Pointing out an Image: Mouse traces and their influence on a Multimodal Image Retrieval Task

Riccardo Falsini[1], **Xiangyu Lu**[1], **Zixuan Xu**[1]

[1]MSc in IT and Cognition, University of Copenhagen
qzt393@alumni.ku.dk, xilu@di.ku.dk, zixu@di.ku.dk

Abstract

This paper proposes a multimodal model for image retrieval tasks that incorporates both text and mouse trace modalities. The full model is compared to a text-only baseline model, both are trained and evaluated on portions of the *Localized Narratives* dataset using only or both caption-text and corresponding mouse trace as input queries and returning recommended images as targets. Through the application of *Welch's t-test*, the study reveals a significant performance improvement over the base model, with top-k recall scores showing a noticeable increase of 3.5% to 6.5%. These results underscore the beneficial impact of integrating mouse traces in text-image retrieval tasks, highlighting its potential for enhancing image retrieval accuracy and effectiveness.

1 Introduction

Machine learning models and artificial intelligence have rapidly become an integral part of our modern world, often discreetly embedding themselves within various aspects of our daily lives. Among the different subsets of machine learning, one that has rapidly grown in importance, playing a crucial role in modern technology is *image retrieval*. Image retrieval is a field focused on the ability to quickly and accurately retrieve relevant images from a large database, based on specific search criteria.

The concept of image retrieval stems from the increasing availability of vast image datasets and the need for efficient methods to search and navigate through them. With the exponential growth of digital images and social media platforms on the internet, there is an increasing demand for effective image retrieval systems that can enable users to find images related to specific objects, scenes or concepts - making them invaluable for tasks such as reverse image search, content-based image retrieval, and organization of large image repositories.

Over time, the evolution of image retrieval has expanded beyond the traditional single-modality approach to incorporate multiple modalities, such as text, audio, and even user prompts or feedback ([1], [2]). This shift has significantly enhanced the performance and capabilities of image retrieval systems. The evolution from content-based image retrieval to multimodal retrieval stems from the recognition that information from different modalities can provide richer representations of images, leading to more accurate and comprehensive search results. For example, by combining complementary textual information with visual features, multimodal retrieval systems can capture both the visual and semantic context of images, enabling more precise and context-aware searches.

Throughout this paper we propose a model capable of performing well on a variation of image-retrieval task (*cross-modal image retrieval*), and explore whether the addition of another modality

to such a model can improve its performance. To that end, we will first go over traditional approaches to various image-retrieval tasks and then expand upon our reasoning behind the additional modality we seek to integrate in our model: *mouse traces*.

1.1 Related Works

Content-based image retrieval.

In content-based image retrieval (CBIR) [3, 4], the most similar image from a database is retrieved given a query image. During the early stages of research, a variety of low-level hand-crafted descriptors (such as color, shape, and texture) played a crucial role in enabling the extraction of relevant features from images [5]. However, such manual approaches were both time-consuming and required significant amounts of human effort. Early methods to address this include: SIFT methods [6] which automatically extract distinctive local features from images, and distance metric learning techniques, such as kernel-based and rank-based learning [7, 8]. However, both have their respective limitations, being high computational complexity and the inability to effectively model non-linear data.

Over the past decade, rapid advancements in deep learning techniques have revolutionized CBIR, becoming state-of-the-art. Instead of relying on hand-crafted features to represent image data, deep learning methods employ a feature extraction approach, which leads to more robust representations. Among these methods, CNN [9] based approaches have been extensively explored [10, 11]. CNNs learn features by downsampling the original image data into a lower-dimensional feature space, which is then utilized for tasks, such as image classification.

Cross-modal text-image retrieval.

Text-Image retrieval refers to retrieving relevant images based on textual queries or vice-versa. Early on, text-based image retrieval methods mainly relied on techniques like keyword matching [12], vector space models [13], and text retrieval algorithms to find images semantically related to the provided text.

Two significant milestones in text-image retrieval are the introduction of cross-model embedding techniques [14], which aim to map both images and text into a shared embedding space, and the use of attention mechanisms [15], which allows models to focus on specific regions or words that are most relevant to the task.

The field of cross-modal image retrieval has garnered significant research interest due to its wide range of real-world applications, coupled with advancements in transformer-based models [16] and pretrained language models [17]. Building upon this, recent approaches have placed more emphasis on data selection [18] and model architecture [19], shifting towards fine-tuning a single cross-modal pretrained model [20], which leverages feature embeddings from multiple modalities as input. This area presents a promising avenue for search systems, enabling the retrieval of relevant images based on descriptive text data, or vice versa.

In light of large transformer models demanding ever-extensive training, researchers have embarked on investigating the incorporation of additional modalities to enhance model performance [1] instead. Fascinating breakthroughs, such as Microsoft's KOSMOS-1 [2], have demonstrated how multimodal prompting can elevate performance in various tasks that were previously thought to have reached a performance ceiling.

