ControlNet Project

Swetha Murali - IMT2022018
Soham Pawar - IMT2022127
Dyuthi Vivek - IMT2022523

Demo Application:
https://colab.research.google.com/drive/1q07Nz1__7Wcor6tPZXMAh28o3ZYetzY2?usp=sharing
GitHub: https://github.com/DyuthiVivek/GenAI-project-2

We tried 2 tasks - thermal to RGB conversion and webpage generation from a rough sketch.

Thermal to RGB

Problem Definition

Thermal imaging cameras capture infrared radiation emitted by objects, providing valuable information about temperature distributions but lacking color and fine visual details.

Task: Given a thermal image as input conditioning, generate an RGB image that preserves the spatial structure while adding realistic colors, textures, and details

Applications:

- Transform thermal camera feeds into realistic RGB-like displays for autonomous driving
- Convert 24/7 thermal surveillance footage into interpretable RGB-style video for security personnel

Related Work

- T2V-DDPM**:** Uses DDPM to translate thermal facial images into visible RGB images. This is applied to faces rather than driving scenes.
- ThermalDiffusion: This translates visual to thermal for autonomous driving. We are trying to achieve the reverse.
- IR2VI: GAN-based unsupervised thermal-to-visible framework
- LadleNet**:** A two-stage U-Net architecture that performs thermal to visible image translation guided by semantic segmentation,

In contrast to prior work, we address thermal-to-visible translation for autonomous driving environments using a ControlNet-conditioned diffusion model

Dataset Creation

https://www.kaggle.com/datasets/deepnewbie/flir-thermal-images-dataset - the dataset contains pairs of thermal and RGB images of outdoor road scenes with trees, vehicles, people, etc.

Pre-processing:

- Certain images in the dataset are extremely bright. An image is not chosen if it is very bright on average (high mean pixel value) and has low contrast (low standard deviation), which indicates a washed-out image frame.
- After a few iterations of ControlNet training, we increased the contrast of the RGB images in the dataset using the ImageEnhance library as the model was generating some washed-out images.
- The dataset also consisted of annotated images with labels of bounding boxes drawn around objects. The annotations were extracted and the counts of each of the objects were used to construct a simple text conditioning prompt for each image. For example: an outdoor scene with trees, two people and many bicycles.

Control Signal

The primary structural feature we aim to preserve is the spatial layout of the driving scene, including road geometry, object boundaries (vehicles, pedestrians, buildings), and relative positioning of scene elements. Preserving this structure is critical for autonomous driving scenarios.

We tried training with 2 different control signals:

1. 8-bit thermal images from the FLIR dataset were used directly as the conditioning input. Each thermal image is stored as a 3-channel grayscale RGB image, where all three channels contain identical thermal intensity values. The thermal intensity encodes temperature distributions where hotter objects (vehicles, people) appear brighter and cooler regions (sky) appear darker.

2. With the previous control signal, in some cases, the structure of the image was not being preserved. Raw thermal provides intensity information but lacks these explicit structural cues, while edge maps provide clear boundaries but lose temperature context. We know that the ControlNet paper shows edges (canny) consistently outperform raw images. So we tried to combine these two into the following 3 channel RGB used as conditioning:

   Red Channel: CLAHE-enhanced thermal intensity for local texture and temperature patterns. CLAHE implementation provided by opencv is used.
   Green Channel: HED-detected edges for explicit structural boundaries. HED is applied using a pretrained edge detection network from the controlnet_aux library.
   Blue Channel: Blended combination of thermal intensity and edges

Training

Model used: stable-diffusion-v1-5

Best hyperparameters:
```
 --resolution=384 \
 --mixed_precision=fp16 \
 --learning_rate=3e-5 \
 --lr_scheduler=constant_with_warmup \
 --lr_warmup_steps=500 \
 --max_train_steps=2500 \
 --train_batch_size=4 \
 --gradient_accumulation_steps=4 \
 --gradient_checkpointing \
 --use_8bit_adam \
 --set_grads_to_none \
 --enable_xformers_memory_efficient_attention \
 --checkpointing_steps=1000 \
 --validation_steps=100 \
 --seed=42 \
```

