Challenges and Opportunities in the Use of CLIP: A Comparative Study for Generic, Medical, and Fashion Contexts

Francesco Dubini
franceso.dubini@mail.polimi.it

Simone Sorrenti
simone.sorrenti@mail.polimi.it

Abstract

In this study, we explored the efficacy of Contrastive Language-Image Pretraining (CLIP) models using different text and image encoders. Our investigation focused on the COCO dataset, revealing that CLIP performance is significantly influenced by the complexity of these configurations: layers, heads and an embedding size. Additionally, we observed that transformer-based image encoders, especially with smaller patch sizes, yield better results. In addition, on the COCO dataset, we used the embeddings generated by CLIP encoders to train a decoder, enabling us to perform image captioning tasks with significant results. We leveraged CLIP models with ViT16 and Bert to conduct image retrieval, zero-shot classification, and clustering tasks on the COCO datase. Subsequently, we extended our study to medical from the Roco dataset and fashion data from the FashionProduct dataset.

1. Introduction

In our study, we investigated the capabilities of Contrastive Language-Image Pretraining (CLIP) models through a series of experiments. We explored CLIP models employing different encoder configurations, including Bert, Vision Transformer (ViT), and ResNet50. Our analyses revealed that CLIP performance is significantly influenced by the complexity of these configurations. These optimized CLIP configurations yielded outstanding results on the COCO dataset, achieving a precision of 74% in Image Retrieval and an accuracy of 71% in Zero-Shot Classification. Moreover, we also explored the application of CLIP in clustering similar images, employing the K-means technique. But the most intriguing aspect of our study was utilizing the encoders learned through CLIP to generate captions for images. Here, we tackled the challenge of integrating image embeddings, directly obtained from CLIP models, with a decoder trained on the contexts of captions associated with the images. However, difficulties arose when attempting to apply CLIP to more specific contexts. When we tested CLIP's capabilities in the medical domain using the ROCO dataset, Image Retrieval did not hold good performances. To further extend the application of CLIP, we turned to the fashion industry, aiming to create a recommendation system based on Image Retrieval. In summary, our study has unveiled the broad potential of CLIP across various contexts but has also underscored significant challenges related to domain specificities and semantic complexities.

2. Related work

In the context of our study, we find relevant connections with two prominent research works. Firstly, "MedCLIP: Contrastive Learning from Unpaired Medical Images and Text" [4] addresses critical challenges associated with utilizing CLIP for medical datasets. The paper underscores the stark contrast in dataset sizes between general images and medical images, which presents a data insufficiency issue. Additionally, the subtle and fine-grained nature of medical distinctions further complicates vision-text pre-training. Moreover, the authors propose a novel approach that replaces the InfoNCE loss with a semantic matching loss based on medical knowledge, enabling the model to capture nuanced medical meanings while eliminating false negatives. Secondly, "Decap: Decoding Clip latens for zero-shot captions via text-only training" [1] introduces the DeCap framework for zero-shot captioning. This framework addresses the challenge of generating captions from visual inputs when the decoder is trained solely on text data. To bridge the modality gap, they devise a training-free mechanism that projects visual embeddings into the CLIP text embedding space. By incorporating this approach, DeCap successfully generates high-quality descriptions matching visual inputs without relying on paired image-text data, making it a flexible and efficient solution, especially in scenarios where alignment between images and texts is less precise.

3. Proposed approach

In our process of training CLIP with various encoders using different architectures, we followed the conventional approach outlined in the existing literature. This architecture involves jointly training a text and image encoder through

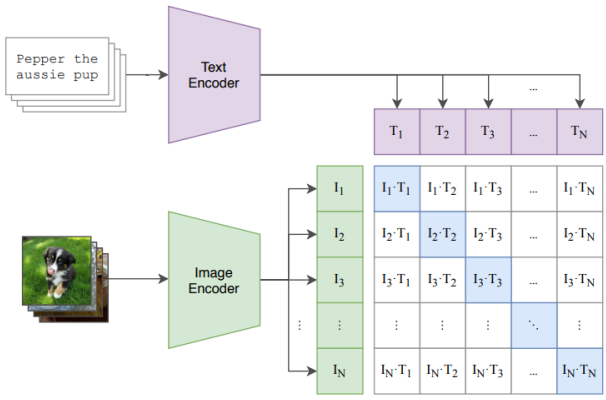

Figure 1. Scheme to explain how a CLIP architecture learns

a shared contrastive loss function, whose objective is to ensure that the embeddings of related images and captions are close to each other in the semantic space.

