Object Recognition in Computer Vision: Sketch-image Classification Challenge

Anonymous submission

Paper ID

1. Introduction

In this report, I have provided my approach and experiments for the Sketch-image classification kaggle challenge, with results.

2. Approach and Experiments

I started with improving baseline and did few experiments and then I used two pre-trained models with different hyper-parameters. Throughout all experiments i used the dataset provided on kaggle. for my experiments. I make use of kaggle notebook with GPU and github repository. Below i have given the details of what experiments i did, what accuracy i get with test set on kaggle and which challenges i faced and how i overcome those.

2.1. Experiments with Extended Baselines

Initially, I modified the baseline by adding convolutional layers. The architecture included 3 convolutional layers (kernels: 5, 5, 3), batch normalization, max pooling, an adaptive average pooling layer, a fully connected layer with dropout, and a final linear layer for classification[2]. With parameters (epochs=10, lr=0.001, batch size=32), the results were poor (test error: 0.00135), performing worse than the baseline. This highlighted the limitations of extending simple baselines for complex datasets, So i didn't tried to do more experiments and moved to pre-trained model

2.2. Fine-Tuning DINOv2

I used pre-trained DINOv2 models (small, base, and large(registers) variants)[4]. The approach involved freezing the backbone and fine-tuning the classification head. I used dinov2 model using torch hub and optimizer AdamW, with cross entropy loss for those experiments[5].

2.2.1. Experimnets

I did experimnets with dinov2_vits14, dinov2_vitb14, and dinov2_vitb14_reg. For data augmentation i used two different transforms:

1) Basic transformations: resized (224), normalize (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).

2) Extended transformation: resized(256), random resized crops(224,, scale=(0.8, 1.0)), rotations(15), and normalize (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])[3].

below is the different experiments with hyperparameters and results with dinov2, i keep changing batch size, learning rate, weight decay and no of epochs by observing the under and over-fitting during training. For simplification, I have only added improved results in Table 1.

Model Variant	Batch Size	Learning Rate	Epochs	Test Accuracy
dinov2_vits14	32	0.001	30	86.1%
dinov2_vits14	64	0.00141	30	86.3%
dinov2_vitb14_reg	64	0.00141	20	87%
dinov2_vitb14	128	0.001	20	90%

Table 1. Model configurations and results with Dinov2

2.3. Fine-Tuning Vit Vision transformer

After Dinov2, i also explored the Google ViT-base (Hugging Face) model and finetune[1] it on provided dataset. In experiments, i used both simple and extended data augmentation mentioned above. In Table 2, i added only the experiments where i get highest accuracy for that model.

2.3.1. Experiments

below table presents the results(test set) and hyperperameters used. here i also used Adamw optimizer with cross Entropy loss, initially i oberserved it's overfitting so i changed my batch size and tried weight decay 0.001 to see if it improves overfitting, but it did improves only 1%.

Model Variant	Batch Size	Learning Rate	Epochs	Test Accuracy
vit-base-patch16-224	32	0.001	20	78.8%
vit-base-patch16-224	64	0.00141	20	78.2%

Table 2. Results with google/vit-base-patch16-224

References

[1] Hugging Face. How to fine-tune vision transformers. https://huggingface.co/blog/fine-tune-vit. 1

[2] GeeksforGeeks. Introduction to convolution neural network. https://www.geeksforgeeks.org/introduction-convolution-neural-network/. 1

[3] Malyaj Mishra. Fine-tuning dinov2: Custom training for your own ai projects. https://medium.com/data-science-in-your-pocket/fine-tuning-dinov2-custom-training-for-your-own-ai-projects-6e8a5a486671. 1

[4] Facebook AI Research. Dinov2: Self-supervised learning with vision transformers. https://github.com/facebookresearch/dinov2/tree/main. 1

[5] Kili Technology. Tutorial: Fine-tuning dinov2 with kili python sdk. https://python-sdk-docs.kili-technology.com/2.157/sdk/tutorials/finetuning_dinov2/. 1