Project Report
Calvin Pugmire
CS 470 Section 001

The problem:

SD: Create a fine-tuned Stable Diffusion LoRA model for generating customized clone troopers.
IC: Create an image captioning transformer-based DNN using the Flickr8k/COCO dataset.

Is this a classification problem? A regression problem?

SD model: Image generation is a generative problem: How do you generate an image based on a given text?
IC model: Caption generation is a generative problem: How do you generate a text based on a given image?

Is it supervised? Unsupervised?

SD model: Supervised via paired labels.
IC model: Supervised via paired images.

What sort of background knowledge do you have that you could bring to bear on this problem?

SD model: I have general DNN knowledge from CS 474, experience using Stable Diffusion, and Star Wars expertise.
IC model: I have general DNN knowledge from CS 474.

What other approaches have been tried? How did they fare?

SD model: There has been one other clone trooper model released, but it performs rather poorly overall. Furthermore, it only contains a limited number of legions and is modeled specifically after “Star Wars: The Clone Wars” troopers.
IC model: Both CNNs and RNNs have been made. These models are simple and effective, but can struggle with long-range dependencies.

The dataset:

Where did it come from? Who published it?

SD model: The dataset I used for this model came from clone trooper images gathered from across the web, paired with captions that I made myself.
IC model: The datasets I used for this model were the Flickr8k and COCO datasets. They were published by adityajn105 on Kaggle.com and various collaborators on COCOdataset.org, respectively.

Who cares about this data?

SD model: Various Stable Diffusion artists have been looking for an easy way to make customized clone troopers since August of 2022, and nobody seems to have made one as of yet.
IC model: Many deep learning engineers use these datasets to train various DNN networks, especially those for labeling images, captioning images, and generating images.

The exploratory data analysis:

Before you start coding, you should look at the data. What does it include? What patterns do you see?

SD model: Around 200 images of clone troopers, each paired with a text-based set of descriptions.
IC model: 8091 (Flickr8k) / 82783 (COCO) images, each paired with 5 human-made captions.

Any visualizations about the data you deem relevant

SD model: N/A. IC model: N/A.

The technical approach:

Background on the approach

SD model: The fine-tuning process for stable diffusion models has become rather streamlined: You gather images, create labels for them, choose a base model to fine-tune, adjust various training parameters in a training app like Kohya-SS, and then run said app on your chosen subjects.
IC model: Transformer models are generally the best solution when it comes to image captioning. They employ a combination of self-attention and encoder-decoder attention to capture long-range dependencies within an image and its generated caption, leading to more coherent and detailed descriptions. Naturally, I decided to use this approach.

Description of the model you used

SD model: The base model I used was v1-5-pruned. It is a well-rounded, general-purpose Stable Diffusion model.
IC model: The model I used was an encoder-to-decoder (image-to-text) transformer model.

Description of the training algorithm you used

SD model: The training algorithm is contained in the Kohya-SS app. I did not have (or need) access to it, as this part of my project focused on the research, data preparation, and testing+analyzing aspects of AI creation.
IC model: For each epoch, the algorithm loops through each batch of training image+caption pairs. The image is fed into the transformer to generate a caption, and then the generated caption is compared to the correct caption to compute the loss. Gradients are then made and applied to the transformer by the optimizer (via backpropagation).

Description of how you partitioned your data into a test/training split

SD model: This was handled by the Kohya-SS application.
IC model: There was no test/training split. Because all of the images had five different captions (and were therefore well-rounded), all of the image+caption combinations could be used as training data.

How many hyperparameters does your model have? What optimizer did you use?

SD model: 87. AdamW8bit.
IC model: 13. Adam.

What topology did you choose, and why?

SD model: A Latent Diffusion Model (fundamental for all Stable Diffusion models), utilizing a U-Net (for denoising) and an autoregressive model (for diversity and overfitting prevention).
IC model: An Encoder-Decoder Model (fundamental for all transformer-based models), utilizing self-attention, encoder-decoder attention, positional encoding, and feed-forward layers.

Did you use any pre-trained weights? Where did they come from?

SD model: Yes. The pre-trained weights came from the v1-5-pruned base model.
IC model: Yes. The pre-trained weights came from a Vision Transformer (ViT) model.

The results:

What was your final RMSE on your private test/training split?

SD model: The model's performance is not reported in the Kohya-SS app. I did not have (or need) access to it, as this part of my project focused on the research, data preparation, and testing+analyzing aspects of AI creation. However, after properly cleaning and labeling my data, the model's performance was satisfactory.

