Cross modal query retrieval using Deep Learning

Nidutt Bhuptani **Shalvi Dessai**

1 Introduction

In this massive information era, where data is constantly being generated, it becomes difficult to sift through gazillion bytes of multimedia data such as images, videos, or audio files to find the desired item. Hence it becomes crucial to establish a semantic correlation to access multimedia data in an efficient manner. Exploring semantic relationships with images, texts, and videos has been an active field of research for the computer vision community.

Our primary objective is to retrieve relevant images based on a text query. It has numerous applications, such as e-commerce lookups using user queries [1], audio and video lookups. We have used an approach where natural language can be utilized for image retrieval. State-of-the-art models have already mentioned techniques to predict entities from a set of predefined object categories. These predefined categories limit our utility of learning directly from raw texts.

As our baseline model, we implemented OpenAI's CLIP [14] which learns to project both the images and the text to the same embedding space. Hence text and images can be combined and projected to the same latent space.To fine-tune and assess our model performance, we use Precision@K and Recall@K as our evaluation metric.

We propose a new model which generates captions for given images by stacking recurrent neural networks cells with LSTM architecture. The RNN architecture implements attention mechanism for weighing the relevant features of the input sentence. After generation of the caption, we generate embeddings out of them. When a text query is received, we extract embedding out of it and compute the semantic similarities between embeddings generated from the captions and text query to return the best image. These captions can be utilized for various downstream tasks such as visual question answering, generating meta data for image clustering, image indexing and building applications for visually impaired people. However a major drawback for this model is underlying overhead generated due to the losses being augmented from two models- caption generator and text encoder.

2 Literature Survey

Majority part of research revolves around automatic media captioning. In recent times bottom-up and top-down approaches[9] have been used majorly for image captioning by intricate analysis and multiple stages of reasoning. Here we evaluate attention at different level of objects and other salient image regions. Bottom-up approach uses Faster-RCNN mechanism and top-down approach determines feature weights.

Recently, Google revolutionized image captioning by crafting an architecture [2] comprising a deep convolution network as an encoder and a language model (Recurrent neural network) as a decoder. This work was improved upon by adding attention mechanism in the encoder [3]. We used this work as the first component of our baseline model.

Historically, cross modal learning has been quite challenging. It processes input from multiple modalities separately and then transform them all into a single feature space. The most

challenging part of this approach is aggregating different modalities whilst retaining their individual characteristics into a single latent feature space . The concept that led to formation of cross-modal query retrieval system is Canonical Correlation Analysis that dates to 1936 [4], its mathematical backbone was developed even further back around 1875 [5]. We came across several approaches proposed over the last decade , starting from early fusion [6], which fashions a joint representation of all modalities and then utilizes a single model to learn the correlation, the next approach that outperforms the former one uses a most recent technique uses deep canonical correlation analysis, which uses deep convolution neural [18] network for extracting feature from all the modalities distinctly and uses CCA to map them into single feature space.

Another method for learning cross-modal representations uses the OSCAR(Object-Semantics Aligned Pre-training for Vision-Language Tasks) [11] which constructs a feature triplet consisting of region-based image features, object tags with respect to anchor points and word tokens extracted from the corresponding annotations.

During the past few years, a lot of tech companies have been focusing on cross modal querying. For instance, in 2019, Spotify [19] was working on retrieving relevant music from the library based on videos being played. It was implemented by projecting videos and audio into the same embedding space and using contrastive loss to train the model. Amazon also worked on the field [20] to revamp cross modal recipe retrieval using transformer and self-supervised learning.Different segments of our work are inspired by the recent developments by OpenAI on contrastive learning [14] which outperformed all the previous cross modal retrievals on Flickr30 and COCO datasets.

3 Methodology

3.1 Dataset and Evaluation Metrics

3.1.1 Training Dataset

We required a dataset that contains image and corresponding text(captions). We considered three most popular datasets :

- Flickr8k
- Flickr30K
- Microsoft COCO

However we used **Flickr30K dataset** as we felt Flickr8k would be a really small dataset whereas the COCO dataset which has 90K images, would be really hard to train.

