|  |
| --- |
| **CLIP Based Multi-Modal Model Report** |

**Rishabh Kaushick**

College of Engineering

Northeastern University

Toronto, ON

*kaushick.r@northeastern.edu*

**Abstract**

This project presents the development of a multi-modal retrieval model inspired by CLIP, targeting the task of aligning images and text within a shared embedding space. Using the InfoNCE contrastive learning loss, the model was trained to match corresponding image-text pairs while distinguishing between unrelated examples. Training results demonstrated high retrieval performance, with Recall@1 reaching over 87% for text-to-image retrieval and 91% for image-to-text retrieval. However, validation metrics were substantially lower, highlighting challenges with overfitting and generalization. Qualitative analyses further confirmed that the model’s retrievals on unseen data lacked semantic consistency. These findings suggest directions for future work focused on enhancing model robustness and generalization capabilities.

**1 Datasets**

For this assignment, I have decided to use the Flickr30k dataset. The reasons for this are as follows:

1. The Flickr30k dataset contains much lesser images as compared to the COCO dataset. This will be easier to train due to the GPU limitations.
2. The Flickr30k dataset contains high quality natural language captions. The COCO dataset has complex captions which could introduce more variability making the training process more difficult. Therefore, in this case the Flikr30k dataset allows for sufficient quantity and feasibility.
3. Flickr30k dataset is suited for retrieval-based tasks.

Table 1: The basic feature of both datasets.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Data Set Characteristics | Attribute Characteristics | Associated Tasks | Number of Instances | Number of Attributes |
| Flickr30k | Multivariate | Real | Image-Text Retrieval | 10,000 | 6 |

**1.1 Data characteristics**

The dataset has the following features:

1. 'image': Image (mode=None, decode=True, id=None),
2. 'caption': List of captions
3. 'sentids': List of unique identifiers for each caption
4. 'img\_id': Unique identifier for each image
5. 'split': Either ‘train’ / ‘val’ / ‘test’
6. 'filename': Unique filename for each image

We can see from the above that for our training process, we will only require the image and the captions.

**1.2 Why is this dataset interesting?**

Flickr was a very popular website from 2005 to 2010 that allowed users to share photos with friends or family. Personally, having used this website during that time, I found it quite interesting. Since Flickr has tens of thousands of images along with relevant real-world captions, I felt that this a good dataset that can be used to train a multi-modal AI model.

**1.3 Train, Validation & Test Split**

Since only the first 10,000 rows of data has been considered the data split initially was:

* Training Set Count: 9375 (93.75%)
* Validation Set Count: 301 (3.01%)
* Testing Set Count: 324 (3.24%)

Here the training set is more than 90%, and the test and validation sets are much smaller in comparison. Therefore, in this project, the training – validation – test split has been updated to:

A pie chart with a pie chart in the middle

AI-generated content may be incorrect.  
Figure 1: Updated train – validation – test split.

**2 Exploratory Data Analysis**

Visualizing a few sample images and their corresponding captions from the training dataset:A dog running on grass

AI-generated content may be incorrect.  
Figure 2: Sample image and caption from the training set.

A yellow parachute in the sky

AI-generated content may be incorrect.  
Figure 3: Sample image and caption from the training set.

**3 Model Design**

The CLIP based multi-modal model has the following components:

1. Image Encoder
2. Text Encoder
3. CLIP Model for Shared Embedding Space

The figure image below shows how the captions go through the text encoder, and the images go through the image encoder. They both have a shared embedding space such that the image embedding of a dog corresponds to the text embedding of words like ‘dog’ or ‘animal’.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: Architecture of the CLIP based model.

**3.1 Image Encoder**

For the image encoder, instead of training a model from scratch the following steps have been followed:

1. The ResNet-18 model will be used as the backbone CNN model for feature extraction.
2. Since ResNet-18 model was pretrained on ImageNet and not on Flickr30k dataset, we must perform transfer learning through fine tuning. To do this:
   1. the final layer of the ResNet18 will be removed.
   2. a projection layer will be added at the end of the ResNet18 model to map data into a 512-dimension shared embedding space.

A screen shot of a computer code

AI-generated content may be incorrect.  
Figure 5: ImageEncoder class to finetune existing ResNet-18 model.

**3.2 Text Encoder**

The text encoder, like the image encoder, does not need to be trained from scratch, and the following steps have been followed:

1. DistilBERT will be used as the backbone model for processing text.
2. Most parameters except the last layer will be frozen to reduce training time. Since the DistilBERT architecture consists of 6 layers (0 to 5), we will freeze layers 0 to 4 and allow the layer 5 to updated during training.
3. It takes the [CLS] token embedding as the sentence representation.
4. The last linear layer (and layer norm) then projects the encoded text to the same 512-dimensional embedding space as the image encoder.

A screen shot of a computer program

AI-generated content may be incorrect.  
Figure 6: TextEncoder class to finetune the DistilBERT model.

**3.3 CLIP Model for Shared Embedding Space**

The CLIP model combines both the ImageEncoder and TextEncoder into a single model. It also has a method to compute the InfoNCE contrastive loss between image and text embeddings.