To aid comprehension, we illustrate the training procedure in fig. 1, employing the same image featured in the original paper by Radford et al. [2].

3.1. Clustering

To make sure our embeddings were correct, we applied clustering and analyzed some sample clusters to see their coherency. We firstly extracted high-dimensional embeddings from images, and secondly applied the K-means clustering algorithm. K-means partitions the data into clusters, each represented by a centroid. To determine the optimal cluster number (K), we utilized the elbow method.

Finally, the determined clusters from K-means were evaluated using within-cluster sum of squares (WCSS), between-cluster sum of squares (BCSS), and silhouette coefficient.

3.2. Image Retrieval

For the image retrieval task, we initiated the process by computing embeddings for all images in the dataset using the CLIP image encoder. To execute image retrieval, we calculated the embedding for a given query and subsequently retrieved and ranked all dataset images based on their cosine similarity between their embedding and the query's embedding. To optimize the efficiency of our retrieval system, we employed Annoy [3], a library designed for approximate nearest neighbor search, to store and retrieve embeddings effectively. Specifically, Annoy allowed us to efficiently organize the high-dimensional embeddings, ensuring swift retrieval times during the search process.

Furthermore, the retrieved images, ordered by their similarity to a specific query, underwent rigorous evaluation using several key metrics: Precision, Recall, Normalized Discounted Cumulative Gain (NDCG), and Average Precision (AP), all computed at the top positions (1, 5, 10).

3.3. Zero shot classification

In zero-shot classification, the goal is to have accurate image classification without the need for explicit training on labeled data.

To achieve this we computed embeddings for all images in the dataset using the CLIP image encoder. Simultaneously, for each label, we calculated its corresponding embedding using the text encoder.

Just as in 3.2, for efficient embedding storage and retrieval, we utilized Annoy, optimizing the entire process and ensuring rapid and scalable performance.

To associate each image with an appropriate class, we employed cosine similarity to measure the similarity between the embeddings of images and the embeddings of the corresponding label queries. Specifically, for every image, we determined the label whose cosine similarity distance with the image's embedding was the smallest, hence the most similar.

The key metrics used to our classifications were precision, recall, and F1 score in both micro and macro modes.

This process is summarized in the picture fig. 2, which was heavily insipired by the original paper.

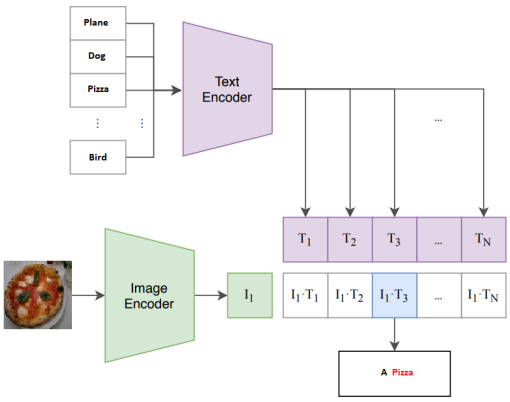

Figure 2. Structure of the zero shot process

3.4. Image captioning

In this part of our study, we investigated the interchangeability of embeddings corresponding to images and associated captions, as they are expected to be similar due to CLIP's training process. Specifically, we focused on testing this hypothesis in the context of image captioning. Our approach involved training a text decoder on contexts extracted from captions, allowing us to predict and generate sentences.

