Department of Electrical & Computer Engineering
ECE 1786: Creative Applications of Natural Language Processing
Professor. Jonathan Rose

Final Report

December 12, 2022

Zixuan Li – 1002922126
Zhenyue Yu – 1003800087
Word Count – 1993 words

1. Introduction

Image captioning is a challenging problem since it concerns generating neural language to describe the input image automatically. Generating text descriptions for images with encoder-decoder frameworks is becoming increasingly popular. Image captions are important since they spark a reader's interest in a full-text story and connect vision and language in a generative fashion.

The project aims to design and implement an image caption model that can automatically generate captions given a dog image as input. The generated captions can recognize the breed of the dog, the actions of the dog, like running or lying, the environment of the image, and the colors of dogs from the input dog image.

2. Project Overall Illustration

Figure 1 is the flow diagram depicting the steps in the pipeline of the caption generation process in this project. Our anticipated model is transformer-based encoder-decoder architecture. The input image is first fed into the backbone layer in the model's encoder to extract essential visual features. The feature map is further encoded to the same size as the word embedding through the transformer-based network in the model's encoder. The model decoder then takes the outputs from the encoder and the generated sequence of word embeddings to recursively produce the next word predictions.

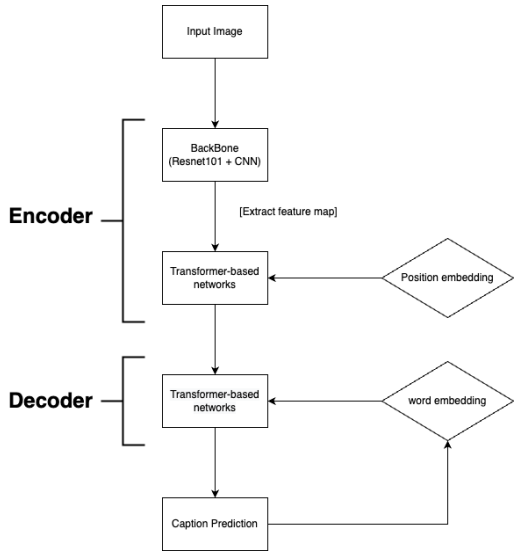

Figure 1: Flow diagram depicting the steps in the pipeline of the caption generation process

3. Background & Related Work

The paper our group found describes Caption TransformR (CPTR), a transformer-based image captioning model architecture. This paper provides the overall architecture of CPTR and a more detailed explanation of the structure of each block in both the encoder and decoder [1]. In addition, the performance of different architectures is compared, and the CPTR is proven to be the optimum solution.

The GitHub repository our group found provided a pre-trained CPTR checkpoint on the COCO dataset [2], and we fine-tunning the model parameters on our manually labelled dog dataset in this project. In

the repository, the model structure is already built, and the trainer function is also provided. In the given trainer algorithm, the model is trained with fixed number of epochs, and the calculation for computing validation loss is missing. Besides, the prediction function in the repo only supports the greedy sampling approach.

4. Data and Data Processing

Our group planned to obtain the source images for training from a Kaggle dataset. This Kaggle dataset contains dog images for 70 different breeds, and each image has the shape of 224*224*3, as shown in Figure 2. We manually selected 16 images for each dog breed and created four captions as the ground truth caption for each image. As shown in Figures 3 and 4, inside each dog breed folder, 14 images were labelled from 1 to 14 for the training dataset, and 2 images were labelled from 1 to 2 for the validation dataset.

