Text Enhanced Image Classification

Jiayun Wang
jw7978@nyu.edu

Lehan Li
ll3745@nyu.edu

Qiwen Zhang
qz2274@nyu.edu

1 Introduction

1.1 Background

Image recognition has been widely investigated in the field of computer vision. ResNet (He et al., 2016), which revolutionized neural networks by introducing the residual connection layer and won the ImageNet competition in 2015, becomes a common baseline architecture in many modern papers.

Bert (Devlin et al., 2018), which takes the encoder part of transformer, learns useful language embeddings using its masked language modeling objective. Lots of variants of Bert were introduced after the occurrence of Bert. DistilBERT(Sanh et al., 2020), as one of the variants, is smaller and more efficient, designed to provide much of the same functionality as BERT but with fewer resources. DistilBERT has about 40% fewer parameters than BERT, which makes it faster and more efficient for training and inference.

Contrastive learning was largely utilized to learn meaningful embeddings and when it is combined with different modalities, such as text and image, models can learn embedding associations between text and image through contrastive learning. Given the large exploration in the pure computer vision models in image classification tasks and inspired by (Xiang et al., 2022) which adds textual information to the action recognition task, we are wondering if we can add the extra information of text to enhance model's capability in learning image representations via contrastive learning. More specifically, we adopt a multi-modal architecture that uses an image encoder to extract image information and a text encoder to extract image category description information. We then design two training objectives for the text and image embeddings. The first objective is to associate the image representation to its corresponding category text representation via contrastive learning. The second objective is to make the image encoder learn the right classification category with cross entropy loss.

For efficient automatic category description generation, we take advantage of the GPT-3.5 (Brown et al., 2020) as the knowledge engine. Recently Large Language Model has become popular in many tasks. It contains rich knowledge that can be useful in many aspects. We used GPT-3.5 for the generative category-level description to save manual labeling work to write the text description for each category in the image classification dataset. We designed four types of prompt to study the effect of text on model's performance.

1.2 Related Work

The attempt of connecting text and images was explored by the the work of Ang Li and his co-authors at FAIR who in 2016 demonstrated using natural language supervision to enable zero-shot transfer to several existing computer vision classification datasets (Ang et al. (2017)), such as the ImageNet dataset, who achieved this by fine-tuning an ImageNet CNN to predict a much wider set of visual concepts from the text of titles, descriptions, and tags of 30 million photos and were able to reach 11.5% accuracy on ImageNet zero-shot.

In 2013, Richer Socher and co-authors at Stanford developed a proof of concept by training a model on CIFAR-10 to make predictions in a word vector embedding space and showed this model could predict two unseen classes (Socher et al. (2013)).

Preprint. Under review.

More recenly, CLIP (Contrastive Language–Image Pre-training) (Radford and Kim (2021)) builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning, which used more modern architectures like the Transformer and improved zero-shot performace to 76.2% on ImageNet.

1.3 Problem Setup

The task is to develop a robust text-enhanced image classification model capable of accurately categorizing images from the CIFAR-10 dataset into one of its 10 predefined classes. The CIFAR-10 dataset is a well-known benchmark for image classification tasks, containing 60,000 32x32 color images across ten classes, with 6,000 images per class.

More specifically, given an image and a piece of descriptive text of the corresponding label of the image, we attempt to predict the probability distribution of the image belongs to each class and output the final prediction as the index of the maximum probability.

2 Model and Methodology

The project combines BERT and ResNet50 to produce a robust classification for CIFAR-10. The entire pipeline of the project is illustrated in Figure 1

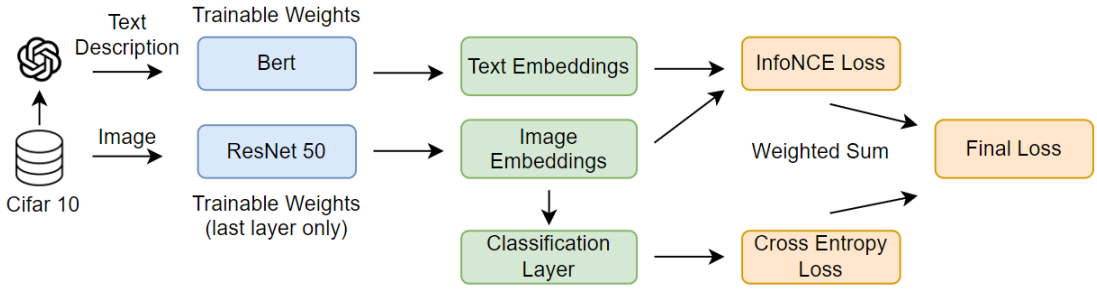

Figure 1: Model Pipeline

2.1 Generation of Descriptive Text

Descriptive text for images is generated using GPT-3.5 (Brown et al., 2020). A block of text is generated for each of the 10 classes; thus, images with the same label share the same descriptive text. To examine the effect of descriptive text, four different prompts are provided to GPT to generate diverse descriptions. The prompts entered into GPT are as follows

- Type 1: Describe what a {} is with one sentence.
- Type 2: Describe what a {} is in detail with one paragraph.
- Type 3: Describe what parts a {} have with one sentence.
- Type 4: Describe what parts a {} have with one paragraph.

