# ThoughtViz: Visualizing Human Thoughts Using Generative Adversarial Network

Praveen Tirupattur
University of Central Florida
praveentirupattur@knights.ucf.edu

Yogesh Singh Rawat
University of Central Florida
yogesh@crcv.ucf.edu

Concetto Spampinato
University of Catania
cspampin@dieei.unict.it

Mubarak Shah
University of Central Florida
shah@crcv.ucf.edu

## ABSTRACT

Studying human brain signals has always gathered great attention from the scientific community. In Brain Computer Interface (BCI) research, for example, changes of brain signals in relation to specific tasks (e.g., thinking something) are detected and used to control machines. While extracting spatio-temporal cues from brain signals for classifying state of human mind is an explored path, decoding and visualizing brain states is new and futuristic. Following this latter direction, in this paper, we propose an approach that is able not only to read the mind, but also to decode and visualize human thoughts. More specifically, we analyze brain activity, recorded by an ElectroEncephaloGram (EEG), of a subject while thinking about a digit, character or an object and synthesize visually the thought item. To accomplish this, we leverage the recent progress of adversarial learning by devising a conditional Generative Adversarial Network (GAN), which takes, as input, encoded EEG signals and generates corresponding images. In addition, since collecting large EEG signals in not trivial, our GAN model allows for learning distributions with limited training data. Performance analysis carried out on three different datasets – brain signals of multiple subjects thinking digits, characters, and objects – show that our approach is able to effectively generate images from thoughts of a person. They also demonstrate that EEG signals encode explicitly cues from thoughts which can be effectively used for generating semantically relevant visualizations.

## CCS CONCEPTS

• **Computer systems organization → Embedded systems**;

## KEYWORDS

EEG, Generative Adversarial Networks, Image Generation

**ACM Reference Format:**
Praveen Tirupattur, Yogesh Singh Rawat, Concetto Spampinato, and Mubarak Shah. 2018. ThoughtViz: Visualizing Human Thoughts Using Generative Adversarial Network. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3240508.3240641


*MM '18, October 22–26, 2018, Seoul, Republic of Korea*

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240641

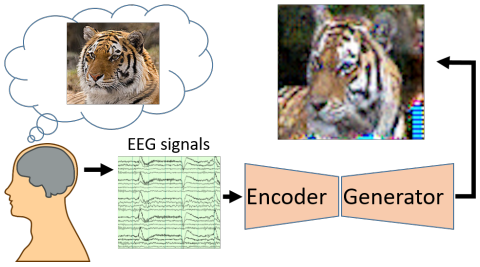


**Figure 1: Overview of our proposed method where an EEG signal from brain is sent to the encoder and the encoded signal is used to generate a visualization corresponding to the captured EEG signal.**

## 1 INTRODUCTION

Deep learning has been very successfully applied to many computer vision and multimedia tasks such as image understanding, speech recognition, and natural language processing. One elusive goal that still remains is to apply deep learning to understand and interpret the inner workings of the human brain. Many of the earlier works in this area focus on decoding informative patterns from brain signals to control machines via a brain computer interface [9, 10, 19, 28] and some medical applications [1]. Brain activity is usually captured by recording the electric potentials produced by neurons using an Electroencephalogram (EEG) or by brain imaging techniques such as MRI (magnetic resonance imaging) and Functional MRI (fMRI). In these earlier studies it has been demonstrated that brain signals contain informative cues reflecting human cognitive processes and can be effectively used in various applications.

Recent works have investigated how to decode visual and linguistic content from brain signals [12, 13, 23, 30] – recorded with EEG or fMRI – of subjects while they are either involved in verbal communication or in a visual task. These works have shown some promising results mainly because they have demonstrated that brain signals contain informative cues corresponding to the visual or linguistic exposure of the subject.

Motivated by these studies, we want to take a step further and study the brain activity of a person's thoughts. Our goal is to extract some cues from the brain activity, recorded using low-cost EEG

devices[1], and use them to visualize the thoughts of a person. More specifically, we attempt to visualize the thoughts of a person by generating an image of an object that the person is thinking about. EEG data of the person is captured while he is thinking of that object and is used for image generation. We use a publicly available EEG dataset [16] for our experiments and propose a generative adversarial model for image generation. We make the following contributions in this work: 1) we introduce the problem of interpreting and visualizing human thoughts, 2) we propose a novel conditional GAN architecture, which generates class-specific images according to specific brain activities; 3) finally, we also show that our proposed GAN architecture is well suited for small-sized datasets and can generate class-specific images even when trained on limited training data.

We demonstrate the feasibility and the effectiveness of the proposed method on three different object categories, i.e., digits, characters, and photo objects, and show that our proposed method is, indeed, capable of reading and visualizing human thoughts.