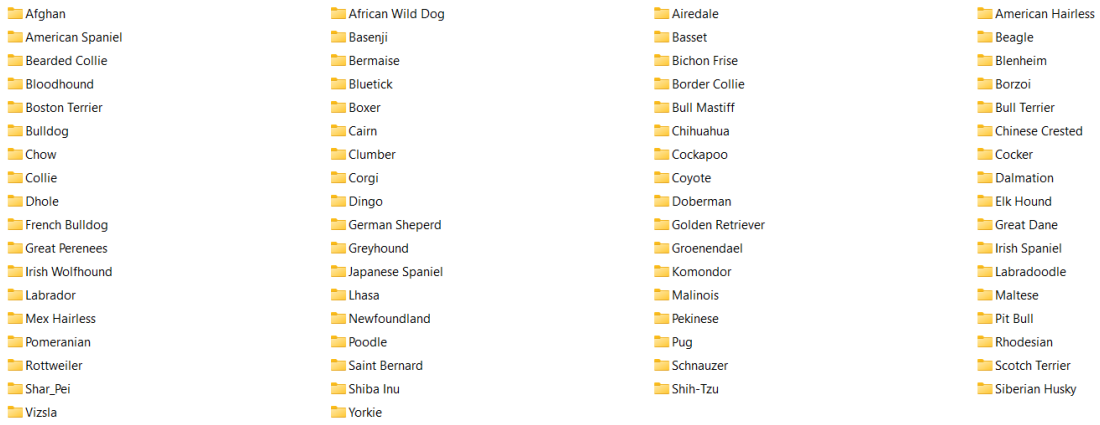

Figure 2: Dog breed folders

Figure 3: Training images for Border Collie dogs

Figure 4: Validation images for Border Collie dogs

Then we created four captions as the ground truth caption for each image in annotations_t.csv and annotations_v.csv, respectively. Figure 5 indicates captions created for the dog breed Border Collie. For the created captions in both dataset files, one of the ground truth captions must at least contain the action of the dog(s). The rest of the ground truth captions should contain more information, such as the dog breed, dog colors, and the environment of the images.

Border Collie/1.jpg	a dog looking to the front .
Border Collie/1.jpg	a Border Collie dog looking to the front .
Border Collie/1.jpg	a white and brown Border Collie dog looking to the front .
Border Collie/1.jpg	a white and brown Border Collie dog looking to the front on the grass .
Border Collie/2.jpg	a sitting dog looking to the front .
Border Collie/2.jpg	a sitting Border Collie dog looking to the front .
Border Collie/2.jpg	a white and black sitting Border Collie dog looking to the front .
Border Collie/2.jpg	a white and black sitting Border Collie dog looking to the front on the grass .
Border Collie/3.jpg	a sitting dog looking to the front .
Border Collie/3.jpg	a sitting Border Collie dog looking to the front .
Border Collie/3.jpg	a white and black sitting Border Collie dog looking to the front .
Border Collie/3.jpg	a white and black sitting Border Collie dog looking to the front in the blank background .
Border Collie/4.jpg	a sitting dog looking to the left .
Border Collie/4.jpg	a sitting Border Collie dog looking to the left .
Border Collie/4.jpg	a white and black sitting Border Collie dog looking to the left .
Border Collie/4.jpg	a white and black sitting Border Collie dog looking to the left on the grass .
Border Collie/5.jpg	a sitting dog looking to the left .
Border Collie/5.jpg	a sitting Border Collie dog looking to the left .
Border Collie/5.jpg	a white and black sitting Border Collie dog looking to the left .
Border Collie/5.jpg	a white and black sitting Border Collie dog looking to the left on the grass .
Border Collie/6.jpg	a sitting dog looking to the front .
Border Collie/6.jpg	a sitting Border Collie dog looking to the front .
Border Collie/6.jpg	a white and black sitting Border Collie dog looking to the front .
Border Collie/6.jpg	a white and black sitting Border Collie dog looking to the front on the grass .
Border Collie/7.jpg	a standing dog looking to the front .
Border Collie/7.jpg	a standing Border Collie dog looking to the front .
Border Collie/7.jpg	a white and black standing Border Collie dog looking to the front .
Border Collie/7.jpg	a white and black standing Border Collie dog looking to the front on the grass .
Border Collie/8.jpg	a standing dog looking to the front .
Border Collie/8.jpg	a standing Border Collie dog looking to the front .
Border Collie/8.jpg	a white and black standing Border Collie dog looking to the front .
Border Collie/8.jpg	a white and black standing Border Collie dog looking to the front in the blank background .
Border Collie/9.jpg	a running dog looking to the front .
Border Collie/9.jpg	a running Border Collie dog looking to the front .
Border Collie/9.jpg	a white and black running Border Collie dog looking to the front .
Border Collie/9.jpg	a white and black running Border Collie dog looking to the front on the grass .
Border Collie/10.jp	a dog looking to the right .
Border Collie/10.jp	a Border Collie dog looking to the right .
Border Collie/10.jp	a white and black Border Collie dog looking to the right .
Border Collie/10.jp	a white and black Border Collie dog looking to the right in the blank background .
Border Collie/11.jp	a dog looking to the front .
Border Collie/11.jp	a Border Collie dog looking to the front .
Border Collie/11.jp	a white and black Border Collie dog looking to the front .
Border Collie/11.jp	a white and black Border Collie dog looking to the front in the blank background .
Border Collie/12.jp	a dog looking to the front .
Border Collie/12.jp	a Border Collie dog looking to the front .
Border Collie/12.jp	a white and black Border Collie dog looking to the front .
Border Collie/12.jp	a white and black Border Collie dog looking to the front in the blank background .
Border Collie/13.jp	a dog looking to the front .
Border Collie/13.jp	a Border Collie dog looking to the front .
Border Collie/13.jp	a white and black Border Collie dog looking to the front .
Border Collie/13.jp	a white and black Border Collie dog looking to the front in the blank background .
Border Collie/14.jp	a sitting dog looking to the front .
Border Collie/14.jp	a sitting Border Collie dog looking to the front .
Border Collie/14.jp	a white and black sitting Border Collie dog looking to the front .
Border Collie/14.jp	a white and black sitting Border Collie dog looking to the front on the grass .

