Multi-Label Classification on MS COCO

Tianyi Chen 490165227, Jiahao Wu 490517596

***Abstract* – This study aims at a multi-label classification task of a large scale dataset whose every entry contains an image and a short caption with 40000 of them. Allow me to introduce you to what may be the ultimate solution for NLP and image recognition field, with super simplicity and competitive accuracy — The contrastive Language-Image Pre-training model, also known as: CLIP [1].**

***Keywords* – multi-label image classification, CV, NLP, multimodality, CLIP**

I. INTRODUCTION

A. Aim of the study

The task is to detect multiple types of objects in the image, which has long been a familiar topic in deep learning. With the development of deep learning, various dazzling networks are implemented every year, many fancy tricks are invented, and records are constantly being refreshed in multiple tasks and datasets.

From Alexnet to Restnet to Transformers, more complexity, more parameters, and more expensive training costs. But in the end, it's all about the same thing: finding a balance between over-fitting and under-fitting. This got me thinking: to fit a data set, even a network of five years ago, is complex enough. Why should we constantly optimize our model for a single task, even if we know that the model is totally useless other than this? Is there a model that solves every task, not just one?

The answer to this question is affirmative, and the answer is CLIP. We want to generalize our ability to solve all problems, and CLIP is our best helper.

B. Significance

Computer vision systems are now commonly trained to predict a predetermined set of object classes. This limited form of supervision restricts their generality and usability, as additional labeled data is required to train any visual concepts that the model has never seen before. We want to find a general solution, an ability to recognize unseen objects with very few rules defined manually, which we call "zero-shot". Once this capability becomes a reality, it will pose a massive challenge to traditional deep learning frameworks. There will be no clear boundaries between different tasks and datasets, and we might be able to jump out of the existing training-fitting process.

After listening to my imagination to the great future, let's come back to reality: CLIP does not have the magical ability that I said it would. It cannot understand the visual concepts of anything that it has never seen before. It doesn't even have any unique technique in the model structure. Just like all networks of the past, it needs data to train. But CLIP has by far the most generalization capability of any network. Using vast amounts of data, it can learn many abstract concepts previously thought to be understandable only to humans. CLIP interprets deep learning from a higher level, which will become a memorable step of AI research.

II. Related work

A. CLIP: contrastive Language-Image Pre-training model

CLIP can efficiently learns visual concepts from literal supervision. CLIP can be applied to ANY visual classification benchmark by simply providing the name or a short description of the visual categories to be recognized, Just like the "zero-shot" capabilities of GPT-2 and GPT-3.

The clip was created to address several long-standing problems in computer vision: it takes a lot of labour to build a data set that provides only a narrow set of visual concepts. In the past, computer visual models required significant effort to train, but the trained models were usually good at and only good at a specific task. Therefore, people began to doubt the cost performance and even the correctness of this training mode of deep learning in computer vision.

CLIP solves this problem by using the abundantly available image resource on the internet and the corresponding natural language property such as caption, name and description as their supervision. By design, the network can be instructed in natural language to perform a great variety of classification benchmarks. At the same time, this capability is generalized and does not require many direct optimizations for each task.

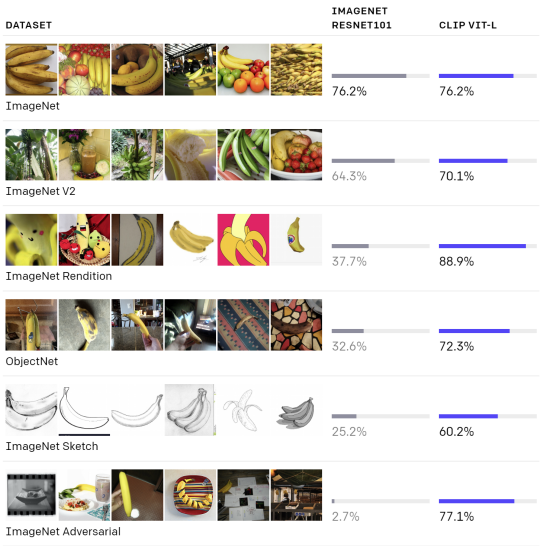

Fig. 1. The classification ability of CLIP on various datasets is demonstrated. We can see that Clip not only outperforms ResNet101 in recognizing highly abstract visual concepts(i.e. sketch, Adversarial etc.), but also performs just as good as ResNet101 in basic visual concepts(ImageNet).