The rest of the paper is organized as follows. In Section 2, we discuss previous works related to Brain Computer Interface (BCI) applications using EEG data and image generation using Generative Adversarial Networks (GAN). In Section 3, we provide the details of our approach for EEG classification and image generation. Section 4 presents the experiments we performed to evaluate our approach, the evaluation metrics used, and discusses the results of the experiments. Finally, in Section 5 we provide the conclusions and possible directions for future work.

## 2 RELATED WORK

Brain activity signals have been widely used mainly for BCI (brain computer interface) applications [9, 10, 19, 28] and for medical applications [3, 32, 33]. Some of the most recent works involve recognizing simple patterns from brain signals to identify the stimuli that evoke specific responses [12, 13, 30]. Brain activity can be recorded using multiple techniques such as fMRI, EEG, and MEG, whose spatial and temporal resolutions have allowed computational methods to decode specific visual and linguistic stimuli [12, 13, 33]. Among the available neuro-imaging techniques, EEG presents several advantages that makes it particularly suited for this kind of research. Indeed, EEG is a low-cost technique which can provide higher temporal resolution than MRI/fMRI.

With the recent rediscovery of deep learning [17] and its success in solving a variety of tasks, different deep learning-based approaches processing EEG data to discriminate semantically different stimuli sources have been proposed [2, 25, 27]. The main outcome of these works is that both convolutional neural networks (CNNs) and recurrent neural networks (such as LSTM) can effectively tackle EEG classification tasks. Accordingly, in this paper we present different CNN and LSTM architectures to a) perform classification of EEG data related to human thoughts and, b) use them for encoding EEG data in order to condition a downstream generative method for converting high-level classes to images.

Image generation from a latent feature space is currently an active research area. Several deep learning approaches have been proposed from variational autoencoders [15] to autoregressive models [22] to generative adversarial networks (GAN) [8]. Among these techniques, GANs have arisen as the most promising paradigm for image generation. A typical GAN framework consists of two networks, a generator G and a discriminator D which are trained adversarially to improve by competing with each other. In this work we mainly focus on developing a generative adversarial network for generating images.

A traditional GAN architecture allows us to synthesize data samples using noise input. A specific type of GAN model is the conditional GAN [18], where the generator is conditioned to generate samples from specific classes. Usually, conditioning is achieved by providing as input to the generator either one-hot vectors or specific features describing classes along with the noise. There are different variants of conditional GAN architectures [4, 7, 18, 21]. In these models, the discriminator is usually tasked with classifying the samples generated by the generator as fake ones. We, instead, propose an architecture with an additional classifier aiming at explicitly classifying the generated samples instead of using the discriminator. We show with experiments that this leads to a faster convergence of the GAN model.

In addition, GAN training is challenging with limited training data, as in the case of cognitive studies involving EEG data recordings. To cope with lack of training data, we extend our GAN model by integrating the idea proposed in [11], where the latent generative space is reparameterized as a mixture model and it's parameters are learned along with the GAN training.

In this paper we tackle the problem of visually synthesizing human thoughts. Visualizing brain activity of a person performing a visual task has also been investigated [13, 14, 14, 20, 23]. In [13, 20] the authors have used fMRI images to visualize the visual stimuli generated in the brain signal when a person is watching a movie clip. The brain activity captured with fMRI has shown great potential in various applications, however it is not cost effective. More recently, the authors in [14, 23] have proposed to use EEG signals to visualize the visual stimuli in brain activity. The difference between these works and ours lie in a) the stimuli that evokes the EEG signals, i.e., thoughts vs images; and b) the conditional GAN architectures; a comparison between our method and [14] is given in Sect. 4.3.

## 3 PROPOSED METHOD

In this work we attempt to visualize the thoughts of a person using brain activity captured by EEG signals. To this end we use a dataset containing EEG signals which were collected from multiple subjects thinking of different objects. The details of the dataset are discussed in section 4.1. For the generation of images from human thoughts, we propose the use of an GAN model conditioned on EEG data. Our approach consists of two phases: 1) classification of EEG signals to identify the object a person is thinking about, and 2) image generation using the EEG encoding learned in the previous phase for conditioning the generation process. In the first step, we devise and train an CNN-based classifier for discriminating EEG signals of thoughts. Then we use this classifier to get an encoding for the EEG signals. This EEG encoding extracted from the trained classifier is then used for conditioning the GAN model. The GAN model is trained using an adversarial and classifier loss, and the trained

