Zero-Shot Text-to-Image Generation for Housing Floor Plans

Shannon Phu
shannon.phu@sjsu.edu
San Jose State University

Shiv Kumar Ganesh
shivkumar.ganesh@sjsu.edu
San Jose State University

Kumuda BG Murthy
kumuda.benakanahalliguruprasadamurt@sjsu.edu
San Jose State University

Raghava D Urs
raghavadevaraje.urs@sjsu.edu
San Jose State University

1. ABSTRACT

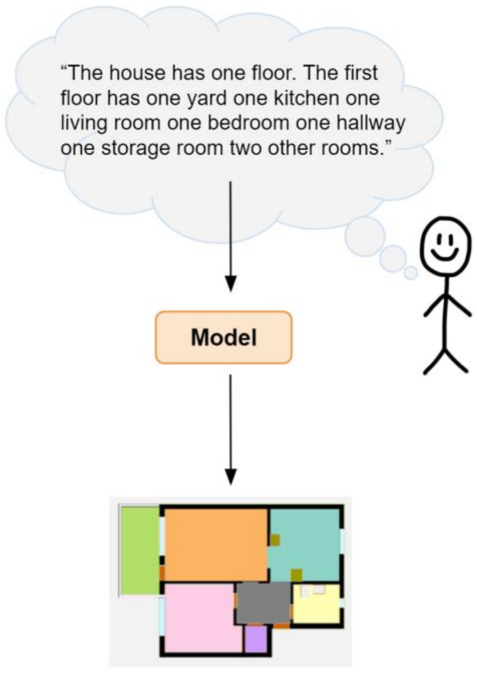
Floor plans are sketches drawn to scale that depict the relationship between rooms, spaces, and physical features as seen from above. They allow you to envision how people would move around the room. Before going further through more elaborate planning or construction phases, floor plans make it easier to check whether the room is appropriate for its intended purpose, work through any possible problems, and redesign. Interior designers, pro builders, and real estate agents can use these floor plans when they are looking to design or sell a new home or property. In this project we propose an approach for the text-to-image generative model on a housing floor plan dataset that automatically generates pictures from natural language sentences. We train a ***variational auto-encoder (VAE)*** to learn how to represent an image as an embedding and a ***DALL-E*** transformer architecture to learn the relationship between text and image pairs in order to generate new floor plan images from natural text.
Keywords: variational autoencoder, DALL-E, generative, computer vision, transformers, wandb, loss, streamlit, housing, floor plans

2. INTRODUCTION

According to marketing industry influencer Krista Neher, the human brain can process images up to 60,000 times faster than words. The point is with a picture, you can convey so much more information than you can with words. Equivalently floor plans make it easier than text description to see if a room is suitable for its intended use. Any future obstacles, redesigning, and accessibility objectives may all benefit from a floor plan. This can be extremely beneficial during the design and construction periods. A text-to-picture system is a technique that automatically converts a natural language text into pictures representing the meaning of that text. The pictures can be static illustrations such as images or dynamic illustrations such as animations. Architects, builders, and even real estate agents can use our model to create floor plans by simply entering some text detailing the floor plan in order to assist in the design and presentation of floor plan concepts to their clients.

In this paper, we will be leveraging an image dataset called CubiCasa5K, a large-scale floor plan dataset and we propose a deep learning methodology using VAE and DALL-E to convert text to image to achieve a better understanding and modelling of housing interiors.

In short, our goal is to accept natural text which describes a house's indoor layout as input and generate a house floorplan image representing that textual description of the house.


"The house has one floor. The first floor has one yard one kitchen one living room one bedroom one hallway one storage room two other rooms."
Model

