Sports Image Classification using CLIP

Saketh B
Visvesvaraya National Institute of Technology, Nagpur

1 Introduction

The given task is about classifying sports images. The total number of unique classes present in the dataset are 22, some of the examples are baseball, chess, wrestling, etc. There are a total of 11040 images in the train set and 2760 images in the test set. The history of image classification algorithms is very vast. One of the very early methods [1] performed image classification on a very large scale and proposed a benchmark dataset called the ImageNet which is widely used even now. In spite of the availability of standard Convolutional Neural Networks (CNNs) like ResNet [2], VGGNet [3], etc, here we implement the CLIP model [4] by OpenAI which can be utilised as zero shot classifier. CLIP (Contrastive Language-Image Pre-Training) is a neural network pre-trained on a variety of (image, text) pairs and can be finetuned for a numerous downstream tasks. The most interesting characteristic of this model that made us choose it is that this model can perform the classification task with great accuracies even though it is not directly optimized for it.

2 Method

The CLIP model consists of 3 main building blocks in its architecture:

- ✓ The image encoder: In our case it is a standard ResNet 50 model [2].
- ✓ The text encoder: In our case it is a DistilBert model [5].
- ✓ A projection head: It is a simple neural network which projects both the image encoding and the text encoding to a common space.

The entire aim of the model is to train it in such a way that given an image or a caption, the encodings or embeddings that are created should be the same as the information present in both of them is the same. A word of caution is that this is not a one - to - one mapping because there can exist multiple meaningful captions for a particular image and vice versa. We design different ablation experiments to see the advantages of using this model for our task. In our task we pre - train the CLIP model on flickr8k dataset which is a collection of common image - caption pairs and then isolate the ResNet 50 encoder from it and couple it with a Multi Layer Perceptron (MLP) to finetune on the sports dataset which will be useful to analyse the change in out of distribution accuracies [6].

3 Architecture

The architecture of the CLIP model can be seen in the Fig. 1 and we code this in PyTorch framework. For a particular ablation, we isolate the image encoder from the model and combine it with an MLP.

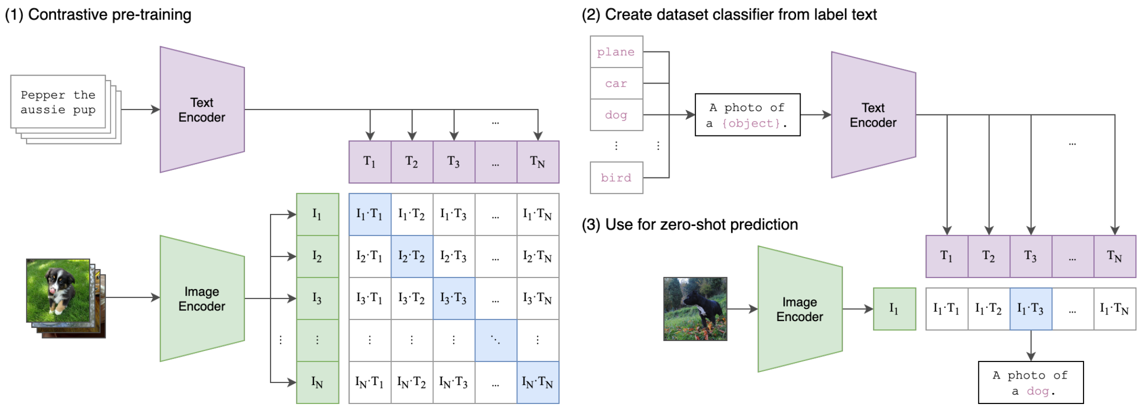
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
$I_2{\cdot}T_1$ $I_2{\cdot}T_2$ $I_2{\cdot}T_3$ ... $I_2{\cdot}T_N$
$I_3{\cdot}T_1$ $I_3{\cdot}T_2$ $I_3{\cdot}T_3$ ... $I_3{\cdot}T_N$
$I_N{\cdot}T_1$ $I_N{\cdot}T_2$ $I_N{\cdot}T_3$ ... $I_N{\cdot}T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
T_1 T_2 T_3 ... T_N
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ ... $I_1{\cdot}T_N$
A photo of a dog.

Figure 1. CLIP architecture [4]

3.1 Ablation

In this project, we will be performing the following tasks:

- ✓ Implement the CLIP model using the PyTorch framework.
- ✓ Pre-training the CLIP model using image - caption pairs present in the the Flickr8k dataset.
- ✓ Check for zero shot accuracy on the sports image classification task.
- ✓ Finetune on the sports dataset by coupling the image encoder of the CLIP model with an MLP and check the performance.

The pre - trained model on the flickr8k dataset gives 35% accuracy on our sports image dataset eventhough it was not directly trained on the sports dataset. This is the power of contrastive pre training. If we would have trained the CLIP model on a larger corpus of image - caption pairs, we would have achieved a better zero - shot accuracy.

3.2 Results

From the experiments, we can observe that we can actually achieve substantial performance on classification tasks without directly optimising it for classification. This method of contrastive pre - training proves to be a good starting point for any downstream tasks and can achieve good performance by finetuning it for just a few epochs. The loss curve and the accuracy curve are shown in Fig. 2 and Fig. 3 respectively. Apart from that, we also show the guided backprop [7] output which makes a heat map according to the spatial activations of the model in Fig. 4 and class activation map [7] in Fig. 5. These visualisations help us understand more in terms of: at which location the model is paying attention to perform classification.