[1] www.emotiv.com/

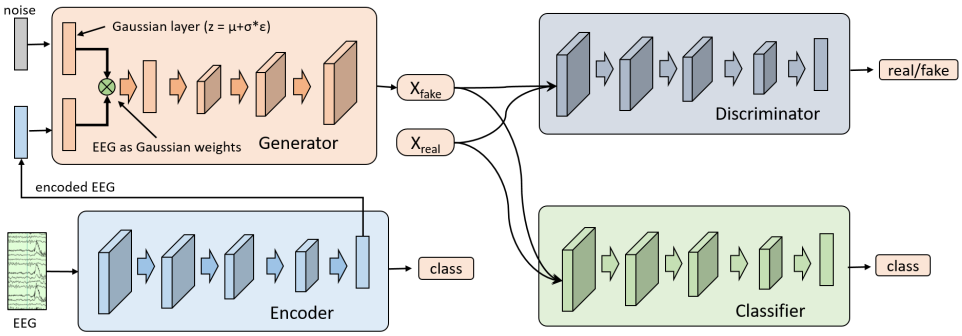


**Figure 2: An overview of the proposed GAN architecture for image generation from EEG signals. The EEG signal is passed to the encoder, and the encoded signal is used as conditioning for the generator. The generated image is passed to the discriminator for an adversarial loss and also to a classifier for classification loss.**

generator is used to generate class-specific images. The complete architecture of our approach is shown in Fig. 2. We will discuss the proposed EEG encoding and the GAN architecture in the following subsections.

## 3.1 EEG Classification

As a first step in our approach we need to encode the EEG signals to a lower dimensional feature vector to be used for conditioning in the next step i.e., image generation. EEG signals are two-dimensional signals where the first dimension corresponds to channels and the second dimension corresponds to time. Each EEG recording corresponds to a specific object which the subject was thinking about. Therefore to encode the EEG signals, we train a classifier to discriminate between these object classes. Later we extract the output from an intermediate layer of the trained classifier model as the EEG encoding.

There can be different ways to classify EEG time-series. In [2, 25, 27] the authors proposed to use an CNN, LSTM, and a combination of these architectures to classify EEG signals for varying tasks. Based on these works, we experimented with various architectures to encode EEG signals. Figure 3 shows four different network architectures we tried on our EEG data for classification.

We observed that a network architecture where 1D-CNN is followed by 2D-CNN performs better for EEG classification as compared with other networks. This corresponds to the first architecture in Figure 3. In this network architecture, we first apply 1D convolutions along the time axis with a kernel size of 4, and then another 1D convolution along the channels axis with kernel size of 14 i.e., the number of channels. These 1D convolutions are followed by two sets of alternating 2D convolution and 2D max pooling layers. Finally, two fully connected layers with 100 neurons are added for classification. The other network architectures shown in Figure 3 do not perform well across all three EEG datasets.

## 3.2 Image generation

We propose a Generative Adversarial Network (GAN) to visualize thoughts which are encoded from EEG signals. A traditional GAN architecture consists of two main components, a generator (G) and a discriminator (D). A generator is used to generate a sample image from a random noise input ($z$), and the discriminator takes this generated sample as input and determines whether it is a generated sample or a real sample. The generator is trained so that it can generate realistic looking samples and the discriminator is trained so that it can distinguish between fake, generated, samples and real samples. The goal of the generator is to fool the discriminator in believing that the generated samples are real samples. This is done by solving the following optimization problem,

$$\min_G V_G(D,G) = \min_G \left( E_{z\sim p_z}[log(1 - D(G(z)))] \right) \tag{1}$$

In the case of the discriminator, we try to maximize the scores for real samples ($D(x)$) and minimize the scores for the fake generated samples ($G(z)$) by minimizing ($D(G(z))$). This can be achieved by solving the following optimization problem,

$$\max_D V_D(D,G) = \max_D \Big( E_{x\sim p_{data}}[logD(x)] + E_{z\sim p_z}[log(1 - D(G(z)))] \Big) \tag{2}$$

The generator G(z;$\theta_g$), learns to generate samples from the target distribution $p_{data}$ by mapping the input noise sample($z$) from a lower dimensional space($p_z$) to the target distribution $p_{data}$. The discriminator D(x;$\theta_d$), learns to distinguish between the samples generated by the generator $p_{gen}$ and the samples from the target distribution $p_{data}$. The overall objective function of a GAN network can be written as,

$$\max_D \min_G V_D(D,G) = \max_D \min_G \Big( E_{x\sim p_{data}}[logD(x)] + E_{z\sim p_z}[log(1 - D(G(z)))] \Big) \tag{3}$$

A traditional GAN architecture can generate sample images from random noise sample, but the generated samples are not class specific. To generate class-specific samples, we have conditional GAN architectures which also consider the class of a sample during generation. There are some existing methods, such as [4, 7, 18, 21], which can generate class-specific samples from noise with additional conditioning on the generator. In one of the most recent works [21], the authors have proposed an architecture where conditioning is

provided to the generator, and the discriminator is used to classify the generated sample as well as to discriminate it as real or fake. In this architecture, the discriminator also plays a non-adversarial role as it helps the generator when it tries to classify the generated sample. Both the adversarial loss as well as the classification loss are back-propagated though the same discriminator network.

We propose a slightly different strategy where we design separate networks for these two roles. In our architecture we have a discriminator which has an adversarial role and a separate classifier which is non-adversarial and helps the generator by classifying the generated images as one of the classes. We observe that apart from generating better images, this network was also able to converge much faster as compared to [21].