Example input text and expected generated image

3. THE DATA

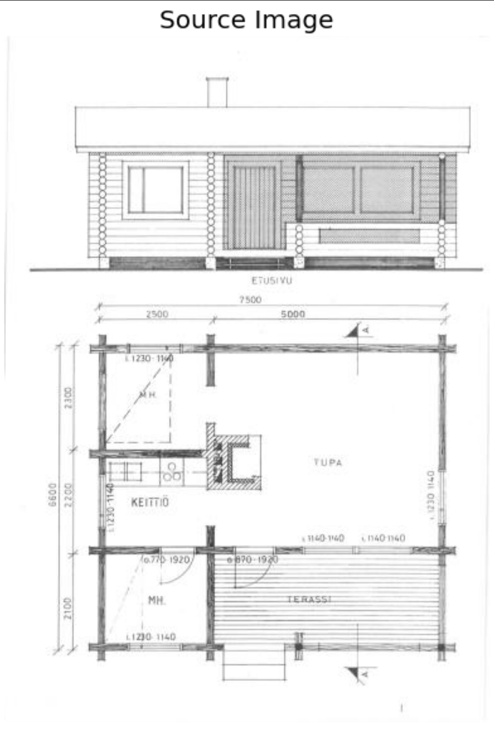
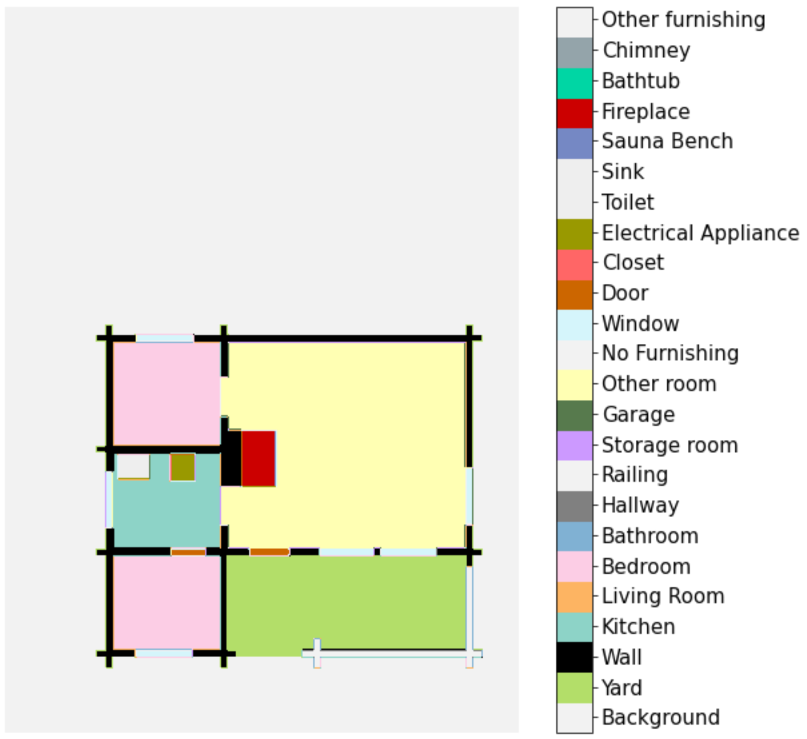
The image dataset, Cubicasa5K [10] contains 5000 samples of house floor plans annotated with over 80 floor plan object categories and dozens of room categories. The data provided consisted of images in Scalable Vector Graphics (SVG) format. We loaded SVGs for thousands of house floor plans which contain information about room placement and size. We converted those SVGs to JPEG images where certain rooms were colored a certain color depending on their room type, such as bedrooms, kitchens, bathrooms, etc. Certain built-in furniture items are also colored in, such as stoves, toilets, sinks, etc.

3.1. DATA PREPROCESSING

The CubiCasa had images for one to four story houses. We filtered our dataset down to one floor houses for our modelling in order to limit the scope of our problem. The images in SVG format were preprocessed to get a colored annotated image. Because the original dataset did not include textual descriptions of the houses, we generated the associated text description representing the house using the data encoded within the SVG.

Below are the steps followed to convert SVG to image:

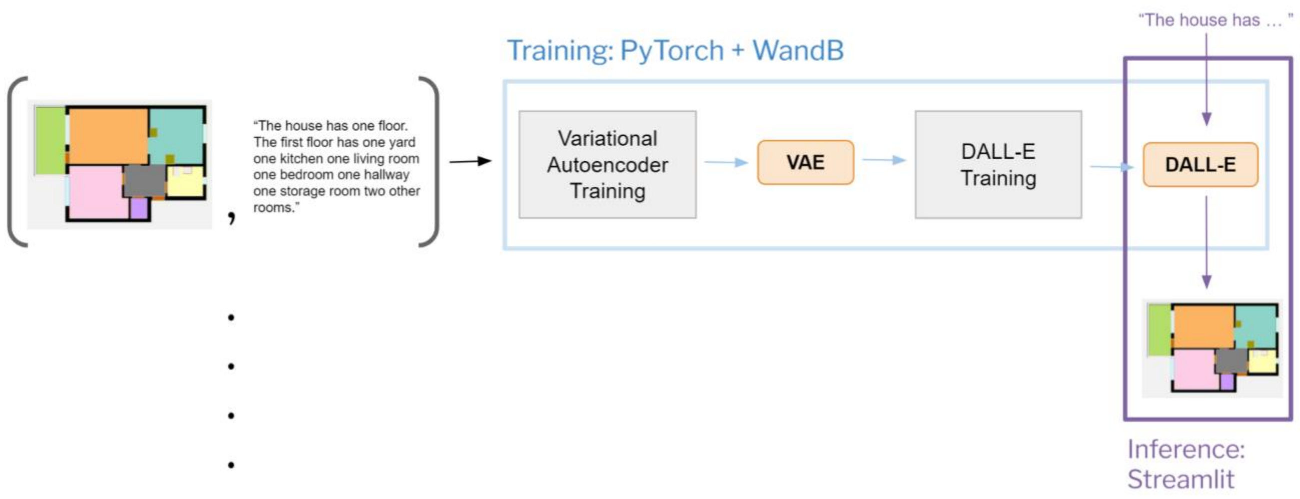
1. Parse floor plan SVG for different floors and room regions
2. Create text description for each house's floor plan based on the number floors and room regions present
3. Integrate the above two steps to generate data from floorplan SVGs
4. Save the generated image processed from SVG