During inference, we firstly attempted a simple switch of the caption embeddings with the embeddings extracted from associated images. Because of inherent errors between the two embeddings (the contrastive loss isn't 0), this approach lead to failure as all the predicted captions made no sense. To circumvent this problem, we explored a projection-based method to bridge the modality gap between text and image embedding spaces. This approach is heavily insipried by [1]. To represent the CLIP text embedding space, we maintained a support memory containing all caption embeddings from the dataset. During inference for a given image, we projected its embedding into the text embedding space using the support memory. This projection was achieved by performing a weighted combination of all embeddings in the support memory.

To determine the weights $\mathbf{w}_i$, we computed the dot product similarity between the image embedding $\mathbf{v}$ and each embedding in the support memory $\mathbf{m}$. Subsequently, we applied a ReLU function to eliminate contributions from embeddings with negative similarity, indicating irrelevant semantic content for the given image. We then performed a weighted sum of the remaining caption embeddings using their respective positive weights:

$$v_{proj} = \sum_{i=1}^{N} w_i * \mathbf{m_i} = \sum_{i=1}^{N} \frac{max((\mathbf{m_i^T v}), 0)}{\sum_{k=1}^{N} max((\mathbf{m_k^T v}), 0)} * \mathbf{m_i}$$

The resulting was then provided to the decoder for generating the corresponding caption describing the image. An overview of the process is available below

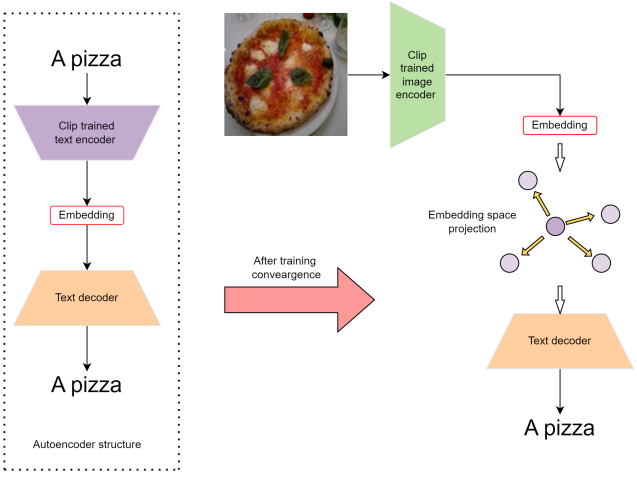

Figure 3. Left: Training of the text decoder. Right: Image caption inference

4. Experiments

4.1. Datasets

We primarily utilized three popular datasets to train various Clip models. Specifically, we employed the COCO dataset for Information Retrieval, Zero Shot Classification, Clustering, and Image Captioning tasks. Additionally, we also attempted to use the ROCO and FashionProduct datasets solely for image retrieval.

4.1.1 COCO

COCO (Common Objects in Context) is a widely used computer vision dataset and it was created to provide a comprehensive and challenging benchmark for image understanding and scene understanding tasks.

In particular, the version of the datasets we used are called "2014 Train images" and "2014 Val images". Each image has 5 different captions associated with it and a varied number of object categories. The 80 possible object categories are of different nature.

We used "COCO 2014 Train images" dataset, to perform the training of the CLIP models, which is composed of 82K images and 414K captions for a total of 414K image-caption pairs with which to train CLIP that have been split into training set, validation set and testing set in the following proportions 70%, 15%, 15%. While we employed "COCO 2014 Val images" for Information Retrieval, Zero Shot Classification, Clustering, and Image Captioning tasks.

The original images have a different size from each other so we applied a fixed resize to all the images to obtain a final size of 128x128 pixels; we applied such a strong resize losing a lot of information for computational reasons. Instead, regarding captions, we applied text vectorization to represent each text sample as a numerical vector.

4.1.2 ROCO

ROCO (Radiology Objects in COntext) is a multimodal image dataset, with the aim of detecting the interplay between visual elements and semantic relations present in radiology images. It contains over 81k radiology images with several medical imaging modalities.

We used a subset of the original training dataset, using only 65K images and each image has only one associated caption, obtaining a total of 65K image-caption pairs. The validation dataset was used to perform the Image Retrieval.

Also for the ROCO dataset, as for the COCO dataset, we applied a division into training, validation and testing sets (70%/15%/15%), resized the images to 128x128 pixels in size and applied text vectorization to the captions.

