CLIP TensorFlow & ROCO
Advanced topics in Deep Learning: the rise of the Transformers

Alessandro La Conca
T2A - Computer Science and Engineering
alessandro.laconca@mail.polimi.it

Pablo Giaccaglia
T2I - Artificial Intelligence
pablo.giaccaglia@mail.polimi.it

Abstract

Deep learning's integration of vision and language understanding is becoming increasingly significant. This report presents the application of CLIP to the ROCO dataset, a collection derived from biomedical articles in the PMC OpenAccess subset, composed of images and medical text. By co-training on both visual and textual data, the model can better align visual features with their textual counterparts, potentially allowing for applications like automated medical report generation. Our work presents a TensorFlow implementation of CLIP and introduces a custom hybrid loss function. Through a comparative analysis, we demonstrate CLIP retrieval and zero-shot capabilities. Code available at https://github.com/Calonca/clip-tensorflow

1. Introduction

The integration of vision and language understanding in deep learning is a task that is constantly increasing in relevance, with applications spanning numerous domains. The CLIP (Contrastive Language-Image Pre-training)[6] multimodal vision and language model, proposed by OpenAI, is one of the pillars of this field. By jointly training on visual and textual data, CLIP not only offers a unified representation where both modalities can benefit from each other but also shows capabilities like zero-shot image learning. The medical domain presents a distinct set of challenges characterized by the intricate relationship between visual data (such as medical images) and associated textual information (like clinical notes or diagnostic descriptions).

The target dataset is a simplified version of the ROCO (Radiology Objects in COntext) [5] dataset, which originates from biomedical articles of the PMC OpenAccess[1] subset and consists of detailed images paired with rich textual annotations. The inherent complexity and the semantic depth of this data make it a suitable candidate for the application of the CLIP model. Utilizing CLIP for such datasets can potentially reduce the need for expensive labeling thanks to its zero-shot capabilities. This is crucial in the medical domain, where obtaining labeled data for every specific task can be costly, time-consuming, and often requires domain expertise. Moreover, the co-training on images and text can allow for a fine-grained alignment between visual features and their corresponding textual descriptions, unlocking effectively downstream tasks such as automated report generation or diagnostic assistance.

This report is about the technical work we conducted for implementing a CLIP model working on such a dataset. We make the following contributions:

1. Minimal CLIP Implementation: We provide a clean and effective implementation of the CLIP model, with an easy-to-use highly modular, and customizable code.

2. Hybrid Loss Function: To address the unique challenges of the provided dataset, we introduce a custom loss function, to mitigate the issues identified when utilizing the standard contrastive loss.

3. Architectural improvements: We present three CLIP models based on different design choices, with different sizes and different performances.

3. Comparative Analysis: We conduct a comparative analysis between the original contrastive loss implementation and our custom loss function within the baseline CLIP model. Finally, we provide a comprehensive evaluation by comparing the baseline CLIP model with our proposed model, equipped with a hybrid loss function. Through this evaluation, we show the improvements attained in tackling medical-specific tasks, showcasing the utility and effectiveness of our contributions.

2. Base model

As the baseline, we decided to implement a Vanilla CLIP model.

Given a batch of N (image, text) pairs, CLIP is trained to predict which of the $N \times N$ possible (image, text) pairings across a batch actually occurred. To do this, CLIP learns a multi-modal embedding space by jointly training an image

encoder and text encoder to maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the $N^2 - N$ incorrect pairings. The text encoder is represented by a base-sized DistilBERT[7] unacased. DistilBERT is a distilled version of the BERT model, designed to retain most of the original model's performance while being significantly smaller and faster. On the other hand, the vision encoder is composed of the EfficientNetV2S[8] architecture, an evolution of the EfficientNet model family, optimized for both accuracy and computational efficiency. To align the embeddings from both encoders into a shared semantic space, we employ two separate Projection Layers. These layers are fully connected neural networks designed to map the embeddings from their respective encoders to a common dimensional space. By doing so, we ensure that the textual and visual embeddings are comparable and can be jointly used. Utilizing two separate projection layers for the text and vision encoders, as opposed to a single shared layer, provides more advantages in terms of generated embedding quality. In fact, separate layers allow for tailored transformations, ensuring that the inherent characteristics of each modality's embeddings are optimally projected. This design choice also offers flexibility in handling different dimensionalities from the encoders, leading to a more robust shared semantic space. This model has been designed to process a single textual input which, in our case, is the preprocessed caption string associated with an image. Lastly, the model is optimized by a contrastive loss.