Original SVG image	**Generated annotated image with text**
 Source Image ETUSIVU 7500 2500 5000 TUPA KEITTIÖ MH. TERASSI 6600 2300 2200 2100	 Other furnishing Chimney Bathtub Fireplace Sauna Bench Sink Toilet Electrical Appliance Closet Door Window No Furnishing Other room Garage Storage room Railing Hallway Bathroom Bedroom Living Room Kitchen Wall Yard Background "The house has one floor. The first floor has one yard one kitchen two bedrooms two other rooms."

4. EXPERIMENT

Converting natural language text descriptions into images is an amazing demonstration of deep learning. Our focus was to train a VAE as well as DALL-E.

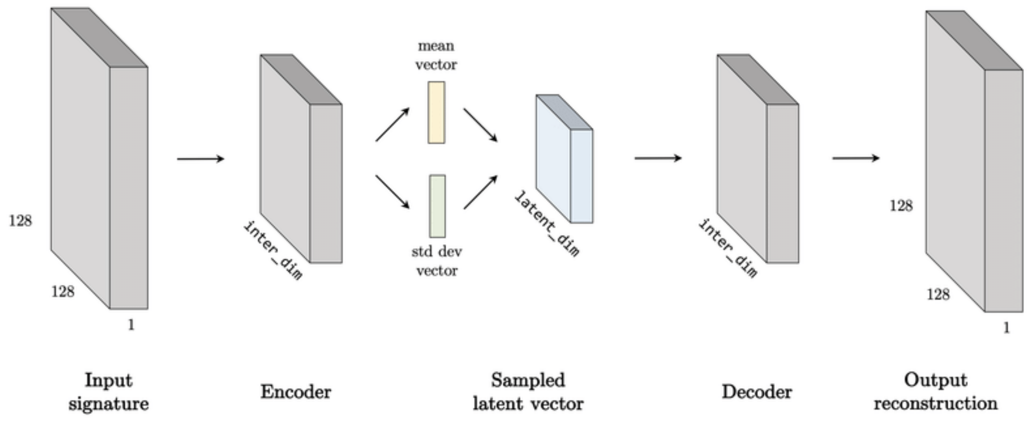
4.1. Architecture


"The house has one floor. The first floor has one yard one kitchen one living room one bedroom one hallway one storage room two other rooms."
Training: PyTorch + WandB
Variational Autoencoder Training
VAE
DALL-E Training
DALL-E
"The house has ... "
Inference: Streamlit

Model architecture for both training and inference pipelines

We designed a training pipeline which accepts pairs of images and text descriptions to first train a discrete VAE then DALL-E transformer. The resulting trained model is then used in our inference pipeline. We deployed our final model using Streamlit and hosted a website for real time inferences. Our stack used PyTorch and leveraged dalle-pytorch [4] to train both the VAE and DALL-E.

VAE: A variational autoencoder is an architecture composed of both an encoder and a decoder and that is trained to minimize the reconstruction error between the encoded-decoded data and the initial data.



System diagram of vae architecture used[9]

However, to introduce some regularization of the latent space, we proceed to a slight modification of the encoding-decoding process: instead of encoding an input as a single point, we encode it as a distribution over the latent space. The model is then trained as follows:

1. First, the input is encoded as distribution over the latent space
2. Second, a point from the latent space is sampled from that distribution
3. Third, the sampled point is decoded, and the reconstruction error can be computed
4. Finally, the reconstruction error is back propagated through the network

We in particular trained a discrete VAE and used reconstruction loss plus KL divergence loss as our objective to minimize.

DALL-E: DALL-E is an extension of GPT-3 by creating images from text descriptions for a wide range of concepts expressible in natural language.

DALL-E is an autoregressive transformer decoder model. It accepts concatenated encoded text tokens with the encoded image embeddings to train the transformer auto-regressively. The training minimizes cross-entropy loss for a combination of the text and image representations.

Weights and Biases (wandb), the developer stack for machine learning practice is used for the entire life cycle of the project. It is leveraged for experiment tracking, hyper parameter optimization, dataset model versioning and sharing results. Adding W&B's lightweight integration to the existing ML code will quickly get us live metrics, terminal logs, and system stats streamed to the centralized dashboard. It can be used to explain how our model works, show graphs of how model versions improved, discuss bugs, and demonstrate progress towards milestones.