The dataset consists of 31,784 images with each image having 5 captions adding upto approximately 150K captions. We used about 25K images for training, about 3K images for validation and 6K images as a test set

3.2 Evaluation Dataset

For evaluation purposes, we required ground truth labels for Flickr30K images. We came across Flickr30K Entities [13] which contains annotated labels for about 81 different entities in all the images. Additionally, they also contain labels for 200 nouns (such as man, woman, dress, baby etc) and 200 Adjectives (such as young, white, black, blue, little etc). The general format of the annotations are as follows:

[/language_entity#unique_entity_number/entity_name caption_entity_text] non entity text here. where :

- **language_entity**:The language in which the entity is annotated. It currently has 2 different values- EN and FR corresponding to English and French.
- **unique_entity_number**: There are 81 different entities in the annotations. Each entity has a unique entity number

- **entity_name**: Name of the entity
- **caption_entity_text**: The caption text referencing to the entity.

We get the entities from the annotations by extracting *entity*. These labels help us validate the results we obtained. A sample format of how they are present are as follows:

[/EN285147/people A woman] is speaking at **[/EN285148/scene a podium] [/EN285149/scene outdoors]**.

We extracted the entities inside the square brackets **[]** and used it for our performance test.

3.3 Evaluation metrics

For the baseline model, we require two form of evaluation

- **Evaluation for Caption Generation using BLEU**:
  The Bilingual Evaluation Understudy Score, or BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. The range of this evaluation metric is from 0 to 1. We used **NLTK** for this using the **sentence_bleu()** function. There are different types of BLEU score - however, we didn't go for the more complex weighted BLEU score which provides different weights for different n-grams.
- **Image retrieval using Precision@K and Recall@K**:
  For evaluating the image retrieval we use two metrics - Precision@K and Recall@K. Precision at k is the proportion of recommended items in the top-k set that are relevant:

  $$\textbf{Precision@K} = \frac{Number\ of\ relevant\ recommended\ items@k}{Number\ of\ recommended\ items@k}$$

  Recall at k is the proportion of relevant items found in the top-k recommendations.

  $$\textbf{Recall@K} = \frac{Number\ of\ relevant\ recommended\ items@k}{total\ number\ of\ relevantitems}$$

3.4 Baseline Model: CLIP

- **Step 1 Convert Image to Embedding** We used **Resnet50** [23] weights to convert the image to 2048 dimension embedding.
- **Step 2 Convert Text to Embedding** Here we used **DistilBert** weights to convert the captions into an embedding of 768 dimensions.
- **Step 3 Projection Head** We pass the output of step 1 and 2 through 2 linear layers to project them in the same latent space. We achieved superior results by passing the output through a dropout layer and a batch normalization layer. The output of this layer is of shape (batchsize, 256).
- **Step 4 Loss Function** This is where the main crux of CLIP model comes in. The *logits* is computed by finding the dot product of projected embedding of image with that of the text which results in a shape of (256,256). Post that we compute the *target* by computing a dot product between the image embedding and the text embedding. After that we pass it to the cross-entropy loss separately and average upon both the losses.
- **Step 5 Inference** During inference we project the text query to 256 dimension embedding space. After that we use **cosine similarity** to measure the similarity between the projected image embedding and the projected text query embedding. The images corresponding to the top k similar embedding are returned.

