MedicalAI : From classification label to text label

Antoine Munier
Ecole Polytechnique Fédèrale de Lausanne, EPFL, Switzerland

***Abstract*—Vision-language models (VLMs) such as CLIP have shown promising performance on a variety of recognition tasks using the standard zero-shot classification procedure – computing similarity between the query image and the embedded words for each category. However, this approach only utilizes the category name and overlooks the rich contextual information that language provides. Additionally, it lacks intermediate insights into why a particular category is chosen and does not allow for adjusting the criteria used for this decision.**

In our work, we first train a CLIP model on both images and their associated text. Following this, we train a classifier using the embeddings generated by CLIP. We then evaluate the performance of this image-embedded classifier to determine if it outperforms a classifier trained solely on image and label without the additional text context. Our method provides a novel approach to leveraging the combined power of image and text embeddings for classification tasks.

Extensive experiments show that our framework not only improves accuracy on MedMNIST across various distribution shifts but also demonstrates the potential to recognize concepts unseen during training. Furthermore, our approach illustrates how integrating descriptive features from text can effectively mitigate bias compared to baseline models trained without this additional context. Note : This report show the work as up to the June 7th. Because of a lot of waiting time on the server. I don't succeed to train the classifier with CLIP embedding yet. We hope it will be done for the final presentation.

I. Dataset

In our study, we utilize the MedMNIST v2 dataset [1], a comprehensive collection of standardized biomedical images. This dataset includes 12 datasets for 2D classification tasks and 6 datasets for 3D classification tasks. Each image in the dataset has been pre-processed to a small size of 28×28 pixels for 2D images or 28×28×28 voxels for 3D images, with corresponding classification labels provided.

Initially, our focus will be on the MedMNIST2D subset, which includes 708,069 images across various biomedical imaging modalities and classification tasks. These tasks range from binary and multi-class classification to ordinal regression and multi-label classification, making it an ideal benchmark for evaluating the performance of our vision-language models (VLMs).

If time permits, we may extend our analysis to the MedMNIST3D subset, which contains 9,998 3D images. This extension would allow us to explore the potential benefits of incorporating 3D spatial information into our classification framework.

II. Multi-label dataset

The first part of our project focused on training a multi-dataset model using the MedMNIST v2 dataset, with the goal of developing a robust classifier capable of handling various biomedical image classification tasks. This process involved several steps and faced significant challenges, primarily due to the limited number of multi-label datasets available, such as CHESTMNIST, which led to overfitting on specific tasks.

To begin, we utilized several 2D datasets from MedMNIST v2, including PathMNIST, DermaMNIST, OCTMNIST, PneumoniaMNIST, RetinaMNIST, BreastMNIST, BloodMNIST, TissueMNIST, OrganAMNIST, OrganCMNIST, and OrganSMNIST. A custom class, TargetOffsetDataset, was created to adjust target labels dynamically, ensuring unique class labels across all datasets. Data transformations were applied, including resizing images to 224x224 pixels when necessary and normalizing them.

Each dataset was then loaded, and target labels were adjusted using the TargetOffsetDataset class. The datasets were concatenated to form unified training, validation, and test sets using PyTorch's ConcatDataset. We experimented with two model architectures: ResNet18 and ResNet50, adapting them for the multi-class classification task. Models were initialized with random weights.

The training procedure involved standard optimization techniques, specifically the Adam optimizer and a MultiStepLR scheduler. The loss function was selected based on the task type, with Cross-Entropy Loss being used for multi-class classification. Extensive hyperparameter tuning was performed, including adjusting learning rates, changing batch sizes, and experimenting with different model architectures by adding or removing layers.

The model's performance was evaluated on the validation and test sets after each epoch, using accuracy and AUC as primary metrics. The best model was selected based on the highest validation AUC achieved during training. However, the multi-task model consistently failed to achieve satisfactory performance, with the highest accuracy being only 0.2 on multi-class tasks. Overfitting was a significant issue, particularly with the CHESTMNIST dataset, leading to poor generalization across different tasks. Despite various strategies, including adjusting dataset weights and modifying the model architecture, we could not effectively overcome the overfitting problem. We choose to only train the model

as a multi-dataset model and remove the multi-class task.

In conclusion, the multi-label-dataset model training highlighted the challenges of handling diverse biomedical image classification tasks within a unified framework.

Despite these challenges, the model demonstrated promising results I, particularly on the test and validation datasets. With an AUC of 0.99433 and accuracy of 0.74205 on the test set, and an AUC of 0.99670 and accuracy of 0.81847 on the validation set, the model shows strong potential in distinguishing between different image types with high precision. These metrics indicate that the model is not only effective during training but also maintains a high level of performance when evaluated on unseen data.

The relatively high AUC scores across all datasets suggest that the model is adept at ranking predictions correctly, which is crucial for applications requiring reliable decision-making. However, the slightly lower accuracy on the test set compared to the training and validation sets hints at potential overfitting, indicating areas for further improvement.

Overall, while there is room for enhancement, the results are encouraging and validate the viability of the approach. Future work will focus on refining the model architecture.

To obtain this metrics you have to run this file train_and_eval_pytorch_multidataset.py with 10 epochs.