Mouse Traces as (spatial information and) 'Attention'

Expanding beyond the integration of image and text, studies have also delved into the impact of virtual "attention" on model enhancement. Transformer models utilizing *multi-head attention mechanisms*, have revolutionized the field, pushing performance boundaries since their introduction in *Attention is all you need* by Vaswani et al. [16] Recognizing the impact of "virtual" attention on modeling, researchers have also sought to explore representations of human 'attention' and how these could augment modeling approaches to yield potential improvements.

Past studies have explored the viability of mouse movements as representations of human "attention". Egner et al. [21] found statistically significant correlations between measures of visual

attention generated through mouse-clicks (mcAT) and those using eye-movement tracking (emAT) during a stimulus viewing task. Computer mouse movements were also found to be a good predictor for 'mind-wandering' - loss of attention during operation span tasks by Dias da Silva et al. 2020 [22]. Studies have also investigated how the addition of mouse movement features can augment modeling approaches, Arapakis et al. [23] found that both recurrent and convolutional neural network models can learn to utilize raw mouse cursor data to predict user attention on search engine pages.

1.2 Research Question

Considering these studies, we have reason to believe that the inclusion of a "mouse trace" modality could significantly benefit cross-modal image retrieval models, as they would not only encapsulate spatial information, but also some level of 'attentional' importance.
Our hypotheses are the following:

- H0 (null): The addition of mouse-trace features **will not** improve the top-k recall performance of a neural network model on a cross-modal image-retrieval task.
- H1 (alternative): The addition of mouse-trace features **will** improve the top-k recall performance of a neural network model on a cross-modal image-retrieval task.

To test this, we plan on constructing and evaluating two models: A 'Base' model trained solely on images and text, and another 'Full' model which utilizes the additional mouse-trace modality, allowing us to explore whether the addition of such modalities can improve the capabilities of image retrieval systems.

2 Data

Datasets	**Subsets**	**Item**	**Size**
Flickr30K	Full set	Image	31,783
LocalizedNarratives	Training set	Caption/Trace	29,783
	Validation set	Caption/Trace	1000

Table 1. Summary of Datasets Used

2.1 Dataset

In this paper, we use *Flickr30K* [24] as our image database. *Flickr30K* is a widely-used benchmark and valuable multi-modal computer vision dataset for image captioning tasks. It consists of over 30,000 images sourced from Flickr's photo sharing platform, each accompanied with 1-5 English descriptions. Our texts and mouse traces come from the corresponding *Flickr30K* subset of the *Localized Narratives* [25] dataset (Fig 1). To generate this data -which combines both visual and linguistic elements- Pont et al. had annotators verbally describe an image while simultaneously moving their mouse cursor over the described region. Each word is precisely localized by synchronizing both voice recording and mouse movement. While this dataset also includes audio for each description as another potential modality to explore, this is not utilized in our paper.

As we seek, to integrate both textual information and mouse trace data as encoded inputs to our model, for our approach, we opted to use novel captions from *Localized Narratives* as textual data over the *Flickr30k* captions. In addition to being represented as complete texts, these captions are also organized into utterances accompanied by their corresponding time segment, which is crucial in order to integrate mouse trace data - as we discuss below.

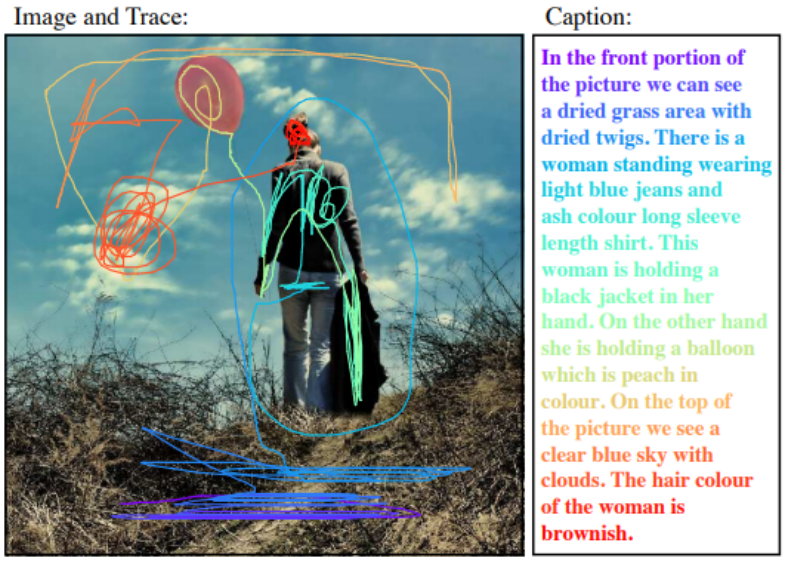


Figure 1. Example from '*Localized Narratives*' dataset [25]

2.2 Creating Bounding Boxes for Mouse Traces

Each utterance in the caption of the *Localized Narratives* dataset is associated with a specific time segment that indicates when the annotator spoke the utterance. This trace data comprises of a sampled sequence of xy coordinates, followed by a timestamp that captures the mouse's location at the given time.