Figure 5: A screenshot of Border Collie dog captions in annotations_t.csv

While training our baseline model, we first resized all images to the dimension of 224*224. Then, we applied the data augmentations on the images: random horizontal flip with a probability of 30% and random rotation with 10 degrees. While training our anticipated model, we first padded all images to the dimension of 299*299. Then, we applied the data augmentations on the images: random horizontal flip with the probability of 50%, random rotation with [0, 90, 180, 270] degrees, and color jitters on [brightness, saturation, contrast]. Finally, while training both models, we normalized the range of pixel values from (0, 255) to (-1, 1).

5. Architecture and Software

The anticipated model is built based on CPTR with some minor changes. As shown in Figure 6, the model encoder consists of two sub-structures: a backbone network and a transformer-based network. The input image is first fed into a pre-trained ResNet101 to extract features from one of its intermedia layers. After several experiments, we found that the intermedia layer contains more condensed visual features if denser and closer to the classifier head. Since the intermedia layer output contains numerous channels, and CNN-based projection layer is applied to further compress the feature map with the kernel size of 1. Afterward, the output from the backbone network is fed into a transformer-based network together with position embeddings, and this transformer-based network brings more learning signals in the encoding process, which share the gradient with the model's decoder during the backpropagation process such that they all can learn some high-level semantic information. In the decoder's design, the generated sequence of word embeddings is first fed into a masked self-attention transformer block; then, a cross-attention transformer block combines the visual and semantic information by taking the outputs from the decoder's self-attention transformer block as the query*(Q)* input and the outputs from the encoder as the key*(K)*, and value(*V)* inputs. The hyperparameter settings used for training our anticipated model are provided in Table 1. The early stopping approach is applied to avoid overfitting problem in the training process. Specifically, the validation loss is computed for each epoch, and the training process is stopped whenever the validation loss increases for two consecutive epochs.