In our proposed approach, along with the generator G and discriminator D, we introduce one other component to the GAN framework, a classifier C(x;$\theta_c$) which is pre-trained and can classify the sample generated by the generator. The generator loss includes both the discriminative loss from D and the classification loss from C. The generator tries to minimize the classification loss while maximally fooling the discriminator. The updated objective function for the proposed architecture can be represented as,

$$\max_D \min_G \min_C V_D(D,G,C) = \max_D \min_G \min_C \Big( E_{x\sim p_{data}}[logD(x)] + E_{z\sim p_z}[log(1-D(G(z)))] + E_{z\sim p_z}[log(C(G(z)))] \Big) \tag{4}$$

The training of a GAN architecture requires a large amount of data, and getting such large-scale data of EEG recordings can be very difficult. In a recent work [11], the authors proposed a variation on traditional GAN models which can be used to train a GAN architecture even on a small sized dataset. The authors introduced a trainable Gaussian layer which makes it easier for the generator to learn mappings for complex distributions. However, their proposed method generates class independent samples. In this work we have adapted their approach for class-dependent sample generation. We have added a trainable Gaussian layer in our generator where the EEG encoding is used as weights for the Gaussian layer. The Gaussian layer has trainable weights in the form of mean ($\mu$) and variance ($\sigma$). With this additional layer, a sampled point ($\epsilon$) from the noise distribution translates to,

$$z = \mu_i + \sigma_i \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0,1) \tag{5}$$

and, we use EEG ($eeg$) as weights for this Gaussian layer and with the conditioning the latent sample then translates to,

$$w = eeg * (\mu_i + \sigma_i \epsilon) \quad \text{where} \quad \epsilon \sim \mathcal{N}(0,1) \tag{6}$$

We call this a weighted Gaussian layer which takes a random noise and EEG encoding as input and $\mu$ and $\sigma$ are trainable network parameters. In our experiments we observed that introducing a weighted trainable Gaussian layer can be very effective when we use small amount of data for training a GAN architecture.

# 4 EXPERIMENTS

In this section we will describe the datasets used, evaluation metrics employed, and the results achieved for the different models used for EEG classification and image generation tasks.

## 4.1 Datasets

In our experiments, we have used EEG data from [16]. The dataset contains EEG recording from 3 different subsets: Digits, Characters and Objects. The Digits dataset has EEG recordings (230 in total) from participants when they were thinking of one of the 10 digits (0-9). Similarly, the Characters dataset consists of EEG signals when participants were thinking of one of the 10 characters chosen from the English language. Finally, the Objects dataset has EEG recordings when participants were thinking of one of the 10 objects shown to them. Each of these subsets have EEG signals from 23 participants for all 10 classes of images and each recording is of 10 seconds. The EEGs were recorded using the Emotiv EPOC+ device which records 14 channels with a sampling rate of 128Hz per channel. Please refer to [16] for more details on the dataset.

We have trained three different generators, one each for the three subsets. To train these, we have used images from 3 different sources. For Digits, we have used the MNIST [17] dataset, for Characters we have used the fonts subset (gray scale) images from the Chars74K [5] dataset, and for Objects we have used images from the ImageNet [6] dataset. As we have EEG recordings for only 10 classes of Characters and Objects, we have picked the images of only those classes from the Chars74K and ImageNet datasets for GAN training.

*4.1.1 Preprocessing.* We apply a sliding window with overlap on the 10 second (128Hz x 10seconds = 1280 samples) EEG recording to split the signal into chunks with a window size of 32 samples and an overlap of 8. The resulting chunks are used for training the EEG classifier. For digit generation, we have used all of 50,000 images from the MNIST training set without any preprocessing. For character generation, we have used a total of 10,000 images with 1000 images per class from the Chars74K dataset. All the images were inverted and resized to 28 x 28 before training to be consistent with the MNIST dataset. The input images, both MNIST and Chars74k, are normalized to have the pixel values in the range [-1, 1]. For the Object generation, we picked 1000 images from the ImageNet dataset, with 100 images per class (10 classes). These images were picked manually to avoid large intraclass variance in the training data. All the images are reshaped to 64 x 64 and normalized to have their pixel values in range [0, 1] before training the generator.

## 4.2 Evaluation Metrics

As mentioned in Section 3, we follow a two step approach. We first train a classifier on the EEG signals to extract the EEG encoding and then train a GAN to generate images using the EEG encoding as conditioning. In this section we present the evaluation metrics employed in both of these steps.

