HIGH SCHOOL EDITION
Journal of Student Research

A Novel Approach to Optimize Memory Reconstruction Using Joint Multimodal Networks

Jeremy Lu[1,2] and Cathy Messenger[2#]

[1]Saratoga High School
[2]Los Gatos High School
[#]Advisor

ABSTRACT

Memories play a crucial role in the human life experience. Whether they are used to make decisions or understand current situations, memories influence the past, present, and future. However, some memories are unreliable; in the court of law, the false memories of victims can cause inaccurate verdicts. To reconstruct human memories directly from the brain, recent studies have used electroencephalographic (EEG) brain signals in machine learning frameworks like generative adversarial networks (GANs), which generates new data from old ones. However, traditional GANs tend to produce the same images, an issue called mode collapse. Therefore, a conditional variational autoencoder-generative adversarial network (CVAE-GAN), which had not been used before in memory reconstruction, was developed to address GAN failures by jointly-training (1) an encoder and (2) GAN. CVAE-GAN correlated the MindBigData dataset's EEG brain signals with the corresponding ImageNet dataset images and produced new memories of participants, such as pandas, humans, and fish. CVAE-GAN's Inception Score, which marked how diverse and distinct its generated images were, was 1.00 on average. The number of floating operations, or FLOPS, was 102 Gigaflops, which was less than the traditional GAN's 198 Gigaflops. While limitations with time, computational memory, and mode collapse prevented the CVAE-GAN from recreating accurate memories, it still generated distinct image colors and general features. Future studies can build on existing network architectures and include more homogeneity in datasets. Ultimately, CVAE-GAN has the potential to advance new understandings of the brain and could elevate the memory reconstruction field.

Introduction

In order to combat the issue of false memories in daily life, the legal system, and making vital decisions in society, recent research has utilized machine learning to reconstruct memories directly from the brain to bypass the medium of true or false communication that must be stated when one conveys their thoughts. In addition to analyzing cognitive behaviors in the brain's visual cortices, human memory reconstruction has several applications in the court of law (Shen et. al 2019). Forensic tools and facial recognition techniques can apply the technique of memory reconstruction to determine an accurate verdict, identify human bodies, and recognize culprits' faces.

In order to generate new memories, machine learning models must be provided brain data and corresponding images. Projecting imagined objects in a visual form in the physical world required the decoding of functional Magnetic Resonance Imaging (fMRI) and EEG (Electroencephalographic) brain signals in many studies. Brain-Computer Interfaces (BCI) have commonly been used in the field of brain-machine learning tasks to provide this data, such as EEG and fMRI brain scans. While fMRI has been widely used due to spatially showing brain activity in different cortices of the brain, fMRI machines are cost-heavy and time-inefficient due to the long process of brain scanning (Shen et. al, 2019). Although EEG brain signals cannot show regional brain activity, they can showcase graphical brain activity involving frequencies of voltage activity in EEG electrode sensors; in addition, EEG brain signals are relatively non-invasive, cost-effective, and time-efficient. However, less studies have applied machine learning to

EEG signals for memory reconstruction tasks due to fMRI's capability to correlate regional brain activity with real-world images. Similarly, the main challenges of reconstructing images with EEG signals lie in the notion that EEG signals do not have immediate visual features that can be directly extracted or corresponded to image features; this, in turn, makes it unclear how to correlate these brain signals with visual images.

Previously, memory reconstruction tasks applicable to machine learning and EEG have generated coherent, detailed images (Kavasidis et al. 2017). Data fed into the machine learning model involved patients witnessing intervals of real-world images while their EEG brain signals were recorded (Fares et al. 2020). After being told that these two data types corresponded to each other, the machine learning frameworks were then assigned with the task of recreating new representations of these real-world images (Ye et al. 2022).

Recent studies have utilized frameworks, such as generative adversarial networks (GAN) and variational autoencoders (VAE) to reconstruct memories of dataset participants (Kavasidis et al., 2017). In these networks, feature extraction and image reconstruction processes are present. Because noise and unimportant frequencies are present in the EEG brain signal data, the VAE's encoder network extracts relevant features from EEG brain signals based on the corresponding image categories. Then, a decoder "upsamples" the EEG features such that it creates a new image or memory; upsampling occurs when the model adds numbers to its inputs to increase the size of the data. Loss functions, which minimize the loss, or difference between a model's desired and its actual output, takes into account the losses of both the encoder and decoder into its holistic calculation in the training process; therefore, this means that the training process allows these models to be trained together. On the other hand, the GAN needs a separate encoder to extract EEG brain signal information. The GAN is more comprehensive: in addition to having a decoder, or generator, model, the GAN framework has a discriminator network. The discriminator takes in the generator's recreated images as input and produces a probability of whether the images are from the original dataset or are, in fact, produced by the generator. The generator tries to fool the discriminator by creating images that resemble the original image dataset's patterns so closely that the discriminator guesses incorrectly. Therefore, the discriminator and generator compete with each other; if each network's parameters are set such that they are balanced, the GAN model is capable of producing images resembling the original dataset.

To evaluate how distinct and diverse these models' generated images were, Inception scores (IS) were used: a higher score means more distinctiveness and diversity, and a lower score means the opposite (Kavasidis et al., 2017). The IS were relatively high with 5.07 for GAN and 4.49 for VAE (Kavasidis et al., 2017). However, certain drawbacks made it difficult to optimize memory reconstruction: mode collapse, an issue where GANs constantly produce similar images, showed internal limitations with network architectures. Inability to most effectively correlate spatial images with the two-dimensional space of EEG brain signals resulted in images that generally resembled, for example, pandas but could not show specific details of an image. However, it could be argued that the exclusion of image details was due to the GAN producing an accurate representation of the image areas that participants, in reality, focus on. Still, failures in relation to generating diverse sets of images still held true due to the encoder's EEG feature extraction being separate from the GAN's image reconstruction process.