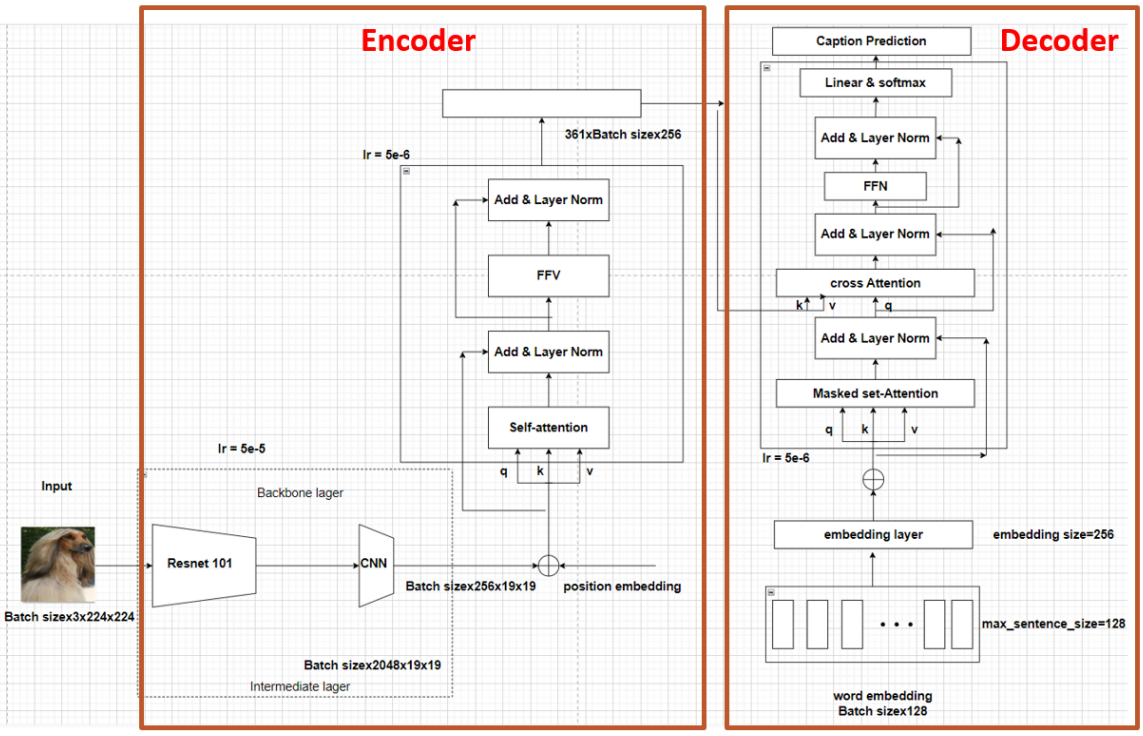

Figure 6: The encoder (left) and decoder (right) architecture for the anticipated model

Learning rate for backbone network	**Learning rate for encoder's & decoder's transformer blocks**	**Word embedding size**	**Tokenizer**	**Maximum sentence length**	**Batch size**
5e-6	5e-5	256	BERT	128	16

Table 1: Hyperparameter settings for training the anticipated model

6. Baseline Model

The baseline model is composed of a CNN-based encoder and a transformer-based decoder. As shown in Figure 7, for the CNN encoder, our group fine tunned the pre-trained MobileNetV2 because it contains fewer parameters and lower floating-point operations per second. Compared to other model architectures, the MobileNetV2 applies inverted residuals and linear bottleneck to reduce memory usage while maintaining the training speed [3]. To fit this model in our project, we replaced the previous classifier head with two fully connected layers, one activation function, and one dropout layer. The decoder is trained from scratch, and its architecture is similar to the GPT-mini in assignment 3, which contains 5 transformer blocks, 5 heads, and an embedding size of 100. Like the anticipated model, the same early stopping technique is applied in the training process, and the hyperparameter settings used for training the baseline model are provided in Table 2.

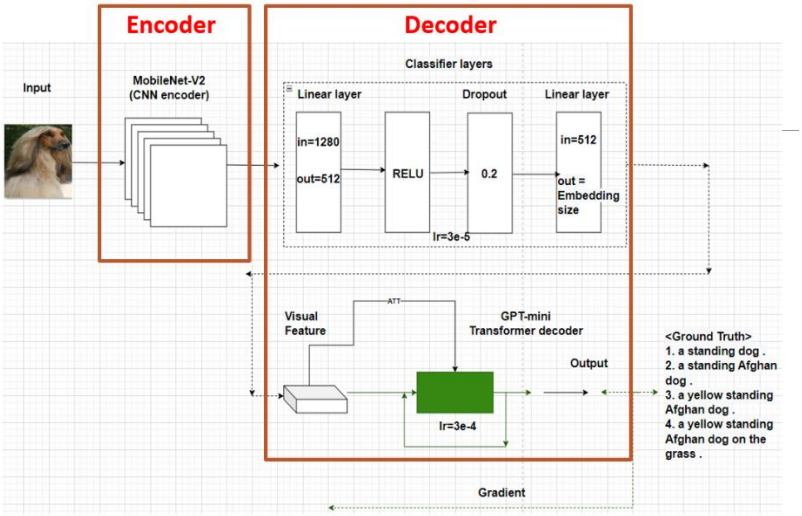

Figure 7: The encoder (left) and decoder (right) architecture for the baseline model