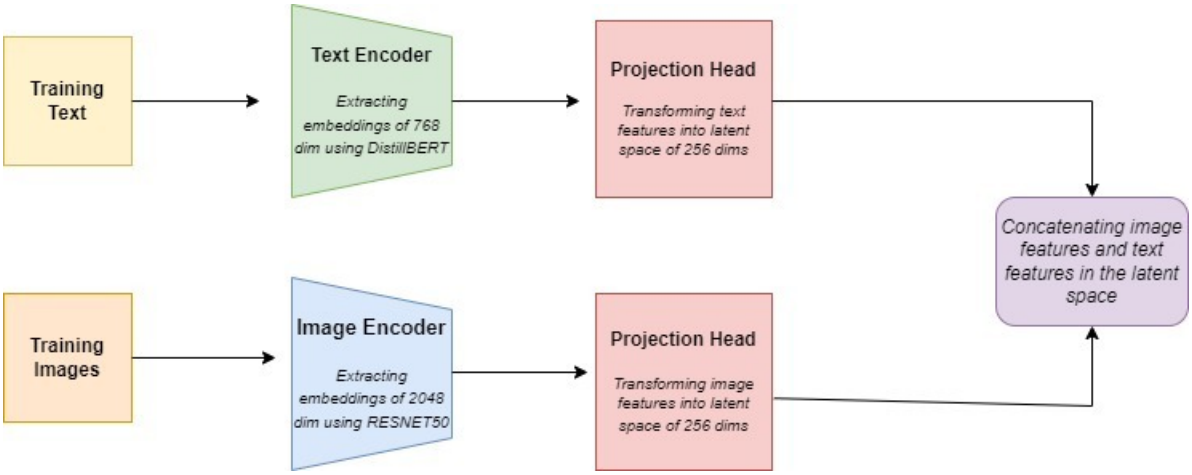

Figure 1: Architecture diagram for CLIP implementation

3.5 Proposed Model

- **Step 1 Image Captioning:** For this we have implemented two approaches inspired from open source solutions for image captioning i.e Show and Tell [2] and , Show, Attend and Tell [3]. We created an encoder-decoder based architecture, which incorporates a vision model in encoder and a text model in decoder. We have used **Resnet101** for this and have evaluated this step using **BLEU score**, which is shown in the results.

- **Step 2 Encode generated caption using HuggingFace** In this step we convert the generated caption over the test set into a 512-length vector using **HuggingFace API over BERT weights.**

- **Step 3 Inference** During inference we converted the query to 512-length vector using the same Bert weights. After that we used **cosine similarity** to measure the similarity between the vectors of test set and the inference query. The images corresponding to the top k similar text are returned.

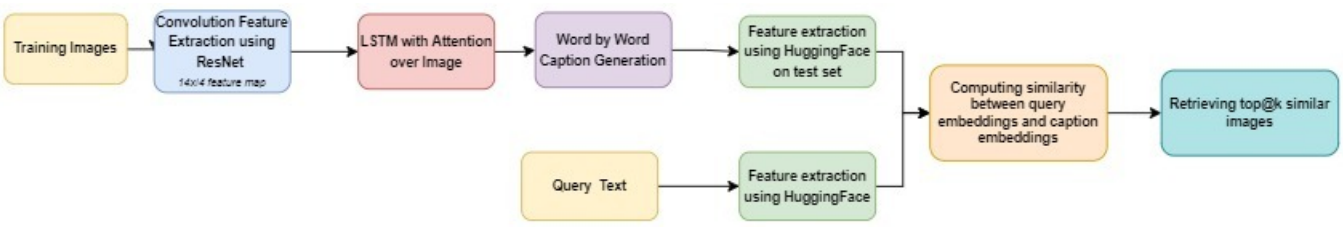

Figure 2: Architecture diagram for CLIP implementation

4 Experimentation and Results

For the CLIP implementation, we tried three different text embedding models- *DistilBERT* [21] and *ALBERT* [22] and *BERT*[15] however, did not see any major performance boost. However we observed a significant increase in training and inference time. For image embeddings, we selected *Resnet50* [23] as per the standard practice for vision tasks. We used unanimous projection heads to fit image and text embedding to a 256 dimension latent space. Note that the activation function used in the projection layer is *GELU* that was utilized in the CLIP implementation. We haven't experimented with different combination of these hyperparameters such as learning rate, text max length, primarily because of lack of computational resources. However, we experimented with different text encoders, batch size, number of epochs and projection dimensions. Table 1 shows our final choice of hyperparameters.

Hyper Parameter	Value
Image Encoder	Resnet
Image Embedding dim	2048
Image Encoder LR	1e-4
Text Encoder	DistilBert and Albert
Text Embedding dims	768
Text encoder lr	1e-5
Projection Head	1
Projection Dims	256
Epochs	3
Batch Size	32
Loss	Cross Entropy

Table 1: Choice of hyperparameters

For our proposed model, we first run an image captioning model based on the implementation of **Show and Tell** [2]. The model does considerably well for image captioning. Table 2 illustrates the best choice of hyperparameters for this model. Figure 3 below also the decreases in loss function over 10 epochs.