Therefore, the present work aimed to discover an optimal method for applying machine learning networks to EEG brain signals. Because previous implementations of CVAE-GAN have not been applied to multimodal data or the memory reconstruction task, a conditional variational autoencoder-generative adversarial network (CVAE-GAN) was developed to solve these issues. Because the VAE and GAN's trained jointly–a process similar to the implementation of loss functions in previous VAE models, the GAN acquired an accurate sense of how each EEG latent feature vector matched with the latent space of an image (Bao et al., 2017). The CVAE-GAN framework consisted of four jointly connected networks: an encoder, generator, discriminator, and classifier. Similar to previous studies' models, the encoder extracted relevant frequency features from EEG signals, and the generator-discriminator system worked in a similar way to traditional GAN frameworks. Since an image category value, or condition, was fed into the encoder and generator, the overall model had a clearer sense of how to reconstruct images based on the EEG feature extraction process. Thus, this allowed the model to produce images respective to their categories, addressing the issue of images blending together due to the ineffective separation of their corresponding categories. In addition, the classifier model, which is extended from the generator, predicted an image category to gauge if the network was holistically factoring in accurate image categories into its image recreation process (Bao et al., 2017).

With this in mind, memory reconstruction with improved luminance distributions in images was accomplished through the implementation of the CVAE-GAN's jointly-connected networks. Aiming to address ethical dilemmas involved with machine learning-memory reconstruction, brain signal inputs were only provided by the user's permission, and images reconstructed would be privatized towards the user in future software.

Methods

The proposed model combined the concept of a multimodal network with the prospects of CVAE-GAN. The CVAE-GAN's encoder network extracted latent features from the EEG data. The extracted features were then sent as inputs to the generator network. For each image in the training process, its respective conditional label was sent into the encoder and generator as a secondary input apart from the EEG signal samples.

The main machine learning library, which provided the necessary algorithm and architectures to train machine learning models, was Keras with a Tensorflow backend; the program was executed by Compute Unified Device Architecture (CUDA®) and CUDA® Deep Neural Network library (cuDNN), which optimized the graphics processing unit, or GPU, usage in the machine learning runtime.

Data: MindBigData's "IMAGENET of the Brain", which used an EEG headset to collect brain signal data, was the public dataset utilized in the present study to conduct machine learning techniques on; the headset itself was built on the standard international 10-20 electrode system. Patients in the data collection process paid close attention to images from the ImageNet database and had their EEG brain signals recorded with 5-channel electrodes. These electrodes were labeled AF4, T7, T8, AF3, and Pz, denoting channels on different locations on the head, which correlated to spatial brain regions.

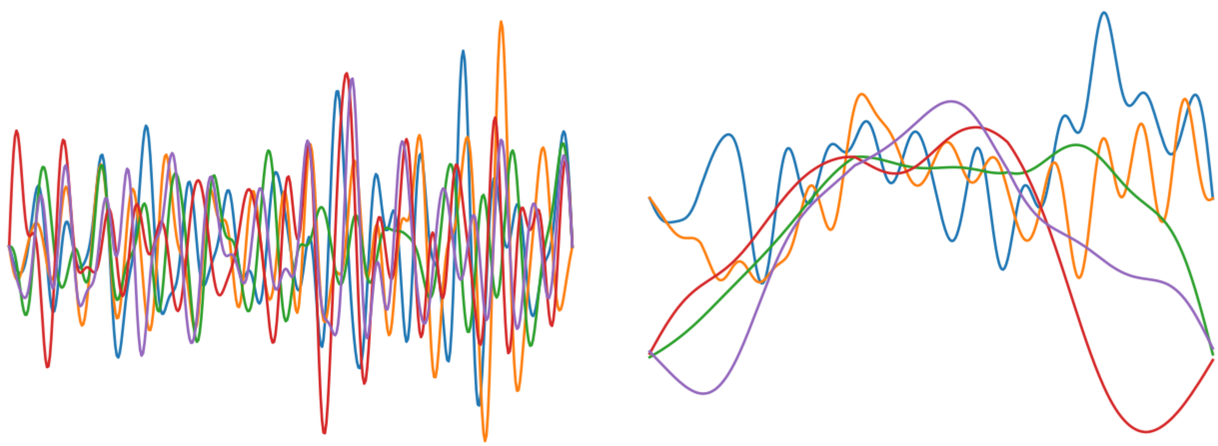

Figure 1. Alpha (left) and delta (right) signals. Samples (Hz) are on the x-axis; peaks (microvolts) are on the y-axis.

Using MNE, a Python library utilized for brain-signal processing, alpha signals were extracted with lower and higher frequencies of 0.4 and 0.5, while delta signals were filtered with lower and higher frequencies of 4 and 8. Delta signals are mostly related to "deep sleep" brain states, which may correspond to deep thinking or memory; alpha signals are mostly associated with relaxed and awake brain states, which can provide the conditions for the user to think about their own memories.

Many libraries were used for assisting the machine learning process. Matplotlib plotted the EEG data and model's produced images; OpenCV, which provided image-processing functions, scaled the images to 128 by 128 pixels; in order to provide floating numbers to the machine learning model in a format it could understand, Numpy was utilized to manipulate data in network architectures and training. MNE analyzed EEG and MEG brain signals, while providing algorithms, such as filters, to efficiently pass through brain data into deep learning models.

After preprocessing the brain signal, image, and string data using MNE and other various libraries described, the encoder network received the EEG data and corresponding conditional labels. It then produced a latent vector that contained only the relevant features of the EEG data.