Learning rate for CNN encoder	**Learning rate for transformer decoder**	**Word embedding size**	**Tokenizer**	**Batch size**
3e-5	1e-4	100	GloVe	64

Table 2: Hyperparameter settings for training the anticipated model

7. Quantitative Results

The Bilingual Evaluation Understudy (BLEU) score is widely used in NLP to evaluate model performance in text generation tasks by comparing one candidate sentence to one or more reference sentences [4]. The n-gram precision score, as shown in equation (1) is the key idea in the BLEU score calculations, where n represents n consecutive words in a sentence. As shown in equation (2), the BLEU is the product of brevity penalty and geometric average precision scores. As indicated in equation (3), the brevity penalty penalized the model generations that are too short comparing to the targets, where c is the length of the generated captions and r is the length of ground-truth captions. Geometric average precision scores compute the precision score up to 4-grams with the uniform wight by default, as shown in equation (4), where p_n is the n-gram precision score.

$$ngram\ precision\ score = \frac{\#\ of\ matched\ n\ consecutive\ words\ btw\ candidate\ \&\ targets}{Total\ \#\ of\ n\ consecutive\ words\ in\ target\ sentences} \quad (1)$$

$$BLEU\ score = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores \quad (2)$$

$$Brevity\ Penalty = \begin{cases} 1, & if\ c > r \\ e^{(1-r/c)}, & if\ c \leq r \end{cases} \quad (3)$$

$$Geometric\ Average\ Precision\ (N = 4, w_n = \frac{1}{4}) = \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (4)$$

To measure the performance of models, our team prepared 140 testing images and generated 4 captions for each image using the baseline and anticipated model, respectively. The top-k sampling method, with k equal to 3, is implemented to recursively generate words for captions with some randomness so that model generations for the same image can be different in each inference. For each testing image, we implemented the BLEU score algorithm to compare each of 4 generated captions to four reference ground-truth captions independently, and the average of 4 BLEU scores are then computed. Finally, we repeated the same procedure on all test images, and the average BLEU score across all testing images for each model is achieved. As shown in Table 3, the anticipated model significantly outperforms the baseline model in terms of the average BLEU score across all testing images.

Average BLEU Score for the Anticipated Model	**Average BLEU Score for the Baseline Model**
0.8161	0.6978

Table 3: Average BLEU score for the baseline and the anticipated model across all testing images

8. Qualitative Results

As shown in Figure 8, two representative images are selected from the test dataset with different actions, background, and breeds. As shown in Figure, the first image is an America Spaniel dog standing on the snow, and the second image is a Border Collie dog running on the grass.

Figure 8: Selected images for qualitative test (America Spaniel dog on the left & Border Collie dog on the right)

The baseline and the anticipated models are implemented to generate 4 captions for each chosen image, respectively, and the model generations and reference captions for each image are summarized in Table 4 and 5. The baseline successfully generates complete sentences; however, its description of images' details is mostly incorrect, including the color, breeds, background, and actions. The main reason for this result is the baseline mode architecture is heterogeneous, and very little amount of visual information are distilled from the CNN encoder to the transformer decoder during the optimization steps. In contrast, the caption generations from the anticipated model are much more meaningful and accurate. For the first image, the anticipate model correctly named the dog breed and described which direction the dog is looking at, what kind of environment the dog is in, and what is the dog's current body position. In the second image, the Border Collie dog only shows two front legs, and the back half of its body is totally hidden in the image. From the model generations, it can observe that the mode is uncertain about whether the dog is standing or running. Overall, the anticipated model produced captions with much higher quality on both images than the baseline model, and the qualitative results match the conclusion we obtained in our quantitative results.