Hyper Parameter	Value
Encoder Architecture	CNN (Resnet)
Embedding dimensions	512
Decoder architecture	RNN
Decoder Layers	4
Decoder Units/Layer	256
Batch Size	128
Loss function	Cross entropy
Vocab Size	10000
epochs	10
learning rate(encoder)	1e-5
learning rate(decoder)	5e-4

Table 2: Choice of hyperparameters for image captioning without attention

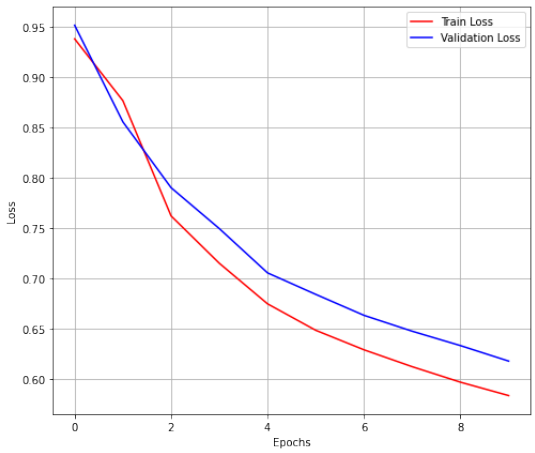

Figure 3: Loss function over epochs

We also tried to improve our image captioning model since performance of the model was strongly dependent on the captioning model. Hence we implemented the Show Attend and Tell model [2] by adding attention to our decoder layer. Table 3 highlights the best set of hyperparameters and while Figure 4 shows the BLEU score:

Hyper Parameter	Value
Encoder Architecture	CNN (Resnet)
Embedding dimensions	2048
Decoder architecture	LSTM
Decoder Layers	3
Decoder Units/Layer	256
Batch Size	80
Loss function	Cross entropy
Attention Dimensions	49
Vocab Size	10000
epochs	10
learning rate(encoder)	1e-5
learning rate(decoder)	5e-4

Table 3: Choice of hyperparameters for image captioning with attention

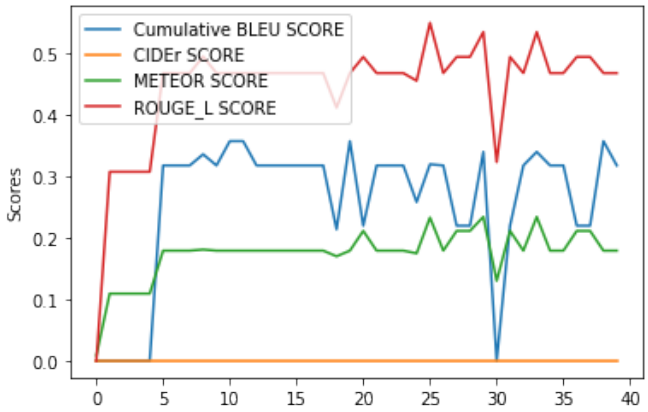

Figure 4: Loss function over epochs

While training our image captioning model, we evaluated the BLEU score of model on the validation set for every epoch. This helped us keep track of whether the model is learning well. Also, we took note of the model performance after every epoch by displaying the generated captions as follows:

Epochs	Caption
1	Explains explains explains explains explains explains explains explains explains
2	A man in a white shirt and a woman in a white shirt are walking down the street
3	A man in a white shirt is standing in front of a group of people
4	A man in a white shirt is standing in front of a crowd of people
5	A man in a white shirt and a woman in a white shirt are walking down the street
6	A man and a woman in a white shirt is standing in front of a crowd of people
7	A group of people in white clothes are walking down the street
8	A group of people in white clothes are near food
9	A group of people in white clothes are near food in the street
10	A group of people in white clothes are near meat in the street

Table 4: Ground truth for image:a group of arabic looking men are selling some kind of meat

Table 5 showcases the BLEU score of our model and compares it with the original implementation. Our results are less significant than mentioned in the paper since we just trained the model for 10 epochs- while the original paper executed it for more then 50 epochs.

Table 5: Comparison of BLEU score

Model	BLEU SCORE	
	BLEU-1	BLEU-2
Original Show and Tell [2]	59.23	39.23
Our implementation Show and Tell	49.23	28.23
Original Show, attend and tell [3]	66.9	43.9
Our implementation Show, attend and Tell	52.8	34.2

Table 6 showcases the result with inference time. We first run simple search query where we address all the entities in the evaluation dataset. We took about 81 entities and query images with the text 'Image of a [Entity name]' where *Entity name* are entities such as people, car, park etc. We observe the model does really well on the simple queries. Most of the entities are recognized correctly- the ones not recognized correctly are the ones with very fewer occurrences during training. For the complex queries we manually wrote about 100 complex queries such as 'image of children playing soccer on field'. In these cases, we do classify the image as a true positive even if the image has most of the characteristics of the query- for example in the previous case even if the image is of children playing in a field, we classify as TP even if the children are playing basketball i.e we have used lenient measures for the purpose of evaluation.