A screen shot of a computer program

AI-generated content may be incorrect.  
Figure 7: CLIPModel combining the Image Encoder & Text Encoder, with contrastive loss.

**4 Training Process (30%)**

To train the CLIP model, the loss function used is the InfoNCE – Info Noise Contrastive Estimation, which is a form of contrastive loss. This loss helps to position positive pairs – the image and correct text close to each other, while pushing the negative pairs – of images which are not related to labels further away from each other.

The training and evaluation loss was captured at each epoch and has been graphed as shown in the figure below:

A graph with a line

AI-generated content may be incorrect.  
Figure 8: Training & validation loss per epoch.

**5 Evaluation and Results (20%)**

To evaluate this model the recall @ k, mean rank and mean reciprocal rank has been calculated. This model will also be evaluated on the two tasks that it can perform:

1. Text to Image
2. Image to Text

**5.1 Recall @ K**

Recall @ k measures how often the correct image is found given the top k texts, or how often the correct text is found from the top k images.

The results for the recall @ k are as follows:

Table 2: Recall @ k for the Training Set

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Recall @ k | Score | Score (%) |
| Text-to-Image Retrieval | Recall @ 1 | 0.8720 | 87.20 % |
| Recall @ 5 | 0.9936 | 99.36 % |
| Recall @ 10 | 1.0000 | 100.00 % |
| Image-to-Text Retrieval | Recall @ 1 | 0.9136 | 91.36 % |
| Recall @ 5 | 0.9888 | 98.88 % |
| Recall @ 10 | 0.9968 | 99.68 % |

Table 3: Recall @ k for the Validation Set

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Recall @ k | Score | Score (%) |
| Text-to-Image Retrieval | Recall @ 1 | 0.0875 | 8.75 % |
| Recall @ 5 | 0.2500 | 25.00 % |
| Recall @ 10 | 0.3750 | 37.50 % |
| Image-to-Text Retrieval | Recall @ 1 | 0.0875 | 8.75 % |
| Recall @ 5 | 0.2750 | 27.50 % |
| Recall @ 10 | 0.3875 | 38.75 % |

We can see clearly that the recall score in the training set is very high, but the score in the validation score is extremely low.

**5.2 Mean Reciprocal Rank (MRR)**

Another metric is to find where at what rank the correct item (text or image) appears. Mean rank is the average of the rank for all the queries. The reciprocal of the rank is known as reciprocal rank and taking the mean of this gives us the mean reciprocal rank (MRR). Therefore, a higher mean reciprocal rank is desirable.

In our project the result of the mean reciprocal rank are as follows:

Table 4: Ranks on Training Set

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Rank | Score | Score % (if applicable) |
| Text-to-Image Retrieval | Mean Rank | 1.23 | - |
| Mean Reciprocal Rank | 0.9251 | 92.51 % |
| Image-to-Text Retrieval | Mean Rank | 1.22 | - |
| Mean Reciprocal Rank | 0.9500 | 95.00 % |

Table 5: Ranks on Validation Set

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Rank | Score | Score % (if applicable) |
| Text-to-Image Retrieval | Mean Rank | 84.79 | - |
| Mean Reciprocal Rank | 0.1713 | 17.13 % |
| Image-to-Text Retrieval | Mean Rank | 83.31 | - |
| Mean Reciprocal Rank | 0.1781 | 17.81 % |