STREAMLIT: Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

4.2. MODELLING

Preparation of image-text pairs**:** The images with different extensions like (.png, .jpg, .jpeg, .bmp) are converted to RGB format, resized to 128 pixels, and converted to tensors. The text descriptions in .txt files were tokenized. Then the tokenized text description and image tensor pairs are used to create our *image-text* dataset.

Train test split: The dataset was split into a training and validation set with a split of 0.9/0.1.

VAE training and validation: VAE training is performed using Adam optimizer and ReLU activation function. Steps performed are as follows:

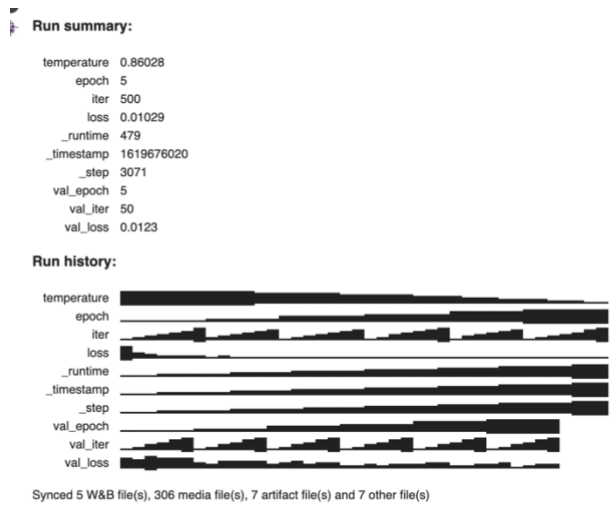
1. Setup VAE parameters like hidden layers, hidden dimension, number of layers and number of tokens for training
2. Initialize wandb experiment run WANDB_OWNER, WANDB_PROJ_NAME, vae parameters and jobtype
3. Initialize DiscreteVAE with VAE parameters
4. Setup Adam optimizer with suitable learning rate
5. Define training loop with loss computation on image reconstruction over mini-batches
6. Perform training using training set, adjust the network parameters from losses obtained during each iteration
7. Log sample images and metrics and save model checkpoints in wandb
8. Calculate the average loss combining the collected loss from multiple iteration
9. Validation of the model using a validation set

DALL-E Training and validation:

1. Load the saved VAE model and get the parameters
2. Set up the DALL-E parameters like text sequence length, number of text tokens, model dimension
3. Initialize DALL-E with the VAE model and DALL-E parameters as arguments
4. Initialize WandB job with WANDB_OWNER, WANDB_PROJ_NAME, DALL-E parameters
5. Train the image-text pairs in mini-batches over multiple epochs
6. Adjust the weights on computing the loss
7. Log samples and metrics in wandb for every 30 iterations.
8. Perform validation using image-text pairs in validation dataset
9. Save model as WandB artifact at end of every epoch
10. Save final trained DALL-E model
11. Deploy model to Streamlit

5. EXPERIMENT RESULTS

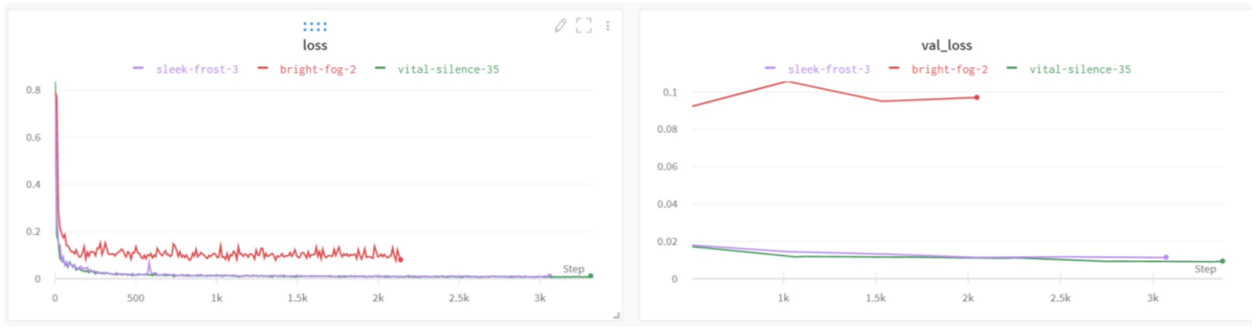
VAE RESULTS



```
Run summary:

 temperature 0.86028
       epoch 5
        iter 500
        loss 0.01029
    _runtime 479
  _timestamp 1619676020
       _step 3071
   val_epoch 5
    val_iter 50
    val_loss 0.0123

Run history:

 temperature
       epoch
        iter
        loss
    _runtime
  _timestamp
       _step
   val_epoch
    val_iter
    val_loss

Synced 5 W&B file(s), 306 media file(s), 7 artifact file(s) and 7 other file(s)
```

MLOps pipeline logging from VAE model training

We experimented with different numbers of epochs and VAE embedding dimensions. We were limited by the amount of computing resources we had available but found that training with more epochs and a higher number of embedding dimensions led to better model performance.