Model	Simple Query				Complex Query				Inference Time
	P@1	P@5	R@1	R@5	P@1	P@5	R@1	R@5	
Baseline Model (CLIP)	92	89	84	79	81	75	79	72	4.30s
Image Caption + Text Embedding	84	77	70	69	71	66	68	61	3.63s
Image Caption(with attention) + Text Embedding	85	80	86	71	74	65	70	63	3.61s

Table 6: Model performance

A sample inference result is given below in the figure below more detailed comparisons on inference can be found in the Appendix section

Figure 5: User query: **Professor teaching in class** result from our proposed model(with attention)

5 Conclusion

Throughout the course of our project, we did numerous experiments for implementation of CLIP and our proposed model. Our conclusion is listed as follow:

- Our baseline model(CLIP) outperforms our proposed architecture. This is mainly because our approach is a concatenation of 2 different model i.e Image captioning and text encoding. During this process the loss by the two different architectures gets added, that results in slightly poor performance of our model.
- While comparing the results between image captioning model with and without attention, it is quite evident that the quality of the images returned depends heavily on the captioning model. A better designed captioning model might further boost the overall performance of our model.
- Our proposed model is about 15% faster then the CLIP model at inference on the same test dataset. Perhaps this is due to the fact that CLIP model projects the test query into an embedding space which might be the bottle neck here.

6 Future Work

While our proposed architecture performs decently, it is still not a viable option in comparison to State of the Art models like CLIP. There are couple of things that can be done to further bolster performance of our model.

- We trained CLIP and the image captioning model for only 3 and 10 epochs respectively. Further training these models with the presence of more computational power would entail better results. This would also indicate we would have to experiment with different set of hyper-parameter for our models.
- Another aspect that needs work is our image caption generation model as it plays a crucial role in our architecture and would demonstrate a significant impact on its performance.
- We are also looking at possible ways to further improve upon the inference time. We believe this could be executed using concepts like Locality Sensitive Hashing(LSH), though it may lead to a significant increase of false negatives.
- We would also want to deploy our model on a website or make an open source API such as HugginFace.
- We want to utilize the captions generated by our image captioning models for other downstream activities such as visual question answering.
- We want to make an attempt to extend our model in the direction of different modalities such as audio and video.

References

[1] MuhammadUmer Anwaa, MartinKleinsteube, "Compositional Learning of Image-Text Query forImage Retrieval", 2021, IEEE Xplorer

[2] Oriol Vinyals, Alexander Toshev, Samy Bengio ,Dumitru Erhan, "Show and Tell: A Neural Image Caption Generator", 2014, Arxiv

[3] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel,YoshuaBengio, "Show,Attend andTell: NeuralImageCaptionGeneration withVisualAttention", 2016, Arxiv

[4] Hotelling, H. (1936). "Relations Between Two Sets of Variates". Biometrika. 28 (3–4): 321–377. doi:10.1093/biomet/28.3-4.321. JSTOR 2333955.

[5] Jordan, C. (1875). "Essai sur la géométrie à n dimensions". Bull. Soc. Math. France. 3: 103.

[6] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In Proceedings of the 13th annual ACM international conference on Multimedia, pages 399–402. ACM, 2005.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever,"Learning Transferable Visual Models From Natural Language Supervision", 2020

[8] Filip Radlinski, Nick Craswell, "Comparing the sensitivity of information retrieval metrics",SIGIR'10

[9] Peter Anderson, Xuedong He, Chris Buehler , Damien Teney, Mark Johnson,Stephen Gould, Lei Zhang , "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering",2018, Arxiv.

[10] Xiaopeng Lu , Tiancheng Zhao, Kyusong Lee ,"VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words",2021 , Arxiv .

[11] Xiujun Li, Xi Yin , Chunyuan Li , Pengchuan Zhang , Xiaowei Hu , Lei Zhang , Lijuan Wang , Houdong Hu , Li Dong , Furu Wei , Yejin Choi , and Jianfeng Gao , "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks" , 2020, Arxiv .