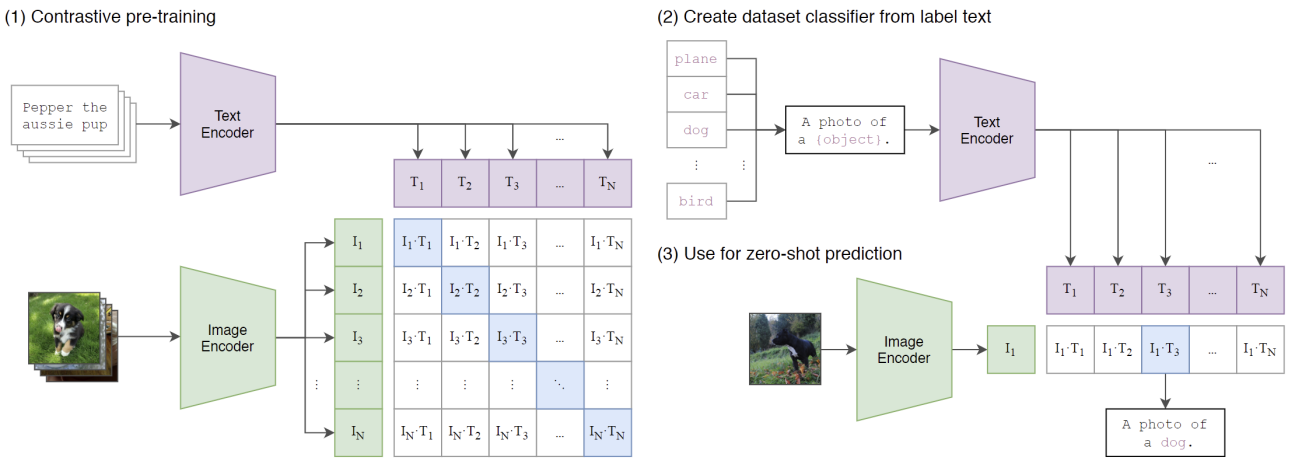

Fig. 2. CLIP's approach. Unlike traditional models that train an image feature extractor and a linear classifier to predict some label, CLIP trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. After then, the learnt text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes. It turns the task of image label classification into the task of image-text matching. This gives CLIP the power to associate images with their names.

Rather than training an image feature extractor and a linear classifier to predict some manually set labels, CLIP was trained with vast visual concepts in images and their relating literal property as their supervision. This gives CLIP the capability to classify almost arbitrary visual concepts as long as they can be described by natural language.

CLIP has solved the following problems that have existed since CV deep learning has arisen: Costly datasets, Poor real-world performance, and Narrow usage. In the past, if we wished to perform a new label recognition task on a pre-trained model, we needed to build a new dataset, add an output head, and fine-tune the model. However, with CLIP we can perform a wide variety of visual classification tasks without needing additional training examples. This generalisation capability is the backbone that we talked about all along with this paper.

B. Text Encoders: Transformer

Clip uses Transformer, which has long been the pearl of NLP, as the base model for text Encoder. The Transformer abandons the recurrent structure and entirely relies solely on the attention mechanism. It breaks through the limitation that the RNN model cannot be used for parallel computation. Compared with RNN, the calculation cost between two positions does not increase with distance. Also, self-attention can make the model more understandable. We can examine the distribution of attention from the model. Therefore we can manipulate the weights to teach the model on different tasks. With these advantages mentioned above and the outstanding performance, Transformer is chosen as the Clip text encoder.

C. Image Encoders: Vit & ResNet

Similarly, Transformer can be used for computer vision, and Clip uses Vision Transformer (ViT) for image encoding. There are many benefits to using VIT, including its superior performance, making joint modelling of vision and NLP easier, etc.

