorded EEG signals in response to the visual image with arithmetic content, we have several deep layers to extract the output vector of the first part of the proposed network to be applied to the next generator adversarial network. The second part consists of generator and discriminative networks in order to map the one-dimensional extracted feature vector of the EEG signal in the first part to the two-dimensional image array. The main layers of GAN in the proposed technique are two-dimensional convolutional blocks to reconstruct the salient images.

The salient images are created using the SALICON approach to apply as the ground reference data of the GAN network. The generative adversarial network is trained, and the network weights will be determined. After this stage of the procedure, tuning the weights of the trained network will be performed and transfer learning will be employed to reconstruct the original visual image stimulation.

The architectural details of the generator and adversarial networks can be seen in Figure 5. The flattened vector with 2500 elements is passed through a dense layer of the generator network and four transposed two-dimensional convolution layers. Also, an additional two-dimensional convolutional layer is needed to fit the output image dimension to the desired dimension. The adversarial consists of three sequential layers of two-dimensional convolutional layers. The output of these layers is imposed to the dropout layer, and then the flattened output vector passes a fully connected layer to judge about fake or real data.

The structural and dimensional details of layers in the generator are represented in Figure 6 and Table 3. The input vector dimension is equal to 2500, and after passing through two dense layers, the output vector dimension is equal to 20,000. The reshaped vector of dense layer output is applied in the first transposed convolution layer. The number of kernels in transposed convolutional layers is considered equal to four to have four two-dimensional outcomes in each layer. Considering different kernels and strides in five

transposed convolution layers according to Table 3, the dimensions of two-dimensional outputs are equal to 50×50 , 100×100 , and 300×300 , as illustrated in this table.

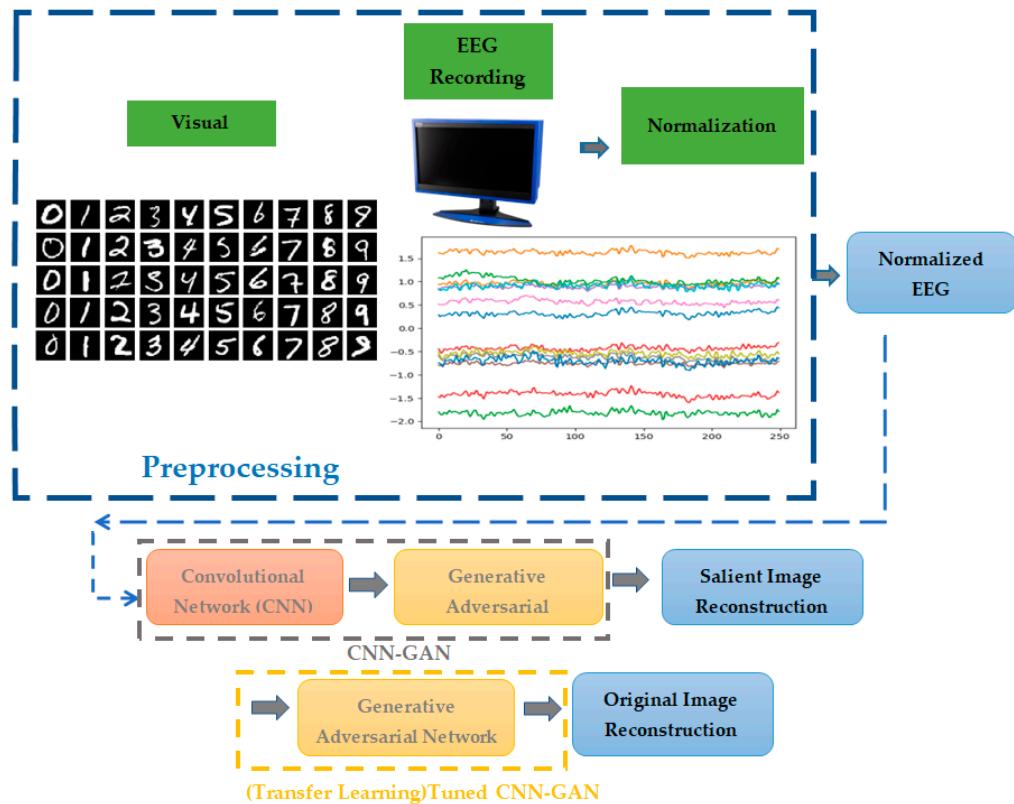


Figure 4. Details of the reconstruction procedure.

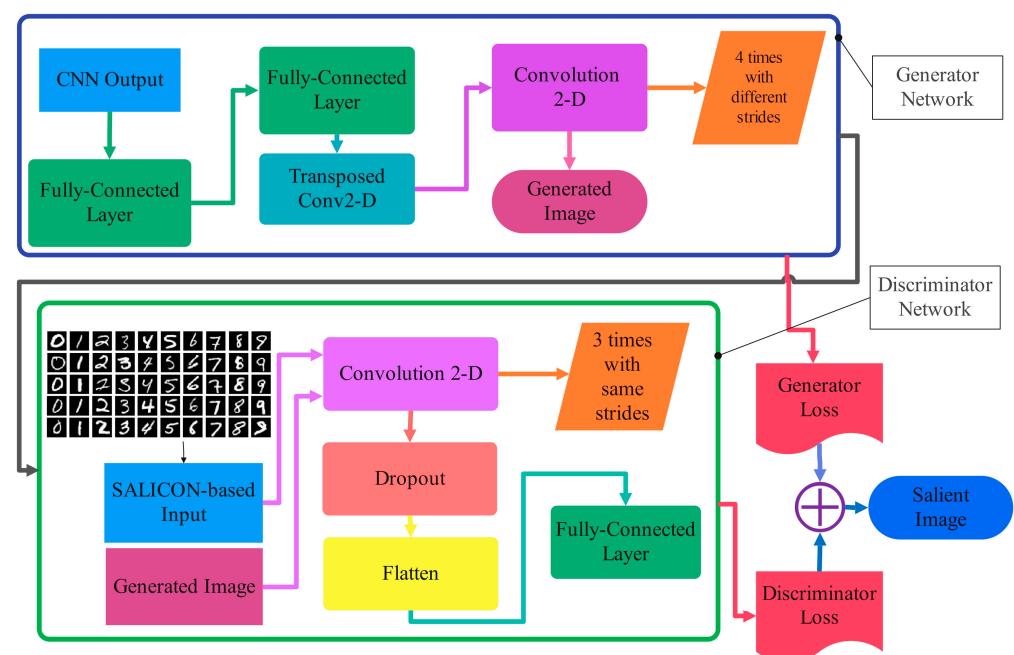


Figure 5. The details of the proposed CNN-GAN for visual salient image extraction.