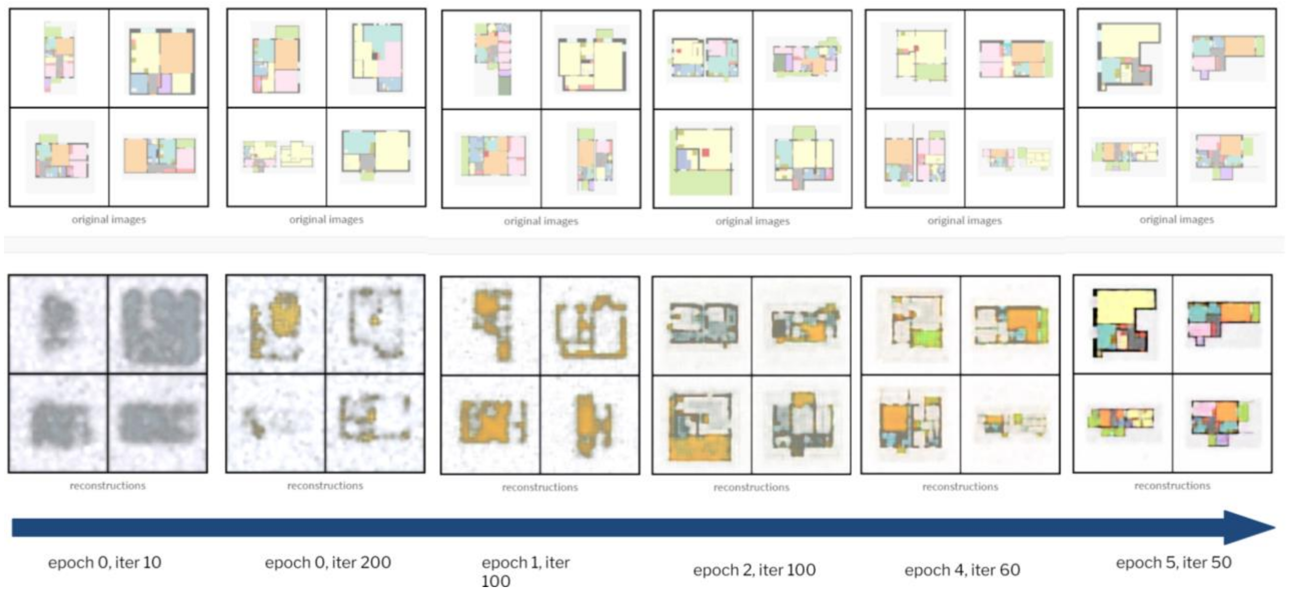


The training and validation loss from different iterations

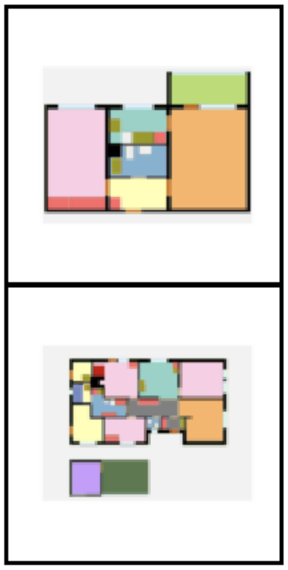
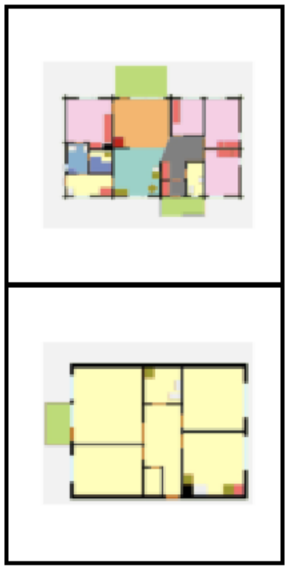
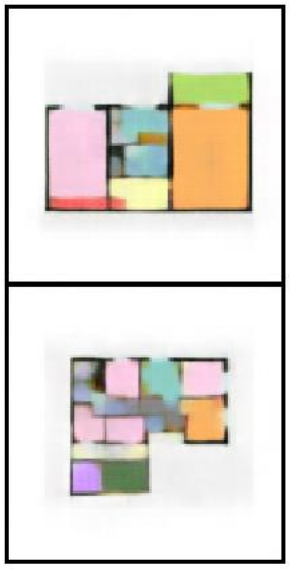
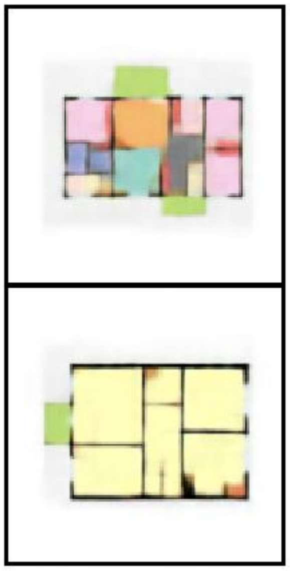
Experiments	Training Loss	Validation Loss
6 epochs, 512 dim	0.012	0.009
5 epochs, 512 dim	0.010	0.012
4 epochs, 256 dim	0.081	0.107