At the same time, due to the large scale of the CLIP network and the costly training process of ViT, ResNet is also selected as an image encoder of Clip. Resnet is a mature network architecture which is lighter and faster.

D. MUTAN fusion model

MUTAN [2] fusion model is used to merge visual and linguistic features, which is based on multimodal Tucker fusion. We came across this model in the field of Visual Question Answering tasks, since the VQA task is a multi-modal task utilising both image and text information, this model is the first choice in our assignment. Specifically, in the architecture of MUTAN, the images are encoded using ResNet and texts are encoded using RNNs to obtain an 1-dimensional representation of each kinds of input. Then it uses Tucker decomposition on the tensor used in the bilinear interaction between visual and linguistic encodings. Since the multi-modal fusion technique is used in MUTAN, we expect this method could also work in our assignment where we need to build relations between images and captions. However, the results are not as good as CLIP gives us since the Visual-and-Language tasks are not pretrianed on MUTAN.

III. Techniques

A. Data augmentation

Data augmentation (DA) refers to techniques for boosting training data variety without collecting more data [3]. DA has been widely employed in CV, where cropping, flipping, and colour jittering are all frequent model training strategies. In this assignment, we use a **vit-gpt2-image-captioning** model to generate additional captions to enrich the dataset. This model is fine-tuned as a proof-of-concept for the huggingface FlaxVisionEncoderDecoder Framework. Specifically, for each sample in the training and testing dataset, we use the image of the sample as the input of the

vit-gpt2-image-captioning model, and use the generated output as augmented caption and add the the existing dataset. For example, for image 1 in the dataset, as shown in figure 3, the original caption is *Woman in swim suit holding parasol on sunny day* and the additional caption generated is *a beautiful young woman holding a pink umbrella*. For object detection purpose, both *woman* are detected, but the original caption fails to provide information of the *umbrella* where the augmented caption provides. Hence, if the *umbrella* is one of the labels needing to be identified, then using the augmented dataset can include more key words to the model.

Fig. 3. An example of the image in the dataset

Moreover, since the pretraining task of the model used for features extraction in our assignment is to minimise the cosine similarity between the mapped embedding of image and text information in the mathematical space. The more relevant captions we use, the more meaningful embedding the model can generate, which could show that the model actually understand the relationship between the image and text, which benefits the downstream task of multi-label classification.

B. Data preprocessing

We first conduct a basic analysis on the given dataset, but the result shows that there are 29996 samples instead of 30000 samples in the training dataset, after converting it to pandas Dataframe object. The reason is because the new line characters are inconsistent in the original training datasets, there are four lines collapsing into one line. Hence, by explicitly handling this issue, we obtain a training dataset of exact 30000 samples where the captions are from the same image of that sample. By doing this data preprocessing, we are able to eliminate bad samples since the collapsed caption in the origianl training set is a combination of four captions for different images, which is not in a correct input format.

C. Pretrained models

The field of AI has become more data-driven with the development of neural networks training, attention mechanism and Transformers, etc. Pretrained models are created in the both Computer Vision and Natural Language Processing domains [4], which does not require higher training costs to achieve state-of-the-art performance on a variety of downstream tasks. Due to the success of utilising Pretrained models in the field of CV and NLP, researchers focus on pre-training large-scale models for the multi-modality across both domains, which are called VisionLanguage Pre-Trained Models (VL-PTMs) [5]. VL-PTMs may learn universal cross-modal representations by pre-training on large-scale image-text corpora, which is useful for obtaining good performance in downstream Vision-and-Language tasks.

In this assignment, we are asked into conduct a multi-label classification of image-text pairs, which is a typical Vision-and-Language task. Using VL-PTMs to extract features could save computational cost and achieve an excellent performance within a feasible time.

D. Extracting features with CLIP

CLIP is a Dual Encoder VL-PTM, which consists of a GPT-2 as text encoder and ViT as image encoder, and. is trained on Cross-Modal Contrastive Learning task with over 400,000,000 images-description pairs. Hence, CLIP is able to learn transferable visual representations and shows a amazing zero-shot transfer to various image related classification task as mentioned before. To this end, CLIP is preferable in our assignment to encode both image and text information and also interact between both modalities. We also leveraged other multi-model VisionLanguage models such as MUTAN, but it is much complex and costly. It has a larger model size and the target task of MUTAN is not exactly the same as in this assignment. Using CLIP for both image and text encoding will surely remains a better consistency. Hence, we chose CLIP to encode both images and texts to extract features.