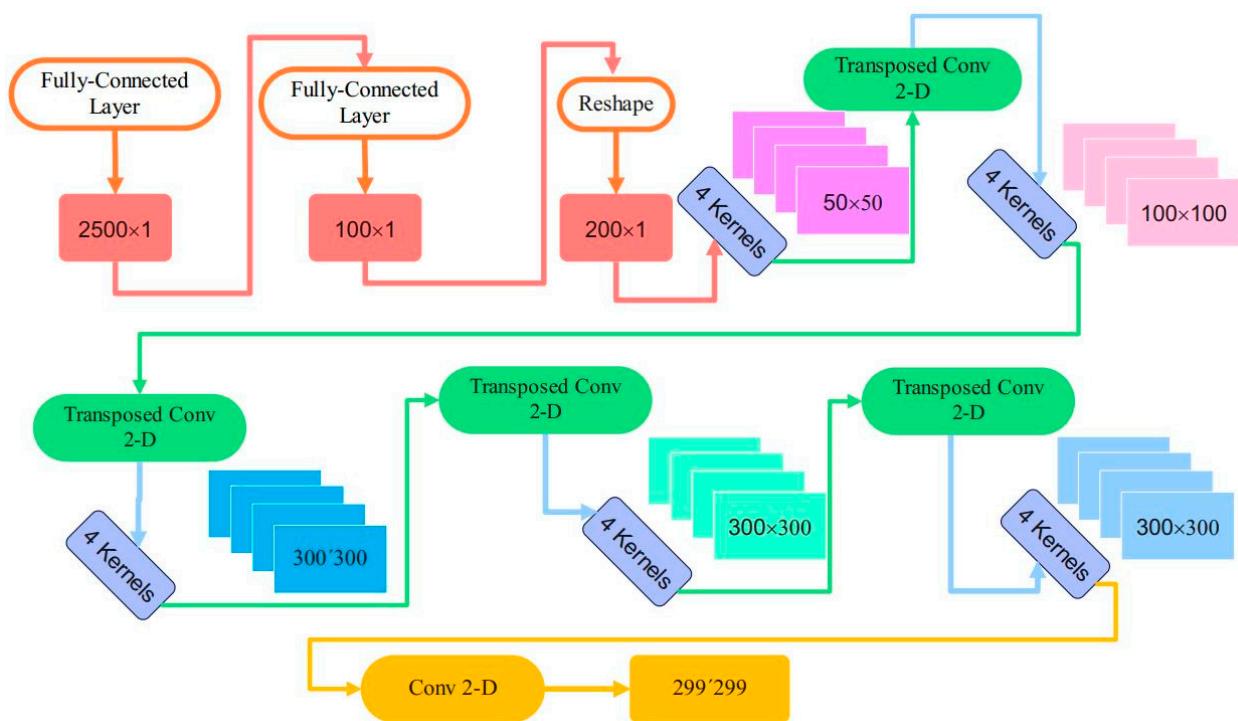


Figure 6. The output shape of layers in generator network of the CNN-GAN.

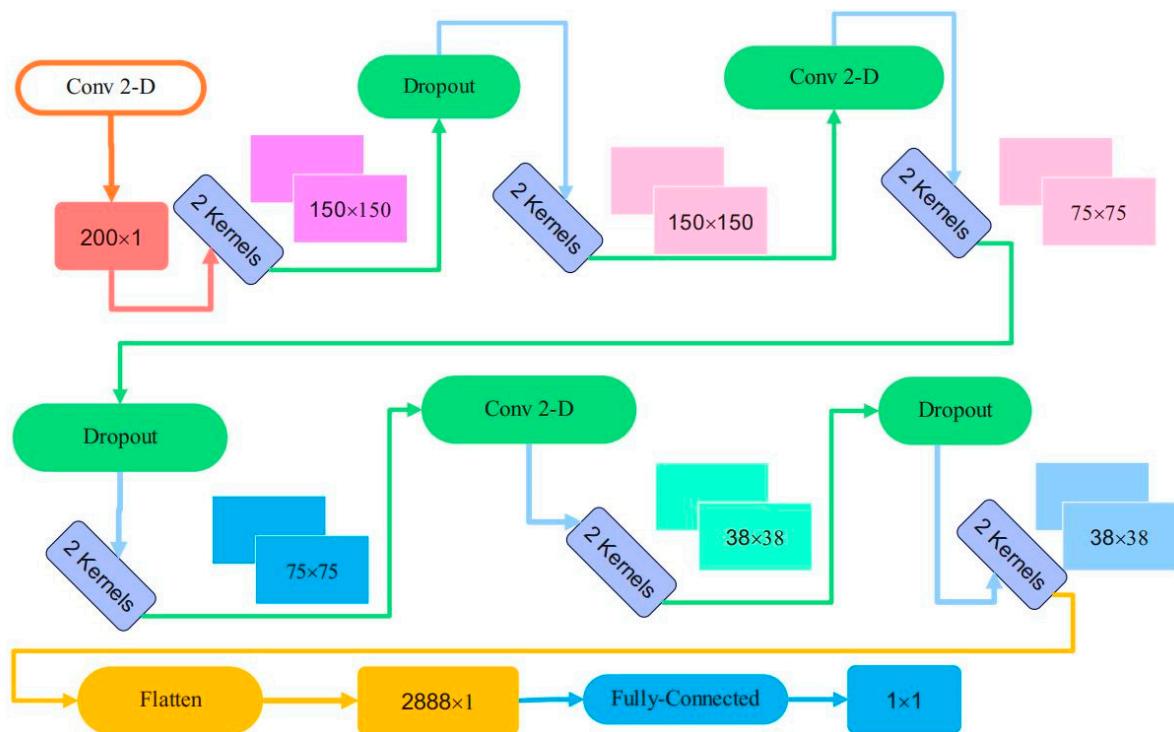
Table 3. The details of the layers in the generator of the proposed CNN-GAN.

Layer Number	Layer Name	Activation Function	Size of Kernel	Number of Kernels	Strides	Total Number of Weights	Output of the Layer
1	Full-connected					250,000	(1, 100)
2	Full-connected	Rectified LU (0.1)				2,000,000	(1, 20,000)
3	Reshape					0	(1, 50, 50, 8)
4	Conv 2-D Transposed	Rectified LU (0.1)	4 × 4	6	2 × 2	768	(1, 100, 100, 6)
5	Conv 2-D Transposed	Rectified LU (0.1)	4 × 4	6	3 × 3	768	(1, 300, 300, 6)
6	Conv 2-D Transposed	Rectified LU (0.1)	4 × 4	6	1 × 1	768	(1, 300, 300, 6)
7	Conv 2-D Transposed	Rectified LU (0.1)	4 × 4	6	1 × 1	768	(1, 300, 300, 6)
8	Conv 2-D	Rectified LU (0.1)	2 × 2	1	2 × 2	33	(1, 299, 299, 1)

Table 4 and Figure 7 describe the convolutional layers' structural details in the proposed framework's discriminator part. Three convolutional layers with kernel size of 4 and stride length of 2 have been considered. Two filters have been applied to construct the output of each convolutional layer. Furthermore, the total number of parameters is illustrated in Table 4. Flattening of the last convolutional layer is performed to apply to the final dense full-connected layer.

Table 4. The details of the layers in the discriminator of the proposed CNN-GAN.

Layer	Layer Name	Activation Function	Size of Kernel	Kernels	Stride	Total Number of Weights	Output Weight
1	Conv 2-D	Rectified LU (0.1)	4	2	2	32	(None, 150, 150, 2)
2	Dropout (rate = 0.2)					0	(None, 150, 150, 2)
3	Conv 2-D	Rectified LU (0.1)	4	2	2	130	(None, 75, 75, 2)
4	Dropout (rate = 0.2)					0	(None, 75, 75, 2)
5	Conv 2-D	Rectified LU (0.1)	4	2	2	130	(None, 38, 38, 2)
6	Dropout (rate = 0.2)					0	(None, 38, 38, 2)
7	Flattening					0	(1, 2888)
8	Full-connected					2889	(1, 1)

**Figure 7.** The output shape of layers in discriminator network of the CNN-GAN.

The procedure to create the original image from brain activity using the salient image extraction deep network is illustrated in Figure 8. Transferring the network's trained parameters for the visual stimulation's salient image to the proposed architecture in this figure would result in the original image of visual stimulation. The initialization of the weights is performed by transferring the weights, and tuning of the weights is performed through cross-validation to acquire the original image.