IC model: Image captioning transformer models use cross-entropy loss and accuracy instead of RMSE. When training on the Flickr8k dataset, I reached a (local) minimum+maximum of loss=5.2 and accuracy=0.19. When training on the COCO dataset (roughly 10x larger), I reached a (local) minimum+maximum of loss=3.69 and accuracy=0.36.

Did you overfit? How do you know?

SD model: No. After thorough testing, I was able to confirm that the model responds well to diverse prompts and also supplies diverse solutions to the same vague prompts.

IC model: No. The model correctly captions images containing diverse subjects.

Was your first algorithm the one you ultimately used for your submission? Why did you (or didn't you) iterate your design?

SD model: No. I had three iterations of my model. The first iteration lacked labels describing the type of shot that was taken (full body, portrait, etc.) and would create an ugly mismatch of an image when asked for something larger than 512x512. The second iteration lacked labels describing the background (hallway, forest, etc.) and would create repetitive backgrounds for nearly all of the images. The third iteration had no notable problems and performs well.

IC model: No. I had four iterations of my model. The first iteration used a dataset class like one from CS 474, which broke the algorithm in multiple, time consuming ways and made it ridiculously difficult to modify and debug. The second iteration was trained on the Flickr8k dataset (8091 images), which was too small and stopped at accuracy=0.10 (with captions that repeated 1-3 words incessantly). The third iteration also used Flickr8k but was fine-tuned to maximize performance, but it still stopped at accuracy=0.19 (with somewhat dismal captions). The fourth iteration was instead trained on the COCO dataset (82783 images) and reached accuracy=0.36 (with satisfactory captions).

Did you solve (or make any progress on) the problem you set out to solve?

SD model: Yes, I have solved the problem I set out to solve. I have made a Stable Diffusion LoRA model that can successfully make customized clone trooper images based on submitted prompts.

IC model: Yes, I have made progress on the problem I set out to solve. I have made an image captioning transformer-based model (using the COCO dataset) that can correctly caption submitted images.

Some outputs:

SD model:

				
Special ops trooper in the nighttime city.	Sniper in the mountains.	Bright orange (for fun).	Yes ma'am.	Christmas trooper.

IC model:

				
'a' 'stop' 'light' 'is' 'lit' 'up' 'on' 'a' 'street' 'light'	'a' 'dog' 'is' 'riding' 'on' 'a' 'surfboard' 'in' 'the' 'ocean'	'a' 'baseball' 'player' 'throwing' 'a' 'ball' 'on' 'a' 'field'	'a' 'large' 'elephant' 'is' 'standing' 'in' 'front' 'of' 'a' 'fence'	'a' 'train' 'is' 'parked' 'on' 'the' 'train' 'tracks'

Hour Log:

Day:	CS 470:
10/23	1 hours: -Stable Diffusion model. -Reading+research: https://www.youtube.com/watch?v=j-So4VYTL98 > Installed Kohya SS Trainer.
10/24	1 hours: -Stable Diffusion model. -Prep work: Dataset collection.
10/31	1 hours: -Stable Diffusion model. -Prep work: Dataset cleaning+preparation.
11/4	1 hours: -Stable Diffusion model. -Prep work: Dataset preparation.
11/6	1 hours: -Stable Diffusion model. -Main work: Coding+debugging.
11/9	0.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/11	0.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/13	0.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/16	0.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/18	0.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/21	2.5 hours: -Image captioning model. -Main work: Coding+debugging.
11/22	3 hours: -Image captioning model. -Main work: Coding+debugging+testing.
11/23	1 hours: -Image captioning model. -Main work: Coding+debugging+testing.
11/24	1 hours: -Image captioning model. -Main work: Debugging+testing.
12/13	3 hours: -Image captioning model. -Main work: Coding+debugging+testing.
12/14	0.25 hours: -Stable Diffusion model. -Reading+research: https://www.youtube.com/watch?v=70H03cv57-o > Installed LoRA addon for Stable Diffusion A1111. 0.5 hours: -Stable Diffusion model. -Main work: Testing+analyzing. 0.5 hours: -Stable Diffusion model. -Prep work: Dataset cleaning+preparation. 0.25 hours: -Image captioning model. -Main work: Coding+testing+analyzing.
12/15	0.25 hours: -Stable Diffusion model. -Prep work: Dataset cleaning+preparation. 0.25 hours: -Stable Diffusion model. -Main work: Testing+analyzing.
Totals:	Read+Res: 1.25 hours. Prep work: 3.75 hours. Main work: 15 hours. Total work: 20 hours.