2.1. Clip Loss

This loss function encourages the model to produce similar embeddings for an image and its corresponding textual description, while simultaneously pushing apart embeddings of mismatched image-text pairs. The objective is to ensure that the model learns to associate semantically relevant image-text pairs closely in the embedding space, thus facilitating tasks like zero-shot learning. First, the embeddings for both text and images are L2-normalized, ensuring they lie on a unit sphere. This makes their dot product a direct measure of the cosine similarity. For now on image and text multimodal embeddings are denoted as i_e and t_e.

$$\begin{aligned} i_{eL2} &= \frac{i_e}{\|i_e\|_2} \\ t_{eL2} &= \frac{t_e}{\|t_e\|_2} \end{aligned} \tag{1}$$

Then, the logits are obtained by computing the dot product between the text embeddings and the transposed image embeddings, scaled by a temperature parameter. The temperature parameter T moderates the sharpness of the softmax distribution: a smaller value sharpens the distribution, making the model more confident, while a larger value makes the distribution flatter, making the model less confident.

$$\text{logits} = \frac{t_{eL2} \cdot i_{eL2}^T}{T} \tag{2}$$

This allows to effectively obtain scaled pairwise cosine similarities. Then the similarities between image embeddings with each other and text embeddings with each other are computed as:

$$\begin{aligned} I_{sim} &= i_{eL2} \cdot i_{eL2} \\ T_{sim} &= t_{eL2} \cdot t_{eL2} \end{aligned} \tag{3}$$

The last thing missing is the "true" distributions for image and text, which are calculated by averaging the two similarity matrices and then applying the softmax function, scaled by the temperature:

$$\begin{aligned} I_{target} &= \text{Softmax}(\frac{I_{sim} + T_{sim}}{T}) \\ T_{target} &= \text{Softmax}(\frac{I_{sim} + T_{sim}}{T})^T \end{aligned} \tag{4}$$

These distributions are derived by blending self-similarities within each modality, thereby ensuring that the model not only aligns text with images but also respects the internal semantics of each modality.

Finally, the loss is computed as the average of the categorical cross-entropy losses between the true distributions and the logits for both image and text loss.

$loss_{CE} = \frac{I_{loss} + T_{loss}}{2}$.

$$\begin{aligned} I_{loss} &= \text{categoricalCE}(I_{target}, \text{logits}^T) \\ T_{loss} &= \text{categoricalCE}(T_{target}, \text{logits}) \end{aligned} \tag{5}$$

The categorical cross-entropy loss promotes the alignment of text and image embeddings such that the dot product between a correct text-image pair is maximized compared to mismatched pairs.

3. Base model & loss variation

The second proposed model preserves the same architectural structure as the base one. This model, as the previous one, as a textual input receives the pre-processed caption string associated with an image. The only difference is in the choice of the loss function design. It still aims to minimize the distance between matching pairs of text and image embeddings while maximizing the distance between mismatched pairs. The peculiar difference is that it directly uses the indices as labels, implying that each text embedding should match directly with its corresponding image embedding, and vice versa.

$$targets = [1, ..., \text{\# batch samples}] \tag{6}$$

This means that a Sparse Categorical Cross-entropy is employed to compute the losses. The obtained loss is denoted as $loss_{SCE} = \frac{I_{loss}+T_{loss}}{2}$.

$$\begin{aligned} I_{loss} &= \text{SparseCategoricalCE}(targets, \text{logits}^T) \\ T_{loss} &= \text{SparseCategoricalCE}(targets, \text{logits}) \end{aligned} \tag{7}$$