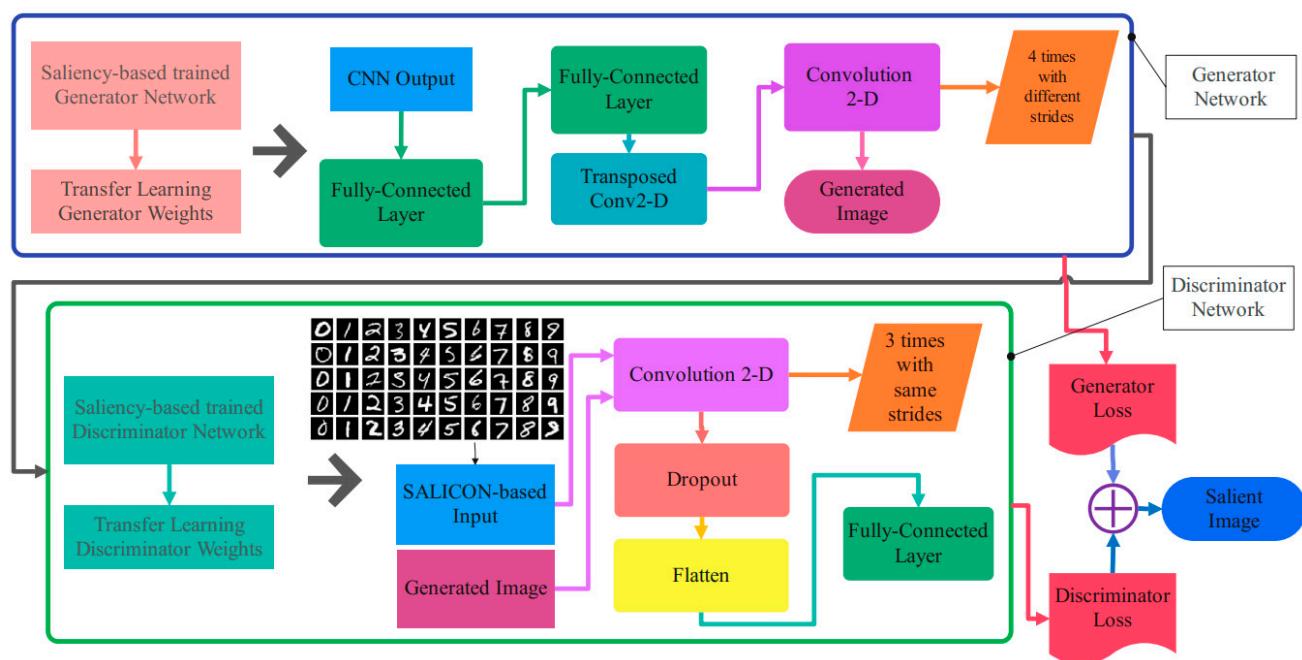


Figure 8. The details of transfer learning for image reconstruction.

4.2. Training and Evaluation

The training procedure is accomplished through cross-validation to adjust the network weights of the proposed CNN to the MindBig dataset. The trained parameters of this part are transferred to the proposed CNN-GAN to extract the salient images of visual stimulation from EEG recordings. Cross-entropy is utilized as a loss function for the training phase of CNN, and binary cross-entropy is employed for the training phase of CNN-GAN. Different parameters have been used through a trial–error technique to find the optimal values of the proposed architecture. Table 5 represents the values as search scope for the optimizer, cost function, and learning rate for the CNN and generative adversarial parts. The corresponding optimal values are obtained with trial and error and illustrated in this table.

Table 5. Search scope and optimal values in the training procedure.

Parameters	Search Scope	Optimal Value
Optimizer for CNN	SGD, Adam	SGD
Loss-function	Cross-Entropy, MSE	Cross-Entropy
Number of convolutional layers	1, 2, 3, 4	3
Learning-rate for CNN	0.001, 0.01, 0.1	0.001
Weight loss of SGD for CNN part	$5 \times 10^{-5}, 5 \times 10^{-3}$	5×10^{-5}
Dropout rate of CNN	0.2, 0.3	0.2
Optimizer for GAN	SGD, Adam	Adam
Learning-rate for GAN	0.01, 0.001, 0.0001, 0.00001	0.0001
Number of 2D-conv transposed layers of generator	4, 3, 2	4
Number of 2D-conv layers of discriminator	4, 3, 2	3
Filters for the first conv-layer in CNN	10, 14, 28	14
Filters for the second conv-layer in CNN	10, 14, 30	10

5. Results and Discussion

The practical implementation results of the proposed CNN-GAN are discussed in this section. A laptop with a GTX 1050 GPU, 16 GB RAM, and a Core i7 2.8 GHz CPU is employed to implement the proposed framework for classification. Furthermore, the reconstruction procedure is accomplished with Python programming in the Google Colaboratory platform with fast GPUs.

The train and test accuracy plots of the proposed CNN to classify the MindBig dataset into 10 categories of visually evoked brain signals corresponding to the numbers between zero and nine are represented in Figure 9. The train and test loss plots of the proposed convolutional network are illustrated in Figure 10.

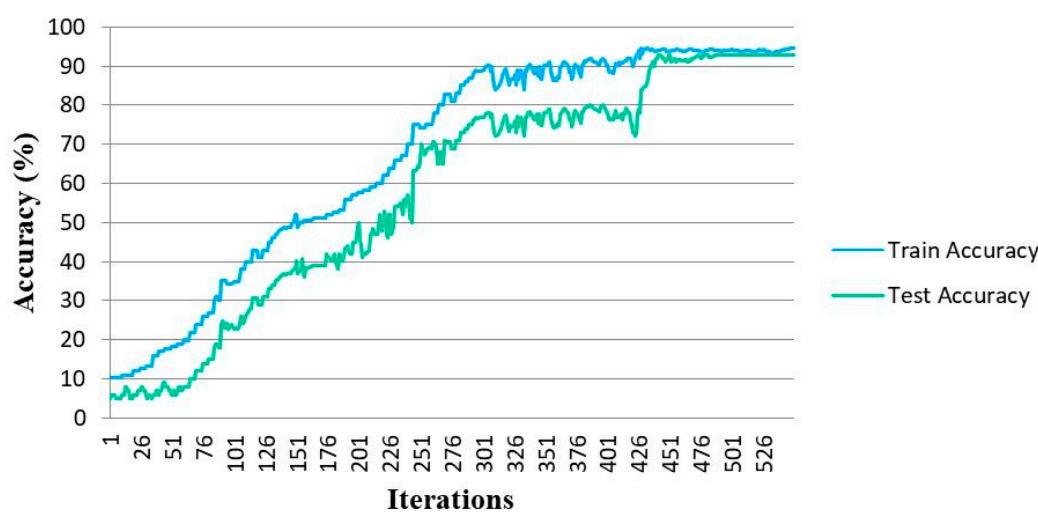


Figure 9. Train and test accuracy plots in CNN.

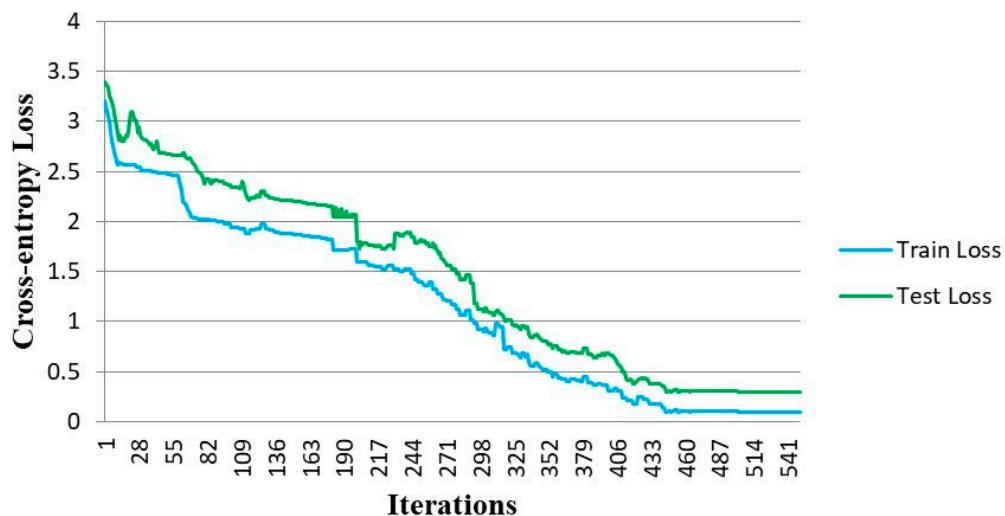


Figure 10. Train and test loss plots in CNN.