To leverage the mouse trace data, we adopt a method proposed by Soravit et al. [26]. In their study, they process the entire dataset and convert the mouse trace segments for each utterance into a bounding box representation, denoted as a 5-dimensional vector $[xmin, xmax, ymin, ymax, area]$. The xy coordinates within the vector indicate the minimum or maximum values for the x and y locations. The rationale behind this approach is that this bounding box, derived from mouse trace data, represents the location and size of an object mentioned in the caption.

An issue the authors observed, is that the mouse trace around the time interval (t_1, t_2) may correspond to only part of an object described by the utterance. To address this issue, they introduced temporal padding to expand the original time interval, updating it to $(t_1 - t_p, t_2 + t_p)$ for a particular utterance. This further expands the bounding box by a spatial padding factor s_p to ensure it adequately covers the entire object within the image (as shown in Fig 2). It is important to note that both t_p and s_p serve as hyperparameters for our model.

3 Proposed Model

This section provides a comprehensive overview of our proposed model for the multimodal image retrieval task. Our model comprises of three primary components: an image branch, a text branch, and a feature fusion layer. Initially, we adopt a 'Base' model (Fig. 3a) architecture inspired by [26], which serves as the baseline for our research. Subsequently, we extend our base model by incorporating an additional mouse trace branch (creating our 'Full' model Fig. 3b) to investigate our research question.

3.1 Base Cross-modal Retrieval Model

We employ the default training setup for cross-modal retrieval, transforming it into a text-image matching task using contrastive loss. Let $\mathbf{X} = [x_1, x_2, ..., x_N]$ denote the images within a

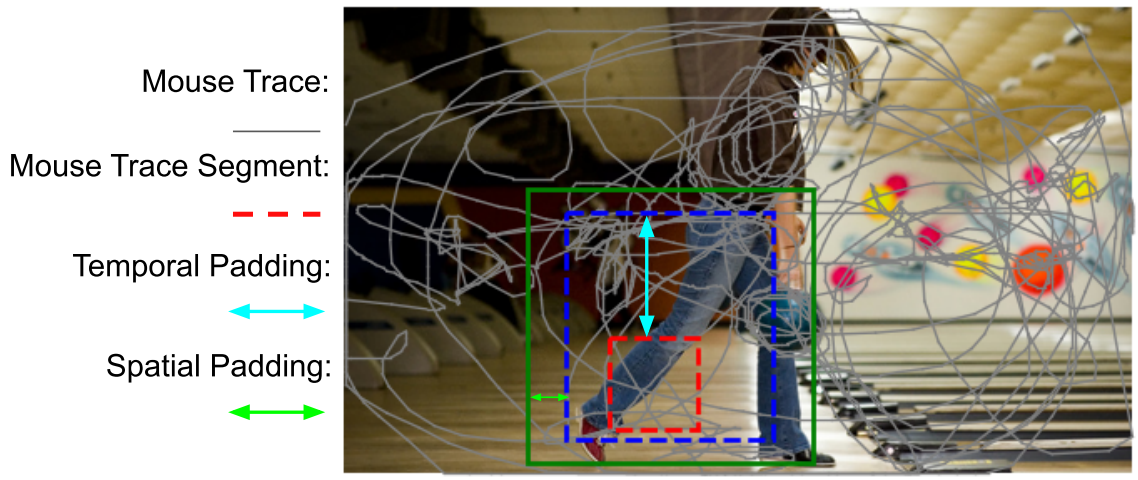


Figure 2. Expanding bounding box for utterance 'jeans': We first expand the original bounding box (red) by adding temporal padding (blue), and then we add extra spatial padding (green).