4. Losses comparison

We decided to experiment with these 2 losses since they have both advantages and disadvantages. In fact the previously presented loss, by considering internal self-similarities within each modality, can potentially handle cases where intra-modality relationships are crucial. For instance, in a medical dataset, images of two types of tumors might look very similar but represent different conditions. By emphasizing their internal differences (how they differ from other images of the same type), the model might be better equipped to distinguish between them. On the other hand, a potential drawback is that the averaging of internal similarities might dilute the effect of cross-modality similarities, especially if one modality's self-similarity is much stronger than the other. For what concerns the second loss, it has the advantage of being more robust in scenarios where each text has a clear corresponding image, thanks to the direct matching. As a con, this loss assumes a perfect one-to-one correspondence between text and images, which might not be optimal for the ImageCLEF dataset, in which there are multiple images for a single caption. This can lead to the case of two identical images with similar captions being present in a batch and leading the loss to learn two different representations that are actually the same.

5. Final model & hybrid loss

The final proposed model has several architectural changes with respect to the previously presented ones. For what concerns the vision encoder we employed an EfficientNetV2M model to exploit its higher vision understanding capabilities with respect to the less-sized EfficientNetV2S. On the other hand, the text encoder is ClinicalBERT [4], an initialized BERT[3] model that was trained in a masked fashion on a large multicenter dataset with a large corpus of 1.2B words of diverse diseases. Moreover, this model is designed to accept as input 2 different textual inputs, which are first processed by the text encoder separately, and then the embeddings are concatenated before being processed by the projector. In our case, the model was trained with both captions and concepts, as 2 separate inputs. The concepts' textual input was generated by concatenating all of them into a single string. This combined approach allows the model to gain a more comprehensive understanding of the visual content. The model optimization happens by combining the two presented loss functions in a weighted summation to take advantage of their strengths for a better embeddings alignment.

$$loss_{hybrid} = \alpha loss_{CE} + (1-\alpha) loss_{SCE} \tag{8}$$

The weighting factor α can be set as a learnable parameter, but we set it to 0.5 in our experiments. Through this combination, we benefit from a dual alignment, ensuring that embeddings align both within modality (mainly due to $loss_{CE}$) and across modalities (mainly due to $loss_{SCE}$)

6. ROCO Dataset

The provided dataset is comprised of 83,275 radiology images., each with a corresponding caption (text describing the image). Each image is also associated with a set of UMLS concepts. The total number of concepts present in the dataset is 8374. The maximum number of concepts present in a single image is 111 while the average number of concepts per image is 16. The most common concepts are "X-Ray Computed Tomography", "Plain x-ray" and "Magnetic Resonance Imaging". The dataset is highly varied in imaging modalities, resolutions, and scales. A preprocessing phase was conducted to handle properly several aspects of the data to ease the learning problem.

During our data exploration phase, we identified several issues within the dataset that introduced complexity to our problem. These issues were addressed as follows:

- Presence of Black or Mostly Black Images: The dataset contains a small percentage of this kind of image, with captions and concepts obviously not related to it. We handled this point by removing images with a sum of pixel values lower than a certain threshold.

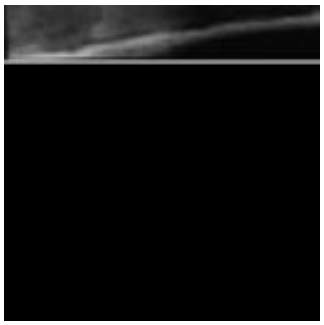

Figure 1: Corrupted image. Caption: reduced distal radius fracture plaster cast

- Medical Abbreviations Expansion: Most of the captions contain medical abbreviations, that, to ease the text encoder job, were expanded using the MEDIALpy Python package. Examples of expansions were "non-steroidal anti-inflammatory drugs" for "nsaid" and "multiple myeloma" for "mm".

- Non-Essential Words in Captions: Not all words in the caption are important for computing text embeddings. We removed words with only one letter, stop words using the NLTK[2] library, and also manually selected popular words that were not deemed very significant, such as "year", "image", and "show".
- Sample Imbalance Across Concepts: Concepts are not balanced across dataset samples. In fact, we observed popular concepts with thousands of samples, like "chests", and rare concepts such as "L2 innervation". The first presented loss was introduced to handle this problem, since it penalizes less for similarity between different samples if the samples are similar, considering both image-to-image and text-to-text similarity computations.