The accuracy and loss plots are tracked for 530 number of iterations. The convergence of training and testing of the proposed CNN through the implemented 10-fold cross-validation is achieved after 420 iterations and the convergence of the CNN training for the classification of the MindBig dataset is acquired.

Furthermore, the proposed CNN's efficiency compared to the other state-of-the-art deep networks is assessed with classification metrics including accuracy, Cohen's Kappa coefficient, F1-score, and precision. The results corresponding to the evaluation metrics for

LSTM [14], GNN [46], and CNN-LSTM are demonstrated in Table 6. A different number of layers has been considered for LSTM as the search scope, and three layers with 84.3% accuracy have been reported in this table. Four layers, as chebconv and graph convolutional layers, have been selected to report the graph neural network (GNN) performance for the MindBig dataset.

Table 6. Comparison of different methods for classification of the MNIST-EEG.

Evaluation Metrics	CNN	CNN + LSTM	GNN [46]	LSTM [14]
Accuracy	95.4%	86.7%	73%	84.3%
Precision	96.7%	87.8%	73.6%	84.52%
F1-score	96.7%	87.8%	73.6%	84.52%
Cohen's Kappa Coefficient	96.7%	87.8%	73.6%	84.52%

In addition to LSTM and GNN, a combination of three LSTM layers with two convolutional layers is considered and analyzed. This table confirms the efficiency of the proposed CNN for the classification of the MindBig dataset.

The accuracy trend plots of train and test procedures in each state-of-the-art method for MindBig dataset are illustrated in Figures 11 and 12, respectively. As illustrated in these two figures, the efficiency of the proposed method against other state-of-the-art methods for classifying the MindBig dataset is observable.

The effect of altering the number of convolution layers in performance and processing time in the training procedure is shown in Figures 13 and 14. As can be seen, three convolution layers result in a desirable compromise between the accuracy and processing time of the proposed CNN.

Another way to assess the performance of the proposed architecture is the representation of the confusion matrix. Figure 15 exhibits the corresponding matrix of the proposed CNN to classify the MindBig dataset. This matrix confirms the efficiency of the proposed CNN.

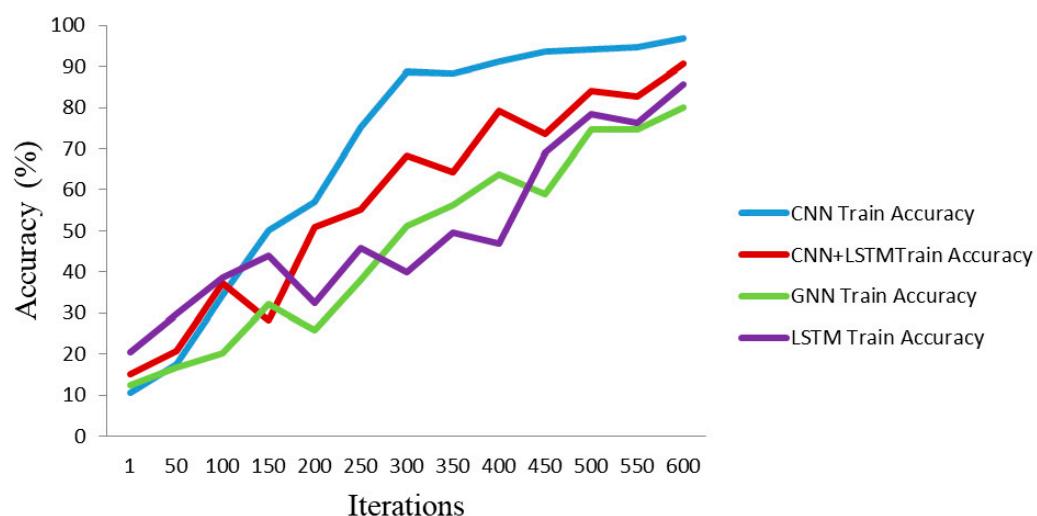


Figure 11. Comparison of train accuracy trends with state-of-the-art methods.

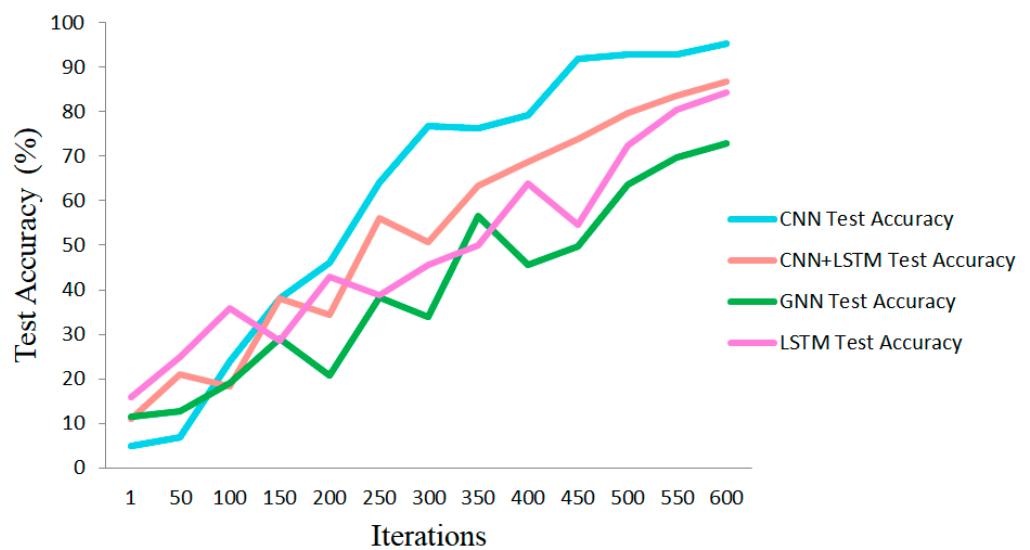


Figure 12. Comparison of test accuracy trends with state-of-the-art methods.

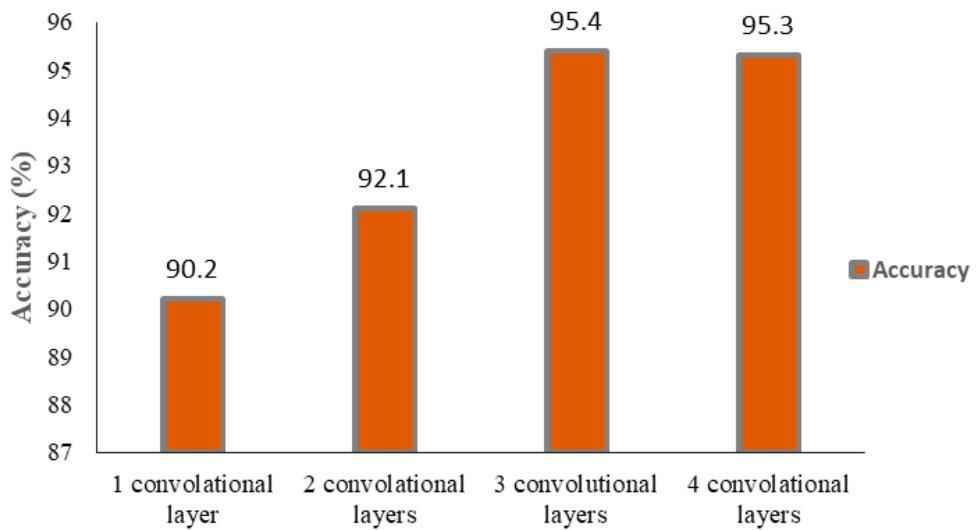


Figure 13. Training accuracy for a different number of convolutional layers in CNN.