To evaluate the performance of an EEG classifier, we use classification accuracy on the test set (20 % of the total data). Metrics for evaluation of the quality of generated images when employing a generative model is still an unsolved problem, and not many

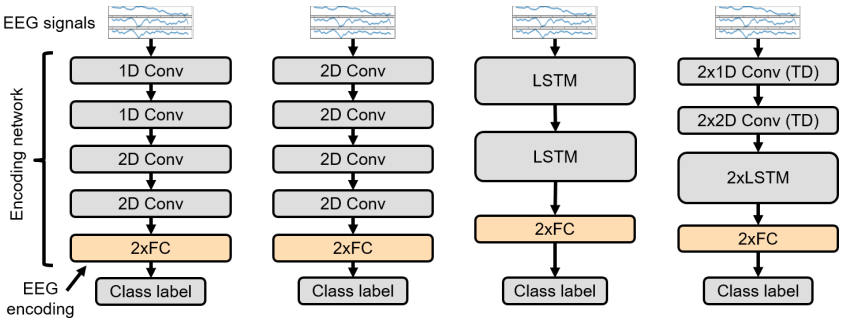


**Figure 3: CNN architectures for classification of EEG signals. The intermediate feature vector from the first fully connected layer is considered as the EEG encoding for a given input EEG signal. We compared the performance of these different architectures and found that the first architecture with 1D-Convolutions followed by 2D Convolutions performs better as compared with others. (TD): These are time-shared convolution networks sharing weights and passing the time-series encoding to the LSTM network.**

approaches are proposed in the literature. The only well know metric is Inception Score [26], which can be used for images of objects. In our work, we employ this metric to evaluate the generated images for the Objects dataset. Apart from this metric, for all the three datasets, we use the classification accuracy of generated images from a trained image classifier as a metric to evaluate the performance of the generator. To this end, we train three image classifiers (one for each of the three subsets: digit, character and object datasets) whose architecture is based on VGG-16 [29]. The image datasets collected for GAN training (MNIST, Chars74K and ImageNet) are split into train(60%), validation(20%) and test(20%) sets and are used to train the image classifiers used for evaluating generator models.

## 4.3 Results

*4.3.1 EEG Classification.* For EEG classification, we tried four different architectures as explained in Section 3.1. Of the four architectures, we picked the architecture which gives the best classification accuracy on the test set. We trained three classifiers with this architecture, one each for the three EEG datasets. We use ReLU activation for all the layers in our network and Softmax for the final classification layer. We also use batch normalization layers after each of the final two fully connected layers. We train the classifiers with SGD optimizer with batch size of 32, a learning rate of 1e-4, momentum set to 0.9 and with decay of 1e-6. We train the network until it converges on the validation set. After training, the classifiers are evaluated by computing the classification accuracy on the test set. We obtained a test classification accuracy of 71-72% for all three EEG datasets. The results are presented in Table 1. With 2D convolution and LSTM architectures we obtained an average classification accuracy of around 40%, and with the time-shared convolution network we obtain an average classification accuracy of around 60%.

*4.3.2 Image Classification.* To evaluate the performance of the generator, we use a trained image classifier as explained in Section 4.2. To this end, we train three image classifiers, one on each of the three image datasets. We use the VGG-16 [29] network architecture for all three classifiers and set the size of the final dense layer to 10. We train these classifiers from scratch with a batch size of 32 using Adam optimizer. We use a learning rate of 5e-4 with decay of 1e-3 and train the network until convergence on the validation set. The test set is used to evaluate the performance of these classifiers. The classification scores on the three datasets are shown in Table 2.

| | Digits | Characters | Objects |
|---|---|---|---|
| Accuracy | 72.88% | 71.18% | 72.95% |

**Table 1: Classification accuracy of EEG classifiers. These EEG classifiers are used to extract conditioning vectors for generator training.**

| | Digits | Characters | Objects |
|---|---|---|---|
| Accuracy | 99.12% | 98.81% | 83.24% |

**Table 2: Classification accuracy of image classifiers. These pre-trained classifiers are used both in GAN training and to evaluate the trained generators.**

*4.3.3 Image Generation.* To evaluate the generator, we employ two metrics : Inception Score and image classification accuracy as explained in Section 4.2. To evaluate the generator using the trained image classifier we follow the same two step process that we followed to train the generator. We first extract the EEG encodings using the trained EEG classifier for the EEG signals from the test set. We use these EEG conditioning vectors from the test set to generate images using a trained generator. The generated images are classified by the trained image classifier to give us the classification accuracies. The ground truth labels used to calculate classification accuracy are the class labels of the EEG signals from the test set. In Table 3 we present the classifier accuracies achieved by our approach on all three datasets. As you could observe from the results, we achieve very high classification accuracies on all three datasets.