4.1.3 FashionProduct

FashionProduct consists of professionally shot high resolution product images and accompanying label attributes. Metadata includes brand names, season, age group, and usage. The dataset comprises 44K images featuring various types of clothing items and accessories. The captions were created by concatenating the product description, brand name, gender, base color, and suitable season information.

Also for the this dataset, as for the COCO dataset, we applied a division into training, validation and testing sets (70%/15%/15%), resized the images to 128x128 pixels in size and applied text vectorization to the captions.

4.2. Experiments setup

Throughout our study, we conducted an extensive experimental analysis by training a total of 13 CLIP models, each with unique configurations of encoders for captions and images. Regarding caption encoders, we explored various options, using variants of the BERT model with different numbers of layers (4, 8, and 12), different numbers of heads (4, 8 and 12) and embedding dimensions (256, 512, and 768). For image encoders, we tested three alternatives: the Convolutional Neural Network (CNN) ResNet50, and Vision Transformers with patch sizes of 16x16 and 32x32.

In terms of model training, we split our dataset into three parts for training, validation, and testing, following a 70% training, 15% validation, and 15% testing ratio. We maintained a constant batch size of 32 for all experimental configurations. The learning rate was set to 1.0e-5, and images were uniformly resized to a dimension of 128x128x3, corresponding to color images in RGB format.

To ensure that the embeddings generated by the encoders had a consistent dimension (768), we applied an additional linear Dense layer, to which an L2 regularization with a value of 1.0e-3 was applied. This step was crucial to prevent overfitting and ensure a coherent representation for both images and captions. Regarding the loss function used, we adopted the original CLIP paper's contrastive loss. We configured the temperature parameter associated with this loss to the value specified in the paper, namely 0.07. Model training was carried out for a maximum of 20 epochs, with an early stopping strategy set with a patience of 2 epochs, to prevent overfitting and identify the optimal point in terms of model performance.

After training the image and text encoders were used to conduct experiments on the COCO validation dataset. These experiments involved tasks such as image retrieval, zero-shot classification, clustering, and image captioning. We explored nine different configurations, varying the number of layers, heads, and embedding sizes for the BERT encoder. Meanwhile, the image encoder remained fixed, utilizing ResNet50. Other configurations included using BERT with 12 layers, 12 heads, and an embedding size of 768 with both Vit16 and Vit32. Finally, we tested information retrieval on the ROCO and FashionProduct validation datasets using Vit16 and BERT with 12 layers, 12 heads, and an embedding size of 768.

Finally, regarding the image captioning task, we trained a text decoder. It was fed with the embedding generated by the caption encoder, predicting the same caption in return. During inference, the caption's embedding was replaced with that of the image.

To achieve this, we needed to vectorize the captions into sequences of 20 integers, where each number represented a word ID in the vocabulary learned from the training dataset, with a maximum vocabulary size of 15,000 words. We used a batch size of 64 during the decoder training. Punctuation was removed from the captions, and the text was converted to lowercase. The dataset was split into training, validation, and testing sets following the usual proportions.

During the training of the decoder, the layers of the encoder were frozen. We trained for a maximum of 20 epochs using early stopping based on prediction accuracy. We used rmsprop as the optimizer with a learning rate of 0.001. The decoder, which received the embeddings, consisted of a single base transformer decoder with an embedding dimension of 768, latent dimension of 2048, and 8 heads. Its output, a 20x768 sequence (representing 20 words, each with a 768-dimensional embedding), passed through a dropout layer with a dropout probability of 0.5. Then, a dense layer with dimensions equal to the vocabulary size and a softmax activation was applied to calculate the likelihood of each word in the vocabulary for each word in the sequence.

4.3. Results and discussion

4.3.1 Image retrieval

Following the previously outlined approach, the performance results of different models in image retrieval on COCO dataset can be found in table 1. In these results, **P** represents precision, **R** indicates recall, **NDCG** stands for normalized discounted cumulative gain, and **MAP** represents mean average precision.