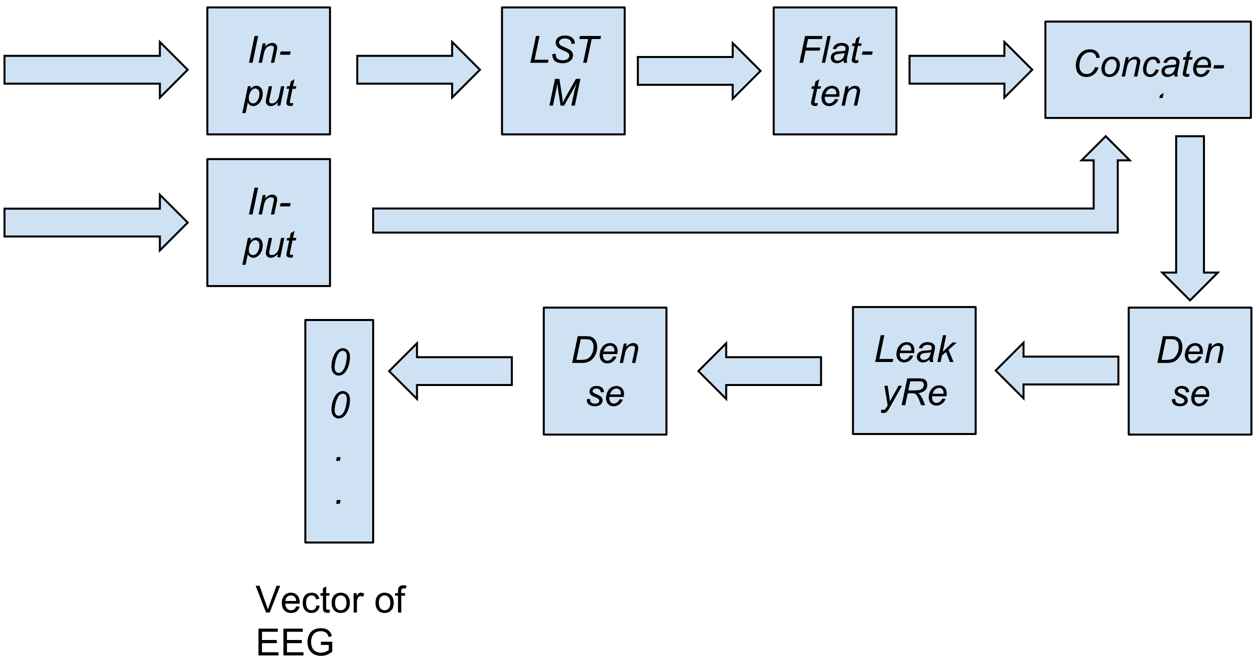

Figure 2. Encoder network flowchart. The EEG signals were fed into a set of LSTM and Dense layers that used kernels to extract data values from the EEG data. It produced an array with 569 values corresponding to predicted image categories for each EEG feature.

<u>Feedforward process</u>: The developed encoder network was a Long-Short Term Memory (LSTM) network consisting of 128 units and one dense layer, which was used to traverse temporally through these EEG sequences and extract features based off of the provided image category that matches with the EEG input. The purpose behind why LSTM was used for the encoder specifically was that the LSTM layers had forget, input, and output gates; these gates served to (1) memorize long-term EEG sequence patterns, (2) filter new EEG data inputs to forget unimportant information relative to the layer's long-term pattern, (3) and update the long-term memory based on relevant patterns. The LSTM layers could be thought of as neural networks themselves, conducting operations within its cells and filtering information by multiplying the inputs by 0 to 1, where the most irrelevant data was multiplied by 0 and the retained data by 1.

Once the EEG feature extraction was completed, the newly encoded array produced by the encoder acted as an image input to the GAN network. The GAN's assigned task was to improve that image such that it resembled the original image data, such as an airplane or pumpkin.

In addition, the conditional label input being sent into the encoder allowed the encoder to extract EEG signals corresponding to their images under one specific category. For example, a sample dataset would include only zebras and human faces. In order to generate images relevant only to human faces, the encoder would independently learn EEG features specific to the visualization of human faces, not zebras. This conditional label was also sent through the GAN to generate images based on their respective categories.

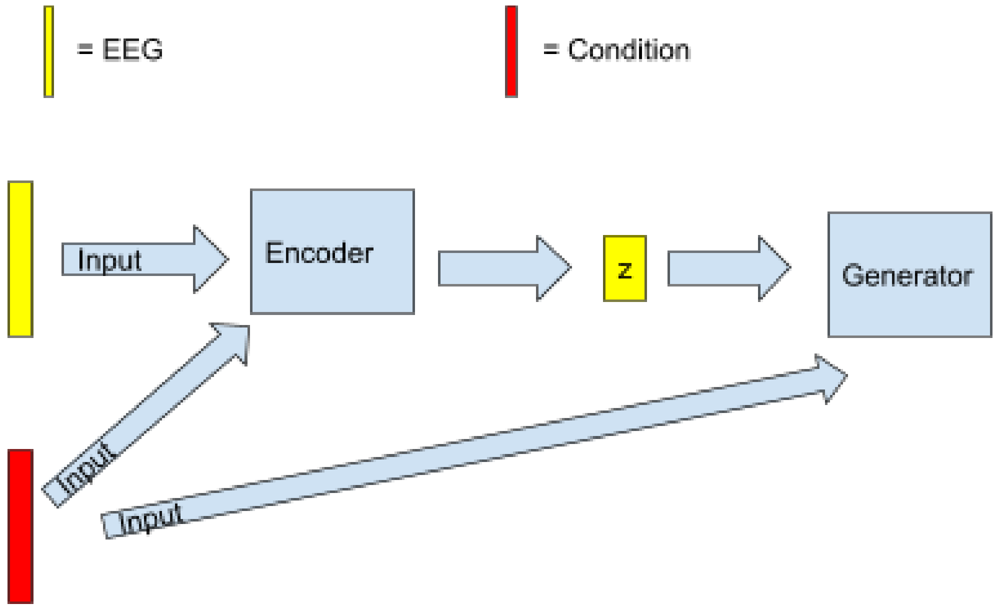

Figure 3. Overall encoder to generator process. EEG signals were fed into the encoder, where output features z were fed into the generator along with conditional integers indicating image categories.