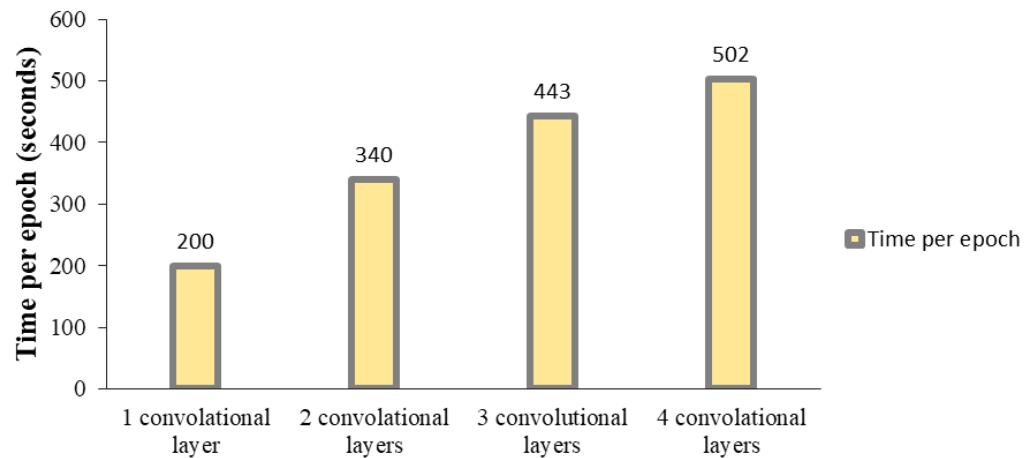


Figure 14. Processing time per epoch for different numbers of convolutional layers in CNN.

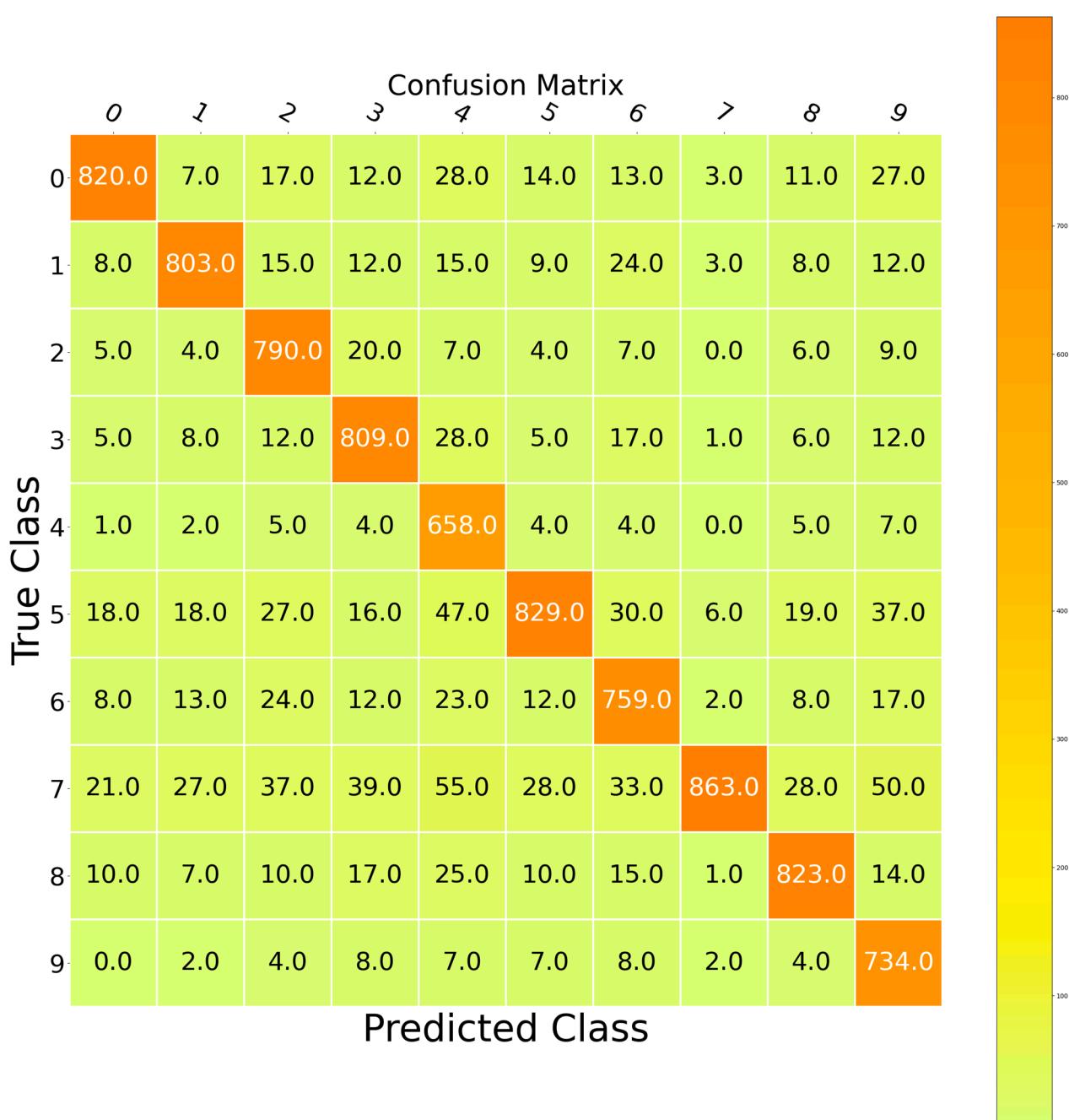


Figure 15. Confusion matrix of CNN.

The MindBig dataset consists of 9120 14-channel EEG signals. We can generate a set of signals through training a generative adversarial network (GAN) and add the generated signals to the base dataset to evaluate the performance of the proposed CNN. We generate 50 sets of 10 14-channel EEG signals according to different categories and add these 500 generated signals to the MindBigData. The generator part of the GAN consists of three transposed convolution 2-D layers, and the discriminator part of the GAN includes three convolution 2-D layers. The details of layers are presented in Tables 5 and 6.

The training of GAN is performed with 9000 iterations. The test accuracy of the proposed CNN with the new dataset after 10-fold cross-validation is equal to 90.3%. The accuracy of the network with the pre-trained weights is performed considering the generated signals, and the obtained test accuracy is equal to 86.9%. Figure 16 confirms the proposed network's efficiency for classifying the new MindBigData with 9620 14-channel

EEG recordings. The details of the layers related to the generative and discriminator subnets are presented in Tables 7 and 8, respectively.