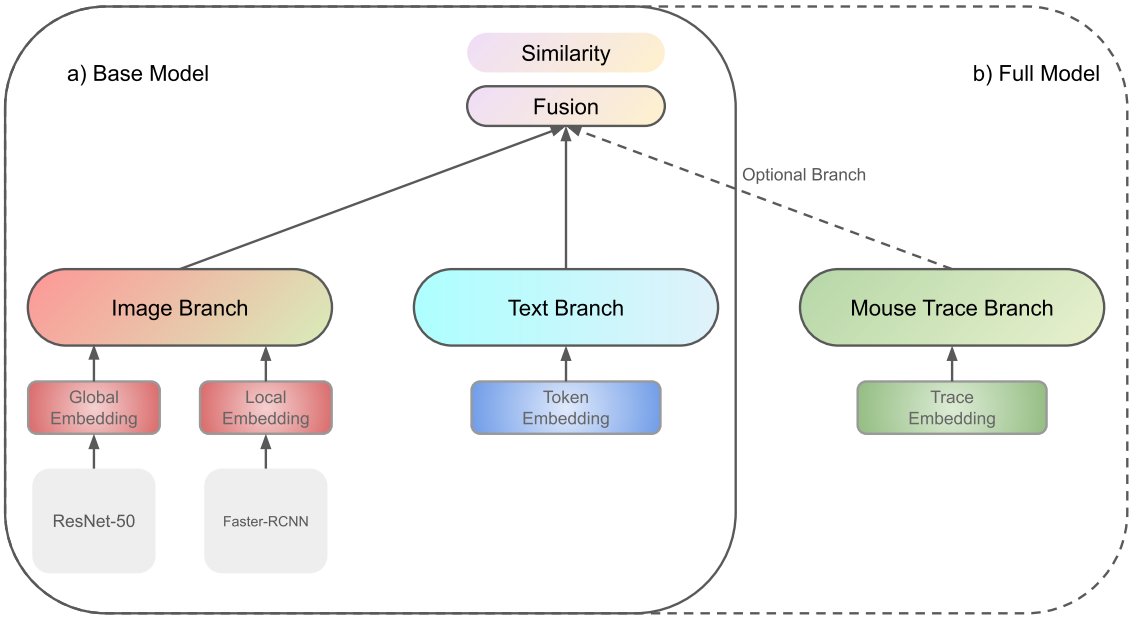


Figure 3. Model: Our full model (b) consists of a base model (a) which only includes an image and a text branch and an optional mouse trace branch. Each branch embeds its corresponding modality and sends the output embedding into the fusion layer.

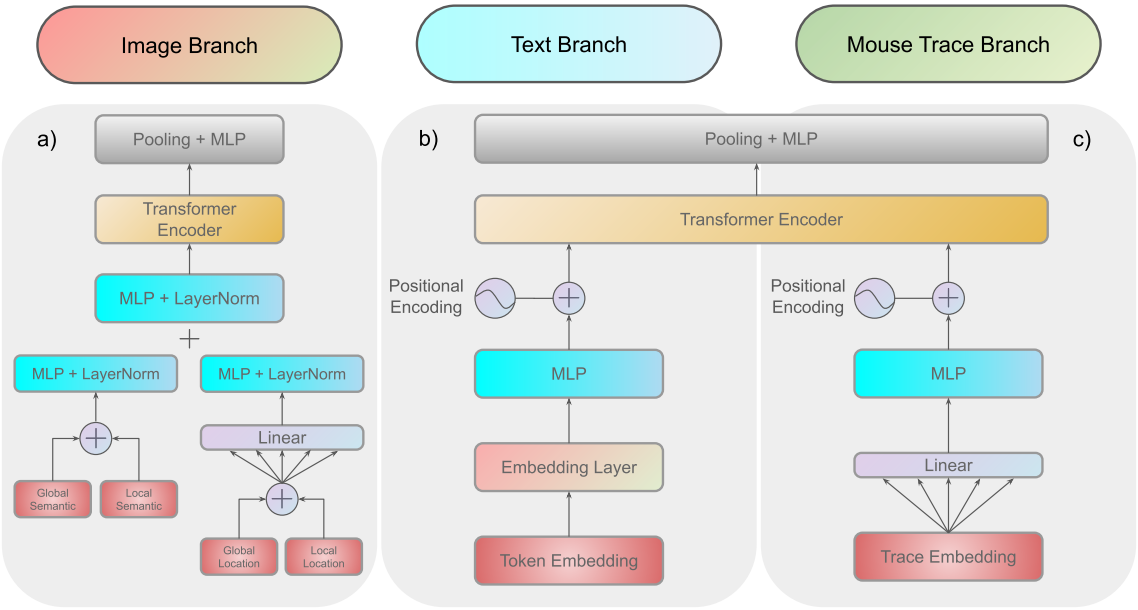


Figure 4. Branches: In the image branch (a), semantic features and location features are combined to form a location-aware semantic embedding. In the text branch (b), token embedding is embedded using an MLP layer. Instead of using random embedding, the mouse trace branch (c) adopts a linear transformation for each element of the trace embedding. Subsequently, all three branches are followed by a transformer encoder and a pooler. Notably, both the text and trace branches share the same transformer encoder, achieved through the concatenation of two output embeddings as inputs.

batch, and $\mathbf{Y} = [y_1, y_2, ..., y_N]$ denote the corresponding texts, where N represents the batch size (paired data share the same index within a batch).

Our base model is trained with the objective of maximizing the similarity score between paired data:

$$f(\mathbf{X}, \mathbf{Y}) = s(g(\mathbf{X}), h(\mathbf{Y}))$$

Here, f, g, h, and s represent our model, image branch, text branch, and similarity function, respectively. Each branch within our model learns the embedding for its respective modality. These embeddings are then used to calculate the similarity score, using cosine similarity.

Each branch contains a feature extractor followed by a vanilla transformer encoder [16]. On the top of the transformer, we adopt an average pooling layer with a multi-layer perceptron. The subsequent sections will provide a detailed explanation of each of these components.