Dataset	AUC	Accuracy
Train	0.99787	0.85833
Validation	0.99670	0.81847
Test	0.99433	0.74205

Table I
PERFORMANCE METRICS FOR TRAIN, VALIDATION, AND TEST DATASETS

III. CLIP TRAINING

A. Dataset Creation

The dataset creation for our study primarily focused on incorporating textual descriptions with biomedical images to train the CLIP model. This approach leverages the unique capabilities of CLIP, which learns from both visual and textual data, enhancing its ability to perform robust image classification tasks.

We used the MedMNIST v2 collection. To effectively utilize these datasets with the CLIP model, we created a custom dataset class called `TextTargetDataset`. This class integrates textual descriptions with the image data, providing the CLIP model with rich contextual information. Each image in the dataset is paired with a text description that explains its content and the associated classification labels.

The `TextTargetDataset` class operates as follows:

- **Initialization**: It receives the dataset, a dictionary mapping labels to textual descriptions (`label_dict`), a general text label for the dataset (`text_label`), the type of classification task (`task`), and the tokenizer used to convert text to token IDs.
- **Length Method**: Returns the total number of samples in the dataset.
- **Get Item Method**: For each sample in the dataset:
  - The image and target label are retrieved.
  - A textual description is generated based on the type of task:
    - For multi-label, binary-class tasks, the description combines the general text label with the specific label of the target.
    - For multi-class tasks, it lists all present labels or indicates no issue if none are present.
  - The generated text is then tokenized using the specified tokenizer to produce token IDs and attention masks, which are used by the CLIP model.

This process ensures that each image is accompanied by a descriptive text, enhancing the CLIP model's ability to learn from both modalities. For instance, an image from PathMNIST may be paired with a description like *"Histological image of colorectal cancer tissue patches with adenocarcinoma,"* providing a clear context for the visual content.

Also, label adjustments were made to ensure unique class identification across the combined datasets. The datasets were then concatenated to form comprehensive training, validation, and test sets. Data loaders were created to facilitate efficient batch processing during training and evaluation.

This detailed dataset creation process, incorporating textual descriptions with images, was crucial for effectively training the CLIP model. By providing rich contextual information alongside visual data, we aimed to leverage the full potential of CLIP's learning capabilities, leading to more accurate and robust biomedical image classification.

B. Fine-Tuning

In our work, we utilized the CLIP (Contrastive Language-Image Pre-training) model [2]. This model leverages natural language supervision to learn robust image representations and has demonstrated state-of-the-art performance across various tasks.

We began by initializing the CLIP model and tokenizer using the "openai/clip-vit-base-patch32" variant. This version combines a Vision Transformer (ViT) for the image encoder with a text transformer for the text encoder, enabling effective learning from a diverse set of image-text pairs.

The data was loaded and preprocessed to ensure compatibility with the CLIP model. We used a function to handle the resizing of images to 224x224 pixels and tokenizing the text inputs. The dataset was then split into training, validation, and test sets.

The optimizer used was Adam, configured with a learning rate of 5e-5, specific beta values, epsilon, and weight decay

to ensure a safe learning rate suitable for fine-tuning the pre-trained model.

The fine-tuning process involved iterating through the training data for ten epochs. For each batch, we zeroed the gradients, then passed the images and text inputs through the model to obtain the logits for images and text. The cross-entropy loss was computed for both image and text logits against the ground truth labels, and the total loss was the average of these two losses. Gradients were backpropagated, and the optimizer step was performed to update the model weights.

This training process allowed us to adapt the CLIP model to the specific requirements of our biomedical image classification tasks, leveraging its pre-trained capabilities while tailoring it to our dataset's nuances.

To train and fine-tune the CLIP model you have to run the file CLIP/train.py

IV. Multi-Dataset with CLIP

The next phase of our project involved training a multi-dataset model using the CLIP model embeddings, leveraging its capability to learn from both images and textual descriptions. This approach aimed to enhance the model's performance in recognizing various biomedical image classification tasks by incorporating rich contextual information.

We began by loading our CLIP trained model. Data was loaded the same way as the multi-dataset model. The datasets were split into training, validation, and test sets, with each image modified with the CLIP embeddings from our image tokenizer.

For the training, we used the Adam optimizer with a learning rate of 0.001, along with specific beta values, epsilon, and weight decay to ensure a safe learning rate suitable for fine-tuning the pre-trained model. The learning rate was scheduled to decay at specific milestones to stabilize the training process.

The fine-tuning process involved iterating through the training data for ten epochs. For each batch, gradients were zeroed out, and the images and text inputs were passed through the model to obtain logits for both images and text. Cross-entropy loss was computed for these logits against the ground truth labels. The total loss, being the average of the image and text losses, was then backpropagated, and the optimizer step was performed to update the model weights.

Throughout the training process, the model's performance was monitored on the validation set to ensure effective learning and to prevent overfitting. After completing the fine-tuning, the model's state dictionary was saved for future use. The performance of the model was evaluated on the test set using metrics such as accuracy and AUC, and the best model was selected based on the highest validation AUC achieved during training.

This meticulous training process allowed us to harness the power of the CLIP model, leveraging both visual and textual information to improve the accuracy and robustness of biomedical image classification tasks.

References

[1] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, Jan. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41597-022-01721-8

[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.