After its feedforward process, the encoder's produced EEG features, which were represented as latent vectors, served as inputs to the GAN's generator network. The generator's purpose was to (1) produce images based on the EEG brain signals and conditional labels it was given and (2) adjust these "fake" images–via the generator's own weights and biases–to accurately depict the original image features.

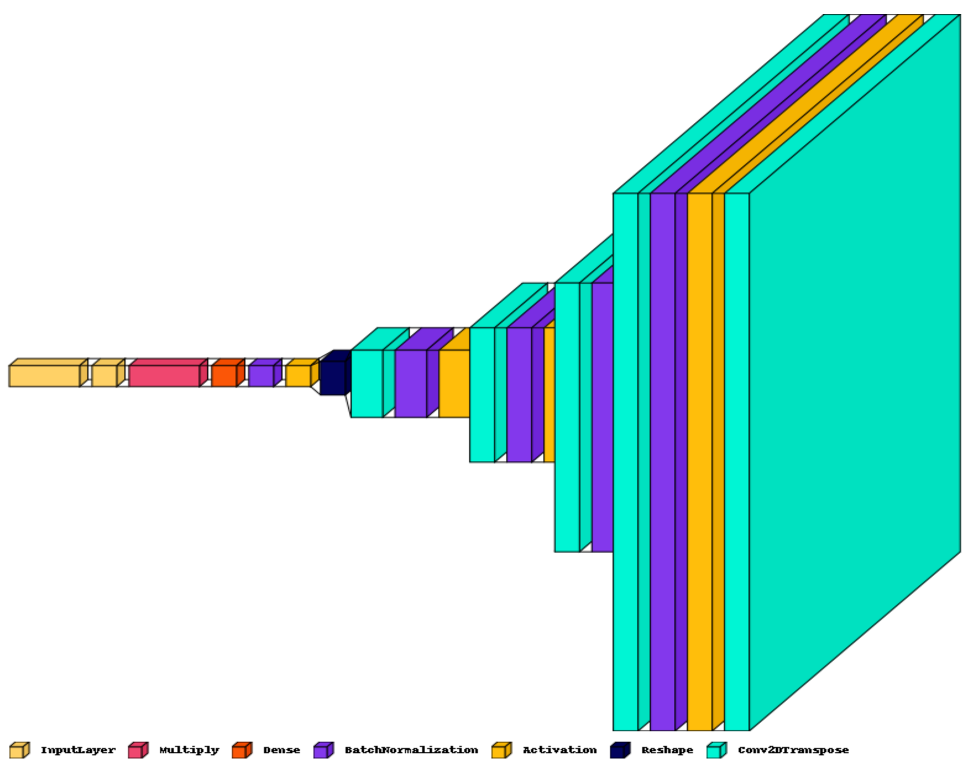

Figure 4. Generator architecture. Input EEG features from the encoder (left-most layer) were upsampled into an image (right-most layer).

The generator model received the EEG input features of shape 384, meaning 384 Hz of samples. The model architecture included an initial input layer that multiplied both the conditional label and EEG feature inputs together into one matrix. This matrix was fed into a new dense layer with 128 units; following this layer was batch normalization, which was a layer that normalized its inputs such that training was more stable, and rectified linear unit (ReLU) activation; activation functions were used in the neural network to activate and deactivate certain nodes for a final prediction. To increase the dimension size of the features while retaining necessary information, convolutional transpose layers–filters being 256, 128, 64, and 32 and kernel sizes being 3x3, 3x3, 5x5, and 3x3 respectively–were used to turn these EEG features into images. The final convolutional transpose layer reshaped the input matrix into an RGB (Red-Green-Blue), or colored, image, and then the hyperbolic tangent activation resided in the output layer. The hyperbolic tangent function was used to normalize the images from integers -1 to 1. Because the original dataset images served as ground truths for the CVAE-GAN's training, these images, along with the generator's produced ones, served as inputs to the discriminator.