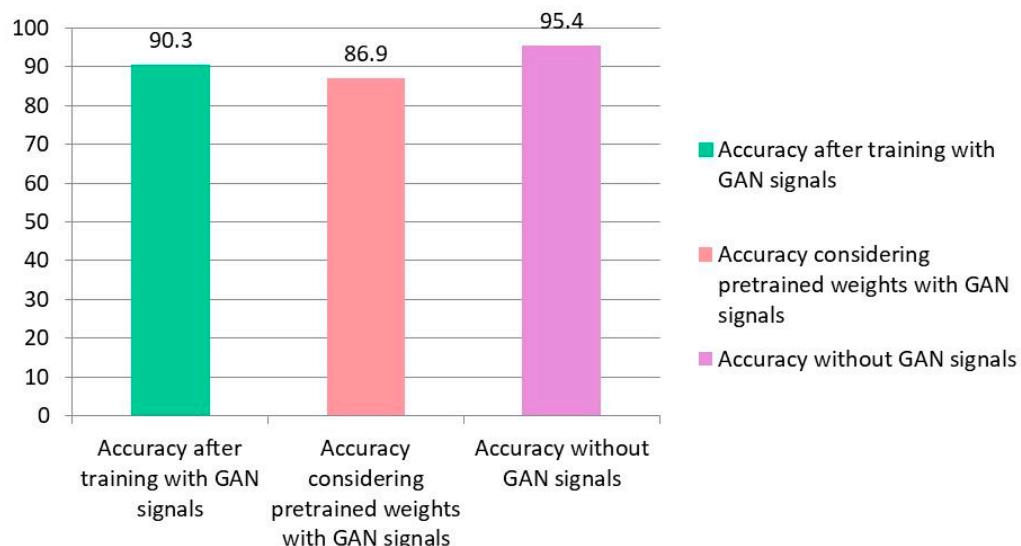


Figure 16. Comparison of computational efficiency of considered networks.

Table 7. The details of layers in the generator of GAN.

Layer Number	Layer Name	Activation Function	Output Size	Size of Kernel	Strides	Number of Kernels	Padding
1	Full-Connected	-	(7 × 125 × 8)				
2	Conv-2D Transposed	ReLU (alpha = 0.3)	(7, 125, 8)	1 × 4	1 × 1	8	Yes
3	Conv-2D Transposed	ReLU (alpha = 0.3)	(7, 125, 8)	1 × 4	1 × 1	8	Yes
4	Conv-2D Transposed	ReLU (alpha = 0.3)	(14, 250, 30)	1 × 4	2 × 2	30	Yes

Table 8. The details of layers in the discriminator of GAN.

Layer Number	Layer Name	Activation Function	Output Size	Size of Kernel	Strides	Number of Kernels	Padding
1	Conv-2D	ReLU (alpha = 0.3)	(1, 14, 250, 6)	1 × 4	1 × 1	6	Yes
2	Dropout (0.2)	-	(1, 14, 250, 6)				
3	Conv-2D	ReLU (alpha = 0.3)	(1, 7, 125, 6)	1 × 4	2 × 2	6	Yes
4	Dropout (0.2)	-	(1, 7, 125, 6)				
5	Conv-2D	ReLU (alpha = 0.3)	(1, 7, 125, 6)	1 × 4	1 × 1	6	Yes
6	Dropout (0.2)	-	(1, 7, 125, 6)				
7	Flatten	-	(1, 5250)				
8	Fully Connected	-	(1, 1)				

The evaluation of the proposed CNN-GAN for salient image extraction is accomplished in a 10-fold cross-validation considering the SSIM and CC for each image category of the MNIST dataset. The classification of the EEG database is performed in the first CNN part of the network, and the extracted feature vector is applied as the input to the next GAN deep network to extract the salient image corresponding to the visual stimulation. The results of SSIM and CC are represented in Table 9 and confirm the good performance of the proposed CNN-GAN for reconstructing the salient visual stimulation.

Table 9. The saliency validation metrics for visual salient reconstruction.

Category Number	Arithmetic Category	SSIM	CC
1	Zero	91.7	95.6
2	One	93.2	98.2
3	Two	95.3	97.1
4	Three	92.4	96.8
5	Four	91.5	96.1
6	Five	91.1	96.8
7	Six	94.8	99.4
8	Seven	93.6	97.7
9	Eight	94.5	99.2
10	Nine	91.8	95.9
-	Average	92.9	97.28

The weights of the trained CNN-GAN for salient image extraction from the brain activity are transferred to the network in order to reconstruct the original image. The initialization of the parameters is performed in the transfer learning procedure, and the adjustment of the new weights to reconstruct the visual stimulation images is accomplished by tracking the loss function of the generator and discriminator networks. The cross-entropy trend curves corresponding to loss function in salient image and original image extraction are illustrated in Figure 17. Furthermore, the plots of tracking the CC and SSIM metrics according to each iteration are represented in this figure considering four categories in the MNIST dataset.

The results of the extracted salient image and reconstructed original image according to four visual stimulation groups are represented in Figure 18. Furthermore, the ground truth image and the actual visual stimulation image are described in this figure. The visual assessment and the evaluation of the SSIM and CC metrics validate the efficiency and good performance of the proposed CNN-GAN framework.

Table 10 compares the performance of the proposed CNN-GAN against other valuable state-of-the-art method methods of SALICON [38], SalNet [39], visual classifier-driven detector [44], neural-driven detector [44], and GNN-based deep network [45] for saliency reconstruction. This table confirms the efficiency of the proposed CNN-GAN method.

One of the restrictions of the proposed method to be overcome in future works is constructing the reference dataset for salient image extraction. This article's reference data for visual saliency is gathered by implementing the SALICON technique in the CAFFE environment compiled to be compatible with the Python programming language. This would be considered in future works to have salient data using an eye-tracker for tracking the pupil position to identify the visual salient part in the images.

Another recommendation to be considered in future works is more complicated arithmetic content for visual stimulation, and EEG records could be analyzed in these complex situations.

Channel selection is another recommendation to be explored in future works. Studying the effects of different EEG channels in classification and salient arithmetic content

extraction would be beneficial. The channels with the most discriminative data could be diagnosed through the experiment.