E. Classification model

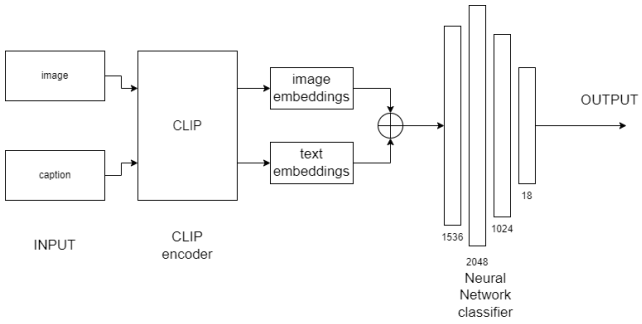

Fig. 4. Model of our classifier. After encoding the image and caption, we concatenate their embedding and feed it into the neural network. It then outputs the probability of the 18 labels.

Firstly, we use CLIP to extract the embedding of image and text respectively, which gives two tensors of 768 in length. After the two tensors concatenate, we get a latent tensor, resulting from the extracted attribute of this entry.

Once we have this latent tensor, we will use it to train our classifier. This classifier uses 30,000 such Latent tensors generated by the training set to train and finally outputs the probability of 18 labels. If it's greater than the preset threshold, we consider that label is positive.

Since latent tensor already has its own extracted information, all the classifier network needs to do is a straightforward one-dimensional vector classification task,

so we use the most primitive shallow neural network. At the same time, we also tried machine learning methods such as KNN and Random Forest, but a simple neural network is the most suitable for our needs due to the size of the dataset.

F. Fine-tuning

There are two ways to fine tune the pretrained model with additional classification layers, by either optimizing the weights in CLIP or update the parameters in the classification layers [6]. We chose the latter strategy for three reasons:

1. The training dataset consists of 30,000 samples which only takes less than 0.0075% of the massive data used for training CLIP, hence, the effectiveness of fine-tuning the weights in the pretrained model may not be significantly rewarded.
2. It is time-costly to include encoding process in the model training process. It takes about five hours to encode training images and texts using CLIP on Tesla V100-SXM2 (Colab). This is a huge obstacle on tuning the pretrained model as we need the feedback constantly, thus the unacceptable time cost is making the hyper-parameter optimisation process even more difficult.
3. Most importantly, we want to emphasize the generalization ability and operability of CLIP, so we do not want to spend too much time finetune for a task, which contrasts with this paper's theme. We demonstrate that clip can be easily used for different classification tasks by optimizing the simple classification network attached to it.

Hence, the choice of fine-tuning lies on the trade-off between performance improvement and computational costs. We chose to save training time and space for further hyperparameter tuning.

G. Warm up

During the training process, warm-up mitigate early overfitting of the model to the mini-batch at the initial stage, keeping the distribution smooth it also helps maintain the stability of the model at a deeper level. Since we obeserved overfitting issue in the testing dataset, we then adopt warm-up to mitigate. In this assignment, we use *transformers.get_linear_schedule_with_warmup* provided by huggingface. During the warm-up period, the learning rate increases linearly from 0 to the initial learning rate in the optimizer. It then creates a schedule after the warm-up phase so that its learning rate decreases linearly from the initial learning in the optimizer to 0. This process is shown in figure 5.