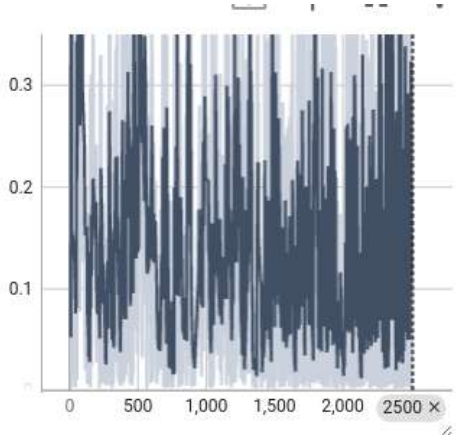
We observed overfitting and loss oscillating after ~2500 steps. We did the following memory optimizations: mixed precision training (fp16), enabled gradient checkpointing, used adam 8 bit optimizer, gradient accumulation to simulate a larger batch, efficient attention and set grads to none. We did validation after every 100 steps and checkpointing after every 1000 steps.

Experiments and Ablations

Attempt 1:

(different hyperparameters than those mentioned above)
Learning rate: 1e-5
Resolution: 512
Batch size: 1
Training steps: 2500
No text prompts

The generated images preserve overall road geometry and scene layout from thermal inputs, indicating that the ControlNet successfully learns coarse structural alignment. However, some regions of the generated images appear blurry/corrupted. Addition of extra objects such as vehicles, traffic cones, etc. is observed and the generated images do not closely match the original.



We observe the loss is oscillating more after 1500.

Attempt 2:

Hyperparameters were modified. Text prompts were not added yet.

- Learning rate was increased from 1e-5 to 3e-5.
- Resolution was decreased from 512 to 384.
- Batch size was increased from 1 to 4.

We ran this for 1500 steps to compare with the previous outputs.

There is higher similarity to the source image and less hallucination, which indicates that the new set of hyperparameters is better.

Attempt 3:

Prompts were added by randomly sampling from:

```
[
    "outdoor street scene with trees and road",
    "road with trees and vegetation",
    "street view with trees and parked cars",
    "outdoor road scene with trees and vehicles"
]
```

The number of training images was increased from 1000 to 1250. Hyperparameters were kept the same as attempt 2.

Prompt: outdoor street scene with trees and road

Prompt: road with trees and vegetation

The images appear even more realistic and similar to the conditioning image. However, the prompts do not always match the images and this causes issues with generation as well. For example, the randomly sampled prompt for the second image is “road with trees and vegetation”. The prompt does not mention cars, hence none of the generated images contain the car in the conditioning image nor the electric poles and cables visible in the scene. The images also appear washed out, probably due to the quality of images in the training dataset.

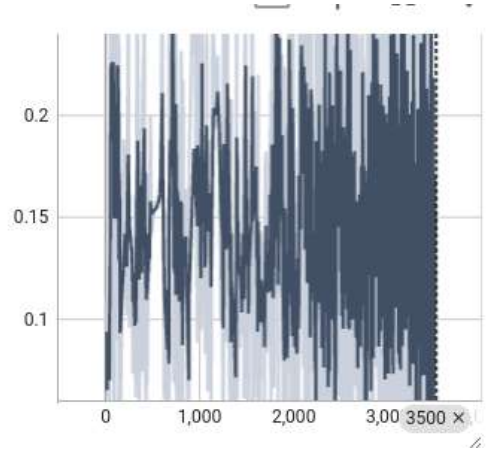
Attempt 4:

Prompts were obtained using the bounding boxes available in the dataset as described in the preprocessing section.

Prompt: outdoor road scene with trees

Prompt: outdoor road scene with trees and several cars

This addresses the issue of missing objects in attempt 3, but the images still appear washed out.



The loss started oscillating after 2000 steps.
This model achieved an SSIM of 0.62

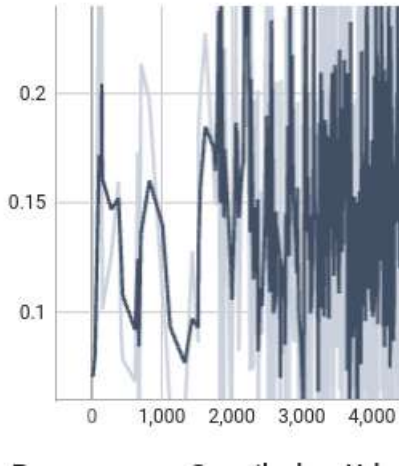
Attempt 5:

Saturation and contrast of RGB images was increased to get better quality images.