Ground-Truth Captions for the First Selected Image (America Spaniel)	
A standing dog looking to the left	
A standing America Spaniel dog looking to the left	
A brown standing America Spaniel dog looking to the left	
A brown standing America Spaniel dog looking to the left on the snow	
Baseline Model Generations	**Anticipated Model Generations**
A white and black standing Elk Hound dog looking to the front	A brown standing American Spaniel dog looking to the left
A white and black sitting Border Collie dog looking to the right on the grass	A standing American Spaniel dog looking to the left
A standing dog looking to the right	A standing American Spaniel dog looking to the left on the snow
A white and black sitting Corgi dog looking to the right on the grass	A standing American Spaniel dog looking to the left on the snow

Table 4: Model generations and ground-truth captions for the first selected image

Ground-Truth Captions for the Second Selected Image (Border Collie)	
A running dog looking to the front	
A running Border Collie dog looking to the front	
A white and black running Border Collie dog looking to the front	
A white and black running Border Collie dog looking to the front on the grass	
Baseline Model Generations	**Anticipated Model Generations**
A yellow and black standing Malinois dog looking to the front in the blank background	A white and black running Border Collie dog looking to the front on the grass
A standing dog looking to the front	A standing dog looking to the front
A standing dog looking to the left	A white and black running Border Collie dog looking to the front
A yellow sitting Cairn dog looking to the front on the ground	A standing dog looking to the front

Table 5: Model generations and ground-truth captions for the second selected image

9. Discussion and Learnings

According to the qualitative and quantitative results, the performance of the anticipated model is beyond our expectations, and the CPTR model architecture indicates a great potential to generate high-quality captions. Its homogeneous architecture allows the gradient to pass through the entire model structure during the backpropagation so that visual and semantic information are efficiently combined while optimizing both the encoder and decoder. Moreover, the transformer can take the entire sentence as the input sample such that the context information can be reserved as much as possible. In the future, in a similar project, we will explore more on this model more by generating captions for more images, including but not limited to dogs.

In addition, we found that the current inference time is time-consuming. In the future, we will try to achieve a comparative performance with a reduced number of transformer blocks and heads. Besides, the ResNet101 used in the backbone network is a deep structure; in order to improve the space and computational efficiency, we could apply the knowledge distillation techniques to replace the current

backbone network with a lighter Student model so as to reduce the model size and computing resources and maintain the accuracy and generalization ability at the same time [5].

Finally, in the qualitative result, we observed the quality of model generations decreases when the dog is partially shown in the image. To further improve the model performance in the future, we could apply random crops on the training images to add more generalization ability to the model.

10. Individual Contributions

(Zixuan Li – mainly worked in master branch)

I created 4 ground-truth captions for all validation images in the data preparation process. Before the proposal presentation, I constructed and trained the first version of the baseline and anticipated model. After that, I designed a top-k sampling method to generate words with some randomness using the trained models recursively. After the final presentation, I added more features to the captions in the training and validation dataset, as suggested by the professor.

(Zhenyue Yu – mainly worked in main branch)

First, I manually selected 16 images for each dog breed and split into training and validation datasets. Then, I created 4 captions for all training images before the proposal presentation. After the proposal presentation, I tunned hyperparameter settings for both baseline and anticipated models to achieve a better performance. Afterward, I computed the qualitative and quantitative test on each model using the BLEU score metric. Finally, I retrained the models on the last version of the datasets updated by my teammate.

Reference

[1] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full Transformer Network for Image Captioning," *arXiv.org*, 28-Jan-2021. [Online]. Available: https://arxiv.org/abs/2101.10804. [Accessed: 26-Oct-2022].

[2] Saahiluppal, "Saahiluppal/CATR: Image captioning using transformer," *GitHub*. [Online]. Available: https://github.com/saahiluppal/catr. [Accessed: 12-Dec-2022].

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginow, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381v4*, 2019. [Online]. Available: https://arxiv.org/pdf/1801.04381v4.pdf. [Accessed: 02-Nov-2022].

[4] K. Doshi, "Foundations of NLP explained - bleu score and WER metrics," *Medium*, 11-May-2021. [Online]. Available: https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b. [Accessed: 12-Dec-2022].

[5] G. Hinton, O. Vinyals, M. Zhu, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv:1503.02531v1*, 2015. [Online]. Available: https://arxiv.org/pdf/1503.02531.pdf. [Accessed: 02-Nov-2022].

Appendix

Permission to post video (Zixuan Li): yes
Permission to post final report (Zixuan Li): yes
Permission to post source code (Zixuan Li): yes

Permission to post video (Zhenyue Yu): yes
Permission to post final report (Zhenyue Yu): yes
Permission to post source code (Zhenyue Yu): yes