7. MLOps

MLOps for this project was a significant consideration, given the computational demands and collaborative nature of the work. Initially, our workflow relied on Google Colab and GDrive for executing notebooks, hosting datasets, and maintaining model weights. However, due to constraints like memory limitations and restricted session durations on the free Colab tier, this setup proved unsuitable for our project's requirements, especially given large batch sizes and extended model training durations. To address these challenges, we transitioned to a more scalable and efficient workflow. Code changes were constantly pushed to a dedicated GitHub repository. Computational resources were rented from Vast.ai. We mainly worked with 48GB RAM A40s. Runs were executed via SSH either on Jupyter Lab or on local IDE. All the experiments were tracked through Weights & Biases. This platform not only facilitated collaboration but also provided a systematic way to compare model performances. We found this workflow to be beneficial in terms of code integration, training speed, and efficiency of models' performance comparisons.

8. Experiments

All the following experiments have been performed using the same data split with a ratio of 68%, 12%, and 20% for training, validation, and test sets. Batch size was set to 80 and images were resized to (128×128). All three models share the following hyperparameters: a dropout rate of 0.1 for the text and image projection layers, learning rate of $1e^{-3}$ for the image encoder, $1e^{-4}$ for the embeddings projection layers, and $1e^{-5}$ for the text encoder. We decided to employ different learning rates to have more control over training stability. The common dimension for the latent representations was set to 512. All the models have been trained using Adam optimizer with a maximum of 20 epochs. This limit was set due to the fact that we observed almost all the experiments ending before that limit due to the usage of Early Stopping with patience of 5 on validation loss. In the Table 1 are reported the epochs and hours taken by each model.

	Epochs	**Time (hours)**
Base Model	10	2.4
Base Model & Loss Variation	7	1.7
Final Model & Hybrid Loss	9	3.4

Table 1: Training epochs and time

We ran several experiments in order to test the performance of the model, both qualitatively and quantitatively. We tested the text-to-image, text-to-text, image-to-text, and image-to-image retrieval, as well as zero-shot classification.

8.1. Text-to-image retrival

Text-to-image retrieval consists of finding the corresponding image given a text query. We implemented text-to-image retrieval and other retrieval tasks in the following way:

1. Pre-compute the embedding of the whole test set in order to speed up computation and avoid recomputing them on each query.
2. Compute the embedding of the query. The text can be arbitrary so we cannot use an existing embedding.
3. Compute the similarity score between the query (an image or a text) and all the images (in case of text-to-image or image-to-image) or all the texts (in case of text-to-text retrieval or image-to-text retrieval)
4. Rank the results decreasingly in terms of similarity and return the ones with the highest similarity.

Figure 2 is an example text-to-image retrieval for of the text query: "chest", qualitatively the model works and it's able to return the expected result.

We assessed the model's performance using recall@K and MRR@K scores, which unfortunately indicated poor results. Upon further investigation, we pinpointed the issue to the ROCO dataset. To validate this, we applied the same metrics to the Flickr Image captioning dataset.

A closer examination of the ROCO dataset revealed a multitude of visually similar samples when searching for the term 'chest' or other words. In contrast, some other samples appeared visually distinct. We believe that the presence of numerous visually similar samples significantly impacted the ranking process, leading to lower scores.

Since we could not use recall scores to evaluate our model we decided to compare them on their performance on Zero-Shot classification.