SSIM : 0.5602 ± 0.0538
LPIPS: 0.6774 ± 0.0623

The images are very similar to the conditioning image. The quality is also better due to increased contrast in the training images.

After training for more than 2000 steps, the loss displayed instability. We are using the checkpoint after 2000 steps.



Attempt 6:

CLAHE-enhanced thermal intensity and HED-detected edges combined was used as the control signal.

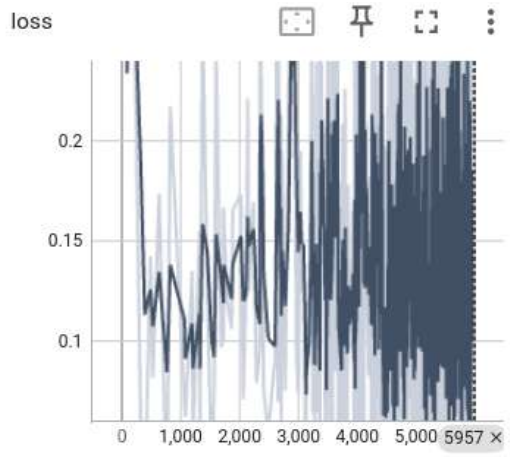
SSIM : 0.6298 ± 0.0564
LPIPS: 0.7235 ± 0.0999

2500 steps:

The loss reduced while training for the first 2000 steps. But as we trained it for more steps, the loss showed instability/oscillation and the quality of the image degraded. We thus use the checkpoint after 2000 steps.

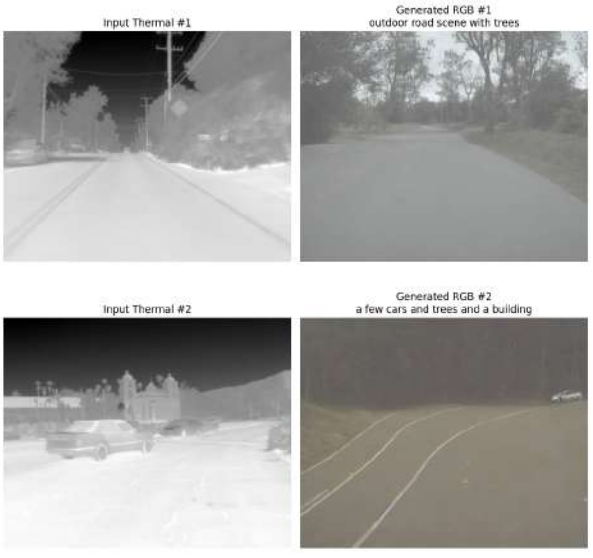
6000 steps:

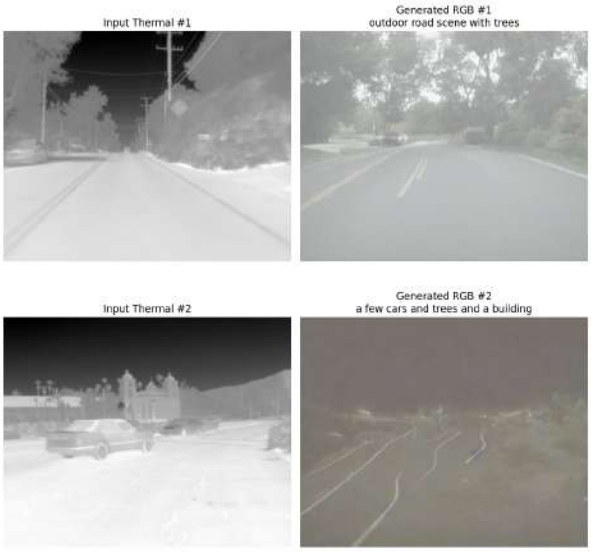
The model likely overfit and memorized the noisy HED edges.