Image Branch. For our image branch (Fig. 4a), we use a feature extractor consisting of two pretrained models: ResNet-50 [27], trained on the ImageNet dataset [28], and Faster-RCNN [29], trained on the COCO dataset [30]. The ResNet-50 backbone is used to extract global semantic features, while Faster-RCNN, extracts the bounding boxes and local semantic features of the top 16 objects detected in an image. To handle scenarios where the number of detected objects is less than 16, we apply padding.

Next, we concatenate the global and local semantic features and transform the bounding box of each object into a 5-dimensional location feature of $[xmin, xmax, ymin, ymax, area]$. Additionally, a special location vector of $[0, 1, 0, 1, 1]$ is concatenated with the local location features of each image, corresponding to the global semantic feature. We then apply linear transformation for each element in the location features which projects it to a high-dimensional space. Subsequently, both the semantic and location features pass through feed-forward networks and are added, yielding a location-aware semantic embedding. Prior to being fed into the transformer, an additional 2-layer MLP is used to introduce more non-linearity.

The resulting embedding is then input to a 6-layer transformer, enabling free interaction be-

tween all global and local features via multi-head attention layers. Instead of taking the output embedding from the [CLS] token, we apply an average pooling layer over all the transformer outputs.

Text Branch. The text branch of our model adopts a simpler architecture. We choose WordPiece [17] as our text tokenizer specifically designed for subwords. After applying a random embedding layer, the token embeddings are transformed using a 2-layer MLP. Additionally, we incorporate positional embeddings using the implementation described in BERT to produce necessary positional information. Similar to the image branch, the text embedding then proceeds through a transformer encoder. Subsequently, an additional pooling layer and feed-forward layer are applied to the resulting outputs (Fig. 4b).

Feature Fusion. The outputs from the two branches correspond to the embeddings of the query text and the target image, respectively. To calculate the similarity between these embeddings, we employ the cosine similarity score. Hence, the feature fusion layer can be expressed as:

$$\mathbf{S} = s(g(\mathbf{X}), h(\mathbf{Y})) = \frac{g(\mathbf{X}) \cdot h(\mathbf{Y})}{\|g(\mathbf{X})\| \, \|h(\mathbf{Y})\|}$$

During implementation, we begin by applying L2-normalization to each sample in both embeddings, followed by computing the dot product between them. The resulting similarity matrix $\mathbf{S}$ comprises cosine similarity scores for all text-image pairs within a batch.

3.2 Bringing in the Mouse Trace Modality

To address our research question in this paper, we aim to integrate mouse traces into our base model. In this section, we will elaborate on the embedding process for the bounding boxes generated for each trace segment, as well as how we fuse them with text embeddings.

Mouse trace branch. Similar to the image branch, we process each element in the 5D bounding box vector by passing it through a linear layer to project it into an embedding space. Subsequently, we concatenate all five projections and feed them through another MLP. As each trace segment is associated with a corresponding timestamp, we adopt the same positional embedding used for text embeddings to represent the trace embeddings (Fig. 4c).

The resulting trace embeddings are then concatenated with the text embeddings derived from the text branch. This combined representation is subsequently inputted into the transformer, which in our base model, is solely employed for processing text data. As a result, we train two separate models from scratch (Fig. 3): one with the mouse trace branch and the other without it. The model with the mouse trace branch becomes ($\mathbf{Z}$ denotes mouse trace) :

$$f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = s(g(\mathbf{X}), h(\mathbf{Y}, \mathbf{Z}))$$

3.3 Contrastive Loss Function

For the loss function, we adopt the symmetric cross-entropy loss from CLIP [31], which treats all incorrect pairs within a batch as negative samples. This loss function was initially introduced as the multi-pair matching loss [32] and later known as InfoNCE loss designed for contrastive learning tasks [33]. The loss function can be represented as (t denotes a learnable temperature parameter):

$$\mathbf{H} = \mathbf{e}^t \times \mathbf{S}$$

$$L_{image \to text} = -\frac{1}{N} \sum_{i}^{N} \log \frac{\exp(\mathbf{H}_{ii})}{\sum_{j=1}^{N} \exp(\mathbf{H}_{ij})}$$

$$L_{text \to image} = -\frac{1}{N} \sum_{i}^{N} \log \frac{\exp(\mathbf{H}_{ii})}{\sum_{j=1}^{N} \exp(\mathbf{H}_{ji})}$$

$$L = \frac{1}{2}(L_{image \to text} + L_{text \to image})$$

4 Experiments

Training settings. In our experimental setup, we trained our models on four Nvidia T4 GPUs provided by UCloud. The Adam optimizer was utilized for optimization purposes. Each training session incorporated a 20-epoch warm-up phase, followed by the learning rate decayed every 25 epochs using a decay rate of 0.95.

To ensure a thorough exploration of hyperparameter settings, we conducted extensive analysis and determined that a learning rate of 0.00005 and a batch size of 128 yielded optimal results. We ended the training process when the number of steps reaches 70k. In order to address the challenge of non-converging loss, we adopted gradient accumulation every 32 steps. This approach effectively leads to a smoother loss curve. As a result of this gradient accumulation configuration, we evaluated our model at intervals of 32 steps. Moreover, we permute the 16 local features extracted from Faster-RCNN.