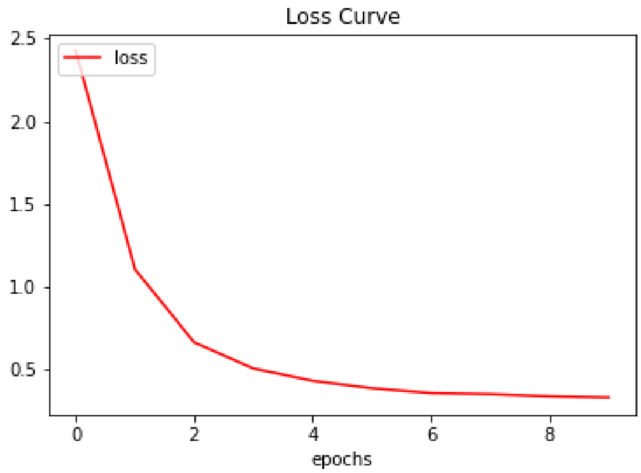

Figure 2. A graph showing the trend of loss values

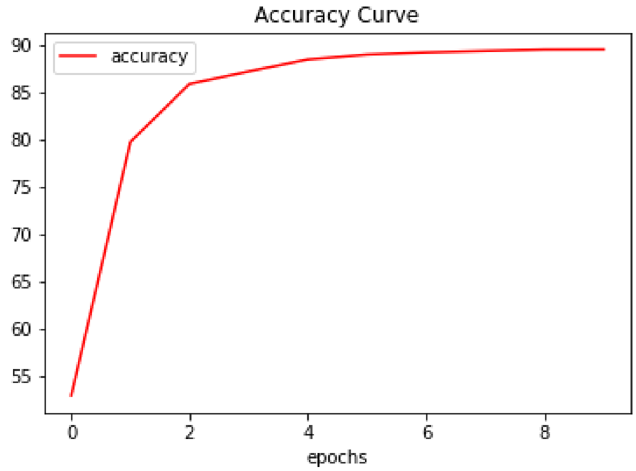

Figure 3. A graph showing the trend of accuracy values

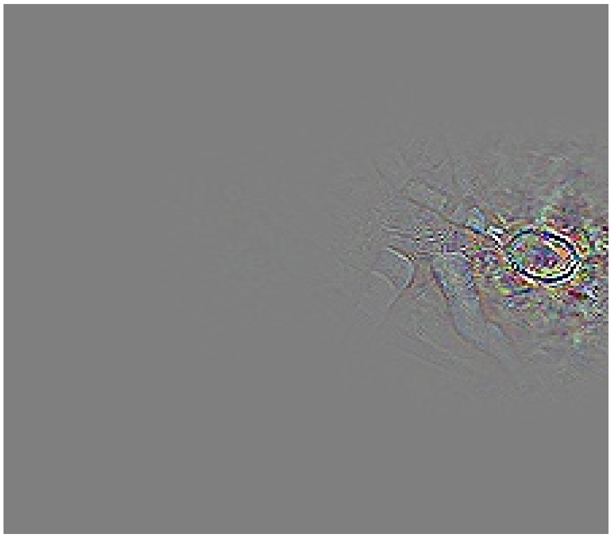

Figure 4. A picture displaying Guided Backprop module [7] output

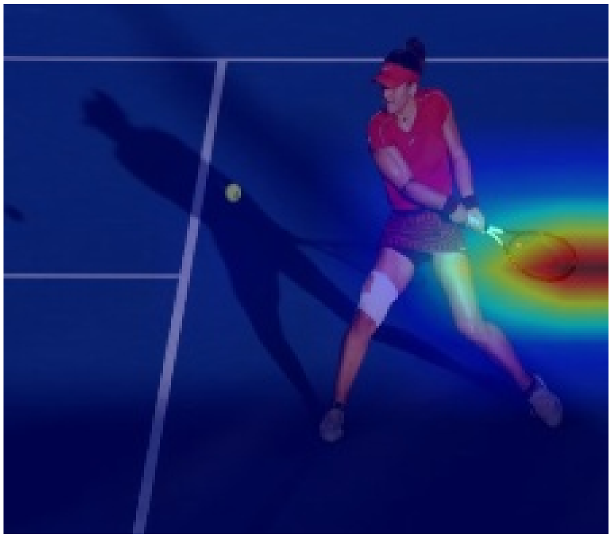

Figure 5. A picture displaying GradCam module [8] output

4 References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[2] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[3] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV].

[4] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 [cs.CL].

[6] M. Wortsman, G. Ilharco, M. Li, *et al.*, *Robust fine-tuning of zero-shot models*, 2021. arXiv: 2109.01903 [cs.CV].

[7] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, *Striving for simplicity: The all convolutional net*, 2015. arXiv: 1412.6806 [cs.LG].

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, 336–359, 2019, ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. [Online]. Available: http://dx.doi.org/10.1007/s11263-019-01228-7.