A Synthetic Task for FGIC

Main Project of Artificial Neural Network

Tianxing Yang, Zihao Liang and Haoyu Chen

School of Computer Science and Engineering, SYSU

June 18, 2023

Content

1. Task 1&2: Baseline and Fine-tune
2. Task 3: Object Detection
3. Task 4: Synthetic Images
4. Task 5: Efficiently use ViT
5. Task 6: Interpretation of the model
6. Task 7: Robustness of the model
7. Discussions

Baseline

- Various training tricks to improve model performance
- Transfer learning: fine-tune pretrained model

Focus on Stanford Dogs dataset for this slide.

Training Settings

- Using ResNet50 as main model
- Resize each image to 512×512 (keep the ratio of images by padding)
- Random gray scale and random flip
- Normalize each image by universial parameter $([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])$
- Batch size $= 16$

Training Tricks

- Use pretrained model
- Using SGD as optimizer and LR scheduler
- learning rate=0.01, momentum=0.9, gamma=0.1
- Epoch num=25, LR scheduler step size=7

Result of Baseline

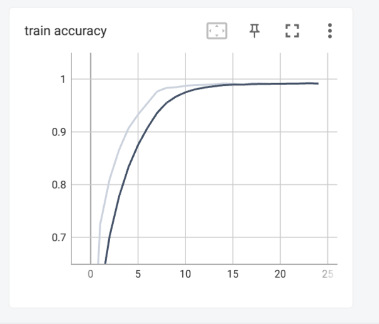


Figure 1: train-acc

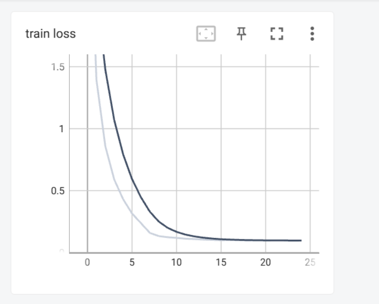


Figure 2: train-loss

Result of Baseline

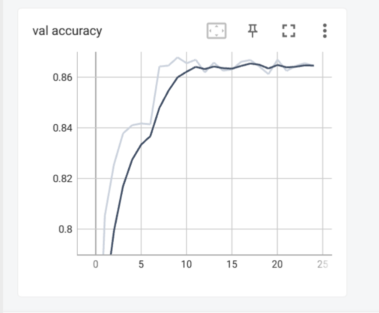


Figure 3: test-acc

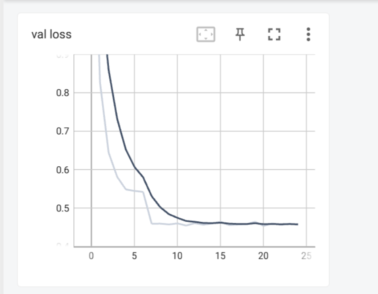


Figure 4: test-loss

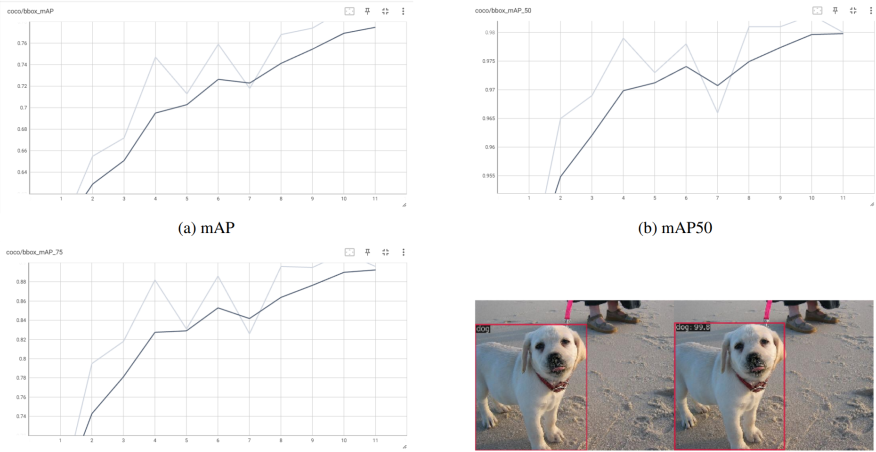
Framework

- Using MMDetection as objection framework
- Use sahi library to transform annnotation of Stanford Dogs dataset to COCO format
- Train/Inference Stanford Dogs dataset as custom dataset by using MMengine

Object Detection

- Training from scratch
- Using Faster RCNN as model
- Using ResNet-101 as model backbone

Object Detection



(a) mAP

(b) mAP50

(c) mAP75

(d) A example of our object detection model

Figure 5: Faster RCNN result

Object Detection

We cropped each images using two different strategies:

- Crop as the predicted bboxes
- Find the nearest square shape to predicted bbox and crop it to square image

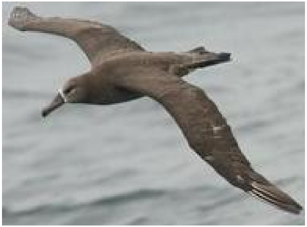


Figure 6: Strategy 1

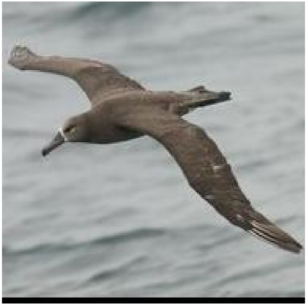


Figure 7: Strategy 2

Object Detection

Question: Accuracy Rate decreases for both datasets after we use cropped images for training and testing.

- Each dataset dropped $1 \sim 2$ percentage accuaracy rate after using cropped dataset
- No matter the cropping strategies

Synthetic Images

Expected Pipeline:

- Getting bbox of trainset images from annotations
- Getting bbox of testset images from inference results
- Cropping each image to bbox and resize it
- Generating more images from cropped images
- Training using images from trainset and generated iamges

Synthetic Images

- Using BigGAN-deep model, introduced by Brock et al. in Large Scale GAN Training for High Fidelity Natural Image Synthesis
- Use previously cropped images as input conditional datasets
- Using MMagic framework as training framework
- Unified resolution of images to 256x256
- Download BigGAN-deep model pretrained by ImageNet-1k dataset from Hugging Face

Synthetic Images

We met problems in this task:

- Requires a lot of computing resources, limited us from trying different methods and comparing them (We trained this model using 4x Nvidia A100, costed $\sim$ 2 days)
- Got unstable FID value during training process, not as the original paper claims (the original paper claims 10 FID value in different datasets)
- Our model generated images of poor quality, which have little value for extra training dataset

Synthetic Images

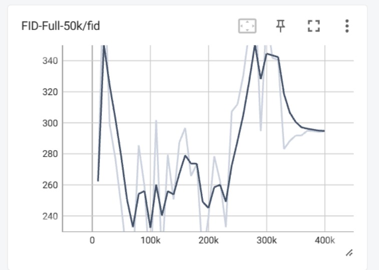


Figure 8: FID value

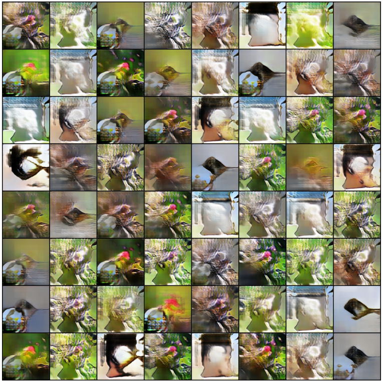


Figure 9: Generated images

Efficiently use ViT

- Explore different configurations (number of layers, number of attention heads, etc.)
- Consider using a hybrid model that combines both CNN and ViT. The CNN could be used to extract features from images, which can then be passed to the transformer for the classification task
- One of the biggest challenges with ViT is its demand for large amounts of data for training. Use transfer learning techniques, take a pretrained ViT model and fine-tune it.

Interpretation of the model

- Use Class Activation Maps to visualize the regions of the image that were important for the model to make its prediction (based on the model's final convolutional layer)
- Visualize attention maps, allow us to see which patches of the image the model paid attention to when making its predictions

Robustness of the model

- Adversarial examples are inputs designed to fool the model by adding small perturbations that lead to incorrect predictions
- Defensive distillation: train a second model (the ‘student’) to mimic the output of the first model (the ‘teacher’)

Robustness of the model

Using Projected Gradient Descent (PGD):

- Fast Gradient Sign Method (FGSM): Computing the gradient of the loss with respect to the input image and creating a new image that shifts each pixel by a small step in the direction of the gradient
- An iterative version of FGSM that applies FGSM multiple times with a small step size
- Projects the perturbed image back into an epsilon ball to ensure that the adversarial example does not go too far from the original image

Discussions

- Acknowledged from other groups, ONLY a conbination of Step 1&2 can achieve 90+% accuracy on two datasets, it is necessary to apply each steps to the model?
- Step 3 (Object Detection) lead to a drop in accuracy, how to explain this phenomenon?
- Step 4 (Synthetic Images) obviously can not improve the performance of the model, what is its usage?
- Robustness often comes at the cost of accuracy, need to make a trade-off between robustness and accuracy
- Use which mertics to measure the robustness of the model (F1 score, Certified Robustness, Lipschitz continuity, etc.)?

Thank you!