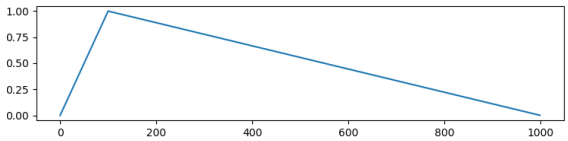

Fig. 5. An example of the learning rate using warm-up

H. Evaluation metrics

For datasets with uneven label distributions, only using accuracy assessment measures does not provide a fair indicator of model performance. This is because they are averaged for each class of assessment and do not take into account the label distribution. Hence, we consider three situations of resulting sample label classifications: true positive, false positive and false negative. We the define **precision** and **recall** as follows:

- Precision: How many selected items are relevant? This is computed by the ratio of true positives on the sum of true positives and false positives. $p = \frac{tp}{tp+fp}$
- Recall: How many relevant items are selected? This is computed by the ratio of true positives on the sum of true positives and false negatives. $r = \frac{tp}{tp+fn}$

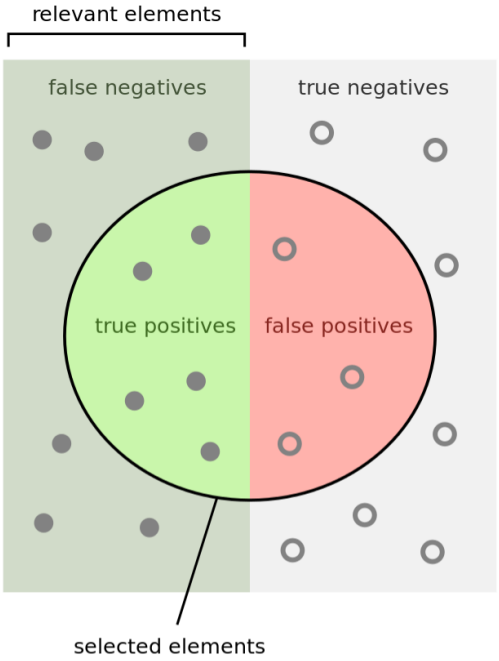

Fig. 6. Precision and Recall

In terms of the precision and recall for a class C, we can define F1, for each class as follows:

$$f1 = 2 \cdot \frac{p \cdot r}{p + r}$$

The maximum possible F-score is 1.0, which indicates perfect accuracy and memory, while the lowest possible number is 0, which indicates either precision or recall is 0.

IV. Experiments and results

A. Dataset

Since there is no validation set provided, if we use part of training data to evaluate the performance of our model, it will be inaccurate. Hence, to tackle this issue, we explicitly split our training dataset into training and validation dataset with first 28000 samples as the new training dataset and use the latter 2000 samples as the new validation set.

B. Accuracy / efficiency

The best F1 score we achieved on our validation set is around 90.2%. The classification report on the validation set of our model is shown in table I and examples of loss plot and f1 plot are shown in figure 7 and figure 8, respectively. From the classification report, we observe that except label 1, the size of samples of other labels are significantly less. In addition, except for label 1 and label 7, the recall score for other labels are not higher than its precision score, and in the most cases, the precision score is significantly higher than its recall score, which means for these low-sample labels, our model is prudent to make any decisions, the labels are predicted only if the model is confident enough.

Label	precision	recall	f1-score	support
1	0.94	0.98	0.96	1548
2	0.88	0.51	0.64	73
3	0.69	0.68	0.68	280
4	0.94	0.77	0.85	83
5	1.00	0.94	0.97	72
6	0.91	0.69	0.79	104
7	0.93	0.96	0.94	78
8	0.61	0.53	0.57	129
9	0.79	0.79	0.79	58
10	0.79	0.61	0.69	96
11	0.97	0.65	0.78	49
13	0.90	0.69	0.78	52
14	1.00	0.53	0.69	17
15	0.70	0.33	0.45	119
16	0.91	0.70	0.79	70
17	0.99	0.93	0.96	83
18	0.91	0.91	0.91	103
19	0.98	0.93	0.95	67
samples avg	0.93	0.90	0.90	3081