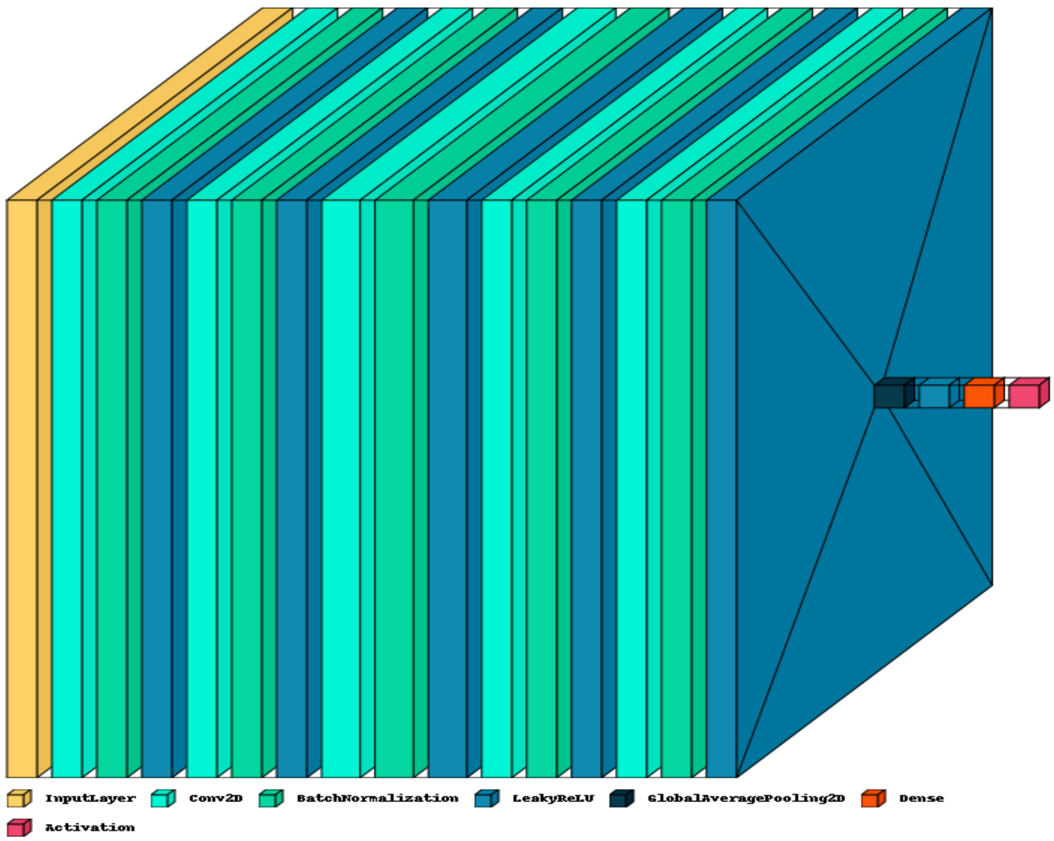

Figure 5. Discriminator architecture. Through a series of convolutional layers, the network produced a probability of an image being produced by the generator.

In order to build the discriminator architecture, four convolutional blocks, in total, were implemented. Following the input layer, which provided a generated and real image to the network, a downsampling block with a series of layers was integrated into the network. Each block had a two-dimensional convolution layer, batch normalization layer, and a leaky ReLU activation layer with an alpha value at 0.3; these blocks served the purpose of extracting image features from the data that were most relevant to predicting their "realness". After being set into the four downsampling blocks, the data was then set into global average pooling, leaky ReLU, dense, and sigmoid activation layers in order to dimensionally reduce the data into a single probability value.

In machine learning, differences between what the model thinks is correct and what is actually correct can occur, which is why parameters called weights and biases are tweaked to find the right spot where the model accurately predicted the true outcome. In the present study, optimizers, such as Adam, served to reduce these differences, while making sure that the training model did not overfit–or train too closely on the image data.

The generator's produced images were also sent to the classifier model as inputs. Since the model was trained end-to-end, the classifier's output affected the generator's. More specifically, the intermediate feature matching layer of the classification network was minimized within the generator's loss function.

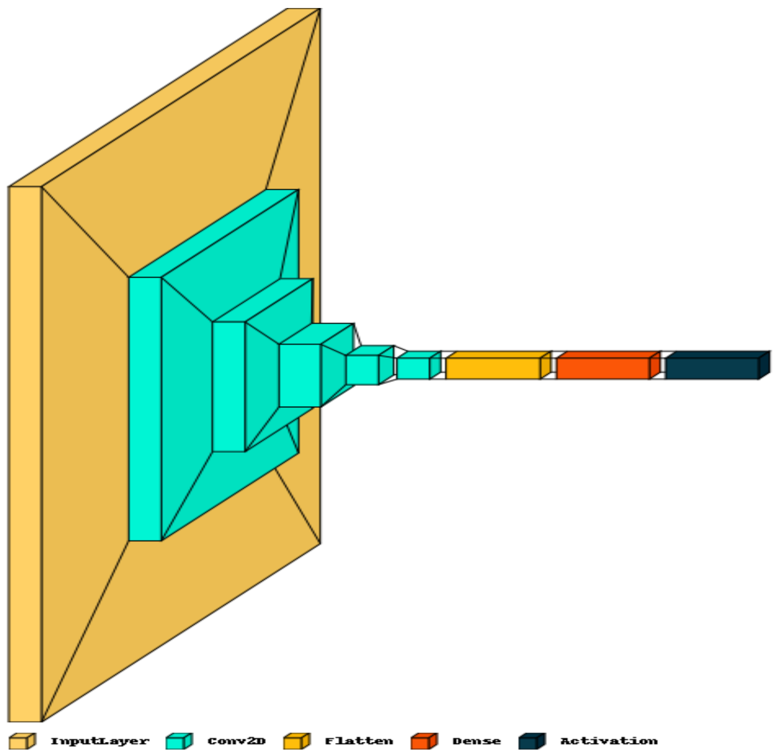

Figure 6. Encoder architecture. The encoder architecture consisted of four convolutional blocks but produced a categorical probability for each image input.

The classifier's architecture consisted of basic convolutional layers with 64, 128, 256, 128, 64 filters with kernel sizes of 3x3, a dense layer, and a final sigmoid activation function; the sigmoid activation function ensures that the output is an integer between 0 and 1, which indicates the probability of an image belonging to a specific image category. The loss function specific to the classifier was softmax, which was not involved in the other CVAE-GAN components. Finally, the output of the classification network was an array of probabilities for each image category denoted by the chance of an image falling into them. The encoder's architecture was similar to the discriminator; however, the labels provided to the classifier were image categories, not real or fake labels fed to the discriminator.

Results

Because qualitative observations could not show whether results indicated progress or simple feedforward GAN generation of images, the use of qualitative metrics were crucial to analyzing the performance of the CVAE-GAN. Inception Score (IS) was used to determine the overall distinctness and diversity in color, features, and brightness of the images. Using the InceptionV3 model from the Keras machine learning library, the pre-trained model was run on the generated CVAE-GAN images, outputting categorical probabilities as to how likely an image belonged to a category

 Volume 12 Issue 3 (2023)

of, for example, cats, dogs, and lions. If the image is distinct, the probability distribution of the categories for that image will not be uniform. Once the individual probability distributions are added up across all images, if all of the images are uniquely distinct, the result should be an approximately uniform distribution.