For additional details regarding the model implementation, please refer to Appendix A.

Evaluation. To evaluate our models we compare their average *top-k recall* scores. Top-k recall is a performance metric which measures the proportion of relevant images that are successfully retrieved within the model's top "k" (1, 5 or 10) recommendations. For each query within a 128-size batch of our validation dataset (using either text only, or both text and mouse traces), our models return the top-k images which the model believes best fit the input query.To calculate recall scores, we count the number of times the correct image is retrieved within the top "k" recommended images and divide this count by the total number of queries made. We calculate this recall score 5 times, on 5 different batches, in order to retrieve an average recall score for each "k". As an example, if the top-5 recall is 0.2, on average the model retrieves the correct image 20% of the time in its top 5 recommendations.

In order to determine whether there is a statistically significant difference between the performance of our two models, we utilize two independent samples *Welch's t-test* (as we don't assume equal population variance in our recall scores), using d.o.f=4 and α=0.001.

5 Results

Model	**Avg.(Std.) Top-k Recall**		
	k=1	k=5	k=10
Text-Only model	0.089(0.003)	0.366(0.008)	0.555(0.004)
Text-Trace model	**0.125**(0.009)	**0.425**(0.009)	**0.620**(0.009)
T-statistic	-7.619	-9.789	-20.37
P-value	7.66e-04	1.19e-05	3.62e-08

Table 2. Summary of evaluation results. Displays avg. top-k recall scores and standard deviations (at k=1,5,10), for both our Base (text-only) and Full (text+trace) models. The table also summarizes the results of our *Welch's t-test* significance testing, displaying both the t-statistic and p-value found when comparing the recall scores of the two models at each 'k' (**Bold** indicates best performance metrics).

.

In Figure 5, we can see example results of the two models given the same query, this gives us a qualitative idea of how the models perform, with the Full model retrieving the correct image as its first recommendation, rather than second. The results of our evaluation are summarized by Table 2. Looking at the average recall scores, we can observe that the Full model performs better than the Base model consistently across all top-k categories. At k = 1, the mean recall score for the Base model was 0.089 (SD = 0.003), whereas the mean for the Full model was 0.125 (SD = 0.009).

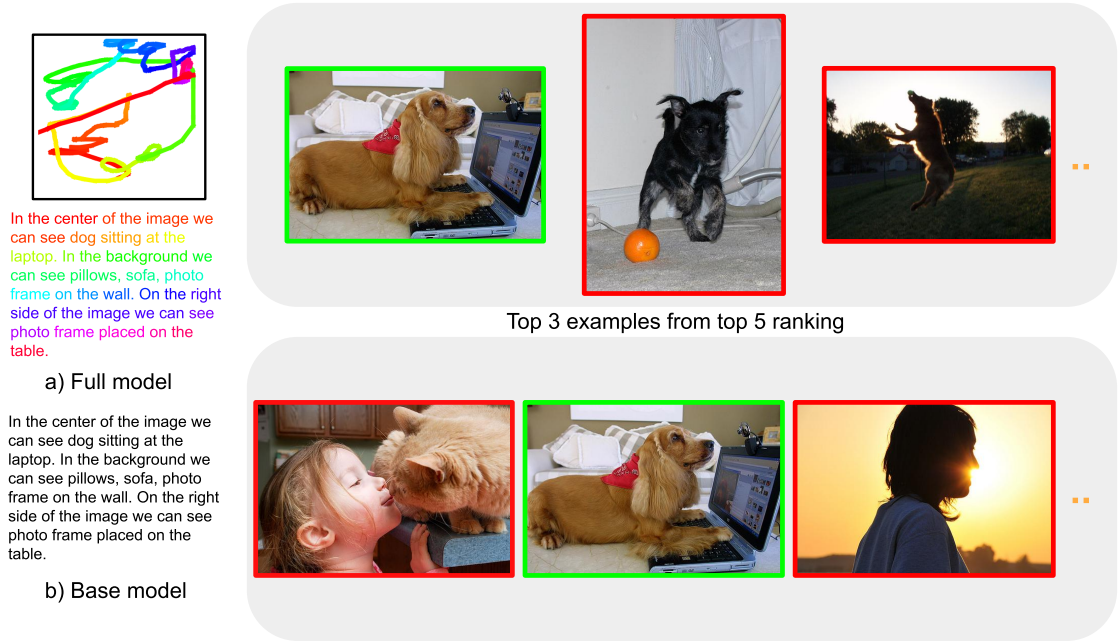


Figure 5. Example Top-k Results from the Two Models: Comparison between the top 3 results of our base and full models on one query. The additional mouse trace adds supplementary location information to the query.