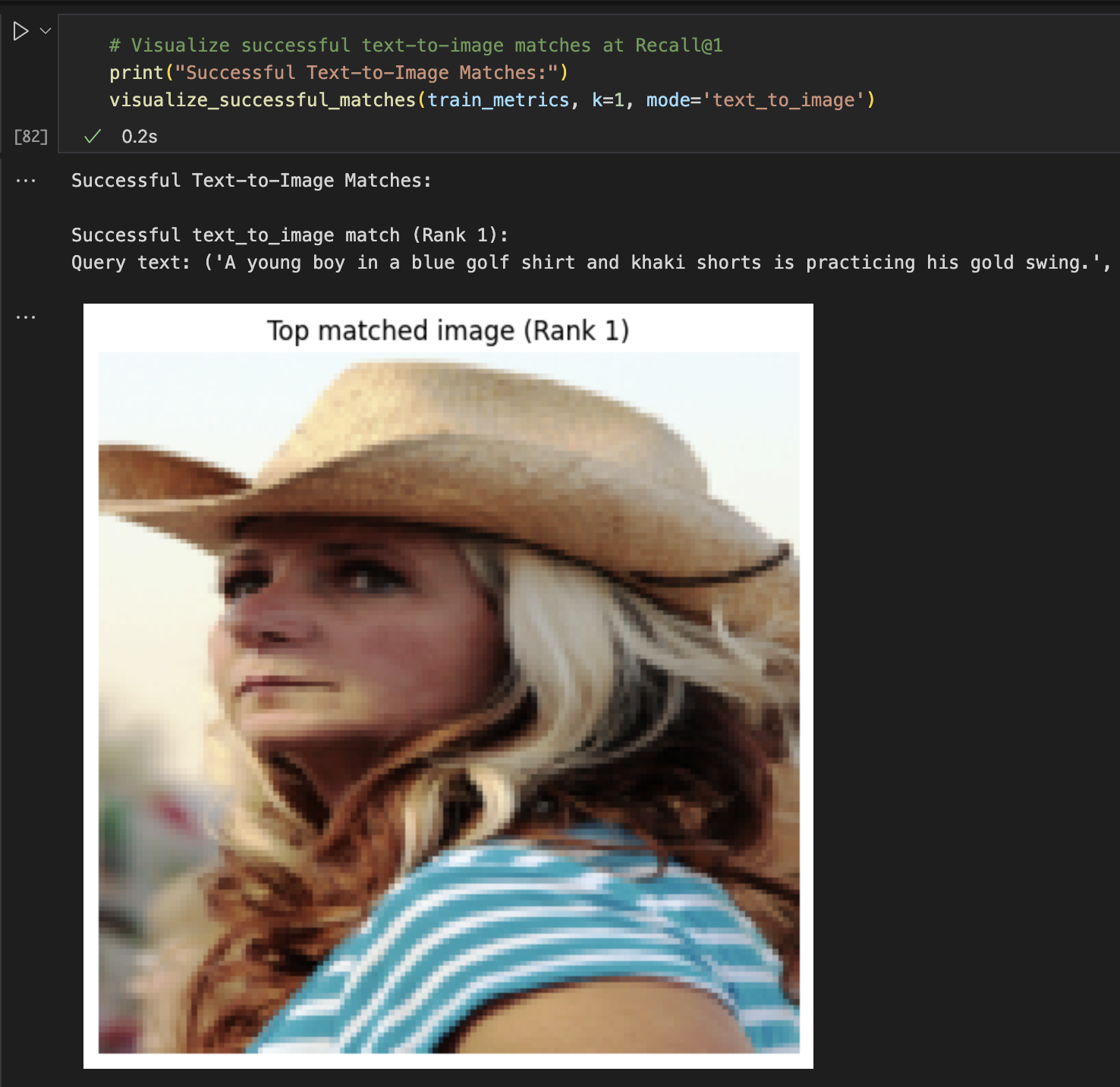
We can see that the rank is very good on training set (almost always 1) and the mean reciprocal rank is also very high (more than 90%). However, the validation test is very low – the mean rank is not good, above 80, and the mean reciprocal rank is also very low around 17%.

**6 Challenges and Insights (10%)**

**5.1 Challenges**

From the evaluation above, we can see that the training retrieval metrics are very high, however the evaluation retrieval metrics are very low. This is also visible from the training and evaluation loss graph in Figure 8. This graph shows that only the training loss keeps reducing. We can see that the validation loss starts initially reducing but later gets worse.

Furthermore, when we visualize the text and image pairs together, we see that they do not make semantic sense as shown below:

  
Figure 9: The picture and the caption at Rank 1 does not match semantically.

**5.2 Future Improvement Areas**

Based on the above information, it is clear to say that this CLIP model has overfit the training data. The model seems to be memorizing the image and text pairs but when looking at new data in the validation set, it is struggling to make sense of it.

The possible Causes of over-fitting are:

1. The training set is too small. In this project, only 10k subset of data has been used, but in the future, we can try this with the entire 30k dataset along with other datasets.
2. Currently the model may be too powerful relative to the data.
3. The training setup is not focusing on generalization. We can add more regularization and train on lesser epochs to avoid this.

**7 Conclusions**

In this project, I built a multi-modal retrieval model inspired by CLIP, aiming to represent image and text representations in a shared embedding space. The model achieved high performance on the training set, with Recall@1 exceeding 87% for text-to-image retrieval and 91% for image-to-text retrieval. These strong training metrics indicate that the model successfully learned to associate image-text pairs within the training distribution.

However, evaluation on the validation set revealed a significant drop in performance, with Recall@1 around 8.75% for both retrieval tasks and a substantially higher mean rank. Furthermore, qualitative analysis showed that retrieved image-text pairs often lacked semantic similarity. These observations suggest that while the model memorized the training set effectively, it struggled to generalize to unseen examples — indicating overfitting.

Future work will focus on improving generalization through techniques such as better regularization, data augmentation, and more robust evaluation strategies.

**References**

1. <https://huggingface.co/datasets/nlphuji/flickr30k>
2. <https://ferdychristant.com/the-rise-fall-and-resurrection-of-flickr-ca1850410ee1>
3. <https://pytorch.org/vision/stable/generated/torchvision.datasets.Flickr30k.html>
4. <https://huggingface.co/docs/datasets/process#rename-remove-cast-and-flatten>
5. <https://pytorch.org/hub/pytorch_vision_resnet/>
6. <https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html>
7. Anthropic. (2025, April 4). Conversation on Building a Clip-Based Multi-Modal Model
8. ChatGPT. (2025, April 4). Conversation with ChatGPT. OpenAI.