Ablations during inference

Attempt 3 model

Conditioning	Image
1.0	

1.5	

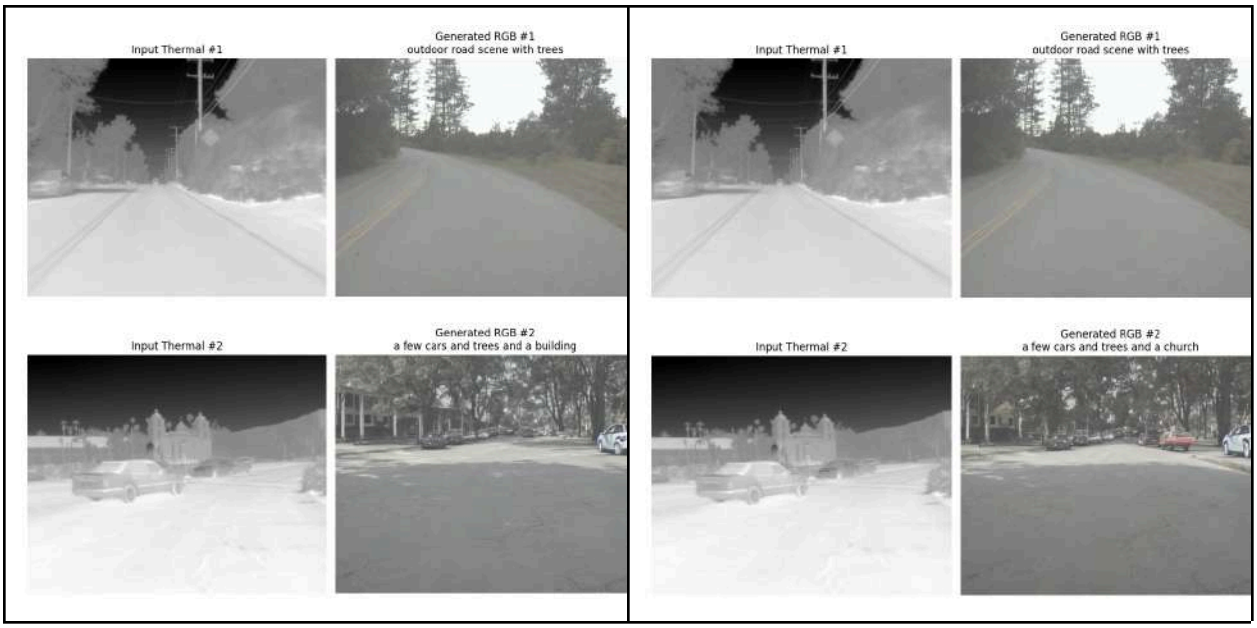
The second image was very different from the images in the dataset, due to which it was unable to follow the structure. On increasing conditioning, the images became blurry.

Without prompts: the model produced unrealistic images.

Attempt 4 model

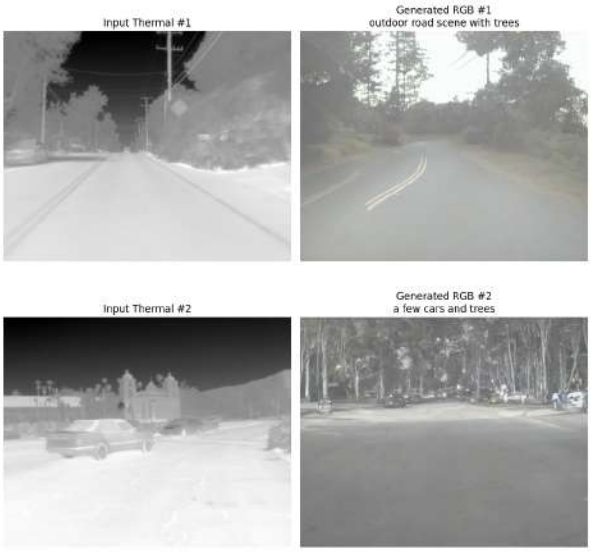
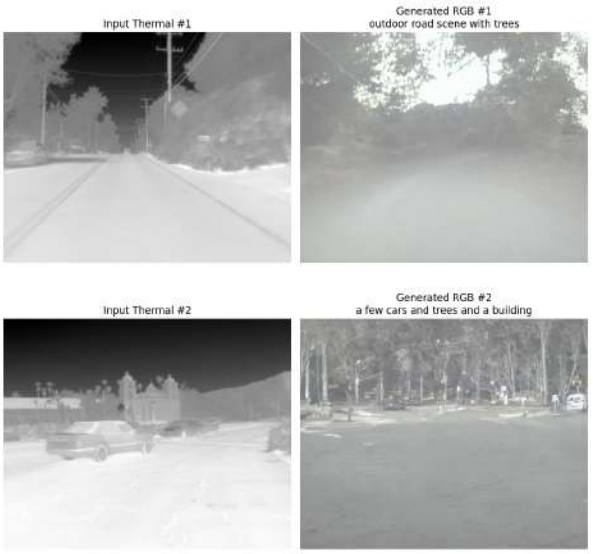
The structure of the second image is closer than the image provided by attempt 3.