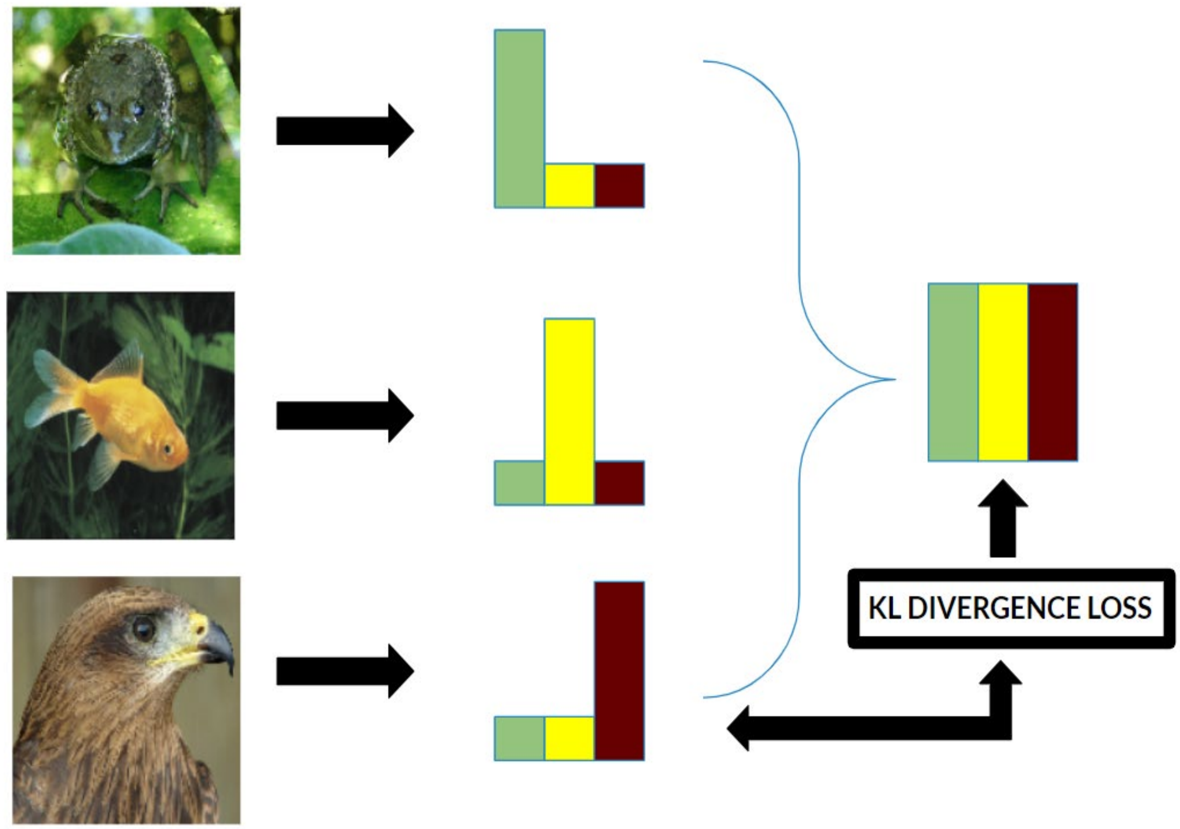

Figure 7. Inception Score calculation. If more distinctness is present in the generated image set, the KL divergence loss will calculate a higher IS score.

By using a Kullback-Leibler (KL) Divergence formula, the calculation was capable of modeling the difference between the individual probability distributions and the overall marginal distribution, contributing to the overall IS score.

$$D_{KL}(P \,||\, Q) = \sum_{x \in X} P(x)\, log(\frac{P(x)}{Q(x)})$$

The KL Divergence equation takes the logarithmic difference of both the individual image's categorical probability distributions and marginal distributions, resulting in the expected value for mean difference. Matrix row operations were conducted in the program to achieve these probability distributions and calculations.

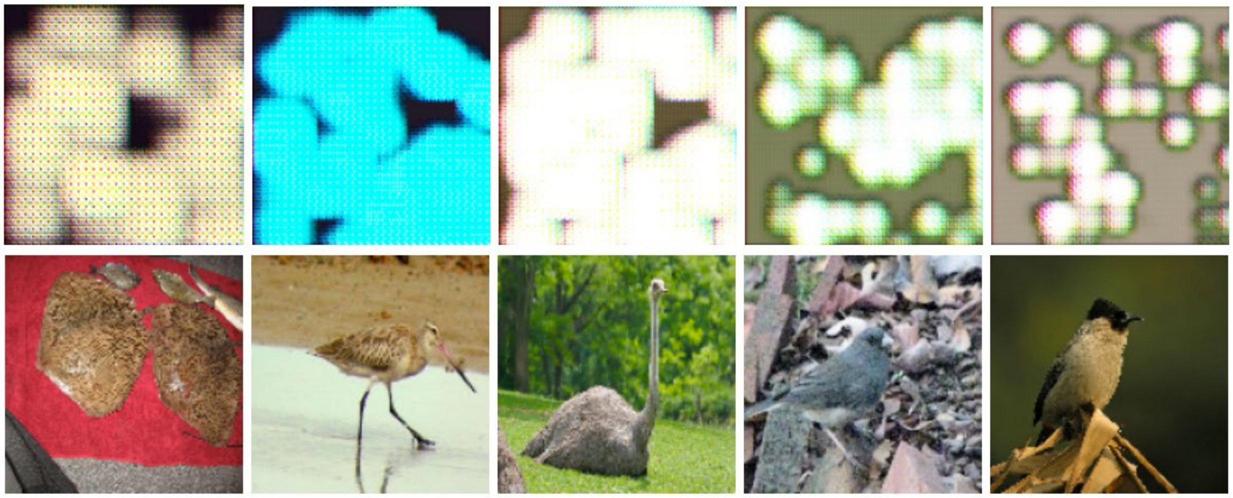

Figure 8. 0 to 100 training iterations: colors present in the generated images (top) were generally matched up with their real image counterparts (bottom), but qualitatively distinct features were not present in the produced images.