TABLE I

F1 scores of the best model

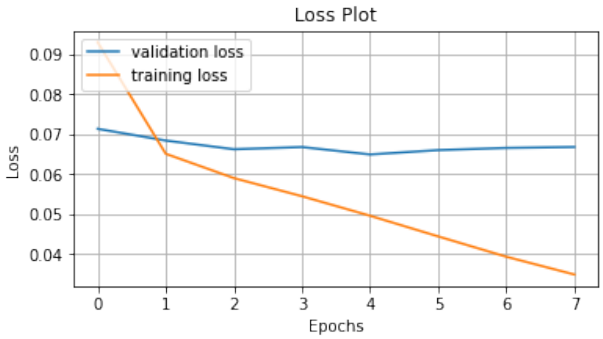

Fig. 7. An example of the loss plot

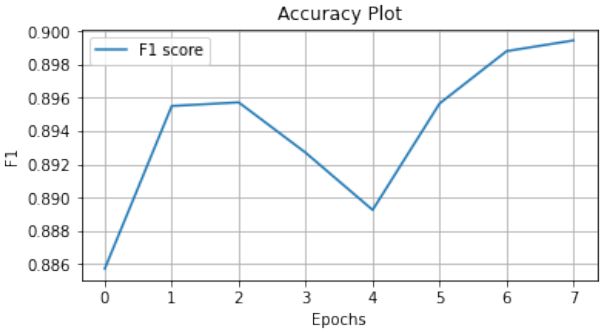

Fig. 8. An example of the acc plot

C. Extensive analysis

1) hyperparameter analysis and comparison methods:

Table II summaries the different hyperparameters used to tune the model and for comparable evaluations.

Number of layers	2, 3,
Hidden size	1024, 2048
Learning rate	5e-4, 7e-4. 9e-4

TABLE II

Different hyperparameters used to tune the model

The results in figure 9 display the accuracy and loss on the validation dataset for classification model with different number of hidden layers. The trend shows that the classification model with 3 layers outperform which with 2 layers. We can also observe from the training losses that with more layers, the model can minimise its training loss even faster.

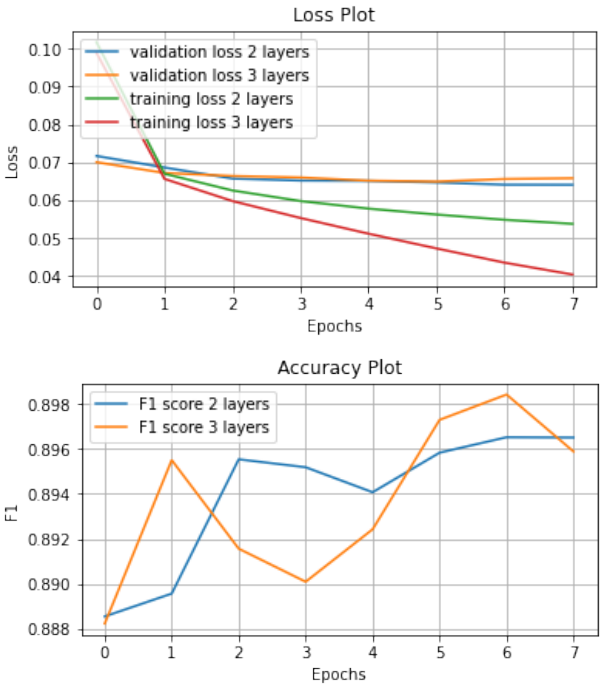

Fig. 9. Number of Hidden Layers Analysis

Figure 10 shows the performance of neural networks with different hidden sizes. It is clearly that a larger size of hidden

layer gives a better result. However, it also suffers from overfitting issue severer than which of size 1048. For the model with hidden size 2048, the validation loss goes up after epoch 4 and the f1 score is significantly lower than which with hidden size 1024. A trade-off decision must be made here and we decide to choose the mix of 2048 and 1024 hidden sizes for 3-layer classification model.

The Learning rate analysis is conducted by using ADAM. Figure 11 depicts that the learning rate of 7e-4 produces the best result as it has a moderate value. Specifically, a small learning rate converges slow and does not reach the optimal value while a large learning rate converges fast but delivers a ordinary result since the oscillation may occur.