Experiment comparison for VAE training

Another interesting observation is the gradual improvement of image reconstruction quality as we trained longer.

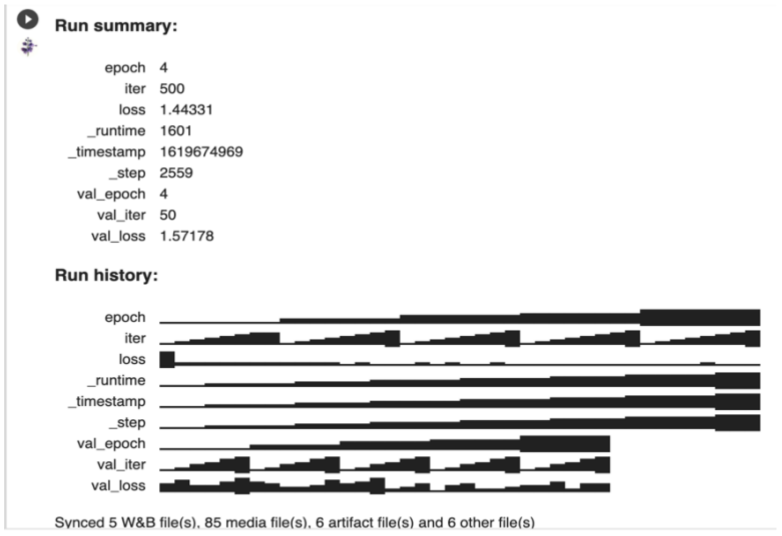


Progress of VAE over training lifecycle

			
Original annotated images fed as input		*The reconstructed images from vae model*	

Examples of VAE reconstructions

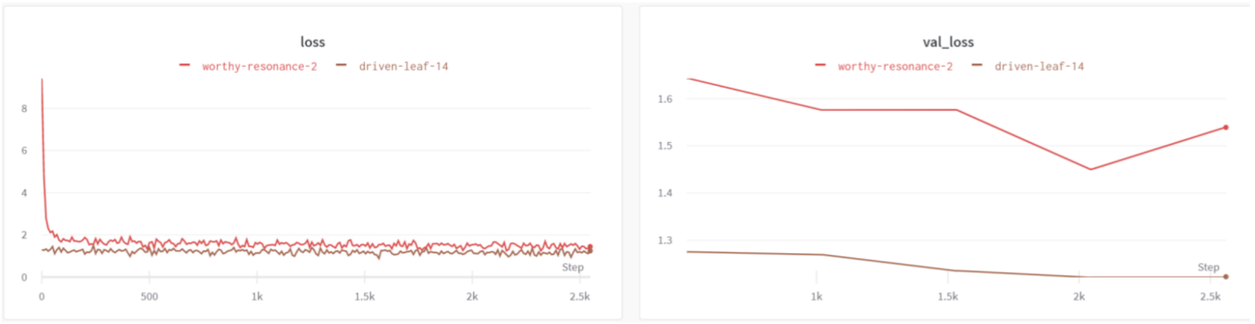
DALL-E RESULTS



```
Run summary:

      epoch  4
       iter  500
       loss  1.44331
   _runtime  1601
 _timestamp  1619674969
      _step  2559
  val_epoch  4
   val_iter  50
   val_loss  1.57178

Run history:

      epoch
       iter
       loss
   _runtime
 _timestamp
      _step
  val_epoch
   val_iter
   val_loss

Synced 5 W&B file(s), 85 media file(s), 6 artifact file(s) and 6 other file(s)
```