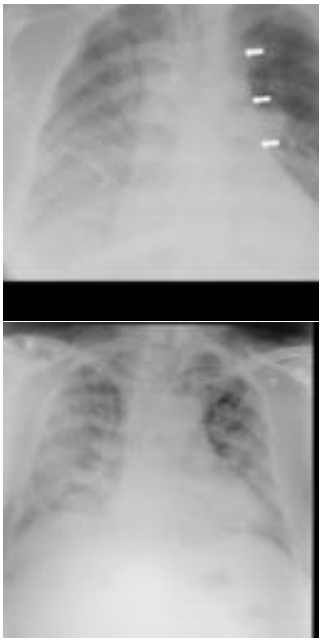

Figure 2: The first two images retrieved by the Final model using text-to-image retrieval for the text **"chest"**.
Image 1 caption: chest radiograph paramediastinal position central venous catheter insert via internal jugular vein
Image 2 caption: chest radiograph two

8.2. Zero-shot classification

We classified the image samples, using the associated concepts. We decided to consider the multi-label classification with a variable number of classes, to observe the performances under different settings. The candidate classes were chosen among the most popular concepts across test samples, to ensure a significant amount of data to use for evaluation. Since the model has been trained on full sentences describing images in some way, to help bridge this distribution gap, we used the following prompt template "A photo of a label.". This is useful since it helps specify the text is about the content of the image. In the Figure 3 are reported the performances of the proposed models.

We can observe that when the number of classes is under 8, the Clip Base model with sparse cross-entropy has the best performance. Interestingly the Final Model, which processes both the captions and the concepts, has the best performance when the number of classes is over 7. Additionally, it is important to note how the accuracy of the Final Model decreases much more smoothly compared to the other models. This is probably due to the fact that is has learned the concepts' vocabulary, which is leveraged for concepts' classification. The Table 2 reports in detail zero-shot accuracy values for different numbers of classes. Note that "Lv" stands for "Loss variation", while "Hl" stands for "Hybrid Loss".

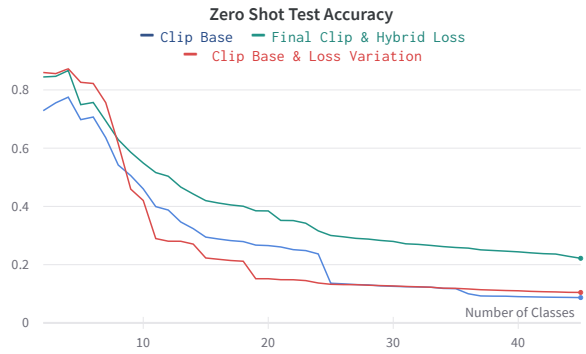

Figure 3: Zero-shot classification accuracy

Model	**2 classes**	**5 classes**	**10 classes**	**30 classes**
Base	0.7287	0.6979	0.4602	0.1362
Base & Lv	0.8598	0.8261	0.4204	0.1326
Final & Hl	0.8445	0.7498	0.5490	0.3000

Table 2: Zero-shot classification accuracy (details)

9. Conclusion

In summary, this report presents our findings from applying the CLIP model to the ROCO dataset. We have analyzed various architectural configurations and loss functions. Our investigation has highlighted the profound influence of the loss function on both the convergence time and zero-shot performance of the model.

Specifically, we have observed that a loss function designed to consider the similarity between concepts leads to faster convergence while maintaining comparable zero-shot performance. We have also discussed the challenges associated with ranking metrics when dealing with highly similar samples within the dataset.

An interesting future research direction is the exploration of ranking metrics that can effectively account for the inherent similarities between samples, which would offer more accurate evaluations in scenarios involving closely related samples. Additionally, addressing the issue of underrepresented samples within the dataset remains an important research challenge, and applying methods to deal with it remains an important direction to follow to further improve the performance of the model.

References

[1] Pmc open access subset. Available from https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/. 1

[2] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 4

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 3

[4] K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*, 2019. 3

[5] O. Pelka, S. Koitka, J. Rückert, F. Nensa, and C. Friedrich. Radiology objects in context (roco): A multimodal image datase, 2018. MICCAI Workshop on Large-scale Annotation of Biomedical Data and Expert Label Synthesis (LABELS) 2018, September 16, 2018, Granada, Spain. Lecture Notes on Computer Science (LNCS), vol. 11043, pp. 180-189, Springer Cham, 2018. 1

[6] 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. 1

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. 2

[8] M. Tan and Q. V. Le. Efficientnetv2: Smaller models and faster training, 2021. 2