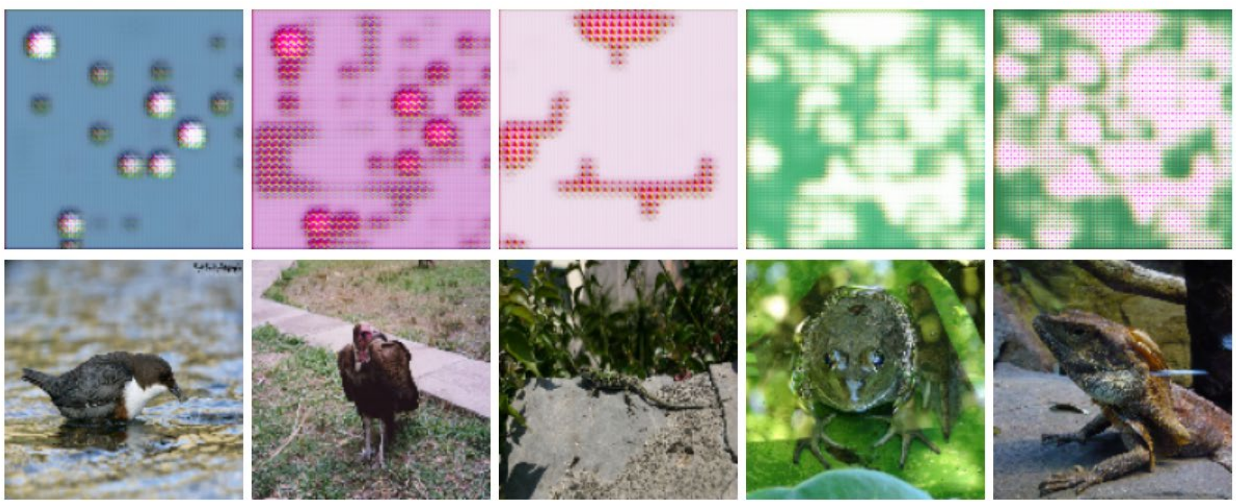

Figure 9. 100 to 200 training iterations: the generated images improved slightly, and certain parts of the images that showed color difference indicated the CVAE-GAN's ROI.

Images shown are samples from the 1047 images generated by the CVAE-GAN. Through observation, image colors changed drastically between one batch of iterations and proceeding ones; this indicated (1) the lack of similarity between the original images being fed in, (2) a medium-high learning rate, and (3) the model's inadequacies in taking into account coarseness, contrast, line-likeness, and roughness components of an image. After testing different values for learning rates, higher values resulted in a drastic change between training batches of generated images, which was to be expected. A learning rate less than the ideal value caused too little adjustment in the generated samples due to undercorrection of the GAN's weights. Whether or not running the training loop for a longer period of time would improve the model performance is unknown; however, due to memory limitations, the program stops short at around 250 iterations.

Table 1. The floating operations (FLOPS) with units of GIGAFLOPS (10^9) were used to standardize computational system performance.

	Traditional GAN	CVAE-GAN	Brain2Image GAN (Kavasidis et al., 2017)	Brain2Image VAE (Kavasidis et al., 2017)
IS Average	8.55	1.00	5.07	4.49
IS Standard Dev.	1.54	1.82 x 10^{-4}	-	-
GIGAFLOPS	198	102	-	-

The IS average and standard deviation were both low–1.00 and 1.82 x 10^{-4} respectively–for the CVAE-GAN, compared to the traditional GAN's 8.55, Brain2Image GAN's 5.07, and Brain2Image VAE's 4.49 average scores. This matches with a qualitative observation of the generated images, where general colors match with the original dataset images but distinct features do not. Changing the architectures of the discriminator and generator or implementing customized loss functions to take into account multiple model losses are viable options to combat this issue. Otherwise, a major issue in the dataset itself is that the lack of images per category makes it difficult to train the GAN with consistency. 569 categories are far too many to be expecting the GAN to create coherent results; an attempt could be made to change the dataset entirely, or condense the images into one category and combining all relevant images into that category, which means that all human faces would go under one category and the model would train on this.

On the other hand, the floating operations (FLOPS) were quite low on the model due to the significantly lowered amount of model parameters and layers within the networks' architectures. However, there was a tradeoff between model performance and computational performance. Whether or not the model would have optimized memory reconstruction while maintaining its computational efficiency was also not yet known due to time constraints.

Conclusion

In the present work, the CVAE-GAN model was used as an optimized alternative to previous machine learning networks, such as GAN, VAE, and conditional GAN, to optimize images in the task of memory reconstruction. The model consisted of encoder, generator, discriminator, and classifier networks–all of which were trained jointly, or all together. The process included the encoder breaking down EEG signals into features; the generator reconstructing images with EEG latent features; the discriminator competing with the generator to improve the GAN's performance; and the classifier predicting image categories, in order to provide the generator an idea of whether it reconstructed images based on their categories well enough or not. Specialized preprocessing and network architectures were implemented in the program with machine learning libraries as the backbone of the CVAE-GAN's development. By virtue of the IS and FLOPS metrics, which were used to determine the image accuracy and computational efficiency of the model, the CVAE-GAN was not able to produce images with distinct features that resembled real-world images; however, it was able to recreate the general colors of the original dataset images. Failure to properly implement image features in the CVAE-GAN suggests not a correlation shortcoming between EEG signals and images, but an issue with the lack of homogeneity in the MindBigData's image dataset and implementation of internal loss functions.

Limitations

A lack of homogeneity, or an abundance of image categories for far fewer images, played a role in the CVAE-GAN's overcorrection in the training process. For example, batches of images were diverse, and the CVAE-GAN generated images with different colors after each training iteration.