A *Welch's two-samples t-test* showed that this difference was statistically significant, returning a t-statistic of t(4) = -7.619, with a $p < 0.001$ (+3.5%). Similar results are seen for recall at other k-values as well, for k=5, the Full model (μ=0.425 and SD=0.009), significantly outperforms the Base model (μ=0.366, SD=0.008), with t(4)=-9.789, and $p < 0.001$ (+5.9%). And again for k=10, the Full model μ=0.620, SD=0.009, performs significantly better than the Base model μ=0.555, SD=0.004 with t(4)=-20.37, $p < 0.001$ (+6.5% increase).

6 Discussion

The purpose of this study was to explore our hypothesis that the inclusion of a mouse-trace modality could improve a text-only model's performance on a cross-modal image retrieval task. In our results, we found that our Full (text+trace) model significantly outperformed our Base (text-only) model across all our top-k recall evaluations. Based on these results, we can reject the null hypothesis and accept our alternate hypothesis (H1) which suggests that the addition of mouse-trace features in our Full model significantly improves the performance of a neural network model (such as our text-only Base model) in a cross-modal image-retrieval task.

Our reasoning for why this is the case is the following: In image retrieval tasks, textual captions primarily emphasize content features, potentially overlooking the positional information present in the images, such as compositional or global relationships in images. While some positional characteristics can be described using specific prepositions like "under", "in front of", and so on, capturing the underlying logic solely through individual pairs of text and image can be challenging.

The addition of mouse-trace features in the Full model addresses this limitation by providing supplemental spatial information. Mouse traces can implicitly capture user attention patterns, revealing insights into the latent logic and spatial relationships within the images. By incorporating both mouse traces and textual information, the Full model is better equipped to learn and leverage this additional positional information, leading to improved performance. The inclusion of mouse-trace features may also contribute to a more comprehensive representation of the images in the Full model. Mouse traces may capture fine-grained details, such as attention patterns, which

might aid in better encoding the images' distinctive features. This enriched feature representation could enable the model to discriminate more effectively between relevant and irrelevant images during the retrieval process.

Furthermore, mouse traces may indirectly convey semantic information about the images that may not be explicitly present in the textual captions. User interactions can reflect the user's understanding of the image's meaning, salient features, or desired attributes. The Full model can capture these implicit semantics from mouse traces and utilize them to improve the relevance of the retrieved images.

7 Limitations and Future works

Although the addition of the mouse trace modality improves the performance of the text image retrieval model, there are still some limitations and potentially valuable future works to consider.

First, the limited availability of computational resources constrained the training time of our model. More abundant resources and extended training period may lead to improved performance. By allowing the models to train for a longer duration, they could potentially learn more intricate patterns, especially since our validation recall curve (as seen in Appendix B) was still slightly increasing by the end of training.

Second, The use of mouse trace data requires the availability of such data during the image retrieval process. In practical scenarios, collecting mouse trace data from users may not always be feasible or accessible. Limitations in data availability can restrict the applicability of mouse trace modalities, particularly in situations where capturing user interactions is challenging or not permitted. Additionally, processing and analyzing mouse trace data alongside textual and visual information requires additional computational resources which may impact the efficiency and scalability of the model.

Lastly, our study did not incorporate the audio modality provided by the *Localized Narratives* dataset. Future work could focus on integrating audio information into existing multimodal frameworks. Audio cues offer valuable contextual details that can provide a more comprehensive understanding of multimedia content. Exploring methods to effectively integrate audio modalities into our model could result in improved retrieval performance.

8 Conclusion

In this paper, we propose a multimodal model for image retrieval tasks that incorporates both text and mouse trace inputs and compare it to a text-only baseline. Through statistical analysis using *Welch's t-test*, we show that the introduction of the mouse trace modality significantly improves the performance of the base model with a noticeable increase of 3.5% to 6.5% in various top-k recall scores. These findings highlight the positive impact that integrating an additional modality such as mouse traces can have on existing models for text-image retrieval tasks.

References

[1] X. Wang, G. Chen, G. Qian, P. Gao, X.-Y. Wei, Y. Wang, Y. Tian, and W. Gao, "Large-scale multi-modal pre-trained models: A comprehensive survey," *arXiv preprint arXiv:2302.10035*, 2023.

[2] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu *et al.*, "Language is not all you need: Aligning perception with language models," *arXiv preprint arXiv:2302.14045*, 2023.

[3] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2687–2704, 2021.

[4] M. M. B. Ismail, "A survey on content-based image retrieval," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 5, pp. 159–170, 2017.

[5] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[7] H. Chang and D.-Y. Yeung, "Kernel-based distance metric learning for content-based image retrieval," *Image and Vision Computing*, vol. 25, no. 5, pp. 695–703, 2007.

[8] J.-E. Lee, R. Jin, and A. K. Jain, "Rank-based distance metric learning: An application to image retrieval," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[10] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 157–166.

[11] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2017.

[12] Y. Deng, B. Manjunath, C. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Transactions on image processing*, vol. 10, no. 1, pp. 140–147, 2001.

[13] A. Vinokourov, D. R. Hardoon, and J. Shawe-Taylor, "Learning the semantics of multimedia content with application to web image retrieval and classification," 2003.