MLOps pipeline logging from DALL-E model training

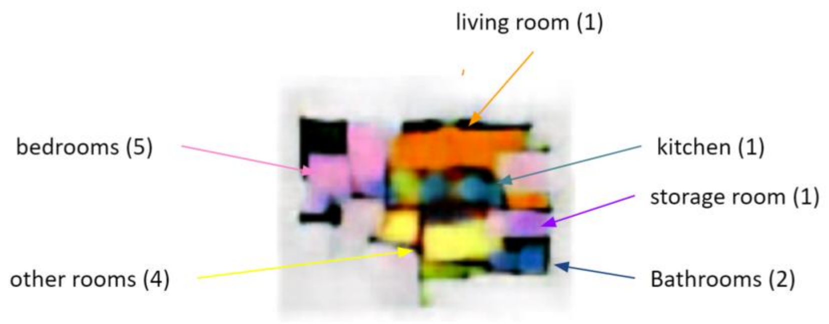
We experimented with different numbers of epochs and embedding dimensions provided from the VAE trained in the previous tep. We were limited by the amount of computing resources we had available but found that training with more epochs and a higher number of embedding dimensions led to better model performance.

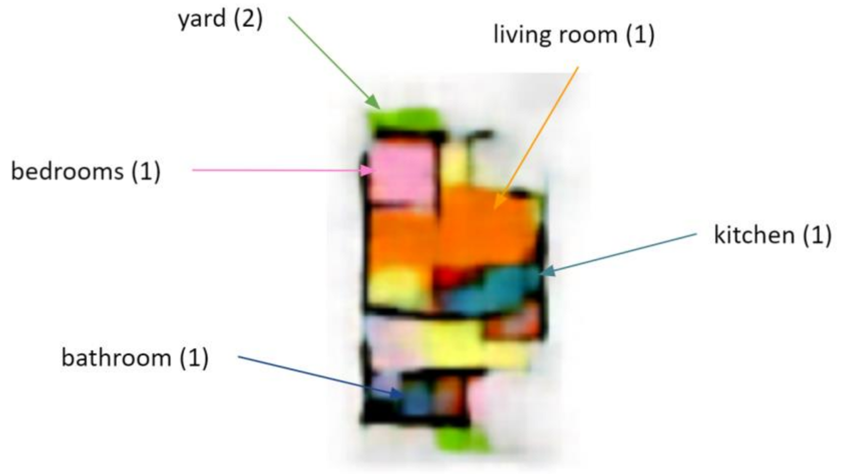
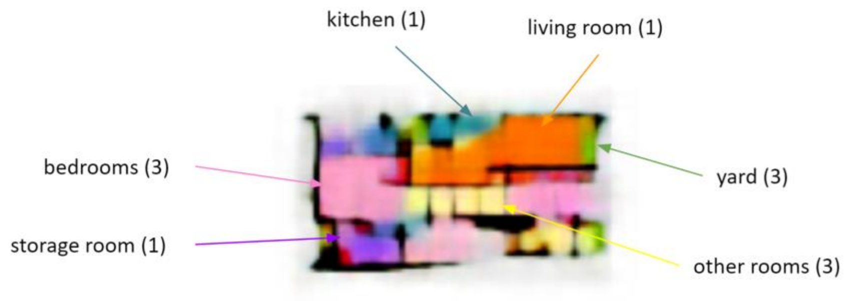


Training and validation loss at different steps for different epochs

Experiments	Training Loss	Validation Loss
5 epochs, 512 dim	1.244	1.252
4 epochs, 256 dim	1.443	1.572

Experiment comparison for DALL-E training

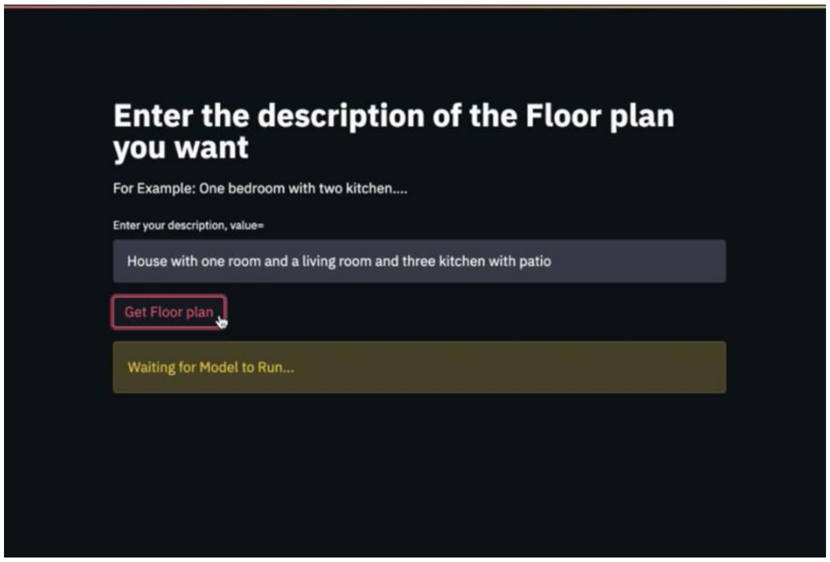
Example 1	 living room (1) bedrooms (5) kitchen (1) storage room (1) other rooms (4) Bathrooms (2) the house has one floor. the first floor has one yard one kitchen one living room five bedrooms two bathrooms three hallways one storage room four other rooms.