In addition, because the Random Access Memory (RAM), which refers to the short-term memory in the Central Processing Unit (CPU) that is read and reset every runtime, was limited to 16 Gigabytes, the amount of training iterations for the CVAE-GAN was limited to 250. If the images were rescaled to 64 by 64 pixels or if less EEG samples were taken, less data would be present while more training iterations would occur. For this reason, previously stated tradeoffs between model and computational performance must be combated; ideally, the model should have been able to run training iterations over the entire dataset.

Another contribution to the CVAE-GAN's inability to produce images resembling the original dataset was the exclusion of customized loss functions (Bao et al. 2017). In order for the amount of error for each of the four networks to be taken into account for the CVAE-GAN's training process, customized loss functions, such as style loss and KL Divergence loss, could have been used to optimize every aspect of the CVAE-GAN. For example, the classifier's predicted image category outputs must have an effect on how the generator decides to produce its images based on the original categories. The encoder's extraction of EEG signal features must influence how the generator produces images from these EEG features and how the classifier predicts corresponding image categories. The lack of interconnectedness led to the networks being unable to systematically train.

Future Studies

The use of several categories are practical in the real-world, but, for pure experimentation of memory reconstruction, implementation of only one or a few categories may constitute the usage of a traditional GAN or VAE, as opposed to a multimodal CVAE-GAN network. Because batches of images across consecutive training iterations had different image categories, future work can instead focus on one image category and record corresponding EEG signals to optimize the model in a more consistent manner.

In order to more effectively correlate EEG features with generated images, the EEG feature vectors themselves can be added into the discriminator network's layers (Kavasidis et al. 2017). In previous studies, the inclusion of EEG features in networks has shown to produce coherent images, but, due to the already complex nature of the CVAE-GAN, this aspect was omitted. Future work may consider reinstating this method into the CVAE-GAN.

On another note, the preprocessing methods could also be altered. In the future, separate networks may run on raw EEG signals to detect frequencies related to memories; these modified models may involve adding more gates to the LSTM network, implementing custom layers specifically meant for EEG processing, or replacing CVAE-GAN itself with other reconstruction models (Fares et al. 2020).

Once these implementations have been made, machine learning models in relation to memory reconstruction could be trained on videos corresponding with EEG signals, which may lead to further advances in smoother transitions in generative models. Furthermore, applications in memory reconstruction technology include not just in forensics techniques to accurately assist court cases, but also in humanistic tasks: for instance, recovering past memories can be the first step to assistance in trauma. As continual development in the field of machine learning shows promise for successful memory reconstruction, future endeavors correlating the brain to the physical world may impact human lives for years to come.

References

1. Bao, J., Chen, D., Wen, F., Li, H., & Hua, G. (n.d.). CVAE-GAN: Fine-Grained image generation through asymmetric training. arXiv:1703.10155.
2. Du, C., Du, C., & He, H. (n.d.). Sharing deep generative representation for perceived image reconstruction from human brain activity. 2017 International Joint Conference on Neural Networks. https://doi.org/10.1109/IJCNN.2017.7965968
3. Fares, A., Zhong, S.-H., & Jiang, J. (n.d.). Brain-media: A dual conditioned and lateralization supported GAN (DCLS-GAN) towards visualization of image-evoked brain activities. ACM DL Digital Library. https://doi.org/10.1145/3394171.3413858
4. Kavasidis, I., Palazzo, S., Spampinato, C., Giordano, D., & Shah, M. (n.d.). Brain2Image. MM '17: ACM Multimedia Conference, 1809-1817. https://doi.org/10.1145/3123266.3127907
5. Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (n.d.). Autoencoding beyond pixels using a learned similarity metric. arXiv:1512.09300. https://doi.org/10.48550/arXiv.1512.09300
6. Miyawaki, Y., Uchida, H., Yamashita, O., Sato, M.-A., Morito, Y., Tanabe, H. C., Sadato, N., & Kamitani, Y. (2008). Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. Neuron, 60(5), 915-929. https://doi.org/10.1016/j.neuron.2008.11.004
7. Multi-task generative adversarial learning on geometrical shape reconstruction from EEG brain signals. (2019). Iconip 2019. https://doi.org/10.48550/arXiv.1907.13351
8. Nemrodov, D., Niemeier, M., Patel, A., & Nestor, A. (2018). The neural dynamics of facial identity processing: Insights from eeg-based pattern analysis and image reconstruction. Eneuro, 5(1), ENEURO.0358-17.2018. https://doi.org/10.1523/ENEURO.0358-17.2018
9. Rashkov, G., Bobe, A., Fastovets, D., & Komarova, M. (2019). Natural image reconstruction from brain waves: A novel visual BCI system with native feedback. bioRxiv. https://doi.org/10.1101/787101
10. Shen, G., Dwivedi, K., Majima, K., Horikawa, T., & Kamitani, Y. (2019). End-to-End deep image reconstruction from human brain activity. Frontiers in Computational Neuroscience, 13. https://doi.org/10.3389/Fncom.2019.00021
11. Shimizu, H., & Srinivasan, R. (2022). Improving classification and reconstruction of imagined images from EEG signals. BioRxiv. https://doi.org/10.1101/2022.06.01.494379
12. Wakita, S., Orima, T., & Motoyoshi, I. (2021). Photorealistic reconstruction of visual texture from EEG signals. Frontiers in Computational Neuroscience, 15. https://doi.org/10.3389/fncom.2021.754587
13. Ye, Z., Yao, L., Zhang, Y., & Gustin, S. (n.d.). See what you see: Self-supervised cross-modal retrieval of visual stimuli from brain activity. arXiv:2208.03666. https://doi.org/10.48550/arXiv.2208.03666
14. Zubarev, I., Vranou, G., & Parkkonen, L. (2022). MNEflow: Neural networks for eeg/meg decoding and interpretation. SoftwareX, 17, 100951. https://doi.org/10.1016/j.softx.2021.100951