The other metric that we use to evaluate the generator for the Objects dataset is the Inception score. The Inception score is a metric for automatic evaluation of the quality of images generated

| | Digits | Characters | Objects |
|---|---|---|---|
| AC-GAN [21] (EEG Conditioning) | 74.10% | 52.57% | 70.36% |
| AC-GAN [21] (1-hot Conditioning) | 82.06% | 79.95% | 62.44% |
| Brain2Image [14] | 28.32% | 17.76% | 12.05% |
| **Our approach** | **99.27%** | **92.23%** | **97.12%** |

**Table 3: Comparison of the image classification accuracy of the proposed approach with the baseline approaches.**

| Object Class | Mean | Standard Deviation |
|---|---|---|
| Apple (n07739125) | 5.477 | 0.065 |
| Car (n02958343) | 5.445 | 0.072 |
| Dog (n02084071) | 5.463 | 0.073 |
| Gold (n03445326) | 5.484 | 0.096 |
| Mobile (n02992529) | 5.511 | 0.068 |
| Rose (n12620196) | 5.470 | 0.088 |
| Scooter(n03791053) | 5.485 | 0.072 |
| Tiger (n02129604) | 5.502 | 0.035 |
| Wallet(n04548362) | 5.439 | 0.067 |
| Watch (n04555897) | 5.448 | 0.046 |
| All | 5.439 | 0.064 |

**Table 4: Mean and standard deviation of Inception scores for each class of objects dataset.**

by generative models. It is calculated using a Inception v3 network [31], which is pre-trained on the ImageNet database. To get an accurate score it should be applied on a large number of generated images, usually 50,000 samples [26]. We computed the Inception score values for each of the 10 classes individually as well for all the 10 classes combined. We obtained an average Inception score of 5.439 on the generated images. The class-wise results are presented in Table 4.

*4.3.4 Qualitative Results.* Apart from the quantitative results, we also present qualitative results of our model. The images generated using our proposed model for digits, characters, and objects are shown in Figure 4, Figure 5, and Figure 6 respectively. In Figure 4, the final column shows random sample images from MNIST dataset and the remaining columns show the images of digits generated by our model. Similarly in Figure 5 and Figure 6, we present the generated samples of characters and objects. The images generated by our model are not only visibly crisper but also diverse and this can be observed across all three datasets.

Our model does a better job in generating characters of some classes than others as shown in Figure 5. Character classes which are visibly distinct are generated better than the ones which look similar like classes H, M and F, P. As a result of this, the overall classification accuracy of the model on the character dataset is comparatively lower than the other two datasets as shown in Table 3.

## 4.4 Comparison

To further evaluate our approach, we compare our performance with the baselines. The first baseline is using AC-GAN [21] with EEG encoding as conditioning and the second baseline is again

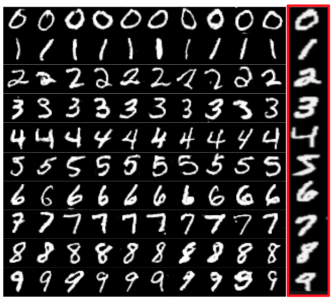

**Figure 4: Images of digits generated by our GAN model. Column 1-10 shows the generated images and the last column is a random sample of same class drawn from training data.**

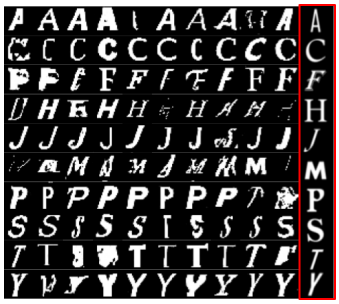

**Figure 5: Sample images of Characters generated by our GAN model. Column 1-10 shows the generated images and the last column is a random image of same class.**

| Method | Inception Score |
|---|---|
| AC-GAN [21] | 4.93 |
| AC-GAN [21] (1-hot) | 3.10 |
| Our Approach | **5.43** |

**Table 5: Comparison of Inception Score values (on all classes) between our approach and the two AC-GAN baselines.**

using AC-GAN [21] but with ground-truth labels (1-hot vectors) as conditioning. We chose the AC-GAN model for comparison as our proposed network architecture is comparable to the AC-GAN network architecture. We train individual networks for both these baselines using the same generator and discriminator models we used to train our GAN architecture. The pre-trained classifier is removed and the discriminator is modified to output both probability distribution over sources and probability distribution over class labels. The third baseline is the Brain2Image [14], where both the generator and discriminator are conditioned. We also trained their GAN model on our datasets for comparison.

We compare our approach with the baselines using the two evaluation metrics we proposed in Section 4.2. In Table 3 we present the comparison of the pre-trained classification accuracies. Our model outperforms all the baselines on all three datasets with a considerable margin. We observed that the Brain2Image model could not generate images of the classes for which the conditioning

**Figure 6: Sample images of 10 classes of objects generated by our GAN model. Column 1-9 are generated images, and the last column is a random image of the same class from training data.**

**Figure 7: Qualitative comparison of our approach with the baselines on the Objects dataset. The first row shows the images generated by our model. The second and third rows show images generated by the two baselines, AC-GAN [21] with EEG as conditioning and AC-GAN [21] with 1-hot conditioning respectively.**

was provided on all three datasets and on the object dataset it fails to generate any meaningful results. Hence, we do not include this baseline model in Inception score comparison.