Example 2	 yard (2) living room (1) bedrooms (1) kitchen (1) bathroom (1) the house has one floor . the first floor has one yard one kitchen one living room one bedroom one bathroom .
Example 3	 kitchen (1) living room (1) bedrooms (3) yard (3) storage room (1) other rooms (3) the house has one floor . the first floor has three yards one kitchen one living room three bedrooms two bathrooms one hallway one storage room three other rooms .

The image generated for the given text description from DALL-E

Inference from Dall-e model in Streamlit

The inference can be viewed on providing text description of floor plan like below and get the annotated floor plan image as result



Enter the description of the Floor plan you want

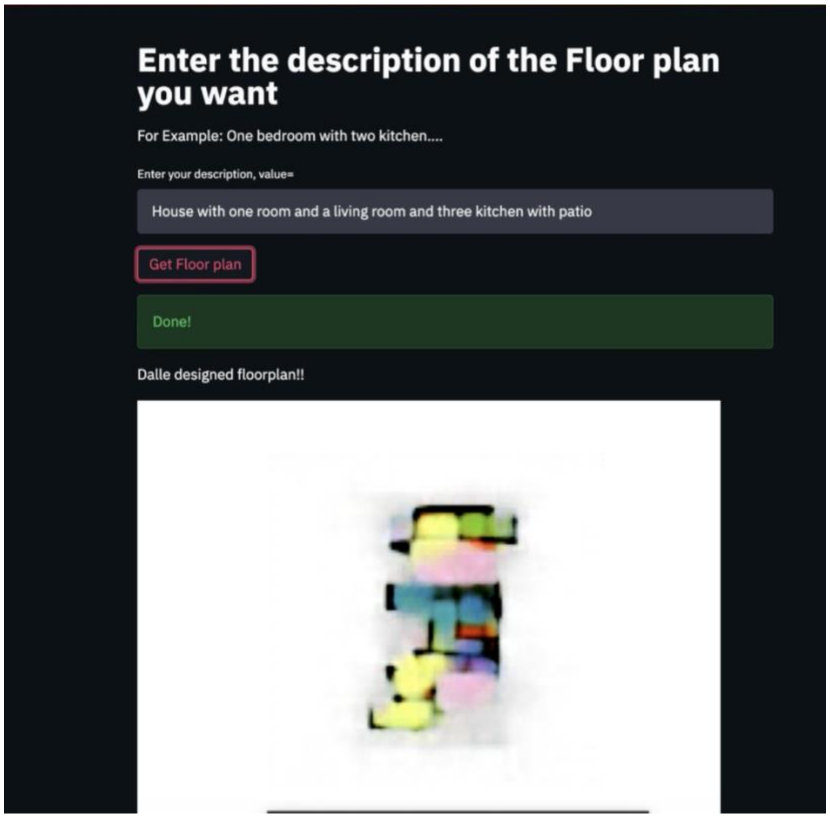
For Example: One bedroom with two kitchen....

Enter your description, value=

House with one room and a living room and three kitchen with patio

Waiting for Model to Run...

On entering text description in streamlit UI



Enter the description of the Floor plan you want

For Example: One bedroom with two kitchen....

Enter your description, value=

House with one room and a living room and three kitchen with patio

Done!

Dalle designed floorplan!!

Dall-e generates floor plan image

6. CONCLUSION

We were able to generate images from text description input using variational autoencoders and DALL-E. The VAE model is used as a pre-trained image embedding layer for the DALL-E model construction. DALL-E used the image text pairs as input data and VAE model achieves the conversion of floor plan description into floor plan image. This approach can be extended to generate different image types like birds, animals and so on by changing the input dataset.

7. ACKNOWLEDGEMENT

The authors wish to thank Professor Vijay Eranti for the continued guidance and support. This project was part of an academic project submitted to the Department of Software engineering, San Jose State University.

8. REFERENCES

[1] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
[2] https://towardsdatascience.com/dall-e-explained-in-under-5-minutes-327aea4813dd
[3] https://openai.com/blog/dall-e/
[4] https://github.com/lucidrains/DALLE-pytorch
[5] https://wandb.ai/home
[6] https://arxiv.org/pdf/1906.02691.pdf
[7] https://towardsdatascience.com/openais-dall-e-and-clip-101-a-brief-introduction-3a4367280d4e
[8] https://daleonai.com/dalle-5-mins
[9] https://www.researchgate.net/figure/System-diagram-of-the-VAE-architecture-used_fig1_332751044
[10] https://zenodo.org/record/2613548