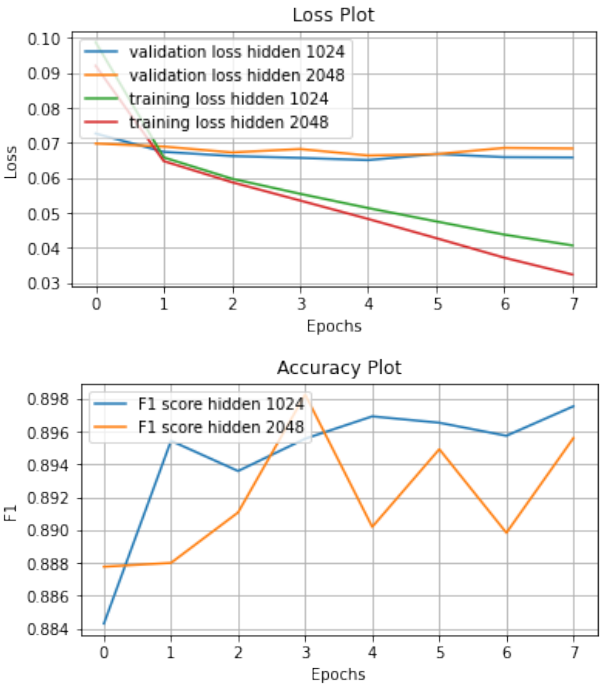

Fig. 10. Size of Hidden Layers Analysis

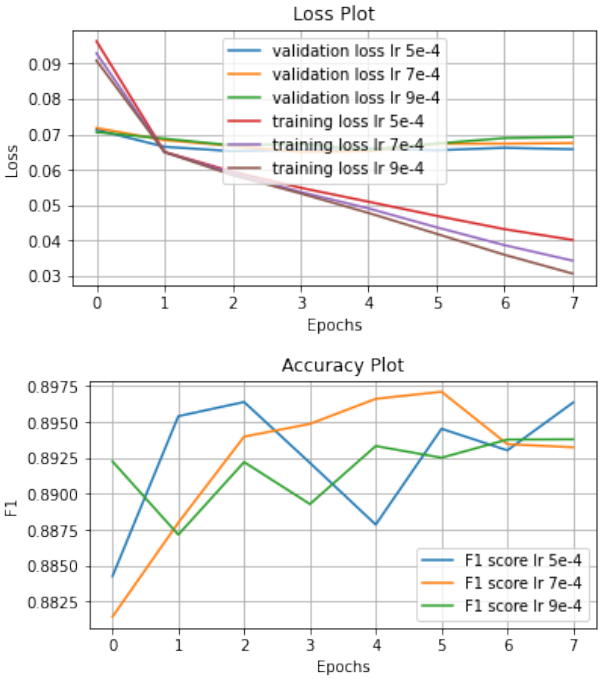

Fig. 11. Learning Rate analysis

2) Ablation studies:

The ablation studies are adapted based on two aspects. We first analyse the effect of utilising a scheduler (warm-up) in training process by comparing the performance between the model with and without warm-up. Then we look into the encoding strategy of CLIP. We test three encoding options: using only image encoding for classification model, using only text encoding for classification model, and using the concatenation of both text and image encoding for classification model.