Thus each image is paired with four different descriptions.

2.2 Models

The BERT model (distilbert-base-uncased) is employed for text-based input data. Pretrained weights are fine-tuned during the training process to generate text embeddings for each data point.

ResNet50 is utilized for image input data. Pretrained weights are employed in the project, and only the last layer of ResNet50 is fine-tuned during training due to computational constraints, while the

weights for all previous layers are frozen.

A projection block is then added after the two embeddings. Since the original embedding dimensions differ, the projection block transforms them into a common space, allowing subsequent steps to seamlessly integrate information from both embeddings without encountering dimensionality issues. The projection block is created using two linear layers with a GELU activation function, dropout, and layer normalization. Thus, the projection block transforms text embeddings from 768 to 256 dimensions and image embeddings from 2048 to 256 dimensions. The transformed embeddings are then evaluated using a contrastive loss function.

In addition to similarity comparisons, image embeddings are directly used for image classification. A classification head is added after the image embeddings to perform image classification. The classification head is a linear layer projecting from 256 to 10 dimensions. The results are evaluated using the cross-entropy loss function.

To assess whether text input enhances model performance, the baseline model is set as ResNet50 with only the last layer trainable just like the image encoder block in our architecture to make them comparable, and the results are compared with those of the enhanced model.

2.3 Calculation of Loss

To explore the capacity enhancement of descriptive text on image features, we design the loss to perform the task of both image classification and contrastive learning between the image and text embeddings, specifically, we use (1) cross entropy as the loss for classification (2) InfoNCE as the contrastive loss between image and text embeddings, where we perform negative sampling to get triplets of (image, positive text, negative next). Since we have a relatively small number of classes, we simply sample a random class different from the target class for each image. Final loss will be the weighted sum of those two components.

$$l_{clf} = -\sum_{i=1}^{N} \frac{1}{N} log\left(\frac{exp(x_{n,y_n})}{\sum_{c=1}^{C} exp(x_{n,c})}\right)$$

$$l_{cont} = -\sum_{i=1}^{N} \frac{1}{N} log\left(\frac{f_k(x_{t+k,c_t})}{\sum_{x_j \in X} f_k(x_j,c_t)}\right) \quad where \quad f_k(x_{t+k,c_t}) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$$

$$l_{final} = w_{clf} l_{clf} + w_{cont} l_{cont} \quad where \quad w_{clf} + w_{cont} = 1$$

3 Experiments

The effect of prompt design and its subsequent influence on model performance is examined. Four models are trained using different sets of descriptive texts generated by distinct prompt instructions. The accuracy is reported in the Table 1. The influence of prompt design shows minimum influence on model performance in terms of accuracy, as well as the time for convergence. It's obvious that regardless of which prompt design to use, the enhanced model has consistent higher accuracy compared to baseline ResNet-50 only model.

The impact of combining weighted loss is assessed. The final loss is a weighted sum of both contrasive InfoNCE loss measuring the similarity between image and text representation and cross entropy loss measuring the accuracy of image based model classification. Various weights are assigned during the combination process to modulate the influence of the two losses on the final loss. The result shows that the model is insensitive to the weights employed in the combination of the loss functions. And the model performance is consistent with the previous experiment.

Baseline Model	
Model	Accuracy
resnet-50	0.8126

Ours		
Prompt Type	Classification VS Contrastive Loss Weight	Accuracy
Experiment I: Effect of Prompt Design		
1	0.8,0.2	0.90035
2	0.8,0.2	0.90025
3	0.8,0.2	0.90135
4	0.8,0.2	0.90085
Experiment II: Effect of Weighted Loss Combination		
1	0.8,0.2	0.90085
1	0.5,0.5	0.90165
1	0.2,0.8	0.90125
Experiment III: Effect of Freezing BERT		
1	0.8,0.2	0.90473
1	0.5,0.5	0.90535
1	0.2,0.8	0.90365

Table 1: Model Result Comparison

To examine the internal mechanism of the model, the weights of classification head from image representation is visualized in Figure 2. The x axis refers to the original 2048 dimensions from ResNet50, and the y axis corresponds to the final prediction of 10 image categories. Some vertical extreme values are observed in the graph (dark purple and bright yellow lines) which is an indication of model's ability to learn using weights in the classification tasks.

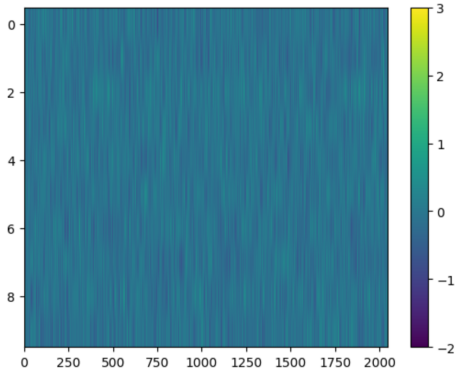

Figure 2: Weights of Classification Head

To better understand how text embeddings from BERT is optimized during the training process, a sample data point is selected and we extract the projected text embeddings of the sample data before training (generated by pretrained weights) and after training (generated by finetuned weights) and visualize the embeddings in Figure 3. A blurry pattern could be recognized in the embeddings from pretrained weights, indicating that the pretrained weights possess inherent contextual comprehension of the text. The pattern becomes more robust and visually apparent after the training process, sug-

gesting that BERT model and projection block is able to capture the contextual meaning specifically related to this input text, and the weights are optimized to better adapt to our dataset.