As expected, it's clear that increasing the number of layers, heads, and embedding dimensions - consequently augmenting the number of parameters and overall complexity of the encoders - leads to improved performance. Another noteworthy observation is that transformer-based image encoders outperform CNN-based ones. Lastly, ViT 16, which extracts smaller patches, outperforms ViT 32, which extracts 32-pixel patches. Furthermore, not all 80 classes exhibit the same performance; there are variations in how well or poorly the models perform across different classes. For instance, the best classes w.r.t. NDCG@5, such as traffic lights, achieve perfect results, whereas the worst class mice

yields poor outcomes. An example of the results follows below

Figure 4. Image retrieval applied to the best and worst class

Later, we aimed to validate the top-performing architecture identified on the COCO dataset, namely Bert with 12 layers and an embedding size of 768 with ViT16, for image retrieval and zero-shot tasks in a more challenging domain: medical imagery. This domain presented added complexities due to the detailed context, false negatives, and limited data. We trained the model using medical images and their corresponding medical descriptions.

Given the absence of a precise label list associated with each image, we conducted a preliminary test with 8 relevant queries using terms such as 'tumor', 'thrombosis', 'aorta', 'lung', 'brain', 'fracture', 'chest', and 'mandibular'. As evident from the results (table 3), we achieved lower performance compared to the results obtained on the COCO dataset. This outcome highlights the challenges posed by more complex domains.

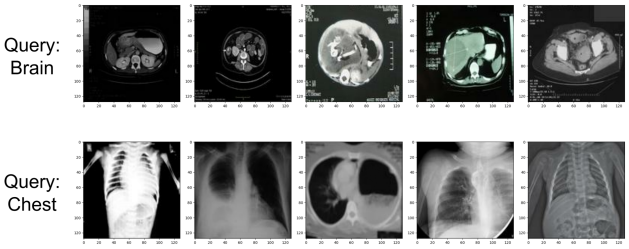

Figure 5. Examples of queries within the medical domain

On the other hand, while experimenting within the fashion domain we obtained more then satisfactory results. An example of two queries done using a fine tuned Bert 768 and ViT16 is available at fig. 6. Unfortunately, due to a lack of labels from the dataset, we were unable to obtain precise metrics. To compensate, we picked 8 sample queries and calculated the metrics by hand classifying the results as relevant or not by ourselves, possibly introducing an additional layer of human error. The hand calculated queries are in the table at table 3, the queried terms were: short skirt, blue t-shirt, white shirt, t-shirt woman, pant yellow, red scarf, red handbag, blue tie.

Figure 6. Examples of queries within the Fashion dataset

4.3.2 Zero shot classification

Similarly, for zero-shot classification on the COCO dataset, enhancing model complexity, utilizing transformer-based encoders, and employing smaller patches like those from ViT16 lead to improved results, as in information retrieval. Moreover, there are variances in performance across different classes, with certain classes showing excellent accuracy while others demonstrate subpar results.

The performance results of different models in zero-shot classification on COCO dataset can be found in table 2. In these results, **P** represents precision, **R** indicates recall, **F1** stands for F1 Score. Furthermore, micro and macro scores were calculated.

4.3.3 Clustering

We aimed to assess the effectiveness of the encoders learned through CLIP, specifically the semantic context they capture, producing similar embeddings for similar contexts. To achieve this, we implemented the method outlined in the previous paragraphs, utilizing K-means and selecting the number of clusters using the elbow method. In this instance, we chose 110 clusters and observed that each cluster is meaningful and coherent, representing strongly similar contexts or objects, as we can see in the embedding space displayed in 3D space (fig. 7). We achieved a Between Clusters Sum of Squares (BCSS) of 1.25, a Within-Cluster Sum of Squares (WCSS) of 0.18 and a Silhouette Score of 0.066.