Conditioning = 1.0, regular prompts	Conditioning = 1.0, detailed prompts



Adding detailed prompts improved the semantics of the image, but did not help the second image follow the structure better.

Conditioning	Image
0.5	

1.5	
2.0	

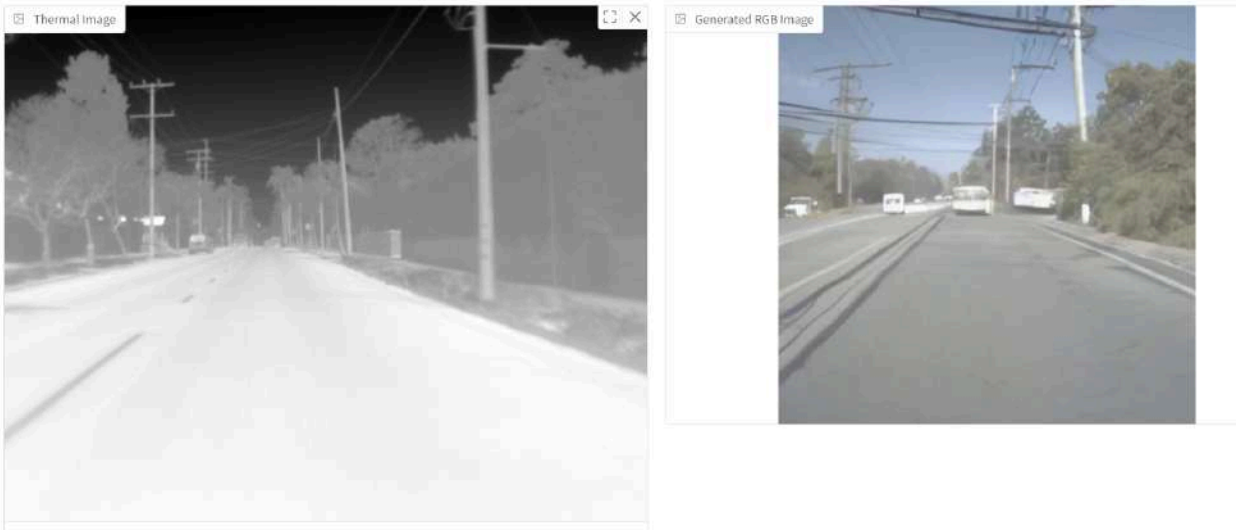
With conditioning = 0.5, the images are sharp but do not adhere to structure completely. With a higher conditioning, the images appear more blurry. Conditioning = 1 appeared to be the best for all the models.

The results produced by this model are better than attempt 3.

Without prompts:

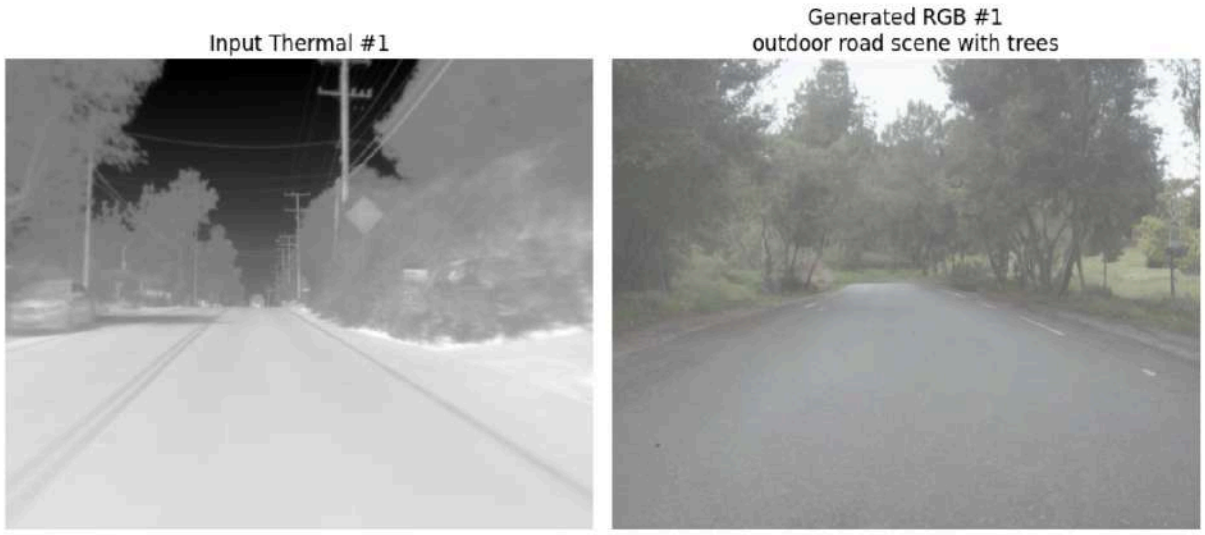
Attempt 5 model

Conditioning scale = 1.0



Attempt 6 model

Conditioning scale = 1.0



Insights and Failure Analysis

- Text prompts significantly improve semantic accuracy and object placement. The model tends to hallucinate when prompts are not added.
- Dataset quality (color saturation, contrast) directly limits output quality
- Training shows optimal performance around 1500-3000 steps before overfitting
- Single-channel thermal conditioning achieves reasonable results but struggles with complex overlapping scenes
- 3e-5 learning rate provides convergence
- Control strength vs. hallucination tradeoff: Increasing the conditioning scale improves structural alignment but makes images blurry and less realistic.
- Attempt 5 has a lower SSIM than Attempt 6 due to contrast enhancement, which changes luminance relative to the ground truth. On the other hand, it had a lower LPIPS than Attempt 6, indicating that Attempt 5 is perceptually better.

Fails:

- The second image in the inference ablations section of attempts 3 and 4 fails to follow the thermal structure due to weak and ambiguous spatial cues in the thermal input, particularly in regions with saturated intensities and overlapping objects. In such cases, the ControlNet receives insufficient structural information, allowing the diffusion model's semantic prior and text prompt to dominate generation.
- Single-channel thermal conditioning lacks explicit edge information. This contributes to a lower SSIM score, as SSIM is sensitive to local structural deviations even when overall scene semantics are preserved.

- To add edge information, we tried HED and also enhanced the image using CLAHE. Despite this, SSIM was low as the edges were noisy and the model was beginning to overfit on this.

Future Directions

- Our current approach in attempt 6 encodes thermal intensity, edges, and contrast into RGB channels of a single ControlNet input. A better extension would be a multi-branch ControlNet, where separate ControlNet modules process different modalities (raw thermal, edge maps, contrast-enhanced thermal) and their outputs are fused at different UNet stages. This would allow the model to learn better.
- Training could be staged by starting with strong conditioning (high ControlNet scale, no text prompts) to enforce structure preservation, followed by gradually reducing conditioning strength and introducing richer text prompts..
- While diffusion training relies on noise prediction loss, losses such as edge consistency loss (between generated and conditioning edges) or perceptual similarity (LPIPS) could be incorporated during fine-tuning. This would penalize structural deviations.

Sketch to Webpage

Problem Definition

Apart from the suggested novel task of Thermal image to RGB image reconstruction, we also tried a custom task of webpage design generation based on the rough sketch of the webpage as input.

Task: Given an image of a rough sketch of the webpage layout as input conditioning, generate a webpage design that preserves the spatial structure while adding realistic elements and details.

Applications:

- Convert a rough idea of the webpage layout into an idea of how the developed webpage would look.
- Enables faster development of frontend for websites and webpages and helps designers to conceptualize early-stage ideas.

Related Work

- Sketch2Code**:** Introduces a benchmark that evaluates state-of-the-art Vision Language Models (VLMs) on automating the conversion of rudimentary sketches into webpage prototypes.

For this task, we use the Sketch2Code dataset which contains the images of rough sketches of the required webpage, the generated HTML code for the webpage and the screenshot of the rendered webpage.

Dataset Creation

https://huggingface.co/datasets/SALT-NLP/Sketch2Code - the dataset contains images of rough sketches of the required webpage, the generated HTML code for the webpage and the screenshot of the rendered webpage.