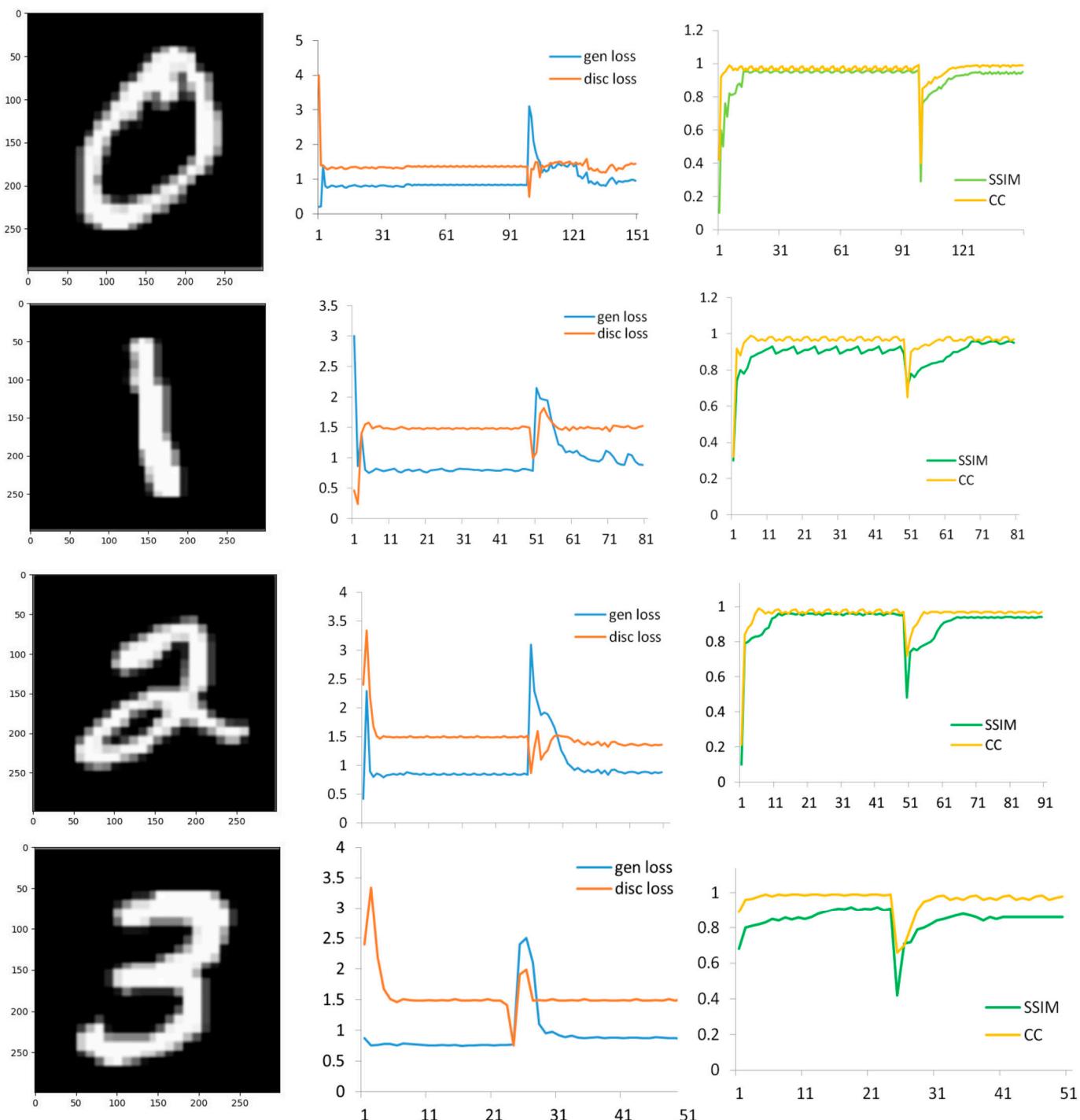


Figure 17. Train accuracy and loss plots of discriminator and generator network for four categories of MNIST dataset; from left to right: MNIST Image, loss trend curves of generator and discriminator for salient image extraction and construction of the original image, CC and SSIM plots for salient image and actual image extraction.

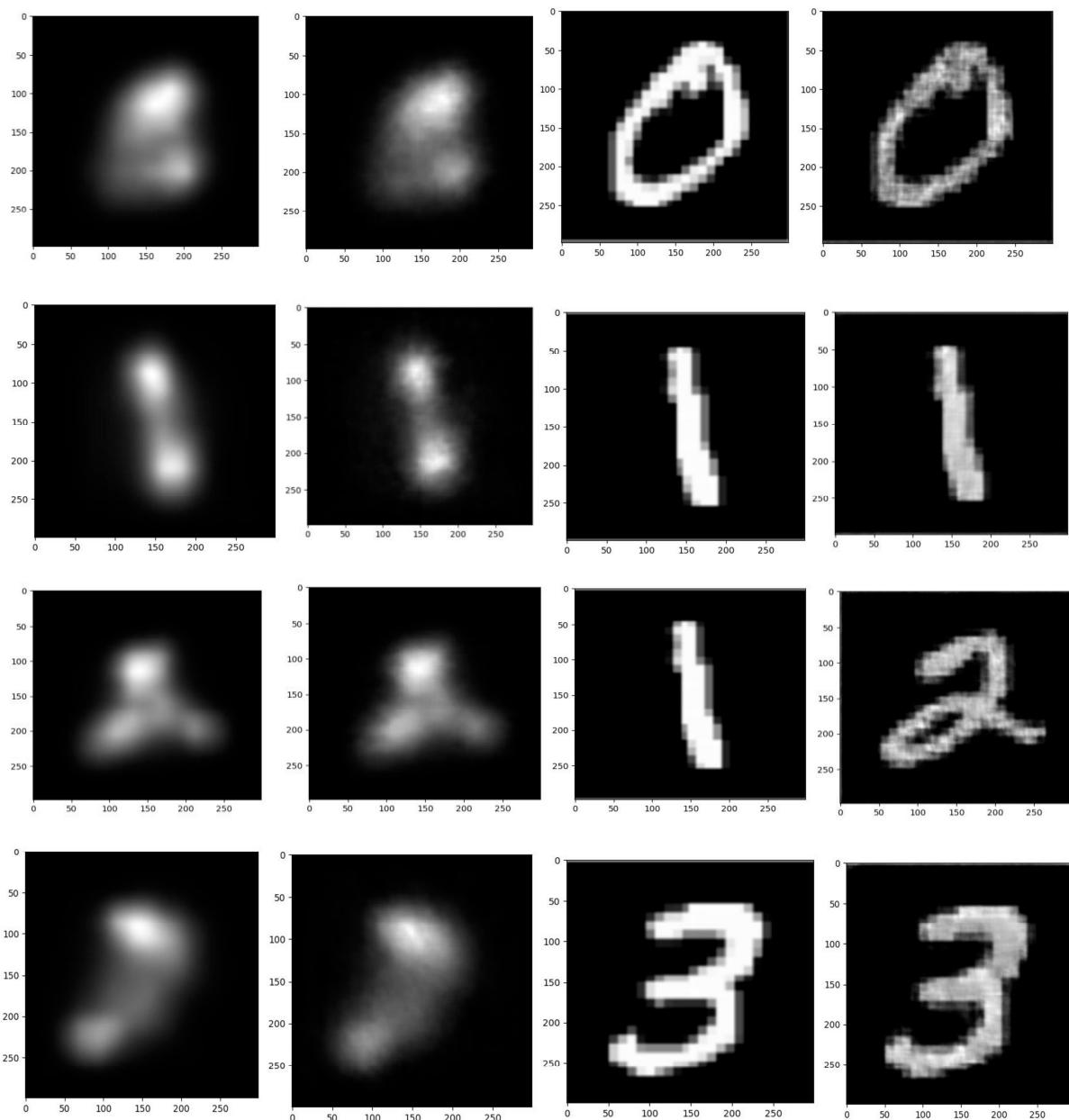


Figure 18. The extracted salient and original images were in four categories, from left to right: the ground reference data, the extracted salient image from brain activity, the original image, and the extracted visual stimulation image using the EEG recordings.

Table 10. Comparison of different methods for saliency detection.

Method	Dataset	SSIM	CC
Visual classifier-driven detector [44]	EEG-ImageNet	-	17.30%
Neural-driven detector [44]	EEG-ImageNet	-	35.7%
SalNet [39]	ImageNet	-	27.10%
SALICON [38]	ImageNet	-	34.8%
GNN-based deep network [45]	EEG-ImageNet	89.46%	99.39%
CNN-GAN	MindBigData	92.9%	97.28%

6. Conclusions

This paper proposes an effective convolutional neural network to extract the arithmetic visual stimulation utilizing the brain activity recordings provoked by images of 10 different groups of the MNIST database. The proposed CNN-GAN is trained to extract the salient arithmetic content corresponding to the visual stimulation utilizing the time samples of EEG signal recordings. The trained parameters are used as the initialization weights of the proposed framework to extract the original version of arithmetic visual stimuli images. The application of this research in BCI projects must be addressed. The implementation of the proposed method in this article in the real world would be helpful for disabled or blind subjects to have better interaction with the surrounding environment.

Author Contributions: Conceptualization, N.K., S.H. and S.S.; methodology, S.S. and S.D.; validation, S.S. and S.D.; writing—original draft preparation, N.K. and S.Z.A.; writing—review and editing, S.S. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Experimental data were not recorded in this research.

Informed Consent Statement: Not applicable.