We also compare the Inception score achieved by our model with the AC-GAN baseline models and the results are presented in Table 5. Qualitative comparison of images generated by our model with the baseline models is shown in Figure 7. We chose the Objects dataset for this qualitative comparison to show the capability of our model in generating complex images of real world objects.

To evaluate the performance of our approach on smaller datasets, we perform an experiment with the MNIST dataset with a reduced number of images per class. We use only 20% of the training data, i.e., 1000 images per class, for generator training in this experiment. We refer to this dataset as MNIST-1K. We also set the number of training epochs in these experiment to 50 to compare the pace of convergence. We use AC-GAN [21] and Brain2Image [14] approaches as baselines for these experiments. We use EEG encoding as conditioning for all three models.

In Figure 9, we present the comparison of classification accuracy of the generated images for the three models after every epoch. As can be seen from the plot, our model not only achieves the best classification accuracy but also converges very fast when compared to the other baselines. Our model achieved the highest classification accuracy after only 30 epochs on the reduced MNIST-1K dataset.

We also present the qualitative comparison of the images generated by the three models after 50 epochs of training in Figure 10. We can observe that not all images generated by the AC-GAN and Brain2Image models are legible where as the images generated by our model are sharp and clear. With the Brain2Image model we observed that the digits generated by the model do not correspond to the class of EEG conditioning resulting in a very low classification accuracy even though the qualitative results look better.

Finally, we perform an experiment to check for signs of memorization by our generator by walking along the latent space [24]. In this experiment, we pick two EEG conditioning vectors (for different classes of objects) and linearly interpolate between them to generate a series of conditioning vectors. We use a constant noise vector (z) and generate a series of images using the conditioning vectors as weights. We present the results in Figure 8, Figure 11 and Figure 12. We can observe that there is a smooth transition from

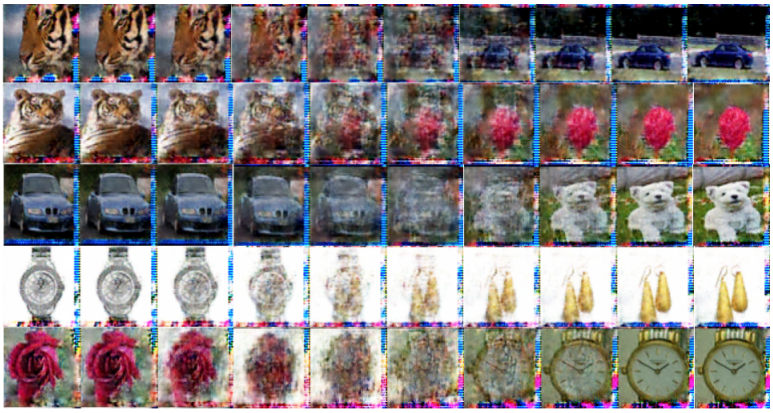

**Figure 8: Interpolation between EEG conditioning vectors of two different classes. The results show that the learned space has a smooth transitions. Here we set the noise vector (z) constant while interpolating between the conditioning vectors.**

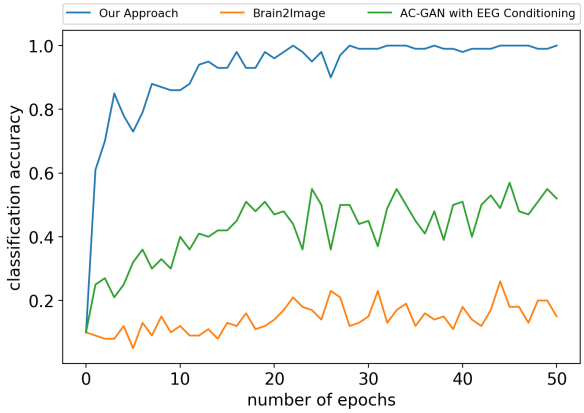


**Figure 9: Comparison of classification accuracies while training the generator on MNIST-1K dataset.**

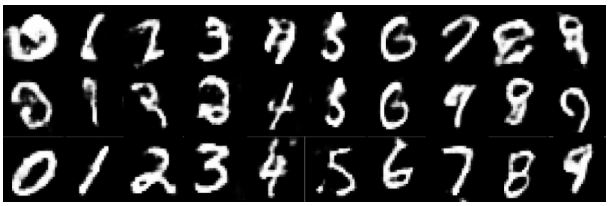

**Figure 10: Qualitative comparison of our approach with AC-GAN [21] and Brain2Image [14] on MNIST-1K images dataset. First and second row show the images generated by AC-GAN and Brain2Image and the final row shows images generated by our model.**

an image of one object to another suggesting that the network has learned relevant features required for image generation.