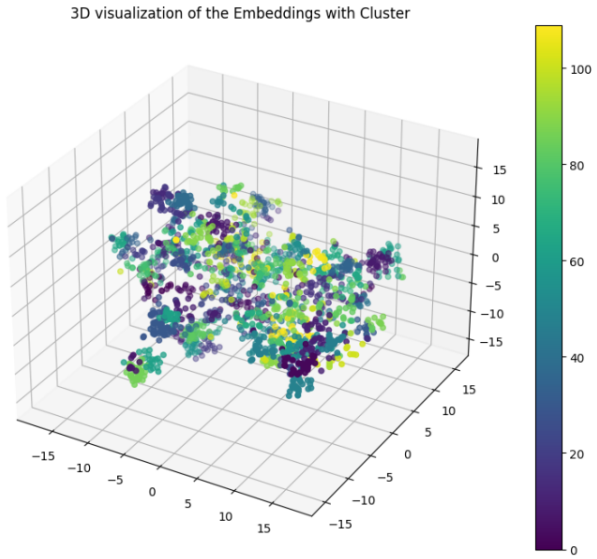

Figure 7. 3D Visualization of the embeddings in the clusters

4.4. Image captioning

Using the approach described earlier, we trained the text decoder on context extracted from captions. During inference, we utilized context from images using the encoders trained via CLIP to predict a sentence describing the image context. As evident from the example below (fig. 8), we achieve excellent results both with context extracted from captions and images.

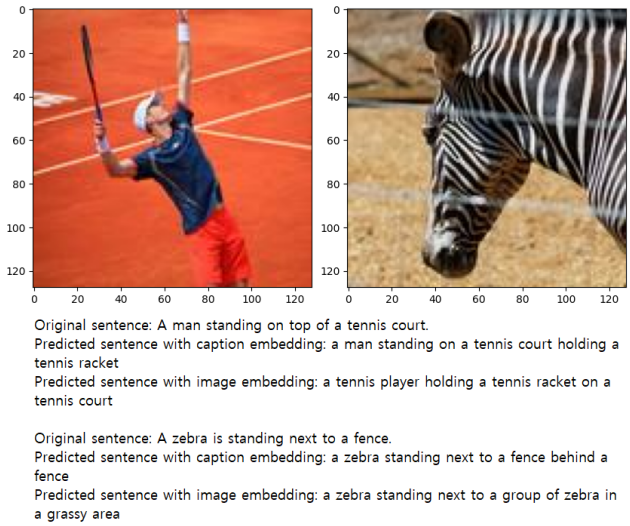

Figure 8. Examples of image captioning

5. Conclusion

In conclusion, our study sheds light on crucial aspects of CLIP-based models for image-text tasks. Increasing the number of layers and embedding dimensions in CLIP significantly enhances its capacity to comprehend intricate patterns and abstract features. Larger embeddings capture richer semantic information, facilitating a deeper understanding of complex relationships between images and text. This expanded capacity empowers CLIP to establish more nuanced and accurate alignments between visual and textual data. Furthermore, our findings underscore the importance of patch size in visual recognition. Smaller patches, exemplified by ViT16, prove invaluable for capturing fine image details, such as subtle color shades and intricate patterns.

Additionally, it is crucial to note that direct image captioning using image embeddings was not feasible. The text decoder was specifically trained with embeddings from associated captions. This discrepancy highlights the subtle diversities even among similar embeddings, impacting the task outcomes. However, we took a more nuanced approach to address this challenge. Instead of directly employing image embeddings for captioning, we opted for a weighted average of captions, where the weights were calculated using dot product similarity. Subsequently, the ReLU function was applied to these weights. This meticulous process allowed us to completely neutralize the impact of entirely irrelevant captions concerning a specific image and context. Through these strategic steps, we effectively mitigated the discrepancies arising from subtle diversities among similar embeddings, ensuring a more accurate and contextually relevant captioning outcome.

Lastly, it is essential to acknowledge that the performance observed through CLIP is context-dependent, varying significantly across domains. CLIP demonstrates robust outcomes in contexts such as the generic realm of everyday images. However, it presents substantial challenges in more specialized domains like medicine, primarily due to false negatives, limited data availability, and the inherent complexity of the domain. These hurdles underscore the need for domain-specific fine-tuning and highlight the nuanced nature of CLIP's applicability across diverse fields.