Data Availability Statement: The EEG dataset is available online at <https://mindbigdata.com/opendb/> (Accessed on 12 February 2020).

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

Nomenclature

BCI	Brain–Computer Interface
CNN	Convolutional Neural Network
EEG	Electroencephalogram
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GNN	Graph Neural Network
LSTM	Long Short-Term Memory
MNIST	Modified National Institute of Standards and Technology
SALIENCY	Saliency in Context
SGD	Standard Gradient Descent
SIM	Similarity
SSIM	Structural Similarity
SSVEP	Steady-State Visually Evoked Potential
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group

References

1. Wang, C.; Xiong, S.; Hu, X.; Yao, L.; Zhang, J. Combining features from ERP components in single-trial EEG for discriminating four-category visual objects. *J. Neural Eng.* **2012**, *9*, 056013. [[CrossRef](#)] [[PubMed](#)]
2. Gilbert, C.D.; Sigman, M.; Crist, R.E. The neural basis of perceptual learning. *Neuron* **2001**, *31*, 681–697. [[CrossRef](#)] [[PubMed](#)]
3. Shenoy, P.; Tan, D.S. Human-aided computing: Utilizing implicit human processing to classify images. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Toronto, ON, Canada, 26 April–1 May 2014; pp. 845–854.
4. Lotte, F.; Bougrain, L.; Cichocki, A.; Clerc, M.; Congedo, M.; Rakotomamonjy, A.; Yger, F. A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update. *J. Neural Eng.* **2018**, *15*, 031005. [[CrossRef](#)]
5. Namazifard, S.; Daru, R.R.; Tighe, K.; Subbarao, K.; Adnan, A. Method for Identification of Multiple Low-Voltage Signal Sources Transmitted Through a Conductive Medium. *IEEE Access* **2022**, *10*, 124154–124166. [[CrossRef](#)]
6. Namazifard, S.; Subbarao, K. Multiple dipole source position and orientation estimation using non-invasive EEG-like signals. *Sensors* **2023**, *23*, 2855. [[CrossRef](#)]

7. Sabahi, K.; Sheykhanvand, 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.* **2023**, *14*, 75–92.
8. Seyed Alizadeh, S.M.; Bagherzadeh, A.; Bahmani, S.; Nikzad, A.; Aminzadehsarikhanbeglou, E.; Tatyana Yu, S. Retrograde gas condensate reservoirs: Reliable estimation of dew point pressure by the hybrid neuro-fuzzy connectionist paradigm. *J. Energy Resour. Technol.* **2022**, *144*, 063007. [[CrossRef](#)]
9. Baradaran, F.; Farzan, A.; Danishvar, S.; Sheykhanvand, S. Customized 2D CNN Model for the Automatic Emotion Recognition Based on EEG Signals. *Electronics* **2023**, *12*, 2232. [[CrossRef](#)]
10. Milani, O.H.; Nguyen, T.; Parekh, A.; Cetin, A.E.; Prasad, B. 0537 Incident Hypertension Prediction in Obstructive Sleep Apnea using Machine Learning. *Sleep* **2023**, *46* (Suppl. S1), A236–A237. [[CrossRef](#)]
11. Woodbright, M.; Verma, B.; Haidar, A. Autonomous deep feature extraction based method for epileptic EEG brain seizure classification. *Neurocomputing* **2021**, *444*, 30–37. [[CrossRef](#)]
12. Ak, A.; Topuz, V.; Midi, I. Motor imagery EEG signal classification using image processing technique over GoogLeNet deep learning algorithm for controlling the robot manipulator. *Biomed. Signal Process. Control* **2022**, *72*, 103295. [[CrossRef](#)]
13. Kwak, N.-S.; Müller, K.-R.; Lee, S.-W. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PLoS ONE* **2017**, *12*, e0172578. [[CrossRef](#)] [[PubMed](#)]
14. 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.
15. 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*, 1–11. [[CrossRef](#)]
16. Cudlenco, N.; Popescu, N.; Leordeanu, M. Reading into the mind’s eye: Boosting automatic visual recognition with EEG signals. *Neurocomputing* **2020**, *386*, 281–292. [[CrossRef](#)]
17. Mathur, N.; Gupta, A.; Jaswal, S.; Verma, R. Deep learning helps EEG signals predict different stages of visual processing in the human brain. *Biomed. Signal Process. Control* **2021**, *70*, 102996. [[CrossRef](#)]
18. Ghosh, L.; Dewan, D.; Chowdhury, A.; Konar, A. Exploration of face-perceptual ability by EEG induced deep learning algorithm. *Biomed. Signal Process. Control* **2021**, *66*, 102368. [[CrossRef](#)]
19. 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. [[CrossRef](#)]
20. Kay, K.N.; Naselaris, T.; Prenger, R.J.; Gallant, J.L. Identifying natural images from human brain activity. *Nature* **2008**, *452*, 352–355. [[CrossRef](#)]
21. 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](#)]
22. 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](#)]
23. Brouwer, G.J.; Heeger, D.J. Decoding and reconstructing color from responses in human visual cortex. *J. Neurosci.* **2009**, *29*, 13992–14003. [[CrossRef](#)]
24. Koch, C.; Ullman, S. Shifts in selective visual attention: Towards the underlying neural circuitry. In *Matters of Intelligence*; Springer: Berlin/Heidelberg, Germany, 1987; pp. 115–141.
25. 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](#)]
26. Achanta, R.; Estrada, F.; Wils, P.; Süstrunk, S. Salient region detection and segmentation. In Proceedings of the International Conference on Computer Vision Systems, Santorini, Greece, 12–15 May 2008; pp. 66–75.
27. 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, 4–6 November 2003; pp. 374–381.
28. 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, 28 November–30 December 2005; pp. 716–724.
29. Rosin, P.L. A simple method for detecting salient regions. *Pattern Recognit.* **2009**, *42*, 2363–2371. [[CrossRef](#)]
30. 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.
31. Neisser, U. Cognitive psychology. Appleton-Century-Crofts. [aJRH] Newell, A. (1982) The knowledge level. *Artif. Intell.* **1967**, *18*, 82127.
32. 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.
33. Zhao, J. Adaptive Q-S synchronization between coupled chaotic systems with stochastic perturbation and delay. *Appl. Math. Model.* **2012**, *36*, 3312–3319. [[CrossRef](#)]
34. Xu, J.; Jiang, M.; Wang, S.; Kankanhalli, M.S.; Zhao, Q. Predicting human gaze beyond pixels. *J. Vis.* **2014**, *14*, 28. [[CrossRef](#)]
35. 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](#)] [[PubMed](#)]

36. 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](#)]
37. 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.
38. 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.
39. 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, 26 June–1 July 2016; pp. 598–606.
40. 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](#)]
41. 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](#)]
42. 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; pp. 287–302.
43. 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*. [[CrossRef](#)]
44. 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**, *43*, 3833–3849. [[CrossRef](#)]
45. Khaleghi, N.; Rezaii, T.Y.; Beheshti, S.; Meshgini, S.; Sheykhhivand, 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. [[CrossRef](#)]
46. Khaleghi, N.; Rezaii, 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](#)]
47. Vivancos, D.; Cuesta, F. MindBigData 2022 A Large Dataset of Brain Signals. *arXiv* **2022**, arXiv:2212.14746.
48. Hubel, D.H.; Wiesel, T.N. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **1959**, *148*, 574. [[CrossRef](#)]
49. Fukushima, K. Self-organization of a neural network which gives position-invariant response. In Proceedings of the 6th International Joint Conference on Artificial Intelligence, Tokyo, Japan, 20–23 August 1979; Volume 1, pp. 291–293.
50. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 13 November 2023).
51. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
52. 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](#)]
53. 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.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.