[12] Chunxiao Liu , Zhendong Mao , Tianzhu Zhang , Hongtao Xie , Bin Wang, Yongdong Zhang , "Graph Structured Network for Image-Text Matching" , 2020 , Arxiv.

[13] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier and Svetlana Lazebnik, "Flickr30K Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models", ICCV, 2015

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, "Learning Transferable Visual Models From Natural Language Supervision", 2021, Arxiv.

[15] Jacob Devlin, Ming-Wei, Chang Kenton, Lee Kristina and Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2019, Arxiv.

[16] Jeffrey Pennington, Richard Socher, Christopher D. Manning, "GloVe: Global Vectors for Word Representation", 2014, Stanford.

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov, "Enriching Word Vectors with Subword Information", 2017, Arxiv.

[18] Galen Andrew, Raman Arora, Jeff Bilmes, Karen Livescu, "Deep Canonical Correlation Analysis", 2013, International Conference on Machine Learning.

[19] Bochen Li, Aparna Kumar, "Query By video: cross modal music retrieval", 2019, ICCV.

[20] Amaia Salvador, Erhan Gundogdu, Loris Bazzani ,Michael Donoser , "Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning", 2021, Arxiv.

[21] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter",2019, Arxiv

[22] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations",2020, Arxiv

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition",2015, Arxiv.

Appendices

We provide additional details for comparing the various models used by us on inference images to further understand the approach in depth. We look at a user query over the same test set and compare the results in 3 different model:

A Inference for CLIP

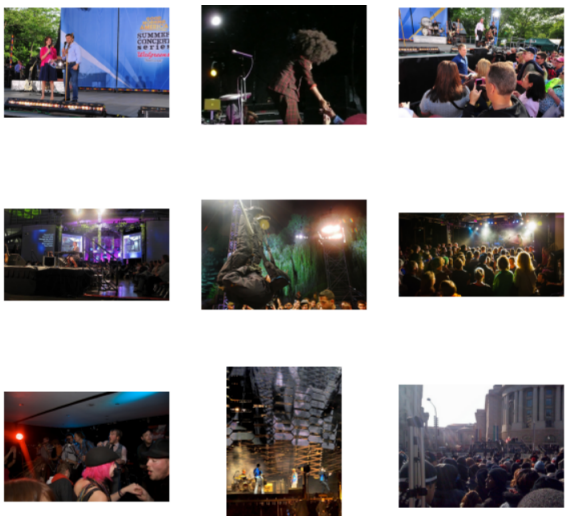

Figure 6: User query: **People in concert** result from our baseline model(CLIP)

B Inference for proposed model(Image captioning(With attention) + Text Embedding)

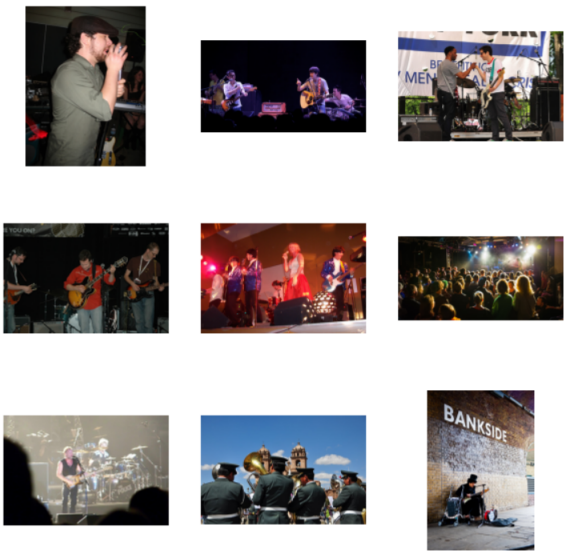

Figure 7: User query: **People in concert** result from our proposed model(with attention)

C Inference for proposed model(Image captioning(Without attention) + Text Embedding)

Figure 8: User query: **People in concert** result from our proposed model(without attention)

Figure 6,7,8 shows the inference results of one of the user query over the same dataset. It is noteworthy to see that the images returned by the CLIP are extremely relevant and useful. The proposed method which is implemented using attention mechanism also has great images returned. However Figure 8 shows that when the image captioning model is not that good(without attention), the quality of returned image also suffers. As a result, the highlighted black bounding boxes corresponds to the irrelevant images.