References

[1] W. Li, L. Zhu, L. Wen, and Y. Yang. Decap: Decoding clip latents for zero-shot captioning via text-only training, 2023. 1, 3

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[3] Spotify and E. Bern. Annoy. https://github.com/spotify/annoy. 2

[4] Z. Wang, Z. Wu, D. Agarwal, and J. Sun. Medclip: Contrastive learning from unpaired medical images and text, 2022. 1

Model	P@1	P@5	P@10	R@1	R@5	R@10	NDCG@1	NDCG@5	NDCG@10	MAP@1	MAP@5	MAP@10
Bert 4/256, Resnet50	0.475	0.474	0.45	0.080	0.494	0.9	0.475	0.568	0.688	0.475	0.585	0.569
Bert 4/512, Resnet50	0.6	0.537	0.53	0.096	0.438	0.9	0.6	0.584	0.706	0.6	0.644	0.617
Bert 4/768, Resnet50	0.6625	0.632	0.606	0.108	0.492	0.925	0.662	0.680	0.775	0.662	0.729	0.698
Bert 8/256, Resnet50	0.587	0.532	0.545	0.101	0.431	0.9	0.587	0.571	0.70	0.587	0.642	0.620
Bert 8/512, Resnet50	0.612	0.635	0.62	0.091	0.501	0.937	0.612	0.686	0.790	0.612	0.719	0.703
Bert 8/768, Resnet50	0.687	0.695	0.678	0.082	0.503	0.975	0.687	0.741	0.829	0.687	0.758	0.746
Bert 12/256, Resnet50	0.55	0.507	0.522	0.113	0.442	0.9	0.55	0.566	0.692	0.55	0.626	0.609
Bert 12/512, Resnet50	0.637	0.674	0.683	0.083	0.463	0.962	0.637	0.683	0.795	0.637	0.724	0.723
Bert 12/768, Resnet50	0.637	0.672	0.664	0.08	0.458	0.937	0.637	0.691	0.790	0.637	0.723	0.715
Bert 12/768, ViT16	0.737	0.685	0.679	0.12	0.502	0.95	0.737	0.746	0.833	0.737	0.80	0.764
Bert 12/768, ViT32	0.487	0.424	0.417	0.108	0.417	0.825	0.487	0.475	0.608	0.487	0.550	0.522

Table 1. Image retrieval results on COCO dataset

Model	macro-P	micro-P	macro-R	micro-R	macro-F1	micro-F1	Accuracy
Bert 4/256, Resnet50	0.26	0.58	0.26	0.49	0.23	0.51	0.49
Bert 4/512, Resnet50	0.29	0.63	0.29	0.53	0.26	0.55	0.53
Bert 4/768, Resnet50	0.32	0.66	0.33	0.59	0.3	0.61	0.59
Bert 8/256, Resnet50	0.32	0.67	0.33	0.56	0.30	0.59	0.56
Bert 8/512, Resnet50	0.3	0.63	0.32	0.54	0.29	0.57	0.54
Bert 8/768, Resnet50	0.38	0.71	0.43	0.62	0.36	0.65	0.62
Bert 12/256, Resnet50	0.34	0.68	0.36	0.59	0.32	0.62	0.59
Bert 12/512, Resnet50	0.38	0.71	0.41	0.62	0.36	0.65	0.62
Bert 12/768, Resnet50	0.35	0.7	0.42	0.6	0.34	0.63	0.6
Bert 12/768, ViT16	0.44	0.79	0.47	0.71	0.43	0.73	0.71
Bert 12/768, ViT32	0.39	0.72	0.4	0.64	0.37	0.66	0.64

Table 2. Zero shot results on COCO dataset

Model	P@1	P@5	P@10	NDCG@1	NDCG@5	NDCG@10	MAP@1	MAP@5	MAP@10
Medical Field	0.25	0.25	0.212	0.25	0.267	0.236	0.25	0.327	0.344
Fashion	0.875	0.85	0.712	0.875	0.872	0.931	0.875	0.919	0.876

Table 3. Image retrieval results on ROCO and Fashion dataset for 8 relevant queries