ELEC 4542: Project

The project can be either an individual project or a group project with 2-3 students. You can choose one topic from the following three topics, or propose your own topic. You are required to submit the code and a project report. If it is a group project, please also provide the responsibility of each student in the project report. If it's a single-person project, we expect you to complete at least 3 sub-questions (a)(b)(c) out of the 4 questions. For group projects, we expect you to complete all of the 4 sub-questions (a)(b)(c)(d).

1. Image Classification.

   (a) Train an image classification model on CIFAR [3] dataset with ResNet-18 [2]. Report the classification accuracy on the test set.

   For this project, we will use the CIFAR10 dataset. It consists of 60000 32x32 color images in 10 classes, with 6000 images per class. ResNet-18[2] is a convolutional neural network that is 18 layers deep. You can load both the dataset and network easily using `torchvision`.

   (b) Use different data augmentation techniques and different ways to avoid overfitting. Report the classification accuracy on the test set with different types of data augmentations.

   Data Augmentation is simply creating variations of training data to increases its size at the same time regularizes the network. Some common data augmentation techniques in image classification include

   - Random Flips and Rotations.
   - Random Crop and Center Crop
   - Normalizing the image
   - ...

   The choice of data augmentation techniques depends on the specific characteristics of the dataset and the problem at hand. It's often beneficial to experiment with different combinations of augmentation techniques to find the ones that work best for your task.

   (c) Visualize the class activation maps of the trained models.

   The concept of Class Activation Map (CAM) is introduced in this paper [8]. It is a technique used to visualize the regions of an image that contribute the most to the prediction made by a convolutional

neural network for a specific class. CAM provides a way to understand which parts of an image are important for the network's decision-making process. For a detailed explanation of CAM and its enhanced versions, as well as example code implementing CAM in PyTorch, please refer to this page

(d) Train an image classification model on CIFAR dataset with Vision Transformers[1] (ViT). (GitHub Page) Report the classification accuracy on the test set.

ViT is a model architecture that applies the Transformer architecture, originally designed for natural language processing tasks, to image recognition tasks. It replaces the traditional convolutional layers with self-attention mechanisms to capture global dependencies in the image. Instead of training the image classification model from scratch, an alternative approach is use a pre-trained model and fine-tune the last few layers. Please use this pretrained model and and fine-tune it for CIFAR image classification.

2. Vision-language model CLIP[5]. (Project Page, GitHub Page)

   (a) Use a pretrained CLIP model to compute the image-text similarity.

   We can extract the image and text features respectively with CLIP, and compute the cosine similarity between the image and text features. Choose 10 image-text pairs to compute the image-text similarity. The example code can be found here. You can use the image and text data from google doc.

   (b) Use a pretrained CLIP model for zero-shot image classification. Explore different prompt designs for the task.

   The pretrained CLIP model will be automatically downloaded when you load the model for the first time. You can find an example here.

   Next, explore the different prompt designs. You can try different variants of prompts (e.g. a photo of a folded chair, a photo of a office chair, a photo of a chair, a high-quality photo of a chair). Some examples of the form of prompts can be found here. Of course, you can also propose more of your own designs. The data for this task is CIFAR10 test dataset.

   (c) Use a pretrained CLIP model for text-based image retrieval.

   Choose one prompt and several text-based images, the target image can be selected by calculating the maximum value of the similarity between this prompt and these text-based images. Use text-based images, which are first 100 images and captions (choose 10 of them) in the COCO[4] validation dataset. You can directly load COCO, please refer to this page. Different versions of pretrained CLIP models (e.g. ViT-B/32, RN50) should be used for the task and compare their performance differences.

   (d) Explore semantic segmentation with the CLIP model.

   MaskCLIP[9] can explore object localization by modifying the CLIP pretrained model. So you can try to do semantic segmentation with this method. There are 10 images from Pascal Context dataset, please refer to this page. Follow Step 1 and Step 3 in this page to visualize the semantic segmentation results.

3. Diffusion models for image generation.

   (a) Use Stable Diffusion [6] to generate images based on text queries.

   Start by reading the Stable Diffusion Blog and Stable Diffusion Docs to gain a deep understanding of the Stable Diffusion method. This will help you understand the theory and implementation details, and you may also find pre-trained models and code implementations that can save you time:

   Stable Diffusion Blog

   Stable Diffusion Docs

   Experiment with different text prompts to generate images. Here are some prompt examples for your reference, and you can also use your imagination to come up with your own prompts. Keep a record

of both successful and unsuccessful cases. This will help you understand the strengths and limitations of the generative model.

(b) Explore different sampling strategy and different number of sampling steps.

Experiment with different sampling strategies, such as the number of sampling steps, guidance scale, and different scheduler (DDPM, DDIM). These parameters can significantly affect the quality and diversity of generated images. Keep track of the results for each sampling strategy.

(c) Use ControlNet [7] to generate images based on text queries and control signals.

Read the ControlNet Paper and ControlNet Docs to understand the principles and implementation details of ControlNet in ControlNet Blog. ControlNet allows you to generate images based on text queries and control signals.

ControlNet Paper

ControlNet Blog

ControlNet Docs

Experiment with different control signals like canny, depth, and hed to see how they affect image generation. This will allow you to create images with various attributes and styles.

Here are some conditioning examples for your reference, the control conditions including canny, depth, segmentation, etc., and multiple conditions for question(d). And you can also try with your own creative thoughts!

(d) Try multiple controls for ControlNet.

Explore multiple control signals to gain a comprehensive understanding of ControlNet's capabilities. You can refer to the Multi ControlNet resource for details on trying out various control options:

Multi ControlNet

Experiment with combining different control signals to create more complex and nuanced image outputs. Here are some examples of multiple conditions for your reference. And keep a detailed record of your experiments, including the control signals used and the results obtained. This will help you identify which combinations of controls work best for different scenarios.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[7] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[8] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.

[9] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022.