[14] Z. Wang, X. Liu, H. Li, L. Sheng, J. Yan, X. Wang, and J. Shao, "Camp: Cross-modal adaptive message passing for text-image retrieval," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5764–5773.

[15] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] Y. Wu, S. Wang, G. Song, and Q. Huang, "Learning fragment self-attention embeddings for image-text matching," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 2088–2096.

[19] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *Advances in neural information processing systems*, vol. 34, pp. 9694–9705, 2021.

[20] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[21] S. Egner, S. Reimann, R. Hoeger, and W. H. Zangemeister, "Attention and information acquisition: Comparison of mouse-click with eye-movement attention tracking," *Journal of Eye Movement Research*, vol. 11, no. 6, 2018.

[22] M. D. da Silva and M. Postma, "Wandering minds, wandering mice: Computer mouse tracking as a method to detect mind wandering," *Computers in Human Behavior*, vol. 112, p. 106453, 2020.

[23] I. Arapakis and L. A. Leiva, "Learning efficient representations of mouse movements to predict user attention," pp. 1309–1318, 2020.

[24] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[25] J. Pont-Tuset, J. Uijlings, S. Changpinyo, R. Soricut, and V. Ferrari, "Connecting vision and language with localized narratives," pp. 647–664, 2020.

[26] S. Changpinyo, J. Pont-Tuset, V. Ferrari, and R. Soricut, "Telling the what while pointing to the where: Multimodal queries for image retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 136–12 146.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[32] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

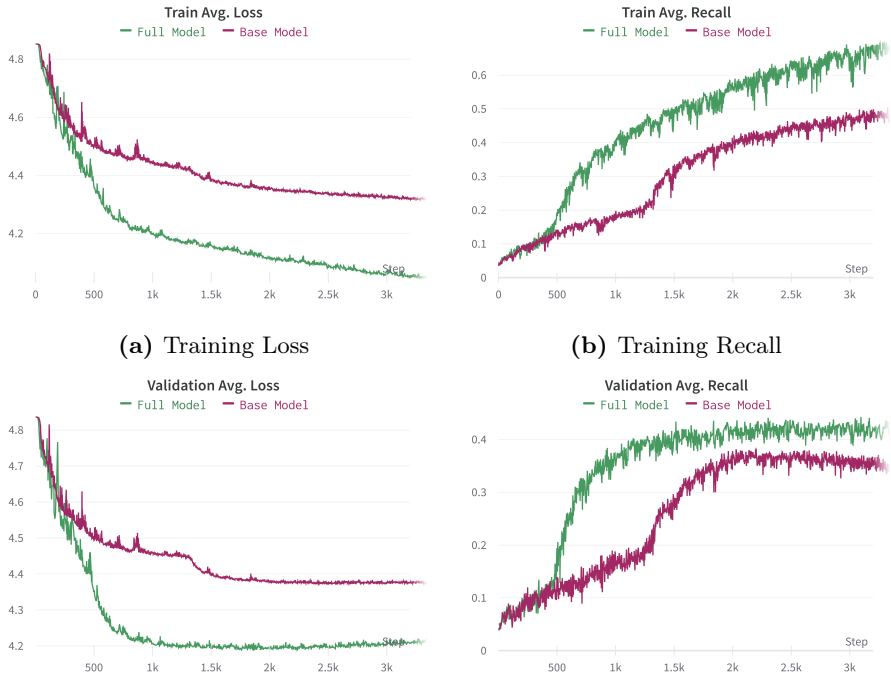
Appendix A

Model details. The extraction of image semantic features was performed by leveraging a pre-trained ResNet-50 backbone, originally trained on ImageNet. Furthermore, our object detection process employed a pre-trained Faster-RCNN with a ResNet-50 backbone, trained specifically on COCO. Each global or local feature was represented as a 2048D vector, subsequently concatenated to form a multi-scale semantic embedding. In the case of location features, we concatenated the local location vectors with a unique global location vector (refer to sec.3 for further details). To alleviate GPU memory usage, we adopted a strategy of pre-storing semantic and location features for both the training and validation sets.

In both branches of our model, MLP consists of two fully-connected layers with a hidden size of 512 followed by a ReLU activation function, and a Dropout layer with a dropout rate of 0.3. Regarding the bounding boxes features, we concatenated all five 512-dimensional projections, corresponding to the components $(x_{min}, x_{max}, y_{min}, y_{max}, area)$, into a 2560-dimensional embedding. Subsequently, we subsampled this embedding to a 512-dimensional representation using an MLP.

Two transformers in both branches adopt the vanilla encoder architecture with 6 layers and 8 heads. The hidden size and filter size were set to 1024, and 4096, respectively. Our code is available at https://github.com/X1angyuLu/Pointing-out-an-Image

Appendix B



(a) Training Loss **(b)** Training Recall

(c) Validation Loss **(d)** Validation Recall

Figure 6. Training Figures from Weights & Biases