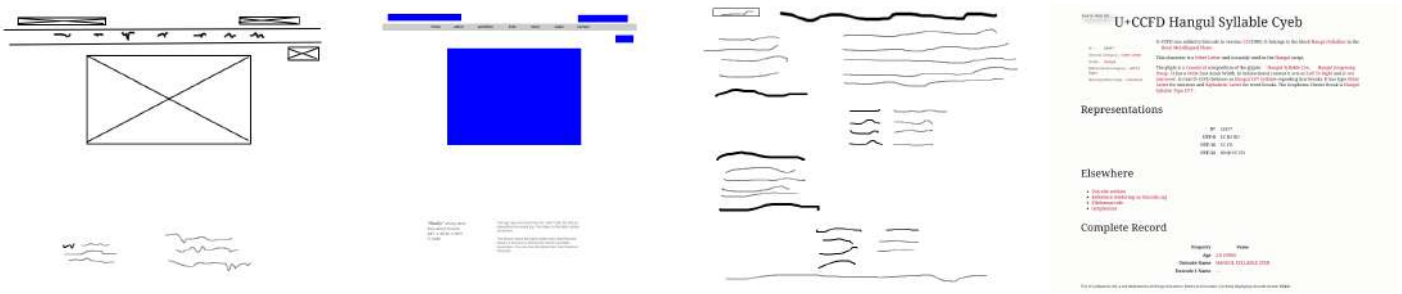
Pre-processing:

- In this dataset, there are multiple sketches for the same webpage. From the original dataset, we generate a CSV file which contains the pairs of the filenames for the sketches and the corresponding webpages. This results in a total of 731 pairs.
- We also make a synthetic dataset where we modified the HTML files to change the colour combination of different HTML elements. From the original HTML files, we generated three new webpage screenshots with different colour combinations. These new images have suffixes of '_blue', '_dark' and '_red'. With the help of this new synthetic dataset, we have considerably more data points (1888 pairs).

Control Signal

We tried training the model with the rough sketches as the conditioning signal. The sketches were converted into uniform dimension black and white images. The CSV files for the original and synthetic datasets were used to load the sketches and target output design images for model training.

An example of the sketch and the screenshot of the corresponding webpage generated can be seen below:



Training

Best hyperparameters:
--resolution=384 \
--mixed_precision=fp16 \
--learning_rate=3e-5 \
--lr_scheduler=constant_with_warmup \
--lr_warmup_steps=500 \
--max_train_steps=2500 \
--train_batch_size=4 \
--gradient_accumulation_steps=4 \
--gradient_checkpointing \
--use_8bit_adam \
--set_grads_to_none \
--enable_xformers_memory_efficient_attention \
--checkpointing_steps=1000 \
--validation_steps=100 \
--seed=42 \

Experiments and Ablations

a) Learning rate: 3e-5
Resolution: 384
Batch size: 4
Training steps: 2000
No text prompts

@Step 1000:

@Step 2000:

b) Learning rate: 3e-5
Resolution: 384
Batch size: 4
Training steps: 4000
No text prompts

@Step 1400:

@Step 2800:

@Step 4000:

c) Learning rate: 1e-5
Resolution: 384
Batch size: 4
Training steps: 2000
No text prompts

@Step 1200:

@Step 2000:

Insights and Failure Analysis

- Dataset diversity such as limited one-to-many pairs of sketches to generated images limits output diversity and quality
- Training shows optimal performance around 2000-3000 steps
- 3e-5 learning rate provides convergence
- Position of text blocks and the length of the text in the generated images is mostly followed from the rough sketches

Fails:
In the generated images, the position of other elements such as the rectangular boxes for colours and logos is not followed well. The elements are either mis-aligned or the size of the element is not followed.

Future Directions

1. Training with prompts

In our current training setting, the model is not provided any text input. This can be changed by giving the model text prompts which describe the layout and relative positioning of the different elements in detail. Text prompts can also include colour assignment for different elements. This can potentially increase the structural similarity in the generated webpage designs with the input rough sketch.

2. Per-element property loss

Task-specific losses such as element relative position loss (among elements in generated images), element absolute position loss (difference in position of elements between conditioning and generated images) or LPIPS could be incorporated during fine-tuning. This would explicitly penalize structural inconsistencies.

3. Multi-branch / multi-channel ControlNet

Multi-conditioned *ControlNet*, where separate ControlNet modules process different properties of the generated image such as the absolute position of elements, colour and texture properties and their outputs are fused at different UNet stages. This would allow the model to better follow multiple required property requirements.