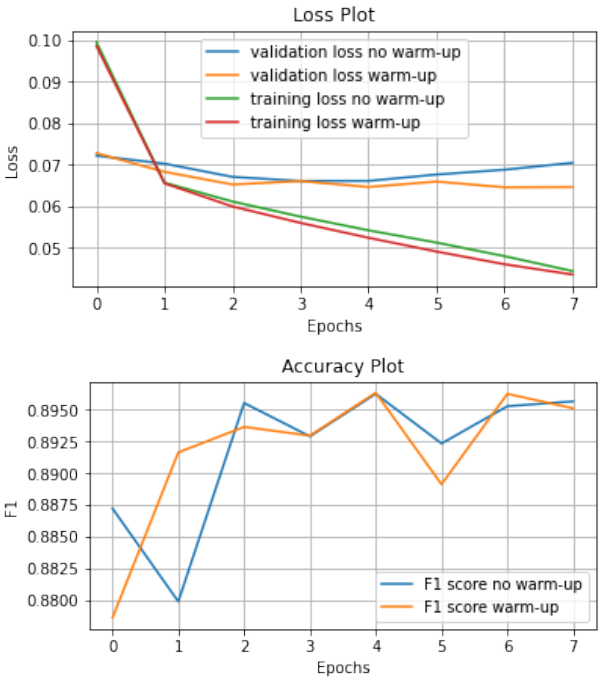

Fig. 12. Ablation Study for warm-up

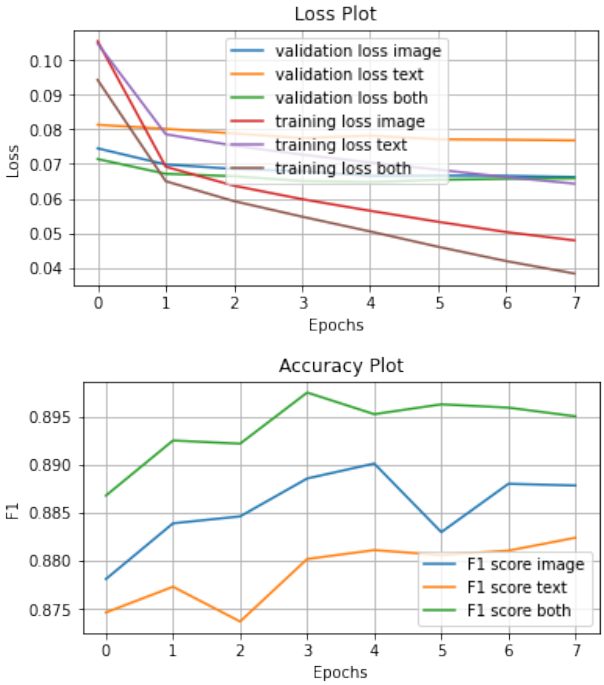

Fig. 13. Ablation Study for encoding options

The results in figure 12 shows the ablation study of whether using warm-up in the training period. Since the weights in the model are randomly initialised at the beginning, without warm-up the model could overfit at initial stage and the weights are updated with a relatively large step, introducing unnecessary oscillation. As shown in the lower chart in figure 12, with warp-up, the accuracy is improved smoothly in early stage.

Figure 13 shows the results of using different encoding from CLIP. It is obvious to see from the accuracy plot that using the concatenation of image and text encoding yields a better performance. We can conlude from the results that although CLIP is trying to minimise the cosine distance between image encoding and text encoding in the mathematical space, the resulting encodings from CLIP is not perfectly close, where the encoding of images contains is relatively more informative than the encoding of texts.

V. Discussion and conclusion

In this study, we verified the 'zero-shot' visual classification capability of the CLIP model on a large-scale dataset. With just a little work, the clip is able to easily outperform the specifically fine-tuned models. This generalization ability made me caught a glimpse of the super AI. We also explored the future of deep learning behind the logic of CLIP. It confirms that task-independent (natural language) supervision can also be used to improve the quality of deep learning models, and this is a great encouragement for us to move forward on this path. Not only that, CLIP, like its predecessor the GPT Family, may turn 'zero-shot' into possible. Although the real 'zero-shot' is far from being realized, the CLIP at least gives us great inspiration on this route. Nowadays with the increasing demands for data in deep learning, many people can't afford the high cost of them. CLIP shows us another way: we can make use of the enormous Internet resources. Will the deep learning practitioners of the future spend their lives on improving the performance on a single dataset, or manipulating the infinite amount of Internet data, or both? We don't know, but this study taught me that deep learning could have many possibilities.

VI. Appendix

A. Instruction on how to run the code

Please refer to README.md under the root directory.

REFERENCES

[1] 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*, pp. 8748–8763, PMLR, 2021.

[2] H. Ben-Younes, R. Cadene, M. Cord, N. Thome, "Mutan: Multimodal tucker fusion for visual question answering", *in Proceedings of the IEEE international conference on computer vision*, pp. 2612–2620, 2017.

[3] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, "A survey of data augmentation approaches for nlp", *arXiv preprint arXiv:210503075*, 2021.

[4] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, "Pre-trained models for natural language processing: A survey", *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[5] Y. Du, Z. Liu, J. Li, W. X. Zhao, "A survey of vision-language pre-trained models", *arXiv preprint arXiv:220210936*, 2022.

[6] H. Zhang, H. Song, S. Li, M. Zhou, D. Song, "A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models", *arXiv preprint arXiv:220105337*, 2022.