## 5 CONCLUSION AND FUTURE WORK

In this work we propose to visualize human thoughts using EEG signals recorded when they were thinking of an object. We propose

**Figure 11: Interpolation between EEG conditioning vectors on digits dataset. In all the rows you see a smooth transition indicating that the generator learned meaningful features required for image generation.**

**Figure 12: Interpolation between EEG conditioning vectors on characters dataset.**

a novel conditional GAN architecture to generate images of the objects that a person is thinking about using the EEG signals as conditioning. Our approach consists of two main steps, we first encode the EEG signals with a deep network and then the encoded EEG signal is used as conditioning to generate images of the objects. We performed our experiments on three different object categories and showed that the proposed GAN architecture can generate class-specific images using EEG signals as conditioning. We also showed that our proposed architecture is well suited for even small sized datasets. The results are a good indication that EEG signals contain informative information corresponding to human thoughts. We believe that the cues from EEG signals can be effectively encoded and used for a wide range of applications. In our future work we want to explore this idea further and visualize human thoughts in the form of a video stream which is more intuitive.

## REFERENCES


[1] U Rajendra Acharya, S Vinitha Sree, G Swapna, Roshan Joy Martis, and Jasjit S Suri. 2013. Automated EEG analysis of epilepsy: a review. *Knowledge-Based Systems* 45 (2013), 147–165.

[2] Pouya Bashivan, Irina Rish, Mohammed Yeasin, and Noel Codella. 2016. Learning representations from EEG with deep recurrent-convolutional neural networks. *ICLR* (2016).

[3] Abhijit Bhattacharyya, Manish Sharma, Ram Bilas Pachori, Pradip Sircar, and U Rajendra Acharya. 2018. A novel approach for automated detection of focal EEG signals using empirical wavelet transform. *Neural Computing and Applications* 29, 8 (2018), 47–57.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2172–2180.

[5] T. E. de Campos, B. R. Babu, and M. Varma. 2009. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 248–255.

[7] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. 2017. Adversarially learned inference. *ICLR* (2017).

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[9] Andrea M Green and John F Kalaska. 2011. Learning to move machines with the mind. *Trends in neurosciences* 34, 2 (2011), 61–75.

[10] Christoph Guger, Werner Harkam, Carin Hertnaes, and Gert Pfurtscheller. 1999. Prosthetic control by an EEG-based brain-computer interface (BCI). In *Proc. aaate 5th european conference for the advancement of assistive technology*. Citeseer, 3–6.

[11] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and V Babu Radhakrishnan. 2017. Deligan: Generative adversarial networks for diverse and limited data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1.

[12] Alexander G Huth, Wendy A de Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature* 532, 7600 (2016), 453–458.

[13] Alexander G Huth, Tyler Lee, Shinji Nishimoto, Natalia Y Bilenko, An T Vu, and Jack L Gallant. 2016. Decoding the semantic content of natural movies from human brain activity. *Frontiers in systems neuroscience* 10 (2016), 81.

[14] Isaak Kavasidis, Simone Palazzo, Concetto Spampinato, Daniela Giordano, and Mubarak Shah. 2017. Brain2Image: Converting Brain Signals into Images. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 1809–1817.

[15] Diederik P Kingma and Max Welling. 2014. Stochastic gradient VB and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*.

[16] Pradeep Kumar, Rajkumar Saini, Partha Pratim Roy, Pawan Kumar Sahu, and Debi Prosad Dogra. 2018. Envisioned speech recognition using EEG sensors. *Personal and Ubiquitous Computing* 22, 1 (2018), 185–199.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[18] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[19] Gernot R Muller-Putz and Gert Pfurtscheller. 2008. Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on Biomedical Engineering* 55, 1 (2008), 361–364.

[20] Shinji Nishimoto, An T Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, and Jack L Gallant. 2011. Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology* 21, 19 (2011), 1641–1646.

[21] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. *ICML* (2017).

[22] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *ICML* (2016).

[23] Simone Palazzo, Concetto Spampinato, Isaak Kavasidis, Daniela Giordano, and Mubarak Shah. 2017. Generative Adversarial Networks Conditioned by Brain Signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3410–3418.

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR* (2016).

[25] Siavash Sakhavi, Cuntai Guan, and Shuicheng Yan. 2015. Parallel convolutional-linear neural network for motor imagery classification. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2736–2740.

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*. 2234–2242.

[27] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. 2017. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human brain mapping* 38, 11 (2017), 5391–5420.

[28] Andrew B Schwartz, X Tracy Cui, Douglas J Weber, and Daniel W Moran. 2006. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* 52, 1 (2006), 205–220.

[29] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[30] Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano, Nasim Souly, and Mubarak Shah. 2017. Deep learning human mind for automated visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6809–6817.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

[32] Lasitha S Vidyaratne and Khan M Iftekharuddin. 2017. Real-Time Epileptic Seizure Detection Using EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25, 11 (2017), 2146–2156.

[33] Dong Wang, Doutian Ren, Kuo Li, Yiming Feng, Dan Ma, Xiangguo Yan, and Gang Wang. 2018. Epileptic Seizure Detection in Long-Term EEG Recordings by Using Wavelet-Based Directed Transfer Function. *IEEE Transactions on Biomedical Engineering* (2018).