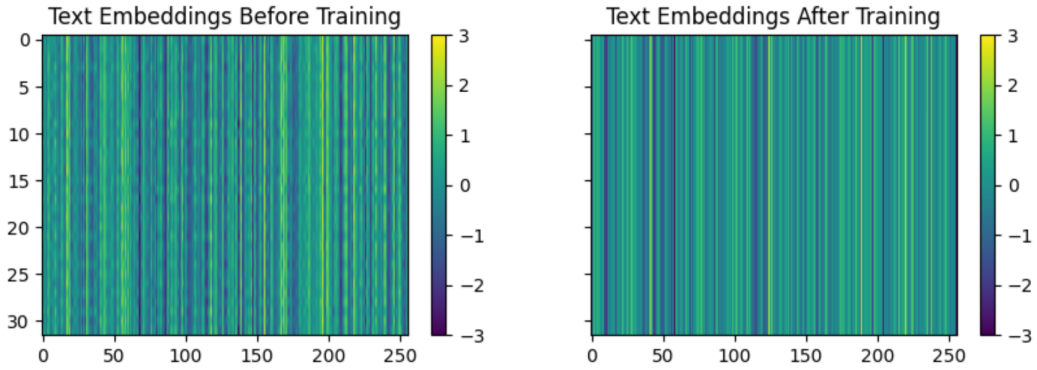

Figure 3: Visualization of Text Embeddings Before and After Training

4 Discussion

4.1 Ablation Study

One ablation study is to investigate the effect of freezing text encoder during training. We can see from table 1 that after freezing the weight of DistilBert, the accuracy increases from 0.90035 to 0.90473, 0.90165 to 0.90535, 0.90125 to 0.90365 for weight combination {0.8, 0.2}, {0.5, 0.5} and {0.2, 0.8}. This might be due to the fact that freezing text encoder makes the sentence embedding of each category description the same throughout training. Because the sentence embedding for each category description falls relatively apart in the embedding space as the model is pre-trained, keeping embedding for each category the same throughout training ensures the image embedding to move towards their corresponding text embedding consistently which boosts the discriminative property among image embeddings of different category and avoids embedding collapse while learning to classify through the regular classification loss. If the weights of DistilBert are not frozen, both text embedding and image embedding would move in the embedding space. In this case, we cannot be sure if image embedding for different category gained that extra discriminative ability from the text information because textual embedding can also move in the embedding space, which might lose the original information that different category description is distant in the embedding space. Hence, fine-tuning DistilBert might lead to a worse result which is what table 1 showed.

4.2 Performance Analysis

The configuration of the prompt exhibits minimal impact on model performance. After manually reading the generated descriptions for each image categories, there is a substantial degree of similarity observed among the four sets of descriptive texts. Given the high degree of resemblance between prompt instructions and output text, it is intuitive that the model's performance doesn't fluctuate greatly based on input text descriptions.

To further investigate the contribution of image encoder and text encoder, we also experimented with different weight assign to the classification and contrastive learning head. Although our initial expectation was that when higher weight is assigned to the classification head, majority of the weight optimization flows through the image encoder, which is eventually used for image classification, thereby is supposed to perform better compared to lower weight assignment. However, empirical results showed that the effect is actually negligible. Our hypothesis to the result is that since our model jointly trains both image and text encoder, the image and text embeddings are both updated in the embedding space throughout the learning process. When either is assigned a higher weight, it manages to learn the embeddings better, meaning different classes are further away from each other and same classes are closer; the one with lower weight assignment will benefit from the training less, meaning the learnt embeddings are less well separated. Nonetheless, under the contrastive

learning setting, this effect might actually counters each other out. For example, if the image encoder is assigned a higher weight such that embeddings are well separated, when contrastively updated according to the distance from the less well separated text embeddings, the eventual setting point of the image embeddings might be similar to the weights being interchanged. Thus, when those image embeddings being used for classification, they carries similar capacities to classification decision-making.

5 Github Link

`https://github.com/QiwenZz/text_enhanced_image_classification/tree/main`

6 References

Ang, L., Jabri, A., and Joulin, A. (2017). Learning visual n-grams from web data.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Radford, A. and Kim, J. W. (2021). Learning transferable visual models from natural language supervision. *arXiv:2103.00020v1*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Socher, R., Ganjoo, M., Manning, C., and Ng, A. Y. (2013). Zero-shot learning through cross-modal transfer. *NeurIPS*.

Xiang, W., Li, C., Zhou, Y., Wang, B., and Zhang, L. (2022). Language supervised training for skeleton-based action recognition. *arXiv preprint arXiv:2208.05318*.